Formalizing, Capturing, and Managing the Context of Statements in the Semantic Web

José Miguel Giménez García

N°d'ordre NNT : 2022LYSES018

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
Université Jean Monnet

Ecole Doctorale 488 Sciences, Ingénierie, et Santé (SIS)

Spécialité : Informatique

Soutenue publiquement le 18/07/2022, par :
José M. Giménez-García

Formalizing, Capturing, and Managing the Context of Statements in the Semantic Web

---

Devant le jury composé de :

Champin, Pierre-Antoine	Associate Professor, Université de Lyon	Rapporteur
Koubarakis, Manolis	Professor, University of Athens	Rapporteur
Hernandez, Nathalie	Professor, Université de Toulouse	Examinatrice
Laforest, Fréderique	Professor, Université de Lyon	Examinatrice
Maret, Pierre.Professor	Professor, Université Jean Monnet	Directeur de thèse
Zimmermann, Antoine	Associate Professor, École des Mines	Co-directeur de thèse
Fernández, Javier D.	Senior Information Architect, Roche	Invité
Martínez-Prieto, Miguel A.	Associate Professor, Universidad de Valladolid	Invité

UNIVERSITÉ DE LYON

DOCTORAL THESIS

Formalizing, Capturing, and Managing the Context of Statements in the Semantic Web.

Author:
José M. GIMÉNEZ-GARCÍA

Supervisors:
Prof. Pierre MARET
Dr. Antoine ZIMMERMANN

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Université Jean Monnet

18/07/2022

Declaration of Authorship

I, José M. GIMÉNEZ-GARCÍA, declare that this thesis titled, "Formalizing, Capturing, and Managing
the Context of Statements
in the Semantic Web." and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: José M. Giménez-García

Date: 18/07/2022

UNIVERSITÉ DE LYON

Abstract

Laboratoire Hubert Curien
Université Jean Monnet

Doctor of Philosophy

Formalizing, Capturing, and Managing the Context of Statements in the Semantic Web.

by José M. GIMÉNEZ-GARCÍA

Representing the context of statements has been a challenge in computer science for more than fifty years, and has been a concern in the Semantic Web since its inception. Ideally, when representing a set of statements in a context, their semantics should remain unchanged but confined within the context. That is, the same inferences that were possible before contextualizing the statements should be possible after the contextualization, but the inferred knowledge should be true only in the same context. Existing solutions in knowledge representation and reasoning usually deal with this issue by separating the contexts in one way or another and adding rules to transfer knowledge between them. These approaches seem unfeasible in the Semantic Web paradigm, where the existing solutions are based on reifying the statements into new individuals, whether by introducing new statements to relate this term to the original statement it represents, or by extending the formalisms of the language. These individuals will be used to make "statements about statements", providing information about the contexts in which the statements are true. This reification approaches, however, have the consequence of either breaking the semantics of the original statements, and/or having inferences that are not confined within the context.

In this dissertation we theorize that it is possible to represent statements within their context, and preserve and keep confined their semantics, without separating in different universes or extending the formalism of the language used to assert them. We formalize the process of contextualizing a set of statements in a *contextualization function*, formalize the properties that a contextualization function can have with regards to how it preserves and separates the semantics of each context, analyze existing contextualization approaches under these properties, and propose a new contextualization approach that better preserve and separate the semantics of the statements. We capture contextual information existing in real-world data, and we argue that the selection of a contextualization approach can depend on the information we want to represent and how these properties are respected. However, the selection of a specific approach heavily impacts how to capture, store, and query the statements. We postulate that it should be possible manage contextualized statements using a representation-agnostic solution, therefore allowing to to capture and represent statements and their context using any desired contextualization approach. We make a first step towards such solution presenting a binary serialization that can store and query contextualized statements using the currently most used contextualization approaches.

UNIVERSITÉ DE LYON

Résumé

Laboratoire Hubert Curien
Université Jean Monnet

Doctorat

Formaliser, capturer, et gérer le contexte des déclarations dans le Web Sémantique.

by José M. GIMÉNEZ-GARCÍA

La représentation du contexte des énoncés est un défi pour l'informatique depuis plus de cinquante ans et constitue une préoccupation pour le Web sémantique depuis sa création. Idéalement, lorsqu'on représente un ensemble d'énoncés dans un contexte, leur sémantique devrait rester inchangée mais confinée dans le contexte. En d'autres termes, les mêmes inférences qui étaient possibles avant la contextualisation des énoncés devraient être possibles après la contextualisation, mais la connaissance inférée ne devrait être vraie que dans le même contexte. Les solutions existantes en matière de représentation des connaissances et de raisonnement traitent généralement ce problème en séparant les contextes d'une manière ou d'une autre et en ajoutant des règles pour transférer les connaissances entre eux. Ces approches semblent irréalisables dans le paradigme du Web sémantique, où les solutions existantes sont basées sur la réification des énoncés en de nouveaux individus, soit en introduisant de nouveaux énoncés pour relier ce terme à l'énoncé original qu'il représente, soit en étendant les formalismes du langage. Ces individus seront utilisés pour faire des "déclarations sur les déclarations", fournissant des informations sur les contextes dans lesquels les déclarations sont vraies. Ces approches de réification ont cependant pour conséquence de briser la sémantique des énoncés originaux, et/ou d'avoir des inférences qui ne sont pas confinées dans le contexte.

Dans cette thèse, nous théorisons qu'il est possible de représenter les énoncés dans leur contexte, et de préserver et garder confinée leur sémantique, sans séparer dans des univers différents ou étendre le formalisme du langage utilisé pour les affirmer. Nous formalisons le processus de contextualisation d'un ensemble d'énoncés dans une *fonction de contextualisation*, nous formalisons les propriétés qu'une fonction de contextualisation peut avoir quant à la façon dont elle préserve et sépare la sémantique de chaque contexte, nous analysons les approches de contextualisation existantes selon ces propriétés, et nous proposons une nouvelle approche de contextualisation qui préserve et sépare mieux la sémantique des énoncés. Nous capturons les informations contextuelles existant dans les données du monde réel, et nous soutenons que la sélection d'une approche de contextualisation peut dépendre des informations que nous voulons représenter et du respect de ces propriétés. Cependant, la sélection d'une approche spécifique a un impact important sur la façon de capturer, de stocker et d'interroger les déclarations. Nous postulons qu'il devrait être possible de gérer les énoncés contextualisés en utilisant une solution indépendante de la représentation, permettant ainsi de capturer et de représenter les énoncés et leur contexte en utilisant toute approche de contextualisation souhaitée. Nous faisons un premier pas vers une telle solution en présentant une sérialisation binaire qui peut stocker et interroger des déclarations contextualisées en utilisant les approches de contextualisation les plus utilisées actuellement.

Contents

[a]This section, as well as future mentions to its content, correspond to new work not included in the published paper.

List of Figures

List of Tables

Introduction

Introduction

Knowledge representation dates back to thousands of years in the past. Aristotle's categories tried to classify all existing beings according to what could be said about them. During the middle ages, and up to the 19th century, the Porphyrian Tree was taught in classes of logic. This tree represented hierarchies of categories and individuals belonging to the most concrete of one of them.

In the 20th century, knowledge representation became an important topic in Computer Science. In the field of artificial intelligence, knowledge representation deals with representing facts as formal statements. These formal statements can then be used by a reasoning system to arrive to new conclusions.

More recently, representing and reasoning taking the context of the facts into account has been addressed as a challenge. For example, we could want to express that some of our facts are true according to a source (say, categories according to Aristotle), but some other are true according to another source (for example, the Porphyrian Tree). Representing all the facts together without taking into account in which context they are true could lead to inconsistencies and/or to logical implications that are not actually true outside the context.

Some approaches exist to represent contextual knowledge about statements. However, most of them involve extending the formalisms of the logic being used, or creating rules to relate different contexts that are considered to exist separately; or they simply ignore possible consequences that the change in representation can have in the reasoning.

In this thesis, we explore the possibility of representing the context of statements within the logical system in which the statements are formulated. The goal is to represent statements in different contexts together, while being able to maintain the logical consequences of the statements in each context. We focus our work in Semantic Web data, but during our first chapters we take a step back to consider First Order Logic and Description Logic, and how solutions in each of them can be applied to the others. We then explore how to capture and manage in practice the context of statements.

This work is framed under the WDAqua *(Answering Questions using Web Data)* project, which had the goal of fostering the state of the art in data-driven question answering, focusing on Semantic Web data. This thesis was conceived with the idea of better using contextual data to improve the answers that rely on that information.

Representing Facts

Data is a fundamental component in Computer Science, therefore how to represent it is a key element. Data is used to encode knowledge, that can be decomposed as sets of facts. As an example let us consider these two facts:

1. *Barack Obama is the President of United States*
2. *A President of the United States needs to be born in USA* territory

These facts can be represented as free text, structured data (such as CSV), or semi-structured data (such as Web tables). These are all means to represent the knowledge, and are—to a different degree—machine-readable.

It is important to have machine-readable data in order to allow automatic processing. One such task is reasoning, i.e. inferring new facts from existing ones. Continuing our example from above the following fact can be inferred:

3. *Barack Obama was born in USA territory*

It is clear that if we represent this knowledge as free text it is quite difficult to automatize this reasoning process because of the non-formal representation of the knowledge. This is why in the last decades computer science developed techniques to represent formally these facts so that automatized reasoning process is made possible.

Facts are represented formally as logical formulas in a logic system. We refer to these formulas that represent facts as *statements* during this dissertation. What information a statement can convey is dependant on the rules of the logic system used. For example, in First Order Logic, a predicate can state a fact between an arbitrary number of individuals. Example 1 states the fact of Obama being born in 4 of August of 1961 in the USA in First Order Logic.

```
BornIn(barackobama, usa, "1961-08-04")
```

Example 1: Example of statements in First Order Logic

In Description Logics (DL), a decidable subset of FOL, it is only possible to state facts about a single individual (through a Class) or between two individuals (through a Role). Example 2 expresses the same fact about the birth date and birth place of Obama in Description Logic.

```
BornIn(barackobama, usa)
BirthDate(barackobama, "1961-08-04")
```

Example 2: Example of statements in Description Logics

In OWL (Ontology Web Language), a family of subsets of DL with different levels of expressivity, one can express fact similarly as with DL and the reasoning capability is dependent on the chosen subset. Example 3 expresses again the birth date and birth place of Obama, this time in OWL.

```
ObjectPropertyAssertion( bornIn barackobama usa )
ObjectPropertyAssertion( birthDate barackobama "1961-08-04" )
```

Example 3: Example of statements in OWL

In the Semantic Web, statements are represented using RDF as triples: tuples of three elements: *subject*, *predicate*, and *object*. The subject is the entity about which the statement is made, the predicate describes what is said about this entity, and the object gives the actual value to the predicate. Example 4 describe the the birth date and birth place of Obama in RDF.

```
ex:barackobama ex:bornIn ex:usa
ex:barackobama ex:birthDate "1961-08-04"
```

Example 4: Example of statements in RDF

Facts and Context

Context is inherent to human knowledge. When humans communicate or reason about facts they know, they don't interpret them devoid of contextual information. Time, location, provenance, and confidence are contexts that usually exist in the knowledge, either explicitly or implicitly. For example, one could say "Barack Obama was the president of United States during 2015". However, if the context of the conversation is about United States during 2015, then the sentence "Barack Obama is the president" would convey the same information. Similarly, using present tense in a conversation could implicitly refer to the present time in a sentence like "Joe Biden is *currently* the president of the United States."

Formally representing this context and be able to reason about it has been a challenge for more than 50 years. As early as 1971, John McCarthy highlighted the importance of formalizing the notion of context to achieve a general AI. According to him, an element to indicate that an axiom was true within a context, but false outside of it, was missing. He did not formalize this element, but his work paved the road to this research topic.

This concept of context is specially important to address when reasoning with the represented knowledge, because new facts derived from the reasoning could not be true in relation with some contexts. For example, imagine that in a knowledge base we have the assertions that both Barack Obama and Donald Trump were presidents of the United States, and that (at any given time) there can only be one president of the United States. If the system can't differentiate between different time contexts, it could reach nonsensical conclusions such as that Barack Obama and Donald Trump are the same person.

In the *Knowledge Representation and Reasoning* field there have been multiple approaches to tackle the problem of context representation. Existing proposals can be generally categorized in two main groups [12]:

- *Compose and Conquer*: This alternative consists in having each context completely separated, possibly even in a different logic system, and having bridge rules that state how formulas in different contexts are related. These bridge rules allow to know how a formula in a context impacts a formula in a different context. For approaches in this group, contexts are not organized in a hierarchy. The works by Ghidini and Giunchiclia are representatives of this kind of approaches [50, 60].

- *Divide and Conquer*: In this alternative the context is a first class citizen. That means that the contexts are reified and used in the statements, and a `is-true` predicate states the context of other elements. This predicate is sometimes used as a modality predicate (with analogous properties as the necessity operator $\Box$). In these approaches, contexts can be contained in other contexts, forming a hierarchy of contexts. There are rules to transcend context (go up in the hierarchy) or to move to an unrelated context. The works by Dinsmore, Buvač, and Guha are representatives of this alternative [36, 19, 64, 17].

These proposals have in common that they maintain (in different manners) each context separate from the others, and implement rules for how knowledge in different contexts is related.

In the Semantic Web community, there has existed an interest to speak about the context of a statement since its inception. This has taken the form of having an additional set of triples that describe statement, and connecting this set with the statement. This connection can be achieved with additional triples or via extending the syntax and/or semantics of RDF. In the W3C recommendation for RDF Schema [15] a vocabulary to model a statement about a statement is presented. Using this vocabulary, the statement is represented by a term, and three properties are used to indicate the subject, predicate, and object of the statement. However, it provides no semantics to actually relate the original triple with the term that represents it. In addition, it is usually considered cumbersome by the community [47]. For these reasons, other approaches were proposed to model statements about statements, to either improve the management of the data, or to be able to give some semantics about the statement.

Numerous proposals have been described to deal with context in the Semantic Web, such as RDF Reification [16, Sec. 5.3], Named Graphs [23], N-Ary relations [110], the Yago model [124], RDF+ [37], Annotated RDF [126], RDF* [67], Singleton Property [108], or Companion Property [47]. These approaches, however, were devised with the idea of just having "statements about statements" that gave additional information about what is stated. The majority do not address reasoning at all, and those that do disregard the implications of reasoning about the statements within contexts.

A number of challenges can be identified with regards to the representation of facts and their context: Reasoning within a context (what will be called *inference preservation* during the thesis), capturing data together with its context, dealing with heterogeneity of context data models, managing contextualized statements in practice (specially scalability and heterogeneity), and dealing with context evolution. These challenges, with the exception of context evolution, will be addressed in this thesis.

Topic of the Thesis

The objective of this thesis is to propose a step forward continuing the research path in context representation, with a focus on the field of Semantic Web.

More specifically, we address the challenge of inference preservation by proposing a novel family of models of representation of context. Also, we address the challenge capturing data together with its context and we propose a method which can be used with any representation model. Finally, we address the challenge of heterogeneity of context data models by proposing an approach to manage contextualized RDF statements in a model-agnostic way.

The thesis statement is defined as follows:

It is possible to capture, represent, and manage contextual information about statements expressed in a formal system and maintain the semantics of the original set of statements without the need to extend the formalisms of the system.

Research Questions

From the thesis statement, we derive the following research questions:

- **R1:** *How to express formally the process needed to add contextual information to a statement or set of statements?* Adding a set of statements to express the context about a statement requires a formal method that, given a set of statements and a context, generates a new set of statements that express the same facts framed under the given context.
- **R2:** *What are the formal properties of a contextualization process with regards to how it affects the semantics of the resulting data?* Given a contextualization process that answers R1, it is important to know what properties characterize the resulting set of statements with regards to how it preserves the semantics of the original set of statements. This allows to compare different contextualization processes and choose one according to the desired properties.
- **R3:** *How to represent contextual information in order to preserve as much semantics as possible from original formalized data?* Once the formal properties of a contextualization process are known, it should be possible to propose a new process that complies with the properties we desire. In our case, those properties that allow inferences as close as possible to the inferences happening in the original set of statements.
- **R4:** *How does a contextualization function affect the efficiency with regards to how the data and their context are stored and queried?* In order to add information about the context, each contextualization process creates a different number of statements in the resulting set. It should be possible, then, to compare the number of statements and their formal properties, and try to find the most efficient contextualization function with the desired properties.
- **R5:** *How to capture existing contextual information that exists implicitly in non-formalized data when transforming it into formal statements?* Most data is not published as formal statements, but in different serializations with different degrees of structure. There exist multiples approaches to capture this data and transform it into formal statements. However, little is done to capture context existing within the data.
- **R6:** *How to efficiently manage contextualized data, independently of the concrete representation used to model it?* There exists a number of approaches to represent the context of statements. However, there is little research about managing in practice the data generated using a specific approach and, to the best of author's knowledge, no common approach to deal in practice with contextualized statements independently of the model use to represent the context.

Contributions

The contributions presented in this thesis with regards to the thesis statement and research questions are the following:

- **C1**: The definition and formalization of *Contextualization Function*: a function that maps a set of statements in a formal system and a context to a set of statements in the same formal system. We use the proposed formalism to express existing reification models in the Semantic Web as contextualization functions. This contribution corresponds to the research question **R1**.

- **C2:** The formal definition of properties of a contextualization function with regards to how it preserves the semantics of the original set of statements. We use these properties to compare the contextualization functions described in *C1*. This contribution corresponds to the research question **R2**.
- **C3:** The family of contextualization functions *Nd**. Each of these functions contextualizes a different subset of terms in the original set of statements (individuals, classes, or properties, in the case of the Semantic Web, for instance). Each contextualization function in *Nd** provides a compromise between how they preserve the semantics and the number of additional terms and statements needed to do it. We compare some of these functions against the contextualization functions of existing models in the Semantic Web, and show that they preserve the semantics of the original data better than any alternative. This contribution corresponds to the research questions **R3**.
- **C4:** Generation and publication of large data sets containing explicit contextual information from non-formalized data with implicit context. This contribution corresponds to the research questions **R4** and **R5**.
- **C5:** A generic approach to capture contextual information from relational tables and transform it into RDF contextualized statements. This contribution corresponds to the research questions **R5**.
- **C6:** An evaluation of how different contextualization approaches affect the data compression and data retrieval using HDT, a well-known binary serialization format with query capabilities for RDF. This contribution corresponds to the research question **R4**.
- **C7:** HDTr, a binary serialization of RDF that extends HDT to allow to manage contextual data in an efficient way, compatible with the majority of the existing contextualization approaches. This contribution corresponds to the research question **R6**.

Organization of the Thesis

The dissertation is presented as a collection of publications. Each chapter, unless otherwise stated, corresponds to a publication. We have updated these chapters homogenizing the writing to American English, homogenizing notation, and correcting typos and references in a transparent manner. We have rewritten the abstract and added footnotes when necessary to put the work in the context of the thesis to help the reader. Original footnotes from the paper will be shown as numeric footnotes, while added footnotes will use Latin letters. Chapters 4 and 6 present extended content with regards to existing publications. Chapters 1 and 9 are original content presented for the first time in this dissertation. Down below we describe the contents of each part and their chapters:

Part I: Formalizing the Context of Statements. This is the core part of the thesis, where we address the first three research questions **R1**, **R2**, and **R3**, and provide the contributions **C1**, **C2**, and **C3**. We use a top-down, step by step, approach from First Order Logic, to Description Logic, to Semantic Web. Each of these steps is covered in the following chapters:

- *Chapter 1: Contextualization of Axioms in First Order Logic* explores the three research questions within First Order Logic and proposes the three aforementioned contributions. We formalize the concept of contextualization function and its properties in First Order Logic. We observe that there are different possibilities to encode the context of a statement, and propose three approaches accordingly. We present one contextualization function that introduces new predicates with higher arity, and two others that rename a subset of symbols, namely the predicates and the constants. This two last approaches can be generalized as a family of contextualization functions that we call *Nd**.
- *Chapter 2: Integrating Context of Statements within Description Logics* performs a similar exploration within Description Logics, and examines how the contributions of the previous chapter can be applied. We see that the renaming approach is applicable, and present *NdTerms*, that can be seen as a instantiation of Nd*.
- *Chapter 3: NdFluents: An Ontology for Annotated Statements with Inference Preservation* presents an instantiation of Nd* that contextualizes the individuals in RDF, and compares against other existing approaches in the Semantic Web.
- *Chapter 4: NdProperties: Encoding Contexts in RDF Predicates with Inference Preservation* gives the formal definition of contextualization function and its properties in RDF, presents an instantiation of Nd* that contextualizes the predicates in RDF, and compares it against other existing approaches in the Semantic Web.

Part II: Capturing the Context of Statements. This part addresses research question **R5** focusing on the Semantic Web world. The first three chapters correspond with work done during the project under which the thesis is framed. In the first two, we generate explicit contextual information about facts from data where this information was implicit, which lead to the realization that we need a formal way of expressing this information as statements about statements. In the third one, we generate a lot of contextual statements about statements of an existing dataset, which lead to the realization of the need of general approaches to perform this transformation. The final chapter is a first step towards such general approach.

- *Chapter 5: Assessing Trust with PageRank in the Web of Data* examines how Linked Open Data datasets make use of other LOD datasets by extracting their links and calculating their PageRank score.
- *Chapter 6: What does Dataset Reuse tell us about Quality?* analyzes how people make use of Linked Open Data datasets and compares it against quality metrics of the datasets, trying to see if there is any correlation.
- *Chapter 7: NELL2RDF: Reading the Web, Tracking the Provenance, and Publishing It as Linked Data* presents how the NELL dataset (a dataset containing hundreds of millions of statements about general knowledge, learned by iteratively reading the web) is transformed into RDF with all the metadata about how the statements were obtained, refined, and given a confidence score using different contextualization functions presented in Part I.
- *Chapter 8: Towards Capturing Contextual Semantic Information About Statements in Web Tables* describes different characteristics of tables containing contextual information about data, and presents general semi-automatic approach to extract it into RDF using different contextualization functions described in Part I.

Part III: Managing the Context of Statements. This part addresses research question **R6**. It contains the following chapter:

- *Chapter 9: HDTr: Managing Reified Triples in Compressed Space* presents HDTr, a binary serialization to encode RDF statements that use a contextualization function to represent context. It extends HDT, and RDF serialization that makes use of succinct data structures to compress the data while allowing to query for triple patterns.

Part IV: Conclusions and Afterword. This part presents a summary of the contributions of the thesis, outlines current and future lines of research that continue the work started with this thesis, and gives some final words.

References

[12] Bouquet, P., Ghidini, C., Giunchiglia, F., Blanzieri, E.: Theories and Uses of Context in Knowledge Representation and Reasoning. Journal of Pragmatics 35(3), 455–484 (2003).

[15] Brickley, D., Guha, R.: RDF Schema 1.1, pp. 91–122. W3C (2014). DOI: 10.1016/B978-0-12-373556-0.00006-X URL: http://www.w3.org/TR/rdf-schema/

[16] Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C (2004). URL: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[17] Buvač, S.: Quantificational Logic of Context. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, August 4-8, 1996, Volume 1. Pp. 600–606 (1996).

[19] Buvač, S., Mason, I.A.: Propositional Logic of Context. In: Proceedings of the 11th National Conference on Artificial Intelligence. Washington, DC, USA, July 11-15, 1993. Pp. 412–419 (1993).

[23] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. Journal of Web Semantics 3(4), 247–267 (2005).

[36] Dinsmore, J.: *Partitioned representations*. Springer (1991).

[37] Dividino, R., Sizov, S., Staab, S., Schueler, B.: Querying for provenance, trust, uncertainty and other meta knowledge in RDF. Journal of Web Semantics 7(3), 204–219 (2009).

[47] Frey, J., Müller, K., Hellmann, S., Rahm, E., Vidal, M.-E.: Evaluation of metadata representations in RDF stores. Semantic Web 10(2), 205–229 (2017).

[50] Ghidini, C., Giunchiglia, F.: Local Models Semantics, or contextual reasoning=Locality+Compatibility. Artificial Intelligence 127(2), 221–259 (2001).

[60] Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics or: How we can do without modal logics. Artificial Intelligence 65(1), 29–70 (1994).

[64] THESIS

[67] Hartig, O.: Foundations of RDF* and SPARQL* (An Alternative Approach to Statement-Level Metadata in RDF). In: Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017. (2017).

[108] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. In: Proceedings of the 23rd International Conference on the World Wide Web (WWW), pp. 759–770. ACM (2014).

[110] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web, pp. 1–1. W3C (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/

[124] Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet. J. Web Sem. 6(3), 203–217 (2008).

[126] Udrea, O., Recupero, D.R., Subrahmanian, V.S.: Annotated RDF. ACM Transaction on Computational Logics 11(2), 10:1–10:41 (2010).

Part I

Formalizing the Context of Statements

Preamble to Formalizing the Context of Statements

The goal of this part is two-fold. First, we want to validate the hypothesis that it is possible to add contextual information to a set of statements without extending the formalism of the system. Second, if it is in fact possible, we want to know how to formally describe the process by which this contextual information can be added, and which properties this process and the resulting data will have.

We address these questions by providing a formal definition for contextualization function—a function that maps a set of statements in a logic system and a context to a new set of statement in the same logic system—and proposing formal properties that a contextualization function can have. These properties are related to how well the contextualization function preserves the semantics of the original data, if it is possible to add new data to a contextualized set of statements, and how the contextualization function is able to maintain different contexts separated. Equipped with these tools, we can compare existing solutions to represent context (now expressed as contextualization functions), and aim to propose new approaches that have whatever properties that are considered desirable. In our case, we are interested in contextualization functions that preserve the semantics, maintain the contexts separated, and in which new data can be added.

Chapter 1, "Contextualization of Axioms in First Order Logic" deals with contextualization functions and their properties in First Order Logic. We give the definitions of contextualization function and its properties for First Order Logic. Then we present three contextualization functions. The first version consists in, for each predicate with arity n, creating a new predicate with arity $n + 1$, where the new element represents the context. The second version consists in renaming predicates; that is, creating a new predicate that is specific to the context and using it in place of the original predicate. The third version is based on renaming constants, with similar meaning. We describe how these functions comply with our desired properties in First Order Logic with and without equality. The last two versions can be seen as two implementations of a family of contextualization functions that contextualize a chosen subset of terms. We call this family *Nd**. This work validates our hypothesis that it is possible to express and formalize a contextualization without extending the formalisms of First Order Logic. We also show that there exists a compromise between the properties a contextualization functions has and other factors such as complexity or number of terms or number of formulas that need to be added to the resulting set of statements.

Chapter 2, "Integrating Context of Statements within Description Logics" studies how the contextualization functions presented in Chapter 1 can be applied to Description Logics and the properties they can have. We give the definitions for contextualization function and its properties for Description Logics. Then we present NdTerms an implementation of Nd* in Description Logics that renames all the terms in the set of statements. We verify that NdTerms complies with our desired properties as long as the set of statements satisfies some basic conditions. We also start

delving into mixing sets of statements in different contexts. Finally, we compare the proposed conceptualization functions against existing approaches in the Semantic Web paradigm expressed in Description Logics. We show that, again, it is possible to annotate sets of statements in Description Logics and preserve their semantics, without the need to extend the syntax or formal semantics. We also show that our proposed contextualization function satisfies our desired properties better than any existing alternative in the Semantic Web community.

Chapter 3, "NdFluents: An Ontology for Annotated Statements with Inference Preservation" presents *NdFluents*, an ontology to implement in OWL an instantiation of Nd* that contextualizes the individuals existing in the set of statements. This approach is analogous to the contextualization of constants in First Order Logic presented in Chapter 1, and could be considered a subset of NdTerms presented in Chapter 2. We also present design patterns representing sets of statements under more than one context simultaneously. In this case, we measure how NdFluents preserves the semantics against other existing approaches by comparing how well they follow inference rules of RDFS and OWL. We show that NdFluents satisfy more rules than any other approach, while being able to combine and differentiate multiple contexts.

Chapter 4, "NdProperties: Encoding Contexts in RDF Predicates with Inference Preservation" provides the formal definition of contextualization function in RDF and its properties, and presents *NdProperties*, a new contextualization function and its associated ontology to add context to a set of statements by renaming RDF predicates. This approach is another instance of Nd*, analogous to the renaming of predicates in First Order Logic presented in Chapter 1. We compare NdProperties against other approaches (including NdFluents) by, once again, measuring how they comply with inference rules in RDFS and OWL. The results of this comparison show that NdProperties satisfies more properties than existing approaches. Against NdFluents, the results are less favorable to NdProperties, but we note that NdProperties needs to create less fresh resources and the resulting design patterns look simpler. This emphasizes our previous observation that there exist a compromise between how much of the semantics are preserved and the size and complexity of the generated data.

All in all, this part tackles with Research Questions **R1:** *How to express formally the process needed to add contextual information to a statement or set of statements?*, **R2:** *What are the formal properties of a contextualization process with regards to how it affects the semantics of the resulting data?*, and **R3:** *How to represent contextual information in order to preserve as much semantics as possible from original formalized data?*. It corresponds to contributions **C1**: *the definition and formalization of* Contextualization Function, **C2**: *the formal definition of properties of a contextualization function*, and **C3**: *the family of contextualization functions* Nd*.

Chapter 1

Contextualization of Axioms in First Order Logic*

In this chapter, we commence our path on representing context about statements in the most high-level logic system analyzed in this thesis: First Order Logic. It is the first step in a top-down approach that will include Description Logics and OWL in the following chapters.

We base our definition of what expressing the context of a statement means in the proposal by McCarthy [99]. That is, given a formula ϕ and a context c, one could express that ϕ is true in c, written $ist(c, \phi)$.

Many existing approaches to formalize the context in First Order Logic do it by extending the semantics to a non-standard logic formalism. Here, we propose a different approach where $ist(c, \phi)$ is interpreted as a pure FOL formula through a *contextualization function*, and define several properties that such an interpretation should have. We first introduce and study two—in principle—simple functions, and investigate possible modifications to make them comply with the defined properties. We show that when equality is added to the logic, many additional axioms have to be added in order to maintain the desirable properties. We then provide an alternative solution based on the idea of fluents that, while more complex, works without change in both first order logic with and without equality.

1.1 Introduction

In his famous work on generality in artificial intelligence, John McCarthy suggested that the notion of context should be formalized (among other things) in order to achieve general AI. He argued that any logical formula ϕ that describes a reality (other than a tautology) can be said to be false when taken out of the context in which it was asserted. Consequently, logic was missing a construct by which one could say that ϕ is true in a context c, written $ist(c, \phi)$. Unfortunately, McCarthy never fully formalized the meaning of the expression $ist(c, \phi)$. However, a number of proposals were made to extend the semantics of first order logic (FOL) to account for such construct [18, 2, 17, 109].

Here, we propose a different approach where $ist(c, \phi)$ is interpreted as a pure first order logic formula. This amounts to finding a function f that maps each pair (c, ϕ) to a FOL formula. We first investigate the properties that such a function should have in order to meaningfully represent the contextual knowledge. Then we successively introduce three concrete functions for representing a FOL formula ϕ in a context c, and study their properties. In a nutshell, the first one consists in adding the

*This chapter is based on unpublished content by Antoine Zimmermann and José M. Giménez-García in 2018.

context as an argument; that is, making each n-ary predicate in ϕ a $n+1$-ary predicate. The second one consists in renaming all predicates, such that predicate symbols in different contexts do not overlap. While these approaches are straightforward and have all desired properties in FOL without equality, they need to be refined when dealing with equality. The third approach consists in renaming constants to separate terms from different contexts. This requires the use of relativization, a technique for ensuring that formulas in a context only describe the elements in the universe of the context. While this approach is more complicated than the other two without equality, it has the advantage of being applicable directly to FOL with equality.

As a running example, let us imagine that we have a set of formulas where we want to represent countries in the world. However, what a country is is not a universal truth, and can depend on the context of the speaker. For instance, let's assume that Palestine is considered a country by the European Union, but not by the United States (which would be expressed using the McCarthian notation as $ist(c_{EU}, Country(palestine))$ and $ist(c_{US}, \neg Country(palestine))$). This could be expressed using the described contextualization functions as follows:

1. $n+1$**-ary:** $Country(palestine, c_{EU})$ and $\neg Country(palestine, c_{US})$
2. **Renaming predicates:** $Country_{EU}(palestine)$ and $\neg Country_{US}(palestine)$
3. **Renaming constants:** $Country(palestine_{EU})$ and $\neg Country(palestine_{US})$

In the end, we relate this contribution to previous work, including our own preliminary results on this problem in the framework of description logics[a], and provide some conclusions where we summarize and discuss the results of this work.

1.2 Contextualization

In this section, we present and formalize the concept of contextualization in first order logic and provide some properties that are desirable for such a concept. But before this, we introduce the notations we will be using throughout the chapter[b]. We assume there is an infinite set $\mathcal{C}$ of contexts. We refer to First Order Logic as FOL and FOL with equality as FOL$^=$. We write $\mathcal{FOLF}$ (resp. $\mathcal{FOLF}_{\mathcal{EQ}}$) for the set of all FOL formulas (resp. FOL$^=$ formulas). The usual symbols $\wedge$, $\vee$, $\neg$, $\exists$, $\forall$ are used in combination with the Latin alphabet for predicates, constants and function symbols. We use Greek letters to denote FOL or FOL$^=$ formulas. The symbol $\doteq$ is used to denote the equality symbol of FOL$^=$. We write $\phi \models \psi$ to mean that ϕ entails ψ (*i.e.*, all models of ϕ are models of ψ). We write the set of all function symbols (resp. constants, predicates) appearing in a formula ϕ as $Func(\phi)$ (resp. $Const(\phi)$, $Pred(\phi)$). Finally, we use a vectorial notation $\vec{x}$ to denote a tuple of variable $x_1, \ldots, x_n$ for some n.

Definition 1.1[c]**.** *We call a* contextualization function *in FOL (resp. FOL$^=$) any function* $f : \mathcal{C} \times \mathcal{FOLF} \to \mathcal{FOLF}$ *(resp.* $f : \mathcal{C} \times \mathcal{FOLF}_{\mathcal{EQ}} \to \mathcal{FOLF}_{\mathcal{EQ}}$*).*

Given a contextualization function f, a context c and a FOL formula ϕ, $f(c, \phi)$ is supposed to encode the idea that ϕ is true in context c. However, it is clear that not

[a] This work is presented in chapter 2 of this dissertation.

[b] We have replaced mentions to the paper to mentions to the chapter along the dissertation for consistency.

[c] In the original paper different symbols where used for the contextualization function along the paper. In this chapter we have homogenized notation to make it consistent.

all such functions express well this intuition. For instance, a function that assigns the same tautology to all input c and ϕ would be a strange and irrelevant contextualization. To discriminate meaningless contextualization, we introduce some properties that we consider desirable. Note, however, that some relevant contextualization approaches may not have these two properties (see [51, 57]).

First, let's suppose that in our running example, people born in a country have the nationality of that country (*i.e.*, $\phi \equiv$ Born_in(x,y) $\wedge$ Country(y) $\rightarrow$ Citizen(x,y)) for both contexts (EU and US). Then, the contextualization of Palestine being a country for the EU should infer that, indeed, people born in Palestine have Palestinian nationality. On the contrary, this should not be inferred for the context that represents the US. We define *entailment conservation* as a first property to represent this.

Definition 1.2. *A contextualization function f* conserves entailments *iff for all $c \in \mathcal{C}$ and all $\phi, \psi \in \mathcal{FOLF}$ (resp. $\mathcal{FOLF}_{\mathcal{EQ}}$), $\phi \models \psi$ iff $f(c, \phi) \models f(c, \psi)$.*

Note that this would cause that the contextualization of any inconsistent formula would itself be inconsistent. Because this can be sometimes not desirable, and we want to study the case of representing inconsistent data, we need a relaxed version of this property, which we call *consistent entailment conservation*.

Definition 1.3. *A contextualization function f* conserves consistent entailments *iff for all $c \in \mathcal{C}$ and all $\phi, \psi \in \mathcal{FOLF}$ (resp. $\mathcal{FOLF}_{\mathcal{EQ}}$) such that ϕ and ψ are consistent, $\phi \models \psi$ iff $f(c, \phi) \models f(c, \psi)$.*

In general, it is desirable for the contextualization of a consistent theory to be itself consistent (and vice versa), and conversely, the contextualization of an inconsistent theory to be inconsistent (and vice versa). We define this property as *consistency conservation*. Note, however, that in some cases it may be desirable to say that a context contains inconsistent information, and expressing it should be consistent. For those cases, this property would not be desirable.

Definition 1.4. *A contextualization function f* conserves consistencies *if for all ϕ and all c, ϕ is consistent iff $f(c, \phi)$ is consistent.*

These properties are closely related. In fact, entailment conservation can be seen as the combination of the other two.

Theorem 1.1. *A contextualization function f conserves entailments iff it conserves consistent entailments and conserves consistencies.*

Proof. If f conserves entailments, it obviously conserves consistent entailments, and it conserves consistencies. If f conserves consistent entailments and conserves consistencies, then for all consistent ϕ and ψ, $\phi \models \psi$ implies $f(c, \phi) \models f(c, \psi)$ (consistent entailment preservation) and if ϕ is inconsistent, then so is $f(c, \phi)$ (consistency preservation) and so $f(c, \phi) \models f(c, \psi)$. □

Now, imagine that we have the two sets of contextualized formulas representing countries in the world, but we want to add information to represent changes in the world. For instance, Catalonia splits from Spain, and we need to add that formula to the contextualizations. It would be desirable that we can apply the contextualization function to the new formula(s) and add the result to the sets. If the contextualization function allows it, we say it is *separable*.

Definition 1.5. *A contextualization function f is* separable *if for all ϕ, ψ and c, $f(c, \phi) \wedge f(c, \psi) \equiv f(c, \phi \wedge \psi)$.*

Finally, a function that assigns a constant formula to all input would not conserve entailment. However, the function $\mathsf{forget} : (c, \phi) \mapsto \phi$ is an entailment-preserving contextualization function. But, continuing with the running example, trying to represent $\mathsf{forget}(c_{US}, \neg Country(palestine)) \wedge \mathsf{forget}(c_{EU}, Country(palestine))$ would lead to an inconsistent result, with Palestine being and not being a country at the same time. Therefore, we need to introduce an additional property.

Definition 1.6. *A contextualization function f is* context-independent *iff for all $c, d \in \mathcal{C}$ and all $\phi, \psi \in \mathcal{FOLF}$ (resp. $\mathcal{FOLF}_{\mathcal{EQ}}$), if $c \neq d$ and ϕ and ψ are both consistent, then $f(c, \phi) \wedge f(d, \psi)$ is consistent.*

Observe that the function forget previously defined does not possess context independence.

Note that context independence does not mean that there cannot be interactions between the knowledge of different contexts. Indeed, in addition to $ist(c, \phi)$ formulas, there could be plain FOL formulas that assert something about c.

In the following sections, we are interested in investigating contextualization functions that have both context-independence and entailment-conservation.

1.3 The $n+1$-ary approach

In this contextualization function, for all natural numbers $n \geq 0$ and all n-ary predicates P, we introduce a distinct $n+1$-ary predicate $\tilde{P}$. Hence, the result of contextualizing $ist(c_{EU}, Country(palestine))$ will be $Country(palestine, c_{EU})$. We will define the contextualization function for FOL without equality, and we will observe that it has the desired properties. However, when dealing with FOL$^{=}$, we will need to introduce a non-negligible number of axioms to achieve context independence.

1.3.1 FOL without equality

The $n+1$-ary contextualization function consist in a simple transformation over the formulas, where we replace each predicate with a similar one, to which we add an additional argument representing the context. This is expressed formally as follows.

Definition 1.7. *We define the $n+1$-ary* contextualization function *f as a function that maps a FOL formula ϕ and a context c to a formula $f(c, \phi)$ obtained from ϕ by replacing all atoms $P(t_1, \ldots, t_n)$ of arity n in ϕ by $\tilde{P}(t_1, \ldots, t_n, c)$.*

We first see that the $n+1$-ary function is consistency-conserving.

Lemma 1.1. *The function f conserves consistencies.*

Proof sketch. Let us assume that ϕ is consistent. We know that ϕ has a model, but we need to ensure that the model assigns an interpretation to the context c. So we consider a tautology $\chi(c)$ that only contains the constant c and a single predicate appearing in ϕ (if such a predicate does not exist, ϕ is a trivial formula and the property obviously hold). From our assumptions, there exists a model m of $\phi \wedge \chi(c)$. Let us define an interpretation $\tilde{m}$ as follows:

- the universe of $\tilde{m}$ is the same as the one of m;
- for all constant a (including c), $\tilde{m}(a) = m(a)$;
- for all function f, $\tilde{m}(f) = m(f)$;

- for all predicate P of arity $n > 0$ in ϕ, $\tilde{m}(\tilde{P}) = \{(x_1, \ldots, x_n, m(c)) \mid (x_1, \ldots, x_n) \in m(P)\}$.

Using the recursive definition of the satisfaction relation, we check that $\tilde{m}$ satisfies the formula $f(c, \phi)$. □

Lemma 1.1 is especially important for its contraposed version: ϕ is inconsistent implies that $f(c, \phi)$ is inconsistent, and vice versa. A consequence of this is the following proposition.

Proposition 1.1. *The $n+1$-ary contextualization function f conserves entailments in FOL without equality.*

Proof sketch. If ϕ is inconsistent, then $f(c, \phi)$ is inconsistent, because of Lemma 1.1. As a result, any formula χ is entailed by $f(c, \phi)$. In particular, $f(c, \phi) \models f(c, \psi)$. Similarly, if $f(c, \phi)$ is inconsistent, so is ϕ, thus $\phi \models \psi$.

Then, if ϕ is consistent and $\phi \models \psi$, then $\phi \wedge \neg\psi$ is inconsistent. Therefore, by what precedes, $f(c, \phi \wedge \neg\psi)$ is inconsistent as well. By definition of f, $f(c, \phi \wedge \neg\psi) = f(c, \phi) \wedge \neg f(c, \psi)$. Consequently, $f(c, \phi) \models f(c, \psi)$. An analogous reasoning applies to show that if $f(c, \phi) \models f(c, \psi)$, then $\phi \models \psi$. □

Proposition 1.2. *The $n+1$-ary contextualization function f is separable.*

Proof. This is immediate from the definition of f. □

Proposition 1.3. *The $n+1$-ary contextualization function f is context-independent.*

Proof Sketch. Let $c, d \in \mathcal{C}$. Let us assume that ϕ and ψ are in FOL without equality and are consistent, then so are $f(c, \phi)$ and $f(d, \psi)$. Let χ be a tautology that contains all constants, function and predicate symbols of ϕ and ψ, as well as c and d. Then $\phi \wedge \chi$ (resp. $\psi \wedge \chi$) is consistent and there exists a Herbrand model $\mathcal{H}_\phi$ (resp. $\mathcal{H}_\psi$) for it. Let us define a Herbrand interpretation $\mathcal{H}$ on the signature of $f(c, \phi) \wedge f(d, \psi)$ as follows: constants and functions are interpreted as usual in Herbrand structures. For each n-ary predicate P of ϕ or ψ, $(x_1, \ldots, x_n, c) \in \tilde{P}^{\mathcal{H}}$ iff $(x_1, \ldots, x_n) \in P^{\mathcal{H}_\phi}$ and $(x_1, \ldots, x_n, d) \in \tilde{P}^{\mathcal{H}}$ iff $(x_1, \ldots, x_n) \in P^{\mathcal{H}_\psi}$. We then check that $\mathcal{H}$ satisfies both $f(c, \phi)$ and $f(d, \psi)$, which is the case by construction. □

1.3.2 FOL with equality

We extend the $n+1$-ary contextualization function f to a function $f^{\doteq}$ that works on formulas with equality as defined in Definition 1.7. In this case, $f^{\doteq}$ has entailment conservation but does not have context independence. We use the symbol $\doteq$ to denote the equality symbol in formulas of FOL with equality.

Proposition 1.4. *The contextualization function $f^{\doteq}$ conserves entailments in FOL$^{=}$.*

Proof. Trivial from proof in FOL without equality. □

Now we show that the function $f^{\doteq}$ does not have context independence.

Proposition 1.5. *There exists $c, d \in \mathcal{C}$ and $\phi, \psi \in \mathcal{FOLF}_{\mathcal{EQ}}$ such that c and d are different constants, and ϕ and ψ are both consistent but $f^{\doteq}(c, \phi) \wedge f^{\doteq}(\psi, d)$ is inconsistent.*

Proof. Take $a \doteq b$ for ϕ, $A(a) \wedge \neg A(b)$ for ψ, then $f^{\doteq}(c, \phi) \wedge f^{\doteq}(d, \psi)$ is inconsistent. □

The problem comes from not being able to contextualize equality. The solution lies in replacing in the contextualization the equality axioms by a predicate for "contextual equality" eq as follows:

Replace $a \doteq b$ by $\text{eq}(a, b, c)$ and add the following axioms:

- $\forall x \forall c\ \text{eq}(x, x, c)$ (reflexivity);
- $\forall x \forall y \forall c\ \text{eq}(x, y, c) \rightarrow \text{eq}(y, x, c)$ (symmetry)
- $\forall x \forall y \forall z \forall c\ \text{eq}(x, y, c) \wedge \text{eq}(y, z, c) \rightarrow \text{eq}(x, z, c)$ (transitivity).

For all predicates P of arity n that appear in ϕ, and for all $1 \leq i \leq n$, add:

$$\forall c \forall y \forall \vec{\mathbf{x}}(\text{eq}(y, x_i, c) \wedge \tilde{P}(\vec{\mathbf{x}}, c)) \rightarrow \tilde{P}(x_1, \ldots, y, \ldots, x_n, c) \tag{1.1}^{\text{d}}$$

For all function f of arity n that appear in ϕ, and for all $1 \leq i \leq n$, add:

$$\forall c \forall y \forall \vec{\mathbf{x}}(\text{eq}(y, x_i, c) \rightarrow \text{eq}(f(x_1, \ldots, y, \ldots, x_n), f(\vec{x}), c)) \tag{1.2}^{\text{d}}$$

We denote the new function f^{eq}. With this new construction, we regain context independence. However, we need to prove that this new contextualization is still entailment-conserving.

Proposition 1.6. *The function f^{eq} conserves entailments.*

Proof sketch. We first prove that ϕ is consistent iff $f^{\text{eq}}(c, \phi)$ is consistent. Let ϕ be a consistent formula. It has a model M with a universe $\mathcal{U}^M$. Using the same trick as before, we can ensure that the model is interpreting the context c. We define an interpretation $\tilde{M}$ of the vocabulary of $f^{\text{eq}}(c, \phi)$ as follows:

- $\mathcal{U}^{\tilde{M}} = \mathcal{U}^M$;
- for all constant a, $\tilde{M}(a) = M(a)$;
- for all function f, $\tilde{M}(f) = M(f)$;
- for all n-ary predicate P, $\tilde{M}(\tilde{P}) = \{(x_1, \ldots, x_n, M(c)) \mid (x_1, \ldots, x_n) \in M(P)\}$;
- $\tilde{M}(\text{eq}) = \{(x, x, M(c)) \mid x \in \mathcal{U}^M\}$.

It can be shown by structural induction over the formula ϕ that $\tilde{M}$ is a model of $f^{\text{eq}}(c, \phi)$. Conversely, if $f^{\text{eq}}(c, \phi)$ is consistent, there exists a model m of it. The axioms of contextual equality are such that the relation $\{(x, y) \mid (x, y, m(c)) \in m(\text{eq})\}$ must be an equivalence relation over the universe $\mathcal{U}^m$. Therefore, for any element $x \in \mathcal{U}^m$, we can define the equivalence class $[\![x]\!]$ for this relation. We then build an interpretation $\bar{m}$ of the vocabulary of ϕ as follows:

- $\mathcal{U}^{\bar{m}} = \{[\![x]\!] \mid x \in \mathcal{U}^m\}$;
- for all constant a, $\bar{m}(a) = [\![m(a)]\!]$;
- for all n-ary function f and all $[\![x_1]\!], \ldots, [\![x_n]\!]$, $\bar{m}(f)([\![x_1]\!], \ldots, [\![x_n]\!]) = [\![m(f)(x_1, \ldots, x_n, m(c))]\!]$;
- for all n-ary predicate P, $\bar{m}(P) = \{([\![x_1]\!], \ldots, [\![x_n]\!]) \mid (x_1, \ldots, x_n, m(c)) \in m(\tilde{P})\}$.

[d] Where y appears in the i^{th} position of $\tilde{P}$.

Proving that $\bar{m}$ is a model of ϕ is more complicated. First, because of the axioms of contextual equality, the interpretation is well defined. Then, checking that this interpretation satisfies ϕ requires a bit of work. We assume that the formula is in CNF and proceed recursively. We only show the proof for $\phi = \neg P(a)$, with a a constant, to convince that it works. We must show that $\bar{m}(a) \in \mathcal{U}^{\bar{m}} \setminus \bar{m}(P)$. This amounts to showing that all values in the equivalence class of $m(a)$ are in $\mathcal{U}^m \setminus m(P)$. If such was not the case, then predicate P would be true for an element x "contextually equal" to an element y for which P is not true. This would violate Equation 1.1. □

Proposition 1.7. *The function f^{eq} is context-independent.*

Proof. If ϕ and ψ are consistent, so are $f^{\text{eq}}(c, \phi)$ and $f^{\text{eq}}(d, \psi)$. Note that neither $f^{\text{eq}}(c, \phi)$ nor $f^{\text{eq}}(d, \psi)$ use equality, so there exist Herbrand models for them. We can then prove the property in a way analogous to Proposition 1.3. □

With this new contextualization function, we have to add $3 + \Sigma_{P \in Pred(\phi)} arity(P) + \Sigma_{f \in Func(\phi)} arity(f)$ formulas and many variables, which may make reasoning, even on a simple subset of FOL, complicated. In fact, this technique is similar to adding the axioms of equality to a formula in order to reason in FOL$^=$ using the algorithms of FOL without equality. This is known to be very inefficient, as noted by, e.g., Beckert [4]. Thus, this contextualization approach may require specialized reasoning machinery to handle contextual equality properly.

1.4 Renaming predicates

Another approach to contextualizing formulas consists in using different sets of names (predicates, constants or functions) when asserting statements in different contexts. We first consider an approach that consists in renaming predicates in ϕ to predicates that are specific to the context c. Continuing with our running example, the result of the contextualization function of Palestine being a country for the European Union will be $Country_{EU}(palestine)$, while the opposite for the US will be $\neg Country_{US}(palestine)$. We will see that, again, this contextualization has both entailment conservation and context independence when applied within FOL without equality. However, as soon as equality is added, the associated contextualization function loses context independence. In order to regain context independence, we will need to follow a similar approach as with the n+1-ary contextualization.

Definition 1.8. *Let us assume an injective function ren_p that takes a context and a predicate as parameter and generates a predicate of equal arity. That is, for all contexts $c \in \mathcal{C}$ and all predicates P, $\mathsf{ren}_p(c, P)$ is a predicate of same arity as P and for all c' and P', $\mathsf{ren}_p(c, P) = \mathsf{ren}_p(c', P')$ iff $c = c'$ and $P = P'$. We extend the function ren_p to any FOL$^=$ formulas by defining $\mathsf{ren}_p(c, \phi)$ as the formula ϕ with all predicates P appearing in ϕ renamed to $\mathsf{ren}_p(c, P)$. Then ren_p is a contextualization function.*

Proposition 1.8. *The function ren_p conserves entailments.*

Proof. Since ren_p simply consists of an injective renaming of predicates, it does not affect the meaning of the formulas. Any model of ϕ can be transformed into a model of $\mathsf{ren}_p(c, \phi)$, and vice-versa, by simply changing the mapping from predicates into an analogous mapping from renamed predicates. □

Proposition 1.9. *The function ren_p is context-independent in FOL, but not in FOL$^=$.*

Proof. Let $c, d \in \mathcal{C}$. Let us assume that ϕ and ψ are in FOL without equality and are consistent, then so are $\mathsf{ren}_p(c, \phi)$ and $\mathsf{ren}(d, \psi)$. Let χ_d (resp. χ_c) be a tautology that contains all constants and function symbols of $\mathsf{ren}_p(d, \psi)$ (resp. $\mathsf{ren}_p(c, \phi)$). Then $\mathsf{ren}_p(c, \phi) \wedge \chi_d$ (resp. $\mathsf{ren}(d, \psi) \wedge \chi_c$) is consistent and there exists a Herbrand model H_c (resp. H_d) for it. H_c and H_d interpret the constants and functions of $\mathsf{ren}_p(c, \phi)$ and $\mathsf{ren}_p(d, \psi)$ in the same way, by definition of a Herbrand model. Moreover, the predicates of $\mathsf{ren}_p(c, \phi)$ are disjoint from those of $\mathsf{ren}_p(d, \psi)$. Therefore, $H_c \cup H_d$ is a valid interpretation for the signature of $\mathsf{ren}_p(c, \phi) \wedge \mathsf{ren}_p(d, \psi)$ and satisfies it by construction. This proves the first part of the proposition.

Herbrand models are only guaranteed to exist on formulas without equality. Consider the following formulas in FOL$^{=}$: $\phi = \forall x \forall y . x \doteq y$ and $\psi = \forall x \exists y . \neg(x \doteq y)$. There is no predicate to rename in either ϕ or ψ, so $\mathsf{ren}_p(c, \phi) \wedge \mathsf{ren}_p(d, \psi) = \phi \wedge \psi$. This formula is clearly inconsistent. □

In order to address context dependence with equality, we introduce *contextual equality* $\doteq_c$, a binary predicate that asserts that two terms denote the same thing. In order to ensure this predicate is interpreted appropriately the contextualization function has to add axioms analogous to those introduced before with the predicate eq, that is:

Replace $a \doteq b$ by $a \doteq_c b$ and add the following axioms:

- $\forall x \; x \doteq_c x$ (reflexivity);
- $\forall x \forall y \; x \doteq_c y \rightarrow y \doteq_c x$ (symmetry)
- $\forall x \forall y \forall z \; x \doteq_c y \wedge y \doteq_c z \rightarrow x \doteq_c z$ (transitivity).

For all predicates P of arity n that appear in ϕ, and for all $1 \leq i \leq n$, add:

$$\forall y \forall \vec{\mathbf{x}} (y \doteq_c x_i) \wedge \tilde{P}(\vec{\mathbf{x}}, c)) \rightarrow \tilde{P}(x_1, \ldots, y, \ldots, x_n, c) \tag{1.3}^{e}$$

For all function f of arity n that appear in ϕ, and for all $1 \leq i \leq n$, add:

$$\forall y \forall \vec{\mathbf{x}} (y \doteq_c x_i) \rightarrow (f(x_1, \ldots, y, \ldots, x_n) \doteq_c f(\vec{x})) \tag{1.4}^{e}$$

Proposition 1.10. *With contextual equality,* ren *is context independent.*

While the addition of the axioms of contextual equality adds complexity to the resulting formula, this approach also has an important drawback: The only way to identify the context from the resulting formula is by knowing the renaming function. The connection with the context c cannot be made explicit.

1.5 Renaming constants

This approach is based on the idea of fluents as modeled in Welty & Fikes [131]. In their work, entities that changes over time can be described as different constants that denote the entity at different time periods. For instance, there may be a constant identifying the fluent Chris Welty, while there are other constants denoting Welty as a child and Welty as an adult. This approach was later generalized to any kind of contexts in [130, 57]: each entity may correspond to many constants denoting the "contextual part" of the entity in a given context. In practice, adopting this approach

[e] Where y appears in the i^{th} position of $\tilde{P}$

consists in renaming all constants such that there are disjoint sets of constants associated with different contexts. We will see that this approach has entailment conservation, but is not context-independent. In order to attain context independence, we will need to resort to a technique called *relativization*. This will be discussed in the following section.

Definition 1.9. *Let us assume an injective function* ren *that takes a context and a constant as parameter and generates a constant. That is, for all context $c \in \mathcal{C}$ and all constant a,* $\mathsf{ren}(c,a)$ *is a constant and for all c' and a',* $\mathsf{ren}(c,a) = \mathsf{ren}(c',a')$ *iff $a = a'$ and $c = c'$. We extend the function* ren *to any FOL$^=$ formulas by defining* $\mathsf{ren}(c,\phi)$ *as the formula ϕ with all constants a appearing in ϕ renamed to* $\mathsf{ren}(c,a)$*. Then* ren *is a contextualization function.*

In our running example, the fact that Palestine is a country for the EU could be modeled as $Country(palestine_{EU})$, which we could interpret as "Palestine, as referred by the EU, is a country". Conversely, the fact that it is not a country for the US could be represented as $\neg Country(palestine_{US})$, with a similar interpretation.

Proposition 1.11. *The function* ren *conserves entailments.*

Proof. Since ren simply consists of an injective renaming of constants, it does not affect the meaning of the formulas. Any model of ϕ can be transformed into models of $\mathsf{ren}(c,\phi)$, and vice-versa, by simply changing the mapping from constants to an analogous mapping from renamed constants. □

Proposition 1.12. *The function* ren *is not context-independent.*

Proof. Let c and d be two contexts. The formulas $\phi = \forall x.P(x)$ and $\psi = \forall x \neg P(x)$ are both consistent but $\mathsf{ren}(c,\phi) \wedge \mathsf{ren}(d,\psi)$ is inconsistent. □

As stated before, context independence can be achieved in this approach by using the notion of relativization, which we present in the next section.

1.6 Relativization

In short, relativization consists in ensuring that all assertions in a context are only expressing truth about the entities that exist in the context. For each context c, we need to introduce a unary predicate $\top_c$ that indicates that something exists in the context c. Continuing with our running example, $ist(c_{EU}, Country(palestine))$ and $ist(c_{US}, \neg Country(palestine))$ would be represented as $Country(palestine_{EU}) \wedge \top_{EU}(palestine_{EU})$ and $\neg Country(palestine_{US}) \wedge \top_{US}(palestine_{US})$ respectively. We will see that extending the renaming of constants with relativization conserves consistent entailments, is separable, and context independent.

Definition 1.10[f]. *Given P an n-ary predicate, a context c and a formula ϕ, we define a function* Rel *recursively as follows. We assume ϕ to be in CNF.*

- *if* $\phi = P(t_1,\ldots,t_n)$*, then* $\mathsf{Rel}(c,\phi) = P(\mathsf{ren}(c,t_1),\ldots,\mathsf{ren}(c,t_n)) \wedge \top_c(\mathsf{ren}(c,t_1)) \wedge \cdots \wedge \top_c(\mathsf{ren}(c,t_n)) \wedge \bigwedge_{f\in Func(\phi)} \forall \vec{x} \top_c(x_1) \wedge \cdots \wedge \top_c(x_n) \rightarrow \top(f(\vec{x}))$*;*
- *if* $\phi = \neg P(t_1,\ldots,t_n)$*, then* $\mathsf{Rel}(c,\phi) = \neg P(\mathsf{ren}(c,t_1),\ldots,\mathsf{ren}(c,t_n)) \wedge \top_c(\mathsf{ren}(c,t_1)) \wedge \cdots \wedge \top_c(\mathsf{ren}(c,t_n)) \wedge \bigwedge_{f\in Func(\phi)} \forall \vec{x} \top_c(x_1) \wedge \cdots \wedge \top_c(x_n) \rightarrow \top_c(f(\vec{x}))$*;*

[f]The definition has been updated to include *"P an n-ary predicate"* and fixing a typo where $\top$ appeared instead of $\top_c$.

- *if* $\phi = t \doteq t'$, *then* $\mathsf{Rel}(c, \phi) = \mathsf{ren}(c, t) \doteq \mathsf{ren}(c, t') \wedge \top_c(\mathsf{ren}(c, t)) \wedge \top_c(\mathsf{ren}(c, t')) \wedge \bigwedge_{f \in Func(\phi)} \forall \vec{x} \top_c(x_1) \wedge \cdots \wedge \top_c(x_n) \rightarrow \top_c(f(\vec{x}))$;
- *if* $\phi = \phi' \wedge \phi''$, *then* $\mathsf{Rel}(c, \phi) = \mathsf{Rel}(c, \phi') \wedge \mathsf{Rel}(c, \phi'')$;
- *if* $\phi = \phi' \vee \phi''$, *then* $\mathsf{Rel}(c, \phi) = \mathsf{Rel}(c, \phi') \vee \mathsf{Rel}(c, \phi'')$;
- *if* $\phi = \forall x \phi'$, *then* $\mathsf{Rel}(c, \phi) = \forall x \top_c(x) \rightarrow \mathsf{Rel}(c, \phi)$;
- *if* $\phi = \exists x \phi'$, *then* $\mathsf{Rel}(c, \phi) = \exists x \top_c(x) \wedge \mathsf{Rel}(c, \phi)$.

If a formula is not in CNF, we assume that Rel consists in normalizing the formula before applying the definition above. Then the function Rel is a contextualization function. Note that it is important that the unary predicates $\top_c$ be distinct from all predicates used in ϕ, otherwise the following proposition would not hold.

Proposition 1.13. *The function* Rel *conserves consistent entailments but does not conserves all entailments.*

Proof. Consider $\phi = \forall x \exists y P(x) \wedge \neg P(y)$. Of course, ϕ is inconsistent. However, $\mathsf{Rel}(c, \phi) = \forall x \top_c(x) \rightarrow (\exists y \top_c(y) \wedge P(x) \wedge \neg P(y))$ is consistent, as it suffices that $\top_c$ be interpreted as the empty set.

Let ϕ and ψ be two consistent formulas such that $\phi \models \psi$. From a model M of ϕ, we build an interpretation $\tilde{M}$ of the vocabulary of $\mathsf{Rel}(c, \phi)$ as follows:

- for all constants a $\tilde{M}(\mathsf{ren}(c, a)) = M(a)$;
- for all function symbols f $\tilde{M}(f) = M(f)$;
- for all predicates P $\tilde{M}(P) = M(P)$;
- $\tilde{M}(\top_c) = \mathcal{U}^M$.

Due to the fact that the predicate $\top_c$ is interpreted as the whole universe of $\tilde{M}$, it is easy to check that the relativized formulas are satisfied. This can be done by structural induction on the formula ϕ. □

We may consider that $\top_c$ represents a "local universe of interpretation". However, as opposed to a normal universe of interpretation, the interpretation of $\top_c$ may be empty. The function could be updated to a function $\mathsf{Rel}^{\exists}$ that includes the subformula $\exists x \top_c(x)$. Then for an inconsistent ϕ, we have that $\mathsf{Rel}^{\exists}(c, \phi)$ is inconsistent.

Proposition 1.14. *The function* Rel *is separable.*

Proof. This follows immediately from the definition. □

Proposition 1.15. *The function* Rel *is context independent in FOL*$^{=}$.

Proof. Let c and d be two distinct contexts, and ϕ and ψ two consistent formulas. There exist two models M and M' of ϕ and ψ respectively. Without loss of generality, we can assume that the universes of M and M' are disjoint. We build a model $\tilde{M}$ of $\mathsf{Rel}(c, \phi) \wedge \mathsf{Rel}(d, \psi)$ as follows:

- $\mathcal{U}^{\tilde{M}} = \mathcal{U}^M \cup \mathcal{U}^{M'}$;
- for all constants a appearing in ϕ, $\tilde{M}(\mathsf{ren}(c, a)) = M(a)$;
- for all constants a appearing in ψ, $\tilde{M}(\mathsf{ren}(d, a)) = M'(a)$;

- for all n-ary function symbols f appearing in ϕ but not in ψ, then we take an arbitrary element $e \in \mathcal{U}^{M'}$ and $\tilde{M}(f) = M(f) \cup \{(\vec{x}, e) \mid \vec{x} \in (\mathcal{U}^{M'})^n\}$;[g]
- for all n-ary function symbols f appearing in ψ but not in ϕ, then we take an arbitrary element $e \in \mathcal{U}^{M}$ and $\tilde{M}(f) = M'(f) \cup \{(\vec{x}, e) \mid \vec{x} \in (\mathcal{U}^{M})^n\}$;[g]
- for all n-ary function symbols f appearing in ϕ and ψ, $\tilde{M}(f) = M(f) \cup M'(f)$;
- for all predicates P appearing in ϕ but not in ψ, $\tilde{M}(P) = M(P)$;
- for all predicates P appearing in ψ but not in ϕ, $\tilde{M}(P) = M'(P)$;
- for all predicates P appearing in ϕ and in ψ, $\tilde{M}(P) = M(P) \cup M'(P)$;
- $\tilde{M}(\top_c) = \mathcal{U}^M$;
- $\tilde{M}(\top_d) = \mathcal{U}^{M'}$.

As opposed to the proof for consistent entailment conservation, the universe is not reduced to the interpretation of $\top_c$, and the interpretation of the predicates and function symbols may differ in $\tilde{M}$ and M or M'. However, since M and M' have disjoint universes, the interpretation of the predicates will always be strictly separated between the $\tilde{M}(\top_c)$ and $\tilde{M}(\top_d)$. This fact added to the relativization technique ensures that the interpretation is a model of $\mathsf{Rel}(c, \phi) \wedge \mathsf{Rel}(d, \psi)$. Indeed, we can check by structural induction over the formula ϕ (resp. ψ) that $\tilde{M} \models \mathsf{Rel}(c, \phi)$ (resp. $\tilde{M} \models \mathsf{Rel}(d, \psi)$.

For conciseness, we assume there are no free variables in ϕ. Free variables can easily be handled by adding an assignment wherever necessary.

If ϕ is an atom $P(t_1, \ldots, t_n)$ then $\mathsf{Rel}(c, \phi) = P(\mathsf{ren}(c, t_1), \ldots, \mathsf{ren}(c, t_n)) \wedge \top_c(\mathsf{ren}(c, t_1)) \wedge \cdots \wedge \top_c(\mathsf{ren}(c, t_n)) \wedge \bigwedge_{f \in Func(\phi)} \forall \vec{x} \top_c(x_1) \wedge \cdots \wedge \top_c(x_n) \rightarrow \top_c(f(\vec{x}))$ by definition. Since M satisfies $P(t_1, \ldots, t_n)$, it means that $(M(t_1), \ldots, M(t_n)) \in M(P)$. We can notice that for any term t_i appearing in ϕ, $\tilde{M}(\mathsf{ren}(c, t_i)) = M(t_i)$ even when t_i is not a constant. Moreover, $M(P) \subseteq \tilde{M}(P)$. This implies that $(\tilde{M}(\mathsf{ren}(c, t_1)), \ldots, \tilde{M}(\mathsf{ren}(c, t_n))) \in \tilde{M}(P)$. So the first part of $\mathsf{Rel}(c, \phi)$ is satisfied. Then it is obvious that $\tilde{M} \models \top_c(\mathsf{ren}(c, t_i))$ as well as $\tilde{M} \models \forall \vec{x} \top_c(x_1) \wedge \cdots \wedge \top_c(x_n) \rightarrow \top_c(f(\vec{x}))$ for all function symbols f in ϕ.

The proof for a negative literal is very similar. These constitute the base cases of the induction. We now make the induction hypothesis that if ϕ is of a size at most n, then $M \models \phi$ implies that $\tilde{M} \models \mathsf{Rel}(c, \phi)$. The case for $\phi \wedge \psi$ and $\phi \vee \psi$ are straightforward. We consider now the case for $\forall x \phi$. Then $\mathsf{Rel}(c, \phi) = \forall x \top_c(x) \rightarrow \mathsf{Rel}(c, \phi)$. This formula is satisfied if all elements $e \in \tilde{M}(\top_c)$ are such that $\tilde{M} \models \phi[x \leftarrow e]$ (that is, ϕ with the assignment of e to x. By our induction hypothesis, this obviously the case since $x \in \top_c$ means $x \in \mathcal{U}^M$. Finally, the case for $\exists x \phi$ uses a similar structure. □

As in the previous approach where predicates are renamed, the connection between the resulting formula and the context is not explicit. However, with this approach, one could introduce a new binary predicate $\mathsf{inContext}$ relating the renamed constants to the context, that is $\mathsf{inContext}(\mathsf{ren}(c, a), c)$.

[g] This is necessary because we need to extend the domain of the function f to $\mathcal{U}^{\tilde{M}}$.

1.7 Related Work

The problem of representing and reasoning with logical formulas and their validity in different contexts was first introduced by McCarthy [99, 100, 101], where the predicate $ist(c, \phi)$ represents that the logical formula ϕ *holds* in the context c. Then the context can be described using a set of logical formulas, that can be themselves contextualized. Contexts can be organized hierarchically, and lifting axioms can be used to transfer knowledge from a context to a more general context in the hierarchy.

Some works have focused in extending the logic in order to support contexts. Guha [64] proposed a model theory and proof theory for the *ist* predicate, and a general framework for lifting knowledge from one context to another. Buvač *et al.* [19, 18] consider $ist(c, \phi)$ as a modal operator, with which they extend propositional logic. Each context has its own vocabulary, which may or may not overlap with the vocabulary of other contexts, and partial truth assignments (*i.e.*, total truth assignments in a three-valued logic). This logical system is sound and complete. This work is later extended to first order quantification [17], at the price of losing decidability. Nossum [109] generalized a context as a set of formulas, with semantics defined by quantification over a set of modalities that depend on the context formula.

Instead of extending the logic system with modal operators, Attardi *et al.* [2] reify the logical formulas, and formalized the notion of context as *viewpoint*: a set of sentences that represents the assumptions of the context. The predicate $in(\phi, c)$ (with a similar meaning as McCarthy's $ist(c, \phi)$) is a regular predicate that can take names of sentences and contexts as arguments. This makes it possible self-referentiality and allows the possibility of paradoxes. They however avoid them by relaxing entailment between contexts: the fact that $in(\phi, c)$ holds does not necessarily mean that it holds in a subcontext.

Other works considered different contexts as separate sets, with a deduction system to describe the relations between them: Giunchiglia [59] proposes a theory of reasoning with contexts based on McCarthy's ideas, where reasoning is formalized as deduction in a system that allows multiple first-order theories. Each context can have a different theory, and a series of *bridge rules* allow to bring deductions from one context to another. His work lead to distributed first order logic by Ghidini *et al.* [51], distributed description logics by Borgida *et al.* [11], and C-OWL [13]. Kutz *et al.* ε-connections [87] describe a method to combine disjoint ontologies, with possibly different ontology languages, by using a super-language that contains link relations between elements of the ontologies. Then they prove that if all components are decidable, then the ε-connections system is decidable too. They argue that distributed description logics can be see as a concrete instance of ε-connections.

The concept of reification to speak about the context has been studied in the Semantic Web too. A first reification approach [16, Sec. 5.3] and a design pattern to model n-ary relations [110] were proposed. However, both lack formal semantics to relate the new resource to the original triple, which prevents any reasoning on the reified statement (*i.e.*, they do not have entailment conservation). Nguyen *et al.* [108], with a small extension to RDF semantics, propose to create property derived from the property of the original triple, where the cardinality of domain and range is limited to one element, and its extension belongs to the extension of the original property. While this approach is sufficient to annotate triples with a context, this context is lost in the inferences, which makes it not context-separable. In a previous work

we proposed a fluent-base approach, NdFluents [57][h], to annotate triples in different contexts to avoid this problems. Inspired by this work, we extended the approach to description logic, worked in formalizing the concept of contextualization function and its properties, and presented the NdTerms approach [138][i], which makes use of the renaming and relativization. In the present work, we extend this research to first order logic. Note that, while similar, there are important differences between definitions in that work and this one (*e.g.*, entailment preservation is a sufficient condition, while entailment conservation is necessary and sufficient).

1.8 Conclusions

In this work, we define and studied the concept of *contextualization*: a function that takes a context and a logical formula in a language, and gives as a result a new logical formula in the same language that encodes the idea of what is true in that context. The first of the two most important properties is *entailment conservation*, which means that anything entailed from the original formula is also entailed in the contextualization of the formula. The second property is *context independence*, and conveys that the contextualization of formulas in one context will not affect the validity of formulas contextualized in other contexts. We also characterize some other useful properties; namely, consistent entailment conservation (a relaxed version of entailment conservation, applicable only in the case when the formulas are consistent), consistency conservation, and separability; and the relations between them. We then describe three different contextualization functions in first order logic, in order of increasing complexity: (1) The $n+1$-ary approach, where an additional argument to represent the context is added to each predicate; (2) renaming predicates, where each predicate is renamed to a different name for each context; and (3) renaming constants, where each constant is renamed to a different name depending on the context. We show that the two first approaches have the desired properties in FOL without equality, but a non-negligible number of axioms need to be added when considering FOL with equality. The third approach requires the use of the so-called *relativization* technique, adding predicates to formulas to express that its terms belong to the context. However it works for FOL with or without equality.

The renaming of constants has one additional advantage over renaming predicates: If using relativization, the connection of the terms with the context is made explicit in the formulas through the $\top_c$ predicate. And even in the case of not using relativization, this connection could be easily made by adding an additional binary predicate $inContext(c, a_c)$. This kind of connection is not possible when renaming predicates.

The two renaming approaches (renaming predicates and renaming constants) can be seen as instantiations of a family of contextualizations where a subset of the terms is renamed. We call this family of contextualizations **Nd***.[j]

A contextualization function allows to have several, possibly contradictory, contexts in a single FOL theory. Then, additional axioms that describe the context, or relations between what is truth in the contexts (*à la* McCarthian lifting rules) without the need to extend the logical system. Note that it is also possible to speak about a

[h] This work is presented in Chapter 3 of this dissertation.

[i] This work is presented in Chapter 2 of this dissertation.

[j] This paragraph was added for this dissertation. This family of contextualizations was not given a name in the publications, but it was a concept that was continually referred. We will keep exploring it for different subsets of terms and different logical systems in the next chapters.

context within another context, *i.e.*, since the result of a contextualization of a FOL formula is itself a FOL formula, it is possible to apply recursively a contextualization function in different contexts: $f(\ldots, f(d, f(c, \phi))\ldots)$.

At several points, we mentioned the idea of contextualizing inconsistent information in a context, so the result can express the inconsistency, but be itself consistent. This corresponds to the intuition that it should be possible to speak and reason about a context, even if we know it contains erroneous information. We also introduced consistent entailment conservation as a necessary property, and showed that the renaming constants with relativization has this property. However, the idea is not fully explored, and remains as a future line of research.

References

[2] Attardi, G., Simi, M.: A formalization of viewpoints. Fundamenta Informaticae 23(2), 149–173 (1995).

[4] Beckert, B.: Semantic tableaux with equality. Journal on Logic and Computation 7(1), 39–58 (1997).

[11] Borgida, A., Serafini, L.: Distributed Description Logics: Assimilating Information from Peer Sources. Journal on Data Semantics 1, 153–184 (2003).

[13] Bouquet, P., Giunchiglia, F., Van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing Ontologies. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, ProceedingsLNCS, vol. 2870, pp. 164–179. Springer, Heidelberg (2003).

[16] Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C (2004). URL: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[17] Buvač, S.: Quantificational Logic of Context. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, August 4-8, 1996, Volume 1. Pp. 600–606 (1996).

[18] Buvač, S., Buvač, V., Mason, I.A.: Metamathematics of Contexts. Fundamenta Informaticae 23(2), 263–301 (1995).

[19] Buvač, S., Mason, I.A.: Propositional Logic of Context. In: Proceedings of the 11th National Conference on Artificial Intelligence. Washington, DC, USA, July 11-15, 1993. Pp. 412–419 (1993).

[51] Ghidini, C., Serafini, L.: Distributed First Order Logics. In: Frontiers Of Combining Systems 2, Studies in Logic and Computation, pp. 121–140. Research Studies Press (1998).

[57] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC), pp. 638–654. Springer, Cham (2017).

[59] Giunchiglia, F.: Contextual Reasoning. Epistemologica 16, 345–364 (1993).

[64] THESIS

[87] Kutz, O., Lutz, C., Wolter, F., Zakharyaschev, M.: E-connections of abstract description systems. Artificial Intelligence 156(1), 1–73 (2004).

[99] McCarthy, J.: Generality in Artificial Intelligence. Communications of the ACM 30(12), 1030–1035 (1987).

[100] McCarthy, J.: Notes on formalizing context. Science 13, 555–560 (1993).

[101] McCarthy, J., Buvač, S.: Formalizing Context (Expanded Notes). Theory & Psychology 15(1), 128–131 (1997).

[108] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. In: Proceedings of the 23rd International Conference on the World Wide Web (WWW), pp. 759–770. ACM (2014).

[109] Nossum, R.: A decidable multi-modal logic of context. Journal of Applied Logic 1(1), 119–133 (2003).

[110] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web, pp. 1–1. W3C (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/

[130] Welty, C.: Context Slices: Representing Contexts in OWL. In: Workshop on Ontology PatternsCEUR Workshop Proceedings, pp. 59–60. CEUR-WS.org (2010).

[131] Welty, C., Fikes, R.: A Reusable Ontology for Fluents in OWL. In: Bennett, B., Fellbaum, C.D. (eds.) Proceedings of the 2006 conference on Formal Ontology in Information SystemsFrontiers in Artificial Intelligence and Applications, pp. 226–236. IOS Press (2006).

[138] Zimmermann, A., Giménez-García, J.M.: Contextualizing DL Axioms: Formalization, a New Approach, and Its Properties. In: Joint Proceedings of the Web Stream Processing workshop (WSP 2017) and the 2nd International Workshop on Ontology Modularity, Contextuality, and Evolution (WOMoCoE 2017) co-located with 16th International Semantic Web Conference (ISWC 2017), pp. 74–85 (2017).

Chapter 2

Integrating Context of Statements within Description Logics*

This chapter is the next step dealing with the representation of the context of a set of statements. In the previous chapter, we concentrated on First Order Logic. In this chapter we address the contextualization of statements in Description Logic.

Our definition of what expressing the context of a statement is remains the same. That is, given a formula ϕ and a context c, one could express that ϕ is true in c, written $ist(c, \phi)$.

We address the problem of providing contextual information about a logical formula (*e.g.*, provenance, date of validity, or confidence) and representing it within Description Logics. Since only unary and binary relations are possible in Description Logics, it is needed to rely on a higher order or non standard formalism, or some kind of reification mechanism. We explore the case of reification, reusing the knowledge of the previous chapter to Description Logics. We formalize the concept of contextualizing logical statements in the case of Description Logics. Then, we define several properties of contextualization that are desirable. No previous approaches satisfy all of them.

Consequently, we define NdTerms, a new way of contextually annotating statements. It is inspired by the renaming approaches of the previous chapter. For First Order logic, two approaches were presented, one where the predicates are renamed and one where the constants were renamed. In this chapter we generalize this idea by defining contextual parts for all the elements in the signature: individuals (the equivalent to constants in FOL), relations (the equivalent to binary predicates in FOL), and classes (equivalent to unary predicates in FOL). This formal construction better satisfies the properties, although not entirely. We show that it is a particular case of a general mechanism that NdTerms and the aforementioned First Order Logic approaches instantiate.

*This chapter is based on the following publication:

- Zimmermann, A., Giménez-García, J.M.: Integrating Context of Statements within Description Logics, Université Jean Monnet (2017). URL: http://arxiv.org/abs/1709.04970 [139]

Part of this content has been published in:

- Zimmermann, A., Giménez-García, J.M.: Contextualizing DL Axioms: Formalization, a New Approach, and Its Properties. In: Joint Proceedings of the Web Stream Processing workshop (WSP 2017) and the 2nd International Workshop on Ontology Modularity, Contextuality, and Evolution (WOMoCoE 2017) co-located with 16th International Semantic Web Conference (ISWC 2017), pp. 74–85 (2017). [138]

2.1 Introduction

The problem of being able to reason not only *with* logical formulas, but also *about* said formulas, is an old one in artificial intelligence. McCarthy [99] proposed to extend first order logic by reifying context and formulas to introduce a binary predicate $\mathsf{ist}(\phi, c)$ satisfied if the formula ϕ is true (ist) in the context c. However, a complete axiomatization and calculus for McCarthy's contextual logic has never been formalized. Giunchiglia [59] proposed the grouping of "local" formulas in contexts, and then using other kinds of formulas to characterize how knowledge from multiple contexts is *compatible*. This idea of locality+compatibility [50] has led to several non standard formalisms for reasoning with multiple contexts [136]. Alternatively, the approach of annotated logic programming [85] considers that a contextual annotation is just a value in an algebraic structure (*e.g.*, a number or a temporal interval). This idea was later applied to annotated RDF and RDFS [126, 141].

The representation of statement annotation has sometimes been thought of as a data model problem without consideration of the logical formalism behind. In particular, several proposals to extend the RDF data model in various ways for allowing annotations have been made: named graphs [23], RDF+ [37], RDF* [69][a], Yago Model [76]. However, the underlying data structures have not a clear formal semantics. Therefore, some authors advocate another approach to representing annotation of knowledge: reify the statement or its context and describe it within the formalism of the statement. This requires modifying the statement so as to integrate knowledge of the context or statement. Examples of such techniques are reification [15, Sec. 5.3], N-Ary Relations [110], Singleton Property [108]), and NdFluents [57]. *This paper provides an abstraction of the reification techniques in the context of Description Logics (DLs) in the form of what we call contextualization functions*. Additionally, we introduce a new technique for the representation of contextual annotations that satisfies better some desirable properties.

After introducing our notations for DLs in Section 2.2, we provide formal definitions that allow us to define verifiable properties of the reification techniques (Section 2.3). Our new technique, named *NdTerms*, is presented in Section 2.4, where we also prove to what extent it satisfies the properties of the previous section. Section 2.5 discuss some of the problems that may occur when combining knowledge having different annotations. In Section 2.6, we present how the other approaches fit in our formalization and why they do not satisfy well the properties. Finally, we discuss this and future work in Section 2.7.

2.2 Preliminaries

In this section, we introduce the notations and definitions we use in relation to Description Logics. Note that we use an extended version of DL where all terms can be used as *concept names*, *role names*, and *individual names* in the same ontology. Using the same name for different types of terms is known as "punning" in OWL 2 [62, Section 2.4.1]. Moreover, we allow more constructs than in OWL 2 DL and make no restriction on their use in order to show that our approach is not limited to a specific DL.

We assume that there is an infinite set of *terms*. Every term is an individual, a role, and a concept. An *individual* is a terms. A *role* is either a term or, given roles R and S, $R \sqcup S$, $R \sqcap S$, $\neg R$, R^-, $R \circ S$ and R^+. A *concept* is either a term, or, given

[a] Now renamed to "RDF-star"[68].

concepts C, D, role R, individuals $u_1, \ldots, u_k$, and natural number n, $\bot$, $\top$, $C \sqcup D$, $C \sqcap D$, $\exists R.C$, $\forall R.C$, $\leq nR.C$, $\geq nR.C$, $\neg C$ or $\{u_1, \ldots, u_k\}$. Finally, we also allow concept product $C \times D$ to define a role.[b]

Interpretations are tuples $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}_u}, \cdot^{\mathcal{I}_r}, \cdot^{\mathcal{I}_c} \rangle$, where $\Delta^{\mathcal{I}}$ is a non-empty set (the domain of interpretation) and $\cdot^{\mathcal{I}_u}$, $\cdot^{\mathcal{I}_r}$, and $\cdot^{\mathcal{I}_c}$ are the interpretation functions for individuals, roles and concepts respectively such that:

- for all $u \in \mathit{terms}$, $u^{\mathcal{I}_u} \in \Delta^{\mathcal{I}}$;
- for all $P \in \mathit{terms}$, $P^{\mathcal{I}_r} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and interpretation of roles is inductively defined by $(R \sqcup S)^{\mathcal{I}_r} = R^{\mathcal{I}_r} \cup S^{\mathcal{I}_r}$, $(R \sqcap S)^{\mathcal{I}_r} = R^{\mathcal{I}_r} \cap S^{\mathcal{I}_r}$, $(\neg R)^{\mathcal{I}_r} = (\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}) \setminus R^{\mathcal{I}_r}$, $(R^-)^{\mathcal{I}_r} = \{\langle x, y \rangle | \langle y, x \rangle \in R^{\mathcal{I}_r}\}$, $(R \circ S)^{\mathcal{I}_r} = \{\langle x, y \rangle | \exists z. \langle x, z \rangle \in R^{\mathcal{I}_r} \wedge \langle z, y \rangle \in S^{\mathcal{I}_r}\}$ and $(R^+)^{\mathcal{I}_r}$ is the reflexive-transitive closure of $R^{\mathcal{I}_r}$ (with R and S being arbitrary roles).
- for all $A \in \mathit{terms}$, $A^{\mathcal{I}_c} \subseteq \Delta^{\mathcal{I}}$ and interpretation of concepts is defined by $\bot^{\mathcal{I}_c} = \emptyset$, $\top^{\mathcal{I}_c} = \Delta^{\mathcal{I}}$, $(C \sqcup D)^{\mathcal{I}_c} = C^{\mathcal{I}_c} \cup D^{\mathcal{I}_c}$, $(C \sqcap D)^{\mathcal{I}_c} = C^{\mathcal{I}_c} \cap D^{\mathcal{I}_c}$, $(\exists R.C)^{\mathcal{I}_c} = \{x | \exists y. y \in C^{\mathcal{I}_c} \wedge \langle x, y \rangle \in R^{\mathcal{I}_r}\}$, $(\forall R.C)^{\mathcal{I}_c} = \{x | \forall y. \langle x, y \rangle \in R^{\mathcal{I}_r} \Rightarrow y \in C^{\mathcal{I}_c}\}$, $(\leq nR.C)^{\mathcal{I}_c} = \{x | \sharp\{y \in C^{\mathcal{I}_c} | \langle x, y \rangle \in R^{\mathcal{I}_r}\} \leq n\}$, $(\geq nR.C)^{\mathcal{I}_c} = \{x | \sharp\{y \in C^{\mathcal{I}_c} | \langle x, y \rangle \in R^{\mathcal{I}_r}\} \geq n\}$, $(\neg C)^{\mathcal{I}_c} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}_c}$, $\{u_1, \ldots, u_k\} = \{u_1^{\mathcal{I}_u}, \ldots, u_k^{\mathcal{I}_u}\}$, where C and D are arbitrary concepts, R an arbitrary role, $u_1, \ldots, u_k$ are individual names, and k and n two natural numbers.
- Roles defined as a concept product are interpreted as $(C \times D)^{\mathcal{I}_r} = C^{\mathcal{I}_c} \times D^{\mathcal{I}_c}$ for arbitrary concepts C and D.

In the following, we slightly abuse notations by defining interpretations as pairs $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where $\cdot^{\mathcal{I}}$ denotes the three functions $\cdot^{\mathcal{I}_u}$, $\cdot^{\mathcal{I}_c}$, and $\cdot^{\mathcal{I}_r}$. Moreover, when we write $x^{\mathcal{I}} = y^{\mathcal{I}'}$, it means "$x^{\mathcal{I}_u} = y^{\mathcal{I}'_u}$ and $x^{\mathcal{I}_c} = y^{\mathcal{I}'_c}$ and $x^{\mathcal{I}_r} = y^{\mathcal{I}'_r}$".

Axioms are either general concept inclusions $C \sqsubseteq_c D$, sub-role axioms $R \sqsubseteq_r S$, instance assertions $C(a)$, or role assertions $R(a, b)$, where C and D are concepts, R and S are roles, and a and b are individual names. An interpretation $\mathcal{I}$ *satisfies* axiom $C \sqsubseteq_c D$ iff $C^{\mathcal{I}_c} \subseteq D^{\mathcal{I}_c}$; it satisfies $R \sqsubseteq_r S$ iff $R^{\mathcal{I}_r} \subseteq S^{\mathcal{I}_r}$; it satisfies $C(a)$ iff $a^{\mathcal{I}_u} \in C^{\mathcal{I}_c}$; and it satisfies $R(a, b)$ iff $\langle a^{\mathcal{I}_u}, b^{\mathcal{I}_u} \rangle \in R^{\mathcal{I}_r}$. When $\mathcal{I}$ satisfies an axiom α, it is denoted by $\mathcal{I} \models \alpha$. Instance assertions and role assertions constitute the ABox axioms.

An ontology O is composed of a set of terms called the signature of O and denoted by $\mathsf{Sig}(O)$, and a set of axioms denoted by $\mathsf{Ax}(O)$. An interpretation $\mathcal{I}$ is a model of an ontology O iff for all $\alpha \in \mathsf{Ax}(O)$, $\mathcal{I} \models \alpha$. In this case, we write $\mathcal{I} \models O$. The set of all models of an ontology O is denoted by $\mathrm{Mod}(O)$. A *semantic consequence* of an ontology O is a formula α such that for all $\mathcal{I} \in \mathrm{Mod}(O)$, $\mathcal{I} \models \alpha$.

In the rest of the paper, we will use `teletype font` to denote known individuals, and normal font for unknown individuals and variables (*e.g.*, $\texttt{City}(\texttt{babylon})$ and $\texttt{City}(x)$).

2.3 Contextualization of Statements

A contextual annotation can be thought of as a set of ABox axioms that describe an individual representing the statement (the anchor) that is annotated. An annotated statement (or ontology) is the combination of a DL axiom (or DL ontology) with a contextual annotation.

[b] This paragraph was updated, since the original one didn't reflect correctly the use of punning.

Definition 2.1 (Connected individuals)**.** *Two terms a and b are* connected individuals *with regards to an ABox A iff a and b are used as individual names in A, and either*

- *a and b are the same term, or*
- *there exists* $R_1, \cdots, R_n$ *and* $z_1, \cdots, z_{n-1}$*, such that:*
  - $R_1(a, z_1)$*, or* $R_1(z_1, a)$
  - $R_i(z_{i-1}, z_i)$*, or* $R_i(z_i, z_{i-1})$*,* $2 \leq i \leq n-2$
  - $R_n(z_{n-1}, b)$*, or* $R_n(b, z_{n-1})$

Example 2.1. *If we consider the ABox* $A = \{P(a,b), Q(c,b), S(d,e)\}$*, the pairs of individuals* $\{a,b\}$*,* $\{b,c\}$*,* $\{a,c\}$*, and* $\{d,e\}$ *are* connected individuals*, but* $\{a,d\}$*,* $\{b,d\}$*,* $\{c,d\}$*,* $\{a,e\}$*,* $\{b,e\}$*, and* $\{c,e\}$ *are not.*

Definition 2.2 (Contextual annotation)**.** *A* contextual annotation $\mathcal{A}$ *is an ABox with signature* $\{\mathbf{a}\} \cup \Sigma$ *where* $\mathbf{a} \notin \Sigma$ *is a distinguished term (called the* anchor*) and* Σ *is a DL signature such that* $\forall x \in \Sigma$*, if x is used as individual name in* $\mathcal{A}$*, then* $\{\mathbf{a}, x\}$ *are* connected individuals.[c]

Example 2.2. *The Abox* $\mathcal{CA} = \{\texttt{validity}(\mathbf{a},\texttt{t}), \texttt{Interval}(\texttt{t}), \texttt{from}(\texttt{t},\texttt{609BC}), \texttt{to}(\texttt{t},\texttt{539BC}), \texttt{prov}(\mathbf{a},\texttt{w}), \texttt{name}(\texttt{w},\texttt{wikipedia}), \texttt{Wiki}(\texttt{w})\}$ *is a* contextual annotation*, where* $\mathbf{a}$ *is the anchor and* $\Sigma = \{\texttt{t}, \texttt{Interval}, \texttt{w}, \texttt{wikipedia}, \texttt{Wiki}, \texttt{609BC}, \texttt{539BC}\}$*.*

Definition 2.3 (Annotated statement)**.** *An* annotated statement *is a pair* $\langle \alpha, \mathcal{A} \rangle$ *such that* α *is a description logic axiom and* $\mathcal{A}$ *is a contextual annotation.*

Example 2.3. *The pair* $\langle \alpha, \mathcal{CA} \rangle$*, where* $\alpha = \texttt{capital}(\texttt{babylon}, \texttt{babylonianEmpire})$ *and* $\mathcal{CA}$ *is the contextual annotation from Example 2.2, is an* annotated statement*.*

Definition 2.4 (Annotated ontology)**.** *An* annotated ontology *is a pair* $\langle O, \mathcal{A} \rangle$ *such that O is a description logic ontology and* $\mathcal{A}$ *is a contextual annotation.*

Each reification technique has an implicit construction plan in order to map an annotated statement to a resulting ontology. A contextualization[d] (Definition 2.5.) represents the procedure that generates a single DL ontology from a given annotated statement or ontology. The procedure must not lose information, especially not the annotation.

Definition 2.5 (Contextualization)**.** *A* contextualization *is a function f that maps each annotated statement* $\alpha_{\mathcal{A}} = \langle \alpha, \mathcal{A} \rangle$ *to a description logic ontology* $f(\alpha_{\mathcal{A}}) = \mathsf{St}(\alpha_{\mathcal{A}}) \cup \mathsf{Cx}(\alpha_{\mathcal{A}})$ *such that:*

- *there exists an individual u in the signature of* $\mathsf{St}(\alpha_{\mathcal{A}})$ *and of* $\mathsf{Cx}(\alpha_{\mathcal{A}})$ *such that:*
  - *for all* $R(\mathbf{a}, x) \in \mathcal{A}$*,* $R(u, x) \in \mathsf{Cx}(\alpha_{\mathcal{A}})$*;*
  - *for all* $R(x, \mathbf{a}) \in \mathcal{A}$*,* $R(x, u) \in \mathsf{Cx}(\alpha_{\mathcal{A}})$*;*
  - *for all* $C(\mathbf{a}) \in \mathcal{A}$*,* $C(u) \in \mathsf{Cx}(\alpha_{\mathcal{A}})$*;*
  - *for all other* $\alpha \in \mathcal{A}$*,* $\alpha \in \mathsf{Cx}(\alpha_{\mathcal{A}})$*.*
- *there is an injective mapping from the signature of* α *to the signature of* $\mathsf{St}(\alpha_{\mathcal{A}})$*.*

We extend f to all annotated ontologies $O_{\mathcal{A}} = \langle O, \mathcal{A} \rangle$ *by defining* $f(O_{\mathcal{A}}) = \bigcup_{\alpha \in O} f(\langle \alpha, \mathcal{A} \rangle)$*.*

[c] This definition has been updated to require that x is used as individual name in $\mathcal{A}$.

[d] See also definition 1.1 in Chapter 1

Example 2.4. *An example contextualization function f_{ex} introduces a fresh term t for each annotated statement with a role assertion $R(a,b)$, where R, a, and b are three therms, creates new axioms* $\texttt{subject}(t,s)$, $\texttt{predicate}(t,R)$, $\texttt{object}(t,o)$, *and finally removes the axiom $R(a,b)$. Notice that this construction requires the punning of term R. This function is analogous to* RDF Reification. *The result of this contextualization, along with other possible known approaches, is described in Section 2.6.*

Those are the only structures that we will consider in this paper. The remaining definitions are desirable properties that a contextualization should satisfy, especially if one wants it to preserve as much of the original knowledge as possible.

Definition 2.6[e] (Soundness)**.** *A contextualization function f is* sound *with regards to a set of annotated ontologies Ω iff for each $O_{\mathscr{C}} = \langle O, \mathscr{C}\rangle \in \Omega$ such that O and $\mathscr{C}$ are consistent, then $f(O_{\mathscr{C}})$ is consistent.*

That is, a contextualization is sound if, when contextualizing a consistent ontology, the result is also consistent. This property avoids that the contextualization introduces unnecessary contradictions that would result in everything being entailed by it. Note that this requirement is not necessary in the opposite direction, *i.e.*, if $f(O_{\mathscr{C}})$ is consistent, it is not required that O and $\mathscr{C}$ are consistent.

Example 2.5. *The contextualization function f_{ex} from Example 2.4 is sound with regards to the set of ontologies Ω, where* $\bigcup_{\omega\in\Omega)} \mathsf{Sig}(\omega) \cap \{\texttt{subject}, \texttt{predicate}, \texttt{object}\} = \emptyset$

Definition 2.7 (Inconsistency preservation)**.** *Let f be a contextualization function. We say that f preserves inconsistencies iff for all annotated ontologies $O_{\mathscr{C}} = \langle O, \mathscr{C}\rangle$, if O is inconsistent then $f(O_{\mathscr{C}})$ is inconsistent.*

Inconsistency preservation means that a self-contradictory ontology in a given context is contextualized into an inconsistent ontology, such that bringing additional knowledge from other contexts would result in no more consistency. If something is inconsistent within a context, then it is not really worth to consider reasoning with this annotated ontology.

Example 2.6. *The contextualization function f_{ex} from Example 2.4 does not preserve inconsistencies. For instance,* $\texttt{capitalOf}$ *can be defined as irreflexive using the following axiom:* $\exists\texttt{capitalOf}.\top \sqsubseteq \forall\texttt{capitalOf}^{-}.\bot$. *Then, the axiom* $\texttt{capitalOf}(\texttt{babylon}, \texttt{babylon})$ *would make the ontology inconsistent. But when applying f_{ex} the result is consistent.*

Definition 2.8 (Entailment preservation)**.** *Let f be a contextualization function. Given two description logic ontologies O_1 and O_2 such that $O_1 \models O_2$, we say that f* preserves the entailment *between O_1 and O_2 iff for all contextual annotations $\mathscr{C}$, $f(\langle O_1, \mathscr{C}\rangle) \models f(\langle O_2, \mathscr{C}\rangle)$. Given a set $\mathfrak{C}$ of contextual annotations, if f preserves all entailments between ontologies in $\mathfrak{C}$, then we say that f is* entailment preserving *for $\mathfrak{C}$.*

In short, a contextualization is entailment preserving if all the knowledge that could be inferred from the original ontology can also be inferred, in the same context, in the contextualized ontology.

Example 2.7. *The contextualization function f_{ex} from Example 2.4 preserves entailments for the TBox of ontologies (because no modifications are made on its axioms), but it does not preserve entailments on role assertions. For instance, the axioms* $\texttt{capitalOf} \sqsubseteq \texttt{cityOf}$, $\texttt{capitalOf}(\texttt{babylon}, \texttt{babylonianEmpire})$ *entails* $\texttt{cityOf}(\texttt{babylon}, \texttt{babylonianEmpire})$, *but this inference is not preserved after applying f_{ex}.*

[e] This definition is equivalent to *Consistency Conservation* for First Order Logic in Chapter 1.

2.4 The *NdTerms* Approach

This section defines the *NdTerms* approach that extends the NdFluent proposal [57].[f] To this end, we assume that terms are divided into three infinite disjoint sets N^{nc}, N^{c}, and N^{a} called the *non contextual terms*, the *contextual terms*, and the *anchor terms* respectively. We also assume that there is an injective function $\mathfrak{A} : \mathfrak{C} \to \mathsf{N}^{\mathsf{a}}$, where $\mathfrak{C}$ is a set of contextual annotations, and for all $\mathcal{C} \in \mathfrak{C}$ there is an injective function $\mathsf{ren}_{\mathcal{C}} : \mathsf{N}^{\mathsf{nc}} \to \mathsf{N}^{\mathsf{c}}$ and two terms $\mathtt{isContextualPartOf}, \mathtt{isInContext} \in \mathsf{N}^{\mathsf{nc}}$. For any $\mathcal{C}$, we extend $\mathsf{ren}_{\mathcal{C}}$ to axioms by defining $\mathsf{ren}_{\mathcal{C}}(\alpha)$ as the axiom built from α by replacing all terms $t \in \mathsf{Sig}(\alpha)$ with $\mathsf{ren}_{\mathcal{C}}(t)$.[g]

2.4.1 Contextualization Function in *NdTerms*

The contextualization needs to combine the ontologies from the statements and the contextual annotation. However, if we naïvely make the union of the axioms, they could contradict, and it would not be possible to ensure the desired properties. For example, an ontology may restrict the size of the domain of interpretation to be of a fixed cardinality, while the contextual annotation may rely on more elements outside the local universe of this context. For this reason we use the concept or relativization: The ontology is modified in such a way that the interpretation of everything explicitly described in it is confined to a set, while external terms or constructs may have elements outside said set. Relativization has been applied in various logical settings over the past four decades (*e.g.*, [122]) and applied to DLs and OWL [27], among others.

The relativization of an ontology can be done systematically by relativizing its concepts and roles, and adding additional terms and axioms to it.[h]

Definition 2.9 (Relativization of concepts and roles)**.** *Given a contextual annotation $\mathcal{C}$, we define a function $\mathsf{rel}_{\mathcal{C}}$ that maps concepts and roles to concepts and roles recursively:*

- $\mathsf{rel}_{\mathcal{C}}(t) = t$;
- $\mathsf{rel}_{\mathcal{C}}(\top) = \top_{\mathcal{C}}$
- $\mathsf{rel}_{\mathcal{C}}(\{u_1, \dots u_k\}) = \{\mathsf{rel}_{\mathcal{C}}(u_1), \dots rel f(u_k)\}$;
- $\mathsf{rel}_{\mathcal{C}}(C \sqcup D) = \mathsf{rel}_{\mathcal{C}}(C) \sqcup \mathsf{rel}_{\mathcal{C}}(D)$;
- $\mathsf{rel}_{\mathcal{C}}(C \sqcap D) = \mathsf{rel}_{\mathcal{C}}(C) \sqcap \mathsf{rel}_{\mathcal{C}}(D)$;
- $\mathsf{rel}_{\mathcal{C}}(\neg C) = \neg \mathsf{rel}_{\mathcal{C}}(C) \sqcap \top_{\mathcal{C}}$;
- $\mathsf{rel}_{\mathcal{C}}(C \times D) = \mathsf{rel}_{\mathcal{C}}(C) \times \mathsf{rel}_{\mathcal{C}}(D)$;
- $\mathsf{rel}_{\mathcal{C}}(R \sqcup S) = \mathsf{rel}_{\mathcal{C}}(R) \sqcup \mathsf{rel}_{\mathcal{C}}(S)$;
- $\mathsf{rel}_{\mathcal{C}}(R \sqcap S) = \mathsf{rel}_{\mathcal{C}}(R) \sqcap \mathsf{rel}_{\mathcal{C}}(S)$;
- $\mathsf{rel}_{\mathcal{C}}(R \circ S) = \mathsf{rel}_{\mathcal{C}}(R) \circ \mathsf{rel}_{\mathcal{C}}(S)$;

[f] While this work was made chronologically after NdFluents, this dissertation follows a top-down description of the contribution from the most general logic system to the most concrete. NdFluents is presented in Chapter 3.

[g] Added "where $\mathfrak{C}$ is a set of contextual annotations".

[h] Definitions 2.9 and 2.10 were presented in the inverse order in the original paper. We have changed the order and rewritten Definition 2.10 to use Definition 2.9 in this chapter. In addition, we have updated references to items in Definition 2.10 along the chapter to be in accordance with this change.

- $\mathsf{rel}_{@}(\neg R) = \neg\mathsf{rel}_{@}(R) \sqcap \top_{@} \times \top_{@}$;
- $\mathsf{rel}_{@}(R^{-}) = \mathsf{rel}_{@}(R)^{-}$;
- $\mathsf{rel}_{@}(R^{+}) = \mathsf{rel}_{@}(R)^{+} \sqcap \top_{@} \times \top_{@}$;
- $\mathsf{rel}_{@}(\exists R.C) = \exists\mathsf{rel}_{@}(R).\mathsf{rel}_{@}(C)$;
- $\mathsf{rel}_{@}(\forall R.C) = \forall\mathsf{rel}_{@}(R).\mathsf{rel}_{@}(C) \sqcap \top_{@}$;
- $\mathsf{rel}_{@}(\geq nR.C) = \geq n\ \mathsf{rel}_{@}(R).\mathsf{rel}_{@}(C)$;
- $\mathsf{rel}_{@}(\leq nR.C) = \leq n\ \mathsf{rel}_{@}(R).\mathsf{rel}_{@}(C)$.

where t is a term, C, D are concepts, R, S are roles, $u_1, \ldots u_k$ are individuals, and k, n are natural numbers.

Example 2.8. *The axiom $\exists\texttt{capitalOf}.\top \sqsubseteq \forall\texttt{capitalOf}^{-}.\bot$ from Example 2.6 is relativized into $\exists\texttt{capitalOf}.\top_{@} \sqsubseteq \forall\texttt{capitalOf}^{-}.\bot \sqcap \top_{@}$.*

Definition 2.10 (Relativization)**.** *Let $@$ be a contextual annotation. Given an ontology O, the relativization of O in $@$ is an ontology $\mathsf{Rel}_{@}(O)$ built from O as follows:*

1. $\mathsf{Sig}(\mathsf{Rel}_{@}(O)) = \mathsf{Sig}(O) \cup \{\top_{@}\}$ *where $\top_{@}$ is a term not appearing in $\mathsf{Sig}(O)$;*
2. *for all concepts and roles, apply function $\mathsf{rel}_{@}$.*
3. *Additionally, for all terms $t \in \mathsf{Sig}(O)$, the following axioms are in $\mathsf{Rel}_{@}(O)$:*
   - $t \sqsubseteq \top_{@}$,
   - $\top_{@}(t)$,
   - $\exists t.\top \sqsubseteq \top_{@}$,
   - $\top \sqsubseteq \forall t.\top_{@}$.

Then, the contextualization in *NdTerms* is done by: (1) creating the replacement of the anchor using the function $\mathfrak{A}$, (2) renaming all the terms in the statement using the ren function, (3) linking them to the original terms by the `isContextualPartOf` relation, and (4) linking the renamed terms to the context using the `isInContext` relation.

Definition 2.11 (Contextualization in *NdTerms*)**.** *Let $@ \in \mathfrak{C}$ be any contextual annotation. Let $\alpha_{@} = \langle \alpha, @ \rangle$ be an annotated statement such that the signatures of α and $@$ are in N^{nc}. We define the contextualization function f_{nd} such that $f_{\mathsf{nd}}(\alpha_{@}) = \mathsf{St}(\alpha_{@}) \cup \mathsf{Cx}(@)$ and:*

- $\mathsf{St}_{@}(\alpha) = \{\mathsf{ren}_{@}(\mathsf{Rel}_{@}(\alpha))\} \cup \{\texttt{isContextualPartOf}(\mathsf{ren}_{@}(t), t) \mid t \in \mathsf{Sig}(\alpha)\} \cup \{\texttt{isInContext}(\mathsf{ren}_{@}(t), \mathfrak{A}(@)) \mid t \in \mathsf{Sig}(\alpha)\}$.
- $\mathsf{Cx}(@)$ *contains exactly the following axioms:*
  - *for all $R(\mathbf{a}, x) \in @$, $R(\mathfrak{A}(@), x) \in \mathsf{Cx}(\alpha)$;*
  - *for all $R(x, \mathbf{a}) \in @$, $R(x, \mathfrak{A}(@)) \in \mathsf{Cx}(@)$;*
  - *for all $C(\mathbf{a}) \in @$, $C(\mathfrak{A}(@)) \in \mathsf{Cx}(@)$;*
  - *for all other axioms $\beta \in @$, $\beta \in \mathsf{Cx}(@)$.*

Similarly to Example 2.4, this construction requires punning, since all terms in the statement are used as individual names in the role assertion $\texttt{isContextualPartOf}(\mathsf{ren}_{@\mathcal{A}}(t), t)$.

Example 2.9. *The NdTerms contextualization of our running example within the context $\mathcal{CA}$ of Example 2.2 contains the following axioms, where* term@$\mathcal{A}$ *is the result of the renaming function* $\mathsf{ren}_{@\mathcal{A}}(\text{term})$*:*

$\texttt{capitalOf@}\mathcal{CA}(\texttt{babylon@}\mathcal{CA}, \texttt{babylonianEmpire@}\mathcal{CA})$
$\texttt{isContextualPartOf}(\texttt{babylon@}\mathcal{CA}, \texttt{babylon})$
$\texttt{isContextualPartOf}(\texttt{babylonianEmpire@}\mathcal{CA}, \texttt{babylonianEmpire})$
$\texttt{isInContext}(\texttt{babylon@}\mathcal{CA}, \texttt{exampleContext})$
$\texttt{isInContext}(\texttt{babylonianEmpire@}\mathcal{CA}, \texttt{exampleContext})$
$\texttt{validity}(\texttt{exampleContext}, \texttt{t})$
$\texttt{Interval}(\texttt{t})$
$\texttt{from}(\texttt{t}, \texttt{609BC})$
$\texttt{to}(\texttt{t}, \texttt{539BC})$
$\texttt{prov}(\texttt{exampleContext}, \texttt{w})$
$\texttt{name}(\texttt{w}, \texttt{wikipedia})$
$\texttt{Wiki}(\texttt{w})$

2.4.2 Soundness of *NdTerms*

In this section, we fix a contextual annotation $\mathcal{K}_{@}$ that has its signature in N^{nc}, since the following theorems and proofs require such a constraint.

The contextualization of *NdTerms* is sound, but only with regards to annotated ontologies that satisfy certain conditions. In order to present the conditions, we need to introduce the following definition, that is also used in several proofs of this paper.

Definition 2.12 (Domain extensibility)**.** *Let O be an ontology. A model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ of O is* domain extensible *for O iff for all sets Δ^{+}, $\mathcal{I}' = \langle \Delta^{\mathcal{I}} \cup \Delta^{+}, \cdot^{\mathcal{I}} \rangle$ is also a model of O. An ontology is said to be* model extensible *iff it has a model that is domain extensible.*

Note that, even if the domain of interpretation of an ontology is infinite, that does not necessarily mean that its models are domain extensible. This notion is closely related to the notion of *expansion* in [27] since if $\mathcal{I}$ is domain extensible, then one can build infinitely many expansions of it.

Theorem 2.1 (Soundness of *NdTerms*)**.** *If the contextual annotation $\mathcal{K}_{@}$ is model extensible, then the contextualization function f_{nd} is sound with regards to annotated ontology $O_{\mathcal{K}} = \langle O, \mathcal{K}_{@} \rangle$, $\mathsf{Sig}(O) \subseteq \mathsf{N}^{\mathsf{nc}}$, and $\mathsf{Sig}(O) \cap \mathsf{Sig}(\mathcal{K}_{@}) = \emptyset$.*

The proof of this theorem requires a few intermediary steps. Theorem 2.2 ensures that, given the right condition, a model of the union of two ontologies can be made from two models of the original ontologies.

Theorem 2.2 (Model extensibility theorem)**.** *Let O and O^{+} be two ontologies such that there exist two models $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ and $\mathcal{I}^{+} = \langle \Delta^{\mathcal{I}^{+}}, \cdot^{\mathcal{I}^{+}} \rangle$ that are domain extensible for O and O^{+} respectively. If $\mathsf{Sig}(O) \cap \mathsf{Sig}(O^{+}) = \emptyset$ then $\mathcal{I}$ and $\mathcal{I}^{+}$ can be both extended into a unique model $\mathcal{I}'$ of $O \cup O^{+}$ such that $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}} \cup \Delta^{\mathcal{I}^{+}}$ and for all $t \in \mathsf{Sig}(O)$, $t^{\mathcal{I}'} = t^{\mathcal{I}}$, and for all $t \in \mathsf{Sig}(O^{+})$, $t^{\mathcal{I}'} = t^{\mathcal{I}^{+}}$.*[1]

[1] Remember from preliminaries that $t^{\mathcal{I}'} = t^{\mathcal{I}}$ means "$t^{\mathcal{I}'_u} = t^{\mathcal{I}_u}$ and $t^{\mathcal{I}'_c} = t^{\mathcal{I}_c}$ and $t^{\mathcal{I}'_r} = t^{\mathcal{I}_r}$".

Proof of Theorem 2.2[i]. Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ and $\mathcal{I}^+ = \langle \Delta^{\mathcal{I}^+}, \cdot^{\mathcal{I}^+} \rangle$ be models of O and O^+ respectively. W.l.o.g., we assume that $\Delta^{\mathcal{I}} \cap \Delta^{\mathcal{I}^+} = \emptyset$. We define a new interpretation $\mathcal{I}'$ where $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}} \cup \Delta^{\mathcal{I}^+}$; for all $t \in \mathsf{Sig}(O)$, $t^{\mathcal{I}'} = t^{\mathcal{I}}$; and for all $t \in \mathsf{Sig}(O^+)$, $t^{\mathcal{I}'} = t^{\mathcal{I}^+}$. Since O and O^+ are model extensible, $\mathcal{I}' \models O$ and $\mathcal{I}' \models O^+$. □

From this theorem it follows that *NdTerms* is sound if both $\mathsf{Rel}_{\mathcal{C}}(O)$ and $\mathcal{C}$ are model extensible.[j] However, this is not a strong restriction, because the relativization of any consistent ontology is model extensible.

Theorem 2.3 (Model extensibility of relativized ontologies)**.** *For any annotated ontology $O_{\mathcal{K}} = \langle O, \mathcal{K} \rangle$ where O is consistent, if $\mathsf{Sig}(O) \subseteq \mathsf{N}^{\mathsf{nc}}$ then $\mathsf{Rel}_{\mathcal{C}}(O)$ is model extensible and all its models are domain extensible.*

Proof of Theorem 2.3. Since O is consistent, there exists a model $\mathcal{I}_o = \langle \Delta^{\mathcal{I}_o}, \cdot^{\mathcal{I}_o} \rangle$. We define an interpretation $\mathcal{I}_{\mathsf{rel}} = \langle \Delta^{\mathcal{I}_o}, \cdot^{\mathcal{I}_{\mathsf{rel}}} \rangle$ of $\mathsf{Rel}_{\mathcal{K}}(O)$ where for all $t \in \mathsf{Sig}(O)$, $t^{\mathcal{I}_{\mathsf{rel}}} = t^{\mathcal{I}_o}$ and $\top_{\mathcal{K}}^{\mathcal{I}_{\mathsf{rel}}} = \Delta^{\mathcal{I}_o}$.

First, we show that $\mathcal{I}_{\mathsf{rel}}$ is a model of $\mathsf{Rel}_{\mathcal{K}}(O)$. Indeed, it clearly satisfies all extra axioms in Definition 2.10. Second, since $\top_{\mathcal{K}}^{\mathcal{I}_{\mathsf{rel}}} = \top^{\mathcal{I}_{\mathsf{rel}}}$, replacing $\top$ with $\top_{\mathcal{K}}$ or adding a conjunction with $\top_{\mathcal{K}}$ does not change the meaning of the formulas between $\mathcal{I}_o$ and $\mathcal{I}_{\mathsf{rel}}$.

Second, we need to show that any extension of $\mathcal{I}_{\mathsf{rel}}$ is still a model for $\mathsf{Rel}_{\mathcal{K}}(O)$, that is, $\mathcal{I}' = \langle \Delta^{\mathcal{I}_o} \cup \Delta^{\mathcal{I}^+}, \cdot^{\mathcal{I}_{\mathsf{rel}}} \rangle \models \mathsf{Rel}_{\mathcal{K}}(O)$ for any set $\Delta^{\mathcal{I}^+}$ such that $\Delta^o \cap \Delta^{\mathcal{I}^+} = \emptyset$. Since the interpretations of the terms are the same in $\mathcal{I}_{\mathsf{rel}}$ and $\mathcal{I}'$, it follows that $\mathcal{I}'$ satisfies the extra axioms in Definition 2.10. We now need to prove that all other axioms of $\mathsf{Rel}_{\mathcal{K}}(O)$ are satisfied, which we can do with a proof by induction taking advantage of the recursive definition of $\mathsf{rel}_{\mathcal{K}}$. However, we need to prove an auxiliary lemma first.

Lemma 2.1. *For all concepts or roles X, $X^{\mathcal{I}_o} = \mathsf{rel}_{\mathcal{K}}(X)^{\mathcal{I}'}$.*

This lemma can be proved by structural induction on the concepts and roles. Because of space restriction, we do not provide a complete proof but remark that every time a construct may lead to a different interpretation due to increasing the domain of interpretation (*e.g.*, by using the concept $\top$ or a negation), the function $\mathsf{rel}_{\mathcal{K}}$ adds a conjunction with $\top_{\mathcal{K}}$, so that the interpretations of the relativized concepts and roles stay confined in the original domain of interpretation.

From this lemma, it follows that if $\mathcal{I}_o \models \alpha$ then $\mathcal{I}' \models \mathsf{Rel}_{\mathcal{K}}(\alpha)$, where α is any axiom with $\mathsf{Sig}(\alpha) \subseteq \mathsf{Sig}(O)$. Consequently, $\mathcal{I}' \models \mathsf{Rel}_{\mathcal{K}}(O)$.

In order to prove that all models of $\mathsf{Rel}_{\mathcal{K}}(O)$ are domain extensible, we consider an arbitrary model $\mathcal{I}$ of said ontology and apply the same construction that lead to $\mathcal{I}'$ and use the same arguments to show that it is still a model of $\mathsf{Rel}_{\mathcal{K}}(O)$. □

With Theorem 2.3 proven, we can now proceed to prove the main Theorem 2.1.

Proof of Theorem 2.1. Let us assume that $\mathcal{K}$ is model extensible. Let O be a consistent ontology such that $\mathsf{Sig}(O) \cap \mathsf{Sig}(\mathcal{K}) = \emptyset$ and $\mathsf{Sig}(O) \subseteq \mathsf{N}^{\mathsf{nc}}$.

Since O and $\mathcal{K}$ are consistent, there exist two models $\mathcal{I}_o = \langle \Delta^{\mathcal{I}_o}, \cdot^{\mathcal{I}_o} \rangle$ and $\mathcal{I}_c = \langle \Delta^{\mathcal{I}_c}, \cdot^{\mathcal{I}_c} \rangle$ of O and $\mathcal{K}$ respectively. W.l.o.g., we can assume that $\Delta^{\mathcal{I}_o} \cap \Delta^{\mathcal{I}_c} = \emptyset$.

[i] We have replaced "Since O^+ is consistent, there exists a model $\mathcal{I}^+$ of O^+ and" from the original paper to "Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ and $\mathcal{I}^+ = \langle \Delta^{\mathcal{I}^+}, \cdot^{\mathcal{I}^+} \rangle$ be models of O and O^+ respectively."

[j] In the original paper we stated that *NdTerms* is sound if both the original ontology and the contextual annotation are model extensible, which is not correct. It is not the original ontology which should be model extensible, but its relativization. We have updated the text to reflect that.

First, from the proof of Theorem 2.3, we know that from a model of O, we can define a model $\mathcal{I}_{\mathsf{Rel}}$ of $\mathsf{Rel}_{\mathcal{K}\!a}(O)$ having the same domain as $\mathcal{I}_o$. Then, $\mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O))$ is just a renaming of the terms in $\mathsf{Rel}_{\mathcal{K}\!a}(O)$, so a new interpretation that maps the renamed terms to the original interpretation of the original terms satisfies the renamed ontology.[2]

Due to Theorem 2.3, $\mathcal{I}_{\mathsf{Rel}}$ can be extended to include any additional elements in its domain while remaining a model of $\mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O))$.

Similarly, $\mathsf{Cx}(\mathcal{K}\!a)$ is like $\mathcal{K}\!a$ with a renamed anchor so an interpretation $\mathcal{I}_{\mathfrak{A}}$ that interprets $\mathfrak{A}(\mathcal{K}\!a)$ as $\mathbf{a}^{\mathcal{I}_c}$ and coincides with $\mathcal{I}_c$ on all other terms must be a model of $\mathsf{Cx}_{\mathcal{K}\!a}$. Moreover, we assumed that $\mathcal{K}\!a$ is model extensible so that $\mathcal{I}_{\mathfrak{A}}$ can be extended to include any additional elements.

We thus define an interpretation $\mathcal{I}' = \langle \Delta^{\mathcal{I}_o} \cup \Delta^{\mathcal{I}_c}, \cdot^{\mathcal{I}'} \rangle$ that extends both $\mathcal{I}_{\mathsf{Rel}}$ and $\mathcal{I}_{\mathfrak{A}}$ such that:

- for all $t \in \mathsf{Sig}(\mathcal{K}\!a)$, $t^{\mathcal{I}'} = t^{\mathcal{I}_c}$;
- for all $t \in \mathsf{Sig}(O)$, $\mathsf{ren}_{\mathcal{K}\!a}(t)^{\mathcal{I}'} = t^{\mathcal{I}_o}$, and $t^{\mathcal{I}'}$ is an arbitrary part of $\Delta^{\mathcal{I}_o} \cup \Delta^{\mathcal{I}_c}$;
- $\mathfrak{A}(\mathcal{K}\!a)^{\mathcal{I}'} = \mathbf{a}^{\mathcal{I}_c}$;
- $\mathtt{isContextualPartOf}^{\mathcal{I}'_r} = \{\langle \mathsf{ren}_{\mathcal{K}\!a}(t)^{\mathcal{I}'_u}, t^{\mathcal{I}'_u} \rangle \mid t \in \mathsf{Sig}(O)\}$;
- $\mathtt{isInContext}^{\mathcal{I}'_r} = \{\langle \mathsf{ren}_{\mathcal{K}\!a}(t)^{\mathcal{I}'_u}, \mathfrak{A}(\mathcal{K}\!a)^{\mathcal{I}'_u} \rangle \mid t \in \mathsf{Sig}(O)\}$.

Let us prove that this interpretation is a model of $f_{\mathsf{nd}}(O_{\mathcal{K}\!a})$. Due to the domain extensibility of $\mathcal{I}_{\mathsf{Rel}}$ and $\mathcal{I}_{\mathfrak{A}}$, $\mathcal{I}'$ remains a model of $\mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O))$ and of $\mathsf{Cx}(\mathcal{K}\!a)$

Additionally, the axioms $\mathtt{isContextualPartOf}(\mathsf{ren}_{\mathcal{K}\!a}(t), t)$ and $\mathtt{isInContext}(\mathsf{ren}_{\mathcal{K}\!a}(t), \mathfrak{A}(\mathcal{K}\!a))$ are satisfied for all $t \in \mathsf{Sig}(\alpha)$ by definition of $\mathtt{isContextualPartOf}^{\mathcal{I}'_r}$ and $\mathtt{isInContext}^{\mathcal{I}'_r}$.

□

2.4.3 Inconsistency Preservation

In this section we prove that *NdTerms* preserves inconsistencies. As in Section 2.4.2, we fix a contextual annotation $\mathcal{K}\!a$ that has its signature in N^{nc}.

Theorem 2.4 (Inconsistency preservation of *NdTerms*)**.** *The contextualization function f_{nd} preserves inconsistencies.*

Proof of Theorem 2.4. We prove the theorem by contraposition, that is, if $f(O_{\mathcal{K}\!a})$ is consistent, then O is consistent.

Let $O_{\mathcal{K}\!a} = \langle O, \mathcal{K}\!a \rangle$ be an annotated ontology. Let us assume that $f(O_{\mathcal{K}\!a})$ is consistent. There exists a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ that satisfies $\mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O))$. By definition, $\mathsf{ren}_{\mathcal{K}\!a}$ is injective so there exists an inverse function $\mathsf{ren}^{-}_{\mathcal{K}\!a} : \mathsf{ren}_{\mathcal{K}\!a}(\mathsf{N}^{\mathsf{nc}}) \to \mathsf{N}^{\mathsf{nc}}$ which is itself injective, so that renaming terms $t \in \mathsf{Sig}(\mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O))) \cap \mathsf{ren}(\mathsf{N}^{\mathsf{nc}})$ by $\mathsf{ren}^{-}_{\mathcal{K}\!a}(t)$ gives us a model $\mathcal{I}' = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}'} \rangle$ of ontology $\mathsf{Rel}_{\mathcal{K}\!a}(O)$ where $\mathsf{ren}^{-}_{\mathcal{K}\!a}(t)^{\mathcal{I}'} = t^{\mathcal{I}}$ for all $t \in \mathsf{Sig}(\mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O)))$.

From $\mathcal{I}'$, we can define another interpretation $\mathcal{I}'' = \langle \top^{\mathcal{I}}_{\mathcal{K}\!a}, \cdot^{\mathcal{I}'} \rangle$. For this to be a valid interpretation, all individuals must be interpreted as elements of $\top^{\mathcal{I}}_{\mathcal{K}\!a}$, all concepts must be interpreted as subsets of $\top^{\mathcal{I}}_{\mathcal{K}\!a}$, and all roles must be interpreted as subsets of $\top^{\mathcal{I}}_{\mathcal{K}\!a} \times \top^{\mathcal{I}}_{\mathcal{K}\!a}$. But this is necessarily the case because the interpretation function is that of $\mathcal{I}'$, and $\mathcal{I}'$ is a model of $\mathsf{Rel}_{\mathcal{K}\!a}(O)$ where the extra axioms in Definition 2.10 guarantee this property.

[2] Since "truth is invariant under change of notation" [61, Section 2, p.7].

Now, we prove that $\mathcal{I}'' \models \mathsf{Rel}_{\mathcal{K}\!a}(O)$. Indeed, as explained before, axioms added by Definition 2.10 are satisfied. Then, it can be checked by structural induction that for all concepts or roles X, $\mathsf{rel}_{\mathcal{K}\!a}(X)^{\mathcal{I}''} = \mathsf{rel}_{\mathcal{K}\!a}(X)^{\mathcal{I}'}$. Moreover, all remaining axioms in $\mathsf{Rel}_{\mathcal{K}\!a}(O)$ (other than those from Definition 2.10) are the result of applying the relativization function on concepts and roles. Thus, all axioms of $\mathsf{Rel}_{\mathcal{K}\!a}(O)$ are of one of the following forms: $\mathsf{rel}_{\mathcal{K}\!a}(X) \sqsubseteq \mathsf{rel}_{\mathcal{K}\!a}(Y)$, $\mathsf{rel}_{\mathcal{K}\!a}(C)(x)$, or $\mathsf{rel}_{\mathcal{K}\!a}(R)(x,y)$, for X, Y two concepts or two roles, C a concept, R a role, and x, y two individuals. Since $\mathcal{I}'$ is a model of $\mathsf{Rel}_{\mathcal{K}\!a}(O)$, if $C \sqsubseteq D \in \mathsf{Ax}(O)$, then $\mathcal{I}' \models \mathsf{rel}_{\mathcal{K}\!a}(C) \sqsubseteq \mathsf{rel}_{\mathcal{K}\!a}(D)$. This, combined with the equality of $\mathsf{rel}_{\mathcal{K}\!a}(X)^{\mathcal{I}''}$ and $\mathsf{rel}_{\mathcal{K}\!a}(X)^{\mathcal{I}'}$ for any concept or role X, ensures that $\mathcal{I}'' \models \mathsf{rel}_{\mathcal{K}\!a}(C) \sqsubseteq \mathsf{rel}_{\mathcal{K}\!a}(D)$. The same line of reasoning holds for $C(x) \in \mathsf{Ax}(O)$ and $R(x,y) \in \mathsf{Ax}(O)$. It follows that $\mathcal{I}'' \models \mathsf{Rel}_{\mathcal{K}\!a}(O)$.

We finish the proof by showing that $\mathcal{I}'' \models O$, which proves that O is consistent. Since $\top^{\mathcal{I}''} = \top_{\mathcal{K}\!a}^{\mathcal{I}''}$, adding $\sqcap \top_{\mathcal{K}\!a}$ to a concept does not change its interpretation, and replacing $\top$ with $\top_{\mathcal{K}\!a}$ has no effect on the interpretation of the concepts or roles. Consequently, for all concepts or roles X, $X^{\mathcal{I}''} = \mathsf{rel}_{\mathcal{K}\!a}(X)^{\mathcal{I}''}$. Thus, when $\mathcal{I}'' \models \mathsf{Rel}_{\mathcal{K}\!a}(\alpha)$ it also satisfies α and therefore, $\mathcal{I}'' \models O$. □ □

2.4.4 Inference Preservation

In [57], we were only able to study entailment preservation in the limited setting of $pD*$ entailment.[k] Here we prove a much stronger theorem for *NdTerms*. As in the two previous subsections, we fix a contextual annotation $\mathcal{K}\!a$ that has its signature in N^{nc}.

Theorem 2.5 (Entailment preservation of *NdTerms*)**.** *Let Ω be a set of ontologies having their signatures in N^{nc} and disjoint from the signature of $\mathcal{K}\!a$. If $\mathcal{K}\!a$ is model extensible, then NdTerms is entailment preserving for $\{\langle O, \mathcal{K}\!a\rangle\}_{O\in\Omega}$.*

In order to prove this theorem, we must show that relativization preserves entailments.

Lemma 2.2. *Let O_1 and O_2 be two ontologies. If $O_1 \models O_2$ then $\mathsf{Rel}_{\mathcal{K}\!a}(O_1) \models \mathsf{Rel}_{\mathcal{K}\!a}(O_2)$.*

Proof of Lemma 2.2. Let O_1 and O_2 such that $O_1 \models O_2$. If $\mathsf{Rel}_{\mathcal{K}\!a}(O_1)$ is inconsistent, then the property is obviously verified. Otherwise, there exists a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$ of $\mathsf{Rel}_{\mathcal{K}\!a}(O_1)$. Using the same arguments as in the proof of Theorem 2.4, $\mathcal{I}' = \langle \top_{\mathcal{K}\!a}, \cdot^{\mathcal{I}}\rangle$ is also a model of $\mathsf{Rel}_{\mathcal{K}\!a}(O_1)$ and $\mathcal{I}'$ interprets concepts and roles equally to their relativized counterparts, *i.e.*, $X^{\mathcal{I}'} = \mathsf{rel}_{\mathcal{K}\!a}(X)^{\mathcal{I}'}$. It follows that $\mathcal{I}' \models O_1$ and since $O_1 \models O_2$, $\mathcal{I}' \models O_2$, and $\mathcal{I}' \models \mathsf{Rel}_{\mathcal{K}\!a}(O_2)$. By Theorem 2.3, we know that $\mathcal{I}'$ is domain extensible for $\mathsf{Rel}_{\mathcal{K}\!a}(O_2)$. Consequently, $\mathcal{I}$ is also a model of $\mathsf{Rel}_{\mathcal{K}\!a}(O_2)$. □

With the lemma proven, we then prove Theorem 2.5.

Proof of Theorem 2.5. Let us assume that $\mathcal{K}\!a$ is model extensible. Let O_1 and O_2 be two ontologies having their signatures in N^{nc} and disjoint from $\mathsf{Sig}(\mathcal{K}\!a)$, such that $O_1 \models O_2$. If $f(\langle O_1, \mathcal{K}\!a\rangle)$ is inconsistent, then the entailment is trivially preserved. Otherwise, there exists a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$ of $f(\langle O_1, \mathcal{K}\!a\rangle)$. By definition of f, it satisfies $\mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O_1)$. Due to Lemma 2.2, $O_1 \models O_2$ implies that $\mathsf{Rel}_{\mathcal{K}\!a}(O_1) \models \mathsf{Rel}_{\mathcal{K}\!a}(O_2)$. Moreover, using the fact that $\mathsf{ren}_{\mathcal{K}\!a}$ is just a renaming of terms, and considering that "truth is invariant under change of notation" [61], we know that $\mathsf{ren}_{\mathcal{K}\!a}(O) \models \mathsf{ren}_{\mathcal{K}\!a}(O')$ is equivalent to $O \models O'$. Therefore, $\mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O_1) \models \mathsf{ren}_{\mathcal{K}\!a}(\mathsf{Rel}_{\mathcal{K}\!a}(O_2)$. Moreover, all axioms in $\mathsf{Cx}(\mathcal{K}\!a)$ are satisfied by $\mathcal{I}$. Finally, for O_2 to be entailed by O_1, the

[k] This work will be presented in Chapter 3.

signature of O_2 must be included in the signature of O_1. As a result, whenever $\texttt{isContextualPartOf}(\mathsf{ren}_{\mathcal{K}}, t)$ or $\texttt{isInContext}(\mathsf{ren}_{\mathcal{K}}(t), \mathfrak{A}(\mathcal{K}))$ are in $f(\langle O_2, \mathcal{K}\rangle)$, then they are in $f(\langle O_1, \mathcal{K}\rangle)$ as well. Consequently, $f(\langle O_1, \mathcal{K}\rangle) \models f(\langle O_2, \mathcal{K}\rangle)$. □

2.5 Annotations in Multiple Contexts

So far, we assumed that all axioms of an ontology are annotated with the same contextual information. In this setting, the core of the contextualization function in *NdTerms* amounts to relativizing the axioms and renaming the terms. The renaming part may seem surprising because, as we said a couple of times already, "truth is invariant under change of notation". However, the usefulness of the renaming part becomes apparent when we want to combine several annotated ontologies having different contextual annotations (say $\mathcal{A}_1$ and $\mathcal{A}_2$). In this case, if the renaming functions $\mathsf{ren}_{\mathcal{A}_1}$ and $\mathsf{ren}_{\mathcal{A}_2}$ are mapping non contextual terms into disjoint sets of contextual terms, then the contextualization function f_{nd} ensures that any inference made in a context will not interact with the knowledge from another context. This avoids the contextualized knowledge to be inconsistent when combining statements in different contexts that contradict each others.

The properties presented in Section 2.3 require a little adaptation when applied to the multi-contextual setting. Indeed, in spite of the soundness theorem of Section 2.4.2, in the general case if a set of annotated ontologies $\{\langle O_i, \mathcal{A}_i\rangle\}_{i\in I}$ are satisfying the constraints of Theorem 2.1, it is still possible that $\bigcup_{i\in I} f_{\mathsf{nd}}(\langle O_i, \mathcal{A}_i\rangle)$ is inconsistent. We expect that the preservation of consistency can be guaranteed if *all* the signatures of $\{O_i\}$ are disjoint from *all* the signatures of $\{\mathcal{A}_i\}$. Studying in more details the case of multiple contextual annotations is planned for future work.

2.6 Other Approaches

Here we briefly present the most relevant reification approaches in the Semantic Web. For all of them, the contextualization only annotates the role assertions, leaving other axioms unmodified.[1]

As seen in Example 2.4, *RDF reification* replaces $\alpha = R(x, y)$ with three new role assertions $\texttt{subject}(\mathbf{a}_\alpha^{\mathcal{A}}, x)$, $\texttt{predicate}(\mathbf{a}_\alpha^{\mathcal{A}}, R)$, and $\texttt{object}(\mathbf{a}_\alpha^{\mathcal{A}}, y)$ and the axioms in the contextual annotation are anchored on $\mathbf{a}_\alpha^{\mathcal{A}}$, which depends on the role assertion R and the contextual annotation $\mathcal{A}$. As shown in Example 2.6 and 2.7, this contextualization method preserves neither inconsistencies (in the sense of Definition 2.7) nor entailments on role assertions.

Example 2.10. *An* RDF reification *contextualization of our running example within the context $\mathcal{CA}$ of Example 2.2 contains the following axioms:*

$\texttt{subject}(\texttt{stbcobe}, \texttt{babylon})$
$\texttt{predicate}(\texttt{stbcobe}, \texttt{capital})$
$\texttt{object}(\texttt{stbcobe}, \texttt{babylonianEmpire})$
$\texttt{validity}(\texttt{stbcobe}, \texttt{t})$
$\texttt{Interval}(\texttt{t})$
$\texttt{from}(\texttt{t}, \texttt{609BC})$
$\texttt{to}(\texttt{t}, \texttt{539BC})$

[1] These approaches were already mentioned but not fully described in Section 1.7 of Chapter 1, since that section was giving a general overview about representing and reasoning with context in logical formulas. Here we focus in existing approaches that can be applied in Description Logics.

$\texttt{prov}(\texttt{stbcobe},\texttt{w})$
$\texttt{name}(\texttt{w},\texttt{wikipedia})$
$\texttt{Wiki}(\texttt{w})$

N-Ary relations replaces $R(x,y)$ by two role assertions $p_1(R)(x,\mathbf{a}_\alpha^{\mathcal{G}})$ and $p_2(R)(\mathbf{a}_\alpha^{\mathcal{G}},y)$, where R is a simple role assertion, and p_1 and p_2 are two injective functions with disjoint ranges that map non-contextual roles to contextual roles. Alternatively, a new concept C_R is added for the role R, and the following assertions are added: $C_R(\mathbf{a}_\alpha^{\mathcal{G}})$, $p_1(R)(\mathbf{a}_\alpha^{\mathcal{G}},x)$, and $p_2(R)(\mathbf{a}_\alpha^{\mathcal{G}},y)$.

Example 2.11. *An* N-Ary relations *contextualization of our running example within the context $\mathcal{CA}$ of Example 2.2 contains the following axioms:*

$\texttt{capitalOf\#1}(\texttt{babylon},\texttt{rbcobe})$
$\texttt{capitalOf\#2}(\texttt{rbcobe},\texttt{babylonianEmpire})$
$\texttt{validity}(\texttt{rbcobe},\texttt{t})$
$\texttt{Interval}(\texttt{t})$
$\texttt{from}(\texttt{t},\texttt{609BC})$
$\texttt{to}(\texttt{t},\texttt{539BC})$
$\texttt{prov}(\texttt{rbcobe},\texttt{w}$
$\texttt{name}(\texttt{w},\texttt{wikipedia})$
$\texttt{Wiki}(\texttt{w})$

Singleton property is using a non-standard semantics of RDF but the same idea can be simulated with DL axioms. For each simple role axiom $R(x,y)$, the following axioms are added: $\mathbf{a}_\alpha^{\mathcal{G}}(x,y)$ (that is, the term for the anchor is used as a role), $\mathbf{a} \equiv \{x\} \times \{y\}$ (which guarantees that the anchor property is a singleton), and $\texttt{singletonPropertyOf}(\mathbf{a}_\alpha^{\mathcal{G}},R)$.

Example 2.12. *A* Singleton Property *contextualization of our running example within the context $\mathcal{CA}$ of Example 2.2 contains the following axioms:*

$\texttt{capital\#1}(\texttt{babylon},\texttt{babylonianEmpire})$
$\texttt{singletonPropertyOf}(\texttt{capital\#1},\texttt{capital})$
$\texttt{validity}(\texttt{capital\#1},\texttt{t})$
$\texttt{Interval}(\texttt{t})$
$\texttt{from}(\texttt{t},\texttt{609BC})$
$\texttt{to}(\texttt{t},\texttt{539BC})$
$\texttt{prov}(\texttt{capital\#1},\texttt{w}$
$\texttt{name}(\texttt{w},\texttt{wikipedia})$
$\texttt{Wiki}(\texttt{w})$

The remaining approach, *NdFluents*, uses a similar approach as *NdTerms* except that it only renames the terms used as individuals and does not relativize the ontology. This ensures interesting properties with regards to entailment preservation [57], but TBox axioms in different contexts are not distinguishable.

2.7 Discussion and Future Work

NdTerms and NdFluents are a concrete instantiations of a family of contextualizations based on contextualizing (parts of) the terms in the ontology. We call this family of contextualizations $\mathbf{Nd}^{*\mathrm{m}}$. Other instantiations would be possible, such

[m]This family of contextualizations was not given a name in the publications, but it was a concept that was continually referred. We have added it here for the sake of clarity and continuity among the chapters in this Part. We will keep exploring it for different subsets of terms in OWL and RDF in the next chapters.

as contextualizing role names (in a similar fashion as the singleton property), class names, or a combination of them. Then, *NdTerms* would be the approach where each and every term is contextualized, while in NdFluents only individuals are.

In the future, we would like to deepen the analysis of contextualization, filling gaps still present in this preliminary work. Especially, the combination of multiple annotations, or annotations of contextualized ontologies, present some interesting challenges. A more systematic comparison of the various approaches remains to be presented.

References

[15] Brickley, D., Guha, R.: RDF Schema 1.1, pp. 91–122. W3C (2014). DOI: 10.1016/B978-0-12-373556-0.00006-X URL: http://www.w3.org/TR/rdf-schema/

[23] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. Journal of Web Semantics 3(4), 247–267 (2005).

[27] Cuenca-Grau, B., Kutz, O.: Modular Ontology Languages Revisisted. In: Honavar, V.G., Finin, T.W., Caragea, D., Mladenic, D., Sure, Y. (eds.) SWeCKa 2007: Proceedings of the IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition, Hyderabad, India, January 7, 2007, (2007).

[37] Dividino, R., Sizov, S., Staab, S., Schueler, B.: Querying for provenance, trust, uncertainty and other meta knowledge in RDF. Journal of Web Semantics 7(3), 204–219 (2009).

[50] Ghidini, C., Giunchiglia, F.: Local Models Semantics, or contextual reasoning=Locality+Compatibility. Artificial Intelligence 127(2), 221–259 (2001).

[57] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC), pp. 638–654. Springer, Cham (2017).

[59] Giunchiglia, F.: Contextual Reasoning. Epistemologica 16, 345–364 (1993).

[61] Goguen, J.A., Burstall, R.M.: Institutions: Abstract Model Theory for Specification and Programming. Journal of the Association for Computing Machinery 39(1), 95–146 (1992).

[62] Golbreich, C., Wallace, E.K.: OWL 2 web ontology language, new features and rationale (second edition), W3C (2012). URL: https://www.w3.org/TR/2012/REC-owl2-new-features-20121211/

[68] Hartig, O., Champin, P.-A., Kellogg, G., Seaborne, A., Arndt, D., Broekstra, J., DuCharme, B., Lassila, O., Patel-Schneider, P.F., Prud'hommeaux, E., Thibodeau, T., Thompson, B.: RDF-star and SPARQL-star, W3C (2021). URL: https://w3c.github.io/rdf-star/cg-spec

[69] Hartig, O., Thompson, B.: Foundations of an Alternative Approach to Reification in RDF, pp. 14–14. arXiv (2014). URL: http://arxiv.org/abs/1406.3399

[76] Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artificial Intelligence 194, 28–61 (2013).

[85] Kifer, M., Subrahmanian, V.S.: Theory of Generalized Annotated Logic Programming and its Applications. Journal of Logic Programming 12(3), 335–367 (1992).

[99] McCarthy, J.: Generality in Artificial Intelligence. Communications of the ACM 30(12), 1030–1035 (1987).

[108] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. In: Proceedings of the 23rd International Conference on the World Wide Web (WWW), pp. 759–770. ACM (2014).

[110] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web, pp. 1–1. W3C (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/

[122] Scott, D.: Identity and existence in intuitionistic logic. In: Fourman, M., Mulvay, C., Scott, D.S. (eds.) Applications of Sheaf Theory to Algebra, Analysis and Topology, Lecture Notes in Mathematics, Vol. 753Lecture Notes in Mathematics, pp. 660–696. Springer-Verlag (1979).

[126] Udrea, O., Recupero, D.R., Subrahmanian, V.S.: Annotated RDF. ACM Transaction on Computational Logics 11(2), 10:1–10:41 (2010).

[136] Zimmermann, A.: Logical formalisms for Agreement Technologies. In: Ossowski, S. (ed.) Agreement Technologies, pp. 69–82. Springer-Verlag (2013).

[138] Zimmermann, A., Giménez-García, J.M.: Contextualizing DL Axioms: Formalization, a New Approach, and Its Properties. In: Joint Proceedings of the Web Stream Processing workshop (WSP 2017) and the 2nd International Workshop on Ontology Modularity, Contextuality, and Evolution (WOMoCoE 2017) co-located with 16th International Semantic Web Conference (ISWC 2017), pp. 74–85 (2017).

[139] Zimmermann, A., Giménez-García, J.M.: Integrating Context of Statements within Description Logics, Université Jean Monnet (2017). URL: http://arxiv.org/abs/1709.04970

[141] Zimmermann, A., Lopes, N., Polleres, A., Straccia, U.: A general framework for representing, reasoning and querying with annotated Semantic Web data. Journal of Web Semantics 11, 72–95 (2012).

Chapter 3

NdFluents: An Ontology for Annotated Statements with Inference Preservation*

In this chapter, we take one of the final steps in our journey from First Order Logic to OWL. In previous chapters, for First Order Logic we have presented several possibilities to represent the context of a statement (increasing the arity of the predicates by one to include the context, renaming predicates, and renaming constants). In the case of Description Logics, where only unary and binary predicates are allowed, we have presented NdTerms, that could be seen as a generalization of the renaming predicates and renaming individuals in First Order Logic. In both cases, the link with the context can be made through the constants/individuals. Since OWL is based on Description Logic, we face the same restriction with regards to the arity of the relations. We decide then to implement an approach similar to renaming constants in OWL, presenting the NdFluents ontology, an extension of Welty and Fikes' 4dFluents ontology—which associated temporal validity to statements—to any number of dimensions. In addition, we provide guidelines and design patterns to implement it on actual data, and compare its reasoning power with alternative representations.

3.1 Introduction

The Resource Description Framework (RDF) represents statements as triples that typically match phrases with a subject, a verb and a complement. However, it is often the case that more complex information has to be encoded, such as qualifying a statement with its origin, its validity within a time frame, its degree of certainty, and so on. In this case, one may have to represent statements about a statement. We describe this as an annotated statement. However, with the RDF model it is only possible to represent binary (or dyadic) relations between subject and object [107]. In order to represent additional data about statements it is usually needed to use

*This chapter is based on the following publications:

- Giménez-García, J.M., Zimmermann, A., Maret, P.: Representing Contextual Information as Fluents. In: Knowledge Engineering and Knowledge Management - {EKAW} 2016 Satellite Events, {EKM} and Drift-an-LOD, Bologna, Italy, November 19-23, 2016, Revised Selected Papers, pp. 119–122. Springer, Cham (2017). [58]
- Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC),. Pp. 638–654. Springer, Cham (2017). [57]

external annotations, extend either the data model [22] or the semantics of RDF [69, 108], or use design patterns to represent that information [110, 48].

On the other hand, RDF Schema (RDFS) and the Web Ontology Language (OWL) extend the formal semantics to RDF, making it possible to infer new statements from pre-existing knowledge. However, when data is annotated using the previous approaches, the inferences in the original dataset are no longer possible, or the new inferred data is missing part of the annotations. For instance, OWL allows to define a relation between two resources as transitive. In that case, if a resource A is related to another resource B using that property, and B is in turn related with another resource C with the same property, then it is inferred that A and C are also related. This inference is not preserved when using Reification, a classic approach to reference a triple and annotate it with metadata, that removes the original triple and replaces it with four new triples to identify the statement and describe the position of each element of the original triple.

Along these lines, Welty and Fikes [131] proposed an ontology for representing temporally changing information using a perdurantist view, where statements are asserted over temporal slices of entities, retaining most reasoning capabilities. This approach can be generalized to annotate data not only with temporal information, but with information from any dimension [130]. However, modeling several context dimensions for a statement is not straightforward and presents some challenges. In this work, we propose a generalization of Welty and Fikes model in the form of a generic ontology that can be extended to implement any number of concrete metadata dimensions, while preserving reasoning capacity relative to each dimension.

The rest of the paper is structured as follows: Section 3.2 presents the 4dFluents ontology for annotating statements with temporal data; Section 3.3 introduces NdFluents, the generalization of 4dFluents to annotate statements with any number of context dimensions; Section 3.4 describes three design patterns that can be used to model a combination of context dimensions; Section 3.5 discusses issues and possible solutions when representing metadata with NdFluents; Section 3.6 compares the reasoning capabilities of NdFluents with other current approaches to represent metadata about statements in RDF; Section 3.7 portrays related work; finally, we present some conclusions in Section 3.8.

3.2 Welty and Fikes' 4dFluents Ontology

Welty and Fikes [131] address the problem of representing *fluents*, *i.e.*, relations that hold within a certain time interval and not in others. They address the issue from the perspective of diachronic identity (that is, how an entity looks to be different at different times), showcasing the two ways of tackling it:

- The *endurantist* (*3D*) view maintains a differentiation between *endurants*, entities that are present at all times during its whole existence, and *perdurants*, events affecting an entity during a definite period of time during the entity's existence.

- The *perdurantist* (*4D*) view argues that entities themselves have to be handled as perdurants, *i.e.*, temporal parts of a four dimensional meta-entity. Instead of making an assertion about some entities, such as *"Paris is the capital of France"*, one should make the assertion about their temporal parts: *"A temporal part of Paris (since 508 up to now) is the capital of a temporal part of France (since 508 up to now)"*.

```
Prefix( 4d:=<http://www.example.com/4dFluents#> )
Ontology( <http://www.example.com/4dFluents>
        Declaration( Class( 4d:Interval ) )
        Declaration( Class( 4d:TemporalPart ) )
        DisjointClasses( 4d:Interval 4d:TemporalPart )

        Declaration( ObjectProperty( 4d:fluentProperty ) )
        ObjectPropertyDomain( 4d:fluentProperty 4d:TemporalPart )
        ObjectPropertyRange( 4d:fluentProperty 4d:TemporalPart )

        Declaration( ObjectProperty( 4d:temporalExtent ) )
        FunctionalObjectProperty( 4d:temporalExtent )
        ObjectPropertyDomain( 4d:temporalExtent 4d:TemporalPart )
        ObjectPropertyRange( 4d:temporalExtent 4d:Interval )

        Declaration( ObjectProperty( 4d:temporalPartOf ) )
        FunctionalObjectProperty( 4d:temporalPartOf )
        ObjectPropertyDomain( 4d:temporalPartOf 4d:TemporalPart )
        ObjectPropertyRange( 4d:temporalPartOf ObjectComplementOf( 4d:Interval ))
)
```

Ontology 3.1: 4dFluents ontology (from [131])

```
Declaration( ObjectProperty( ex:capitalOf ) )
SubObjectPropertyOf( ex:capitalOf 4d:fluentProperty )
ClassAssertion( 4d:TermporalPart ex:Paris@508 )
ClassAssertion( 4d:TermporalPart ex:France@508 )
ClassAssertion( 4d:Interval ex:year508) )
ObjectPropertyAssertion( ex:capitalOf ex:Paris@508 ex:France@508 )
ObjectPropertyAssertion( 4d:temporalExtent ex:Paris@508 ex:year508 )
ObjectPropertyAssertion( 4d:temporalExtent ex:France@508 ex:year508 )
ObjectPropertyAssertion( 4d:temporalPartOf ex:Paris@508 ex:Paris )
ObjectPropertyAssertion( 4d:temporalPartOf ex:France@508 ex:France )
```

Ontology 3.2: Expressing a fact about a fluent entity with the 4dFluents ontology

Welty and Fikes adopt the perdurantist approach to create the *4dFluents* ontology, representing *entities at a time* and using them as resources for their statements. The 4dFluents ontology expressed in OWL2 Functional Syntax is shown in Ontology 3.1.

In order to use the ontology for describing fluents, one has to introduce axioms at the terminological level (TBox) as well as assertions in the knowledge base (ABox). For instance, if one wants to say that *"Paris is the capital of France"* since 508, the relation "capital of" has to be a subproperty of `fluentProperty` and new individuals have to be introduced for the temporal part of Paris and of France, as shown in Ontology 3.2.

In this way, temporal information can be represented with standard OWL semantics, preserving reasoning capabilities.

3.3 The NdFluents Ontology

A temporal part of an entity can be viewed as an individual context dimension of the entity. A similar approach can then be used to represent different dimensions, such as provenance or confidence. Continuing with our running example, if Wikipedia states that *"Paris is the capital of France"*, we can articulate that fact as *"Paris as defined by Wikipedia is the capital of France as defined by Wikipedia"*. Different context dimensions of an entity could then be combined if applicable, allowing the representation

```
Prefix( nd:=<http://purl.org/NET/NdFluents#> )
Ontology( <http://purl.org/NET/NdFluents>
        Declaration( Class( nd:Context ) )
        Declaration( Class( nd:ContextualPart ) )
        DisjointClasses( nd:Context nd:ContextualPart )

        Declaration( ObjectProperty( nd:contextualProperty ) )
        ObjectPropertyDomain( nd:contextualProperty nd:ContextualPart )
        ObjectPropertyRange( nd:contextualProperty nd:ContextualPart )

        Declaration( ObjectProperty( nd:contextualExtent ) )
        ObjectPropertyDomain( nd:contextualExtent nd:ContextualPart )
        ObjectPropertyRange( nd:contextualExtent nd:Context )

        Declaration( ObjectProperty( nd:contextualPartOf ) )
        FunctionalObjectProperty( nd:contextualPartOf )
        ObjectPropertyDomain( nd:contextualPartOf nd:ContextualPart )
        ObjectPropertyRange( nd:contextualPartOf ObjectComplementOf( nd:Context ))
)
```

Ontology 3.3: The NdFluents ontology

of complex information, such as: *"A temporal part of Paris as defined by Wikipedia is the capital of a temporal part of France as defined by Wikipedia"*.

We use this idea to extend the 4dFluents ontology for an arbitrary number of context dimensions in the *NdFluents* ontology. The ontology, shown in Ontology 3.3, and published in `https://github.com/jm-gimenez-garcia/NdFluents`[a], is a generalization from temporal parts to contextual parts.

The concept of creating a fresh term that depends on both the context and the individual that we want to contextualize is a concept equivalent to the *renaming function* given in Definition 1.9 of Chapter 1. This function has its equivalent in Description Logics in Section 2.4 of Chapter 2, and will be formalized for RDF in Chapter 4. The individuals of class `nd:ContextualPart` are used to link the statement to its context. They are what we call the *Anchor* in Chapter 2.[b]

Note that `FunctionalObjectProperty( nd:contextualExtent )` axiom is not present in the ontology. This axiom should appear if the ontology was a direct translation from temporal dimension to a generic context dimension, but it is no longer applicable when we have more than one dimension simultaneously.

The NdFluents ontology is meant to be implemented for different context dimensions in a modular way. In this sense, the 4dFluents ontology can be seen as a concrete implementation of NdFluents, as we show in Ontology 3.4. In Figure 3.1a we show the representation of a statement with temporal annotations using this ontology. The non-dashed parts are equivalent to the original 4dFluents ontology, while the dashed parts correspond to the NdFluents extension. Other dimensions, such as provenance, can be modeled similarly to the temporal dimension by replacing `TemporalPart` with `ProvenancePart`, `temporalExtent` with `provenanceExtent`, `Interval` with `Provenance`, and `temporalPartOf` with `provenancePartOf`. Additionally, an assertion like *"Paris is the capital of France, according to Wikipedia"* can be modeled following the same pattern as in Ontology 3.2, replacing the property and class names with their counterparts in the provenance dimension.

[a] Updated from the original paper.

[b] This paragraph has been added to the chapter to make the link with the rest of the work in the dissertation.

```
Prefix( nd:=<http://purl.org/NET/ndfluents#> )
Prefix( 4d:=<http://purl.org/NET/ndfluents/4dFluents#>)
Ontology( <http://www.example.com/4dFluentsV2>
        Import( <http://www.example.com/NdFluents> )

        Declaration( Class( 4d:Interval ) )
        SubClassOf( 4d:Interval nd:Context )
        Declaration( Class( 4d:TemporalPart ) )
        SubClassOf( 4d:TemporalPart nd:ContextualPart )

        Declaration( ObjectProperty( 4d:temporalExtent ) )
        SubObjectPropertyOf( 4d:temporalExtent nd:contextualExtent )
        ObjectPropertyDomain( 4d:temporalExtent 4d:TemporalPart )
        ObjectPropertyRange( 4d:temporalExtent 4d:Interval )

        Declaration( ObjectProperty( :temporalPartOf ) )
        SubObjectPropertyOf( 4d:temporalExtent nd:contextualPartOf )
        ObjectPropertyDomain( 4d:temporalPartOf 4d:TemporalPart )
)
```

Ontology 3.4: 4dFluents ontology as implementation of NdFluents

3.4 Design Patterns

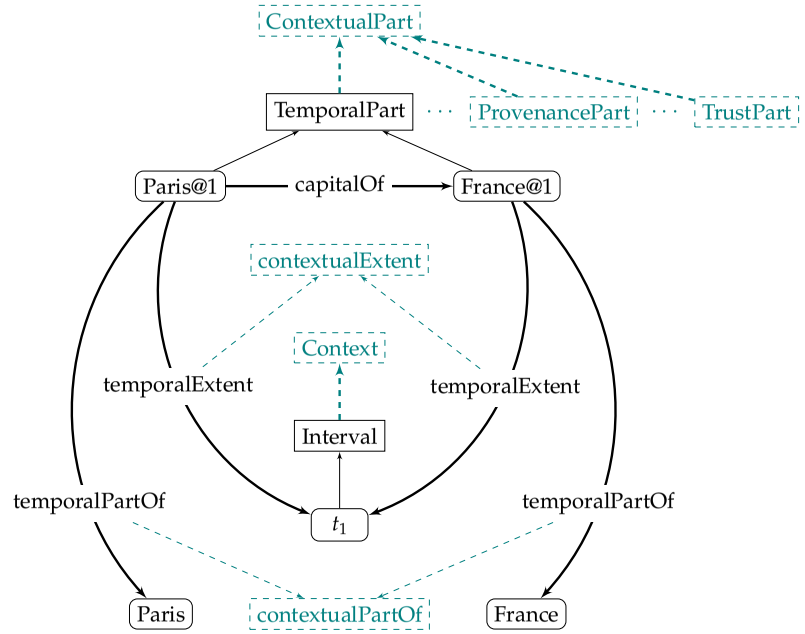
An important scenario where NdFluents becomes relevant is when the necessity of combining two or more context dimensions arises, such as *"According to Wikipedia, Paris is the capital of France since 508"*. In this section we present three design patterns to combine different dimensions, along with added axioms that can be necessary depending on the modeling needs. Methodological support for choosing and implementing a design pattern can be found at Giménez-García, Zimmermann, and Maret [56]

3.4.1 Contexts in Context

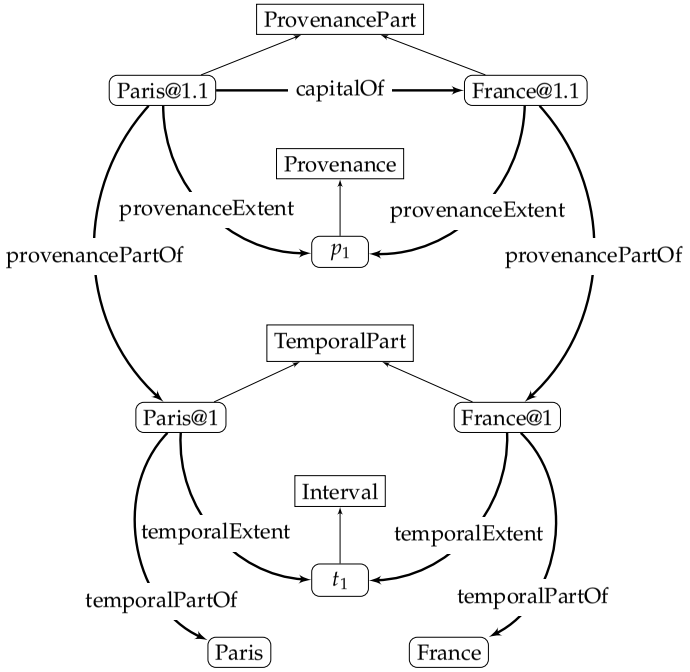
One possible model to represent information using different context dimensions is to relate a `ContextualPart` to another `ContextualPart`. This approach can be taken when the "first level" annotations are relevant facts of the knowledge base, and the intention is to state additional information about them. To be able to reason about different annotation levels of any entity, it is desirable for the `contextualPartOf` property to be transitive, which can be achieved by adding the axiom of Ontology 3.5.

The fact that statements can be made between contextual parts of different levels without affecting other annotation levels allows this model to be more fine-grained, but it can also make it grow in complexity. For example, in Figure 3.1b the statement `capitalOf` is related to the `ProvenancePart` `Paris@1.1`. This information is in no way related to the `TemporalPart` `Paris@1`. In this case this is probably the intended goal of the representation, and nothing else needs to be done. However, it is also true that statements about contextual parts in higher levels do not extend to their own contextual parts. That means that statements related to `Paris@1` have no effect on the information about `Paris@1.1`. While this kind of statements could be duplicated, this can become unfeasible when we start adding more contextual parts to the data. We believe that this pattern can be useful in some specific cases, but it is usually too cumbersome.[c]

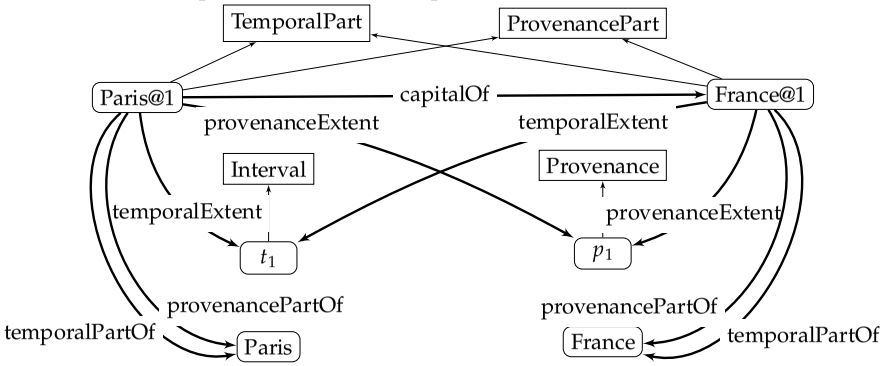
[c]This paragraph has been rewritten from the original paper to better explain the pros and cons of this approach.



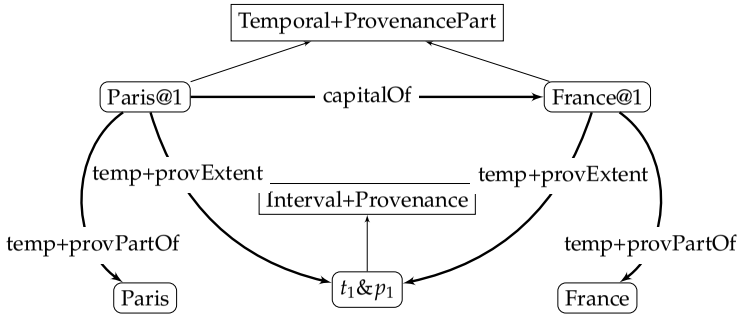
(a) Example of 4dFluents as implementation of NdFluents



(b) Contexts in context



(c) Multiple Contexts on each ContextualPart



(d) Combination of different dimensions on one contextual extent

Figure 3.1: NdFluents ontology and design patterns

```
Prefix( nd:=<http://purl.org/NET/ndfluents#> )
Ontology( <http://purl.org/NET/ndfluents/transitivecontextualpartof>
        TransitiveObjectProperty( nd:contextualPartOf )
)
```

Ontology 3.5: Transitive axiom for NdFluents ontology

```
Prefix( nd:=<http://purl.org/NET/ndfluents#> )
Ontology( <http://purl.org/NET/ndfluents/functionalcontextualExtent>
        FunctionalObjectProperty( nd:contextualExtent )
)
```

Ontology 3.6: Functional contextual extents axiom for NdFluents ontology

3.4.2 Use Multiple Contextual Extents on each Contextual Part

A generic approach for representing entities with more than one context dimension is to have `ContextualParts` with more than one contextual extent. Using this model, only one `ContextualPart` is created for a combination of context dimensions. This `ContextualPart` is then related to all related contextual extents, as shown in Figure 3.1c. This pattern is easier to model: Relating the `ContextualPart` with the context dimensions is straightforward. It also avoids ambiguity when modeling annotations related to more than one dimension, and reduces the number of resources in the ontology (*i.e.*, while the previous model needed one `ContextualPart` for each dimension involved, this approach only requires one). Note that `contextualPartOf` is a functional property, which means that there cannot be a `contextualPartOf` of more than one entity.

3.4.3 Combine Different Contexts on one Contextual Extent

Finally, a third possibility is to create compound `Contexts`, and enforce a limit of only one Context per `ContextualPart`. This model adds a layer of complexity to the previous approach, but it can be useful to require a specific combination of dimensions on a set of `ContextualParts`. This can be achieved by adding the axiom in Ontology 3.6.

We show an example of this approach on Figure 3.1d. Note that the combined classes and properties are subclasses and subproperties of the corresponding classes and properties of the two context dimensions they are combining (*e.g.*, `Temporal+ProvenancePart` is subclass of `TemporalPart` and `ProvenancePart`). As a result, querying and reasoning can be performed in an identical way as the previous approach.

3.5 Additional Considerations

In this section we discuss issues that may arise when modeling annotations using fluents, and possible approaches to deal with them if they exist. While the first one is common to the original 4dFluents ontology, the second is only relevant when dealing with more than one context dimension.

```
Prefix( nd:=<http://purl.org/NET/ndfluents#> )
Ontology( <http://purl.org/NET/ndfluents/annotatedDatatypeProperty>
        Declaration( DataProperty( nd:annotatedDatatypeProperty ) )
        DataPropertyDomain ( nd:annotatedDataProperty nd:ContextualPart )
)
```

Ontology 3.7: Datatype axioms for NdFluents ontology

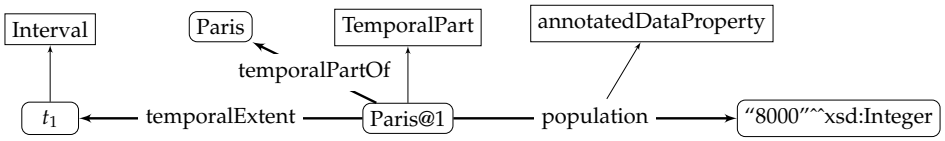


Figure 3.2: Example of Annotated Datatype Property

3.5.1 Dealing with Datatype Properties

The original 4dFluents ontology does not provide any information for modeling datatype properties. While there is nothing that prevents using regular datatype properties with `ContextualParts` of an entity, it may be desirable to declare explicit axioms for annotation properties to facilitate reasoning on that information. In that case, the statements of Ontology 3.7 need to be added to the NdFluents ontology. Figure 3.2 shows an example where a annotated property is used to state the population of Paris in a specific temporal interval. Note that it is also possible to create specific `contextualProperty` subproperties for different context dimensions (*i.e.*, `temporalProperty` for `TemporalPart`) for properties related to concrete context dimensions.

3.5.2 Relations between `ContextualParts` of Different Dimensions

The NdFluents ontology presented thus far allows the modeling of relations among different `ContextualParts` of different dimensions (*i.e.*, a `TemporalPart` of Paris could be the capital of a `ProvenancePart` of France). While this can be convenient for individual cases, it is often needed for a `contextualProperty` to be related to `ContextualParts` of the same dimension. In this case, it is necessary to add the appropriate axioms to the ontology. In Ontology 3.8 we show the needed axioms to include this restriction on the `TemporalParts`. Conversely, if there are datatype properties related to specific dimensions, axioms from Ontology 3.9 should be added.[d]

In a similar fashion, it is usually desirable that `ContextualParts` of the same dimension relate to the same `Context`. That is, if a Provenance Part of Paris relates to

[d]Note that this means that it can be necessary to have different versions of the same property. This would, in essence, lead to an implementation of Nd* where individuals and properties are renamed.

```
Prefix( nd:=<http://purl.org/NET/ndfluents#> )
Prefix( 4d:=<http://purl.org/NET/ndfluents/4dFluents#>)
Ontology( <http://purl.org/NET/ndfluents/4dFluents/temporalpartrestriction>
        Declaration( ObjectProperty( 4d:fluentProperty ) )
        SubObjectPropertyOf( 4d:fluentProperty nd:contextualProperty )
        ObjectPropertyDomain( 4d:fluentProperty 4d:TemporalPart )
        ObjectPropertyRange( 4d:fluentProperty 4d:TemporalPart )
)
```

Ontology 3.8: Temporal restriction on object properties 4dFluents ontology

```
Prefix( nd:=<http://purl.org/NET/ndfluents#> )
Prefix( 4d:=<http://purl.org/NET/ndfluents/4dFluents#>)
Ontology( <http://purl.org/NET/ndfluents/4dFluents/temporalpartrestriction>
        Declaration( DataProperty( 4d:fluentDataTypeProperty ) )
        SubDataPropertyOf( 4d:fluentDataTypeProperty nd:contextualProperty )
        DataPropertyDomain( 4d:fluentProperty 4d:TemporalPart)
)
```

Ontology 3.9: Temporal restriction on datatype properties 4dFluents ontology

a `ProvenancePart` of France, their `provenanceExtent` properties should have the same `ProvenancePart` object. However, this restriction cannot be expressed in OWL. If needed, a rule language (such as SWRL [80] or RIF[84]) can be used for this purpose, but this case goes beyond the scope of this paper.[e]

3.6 Reasoning with Annotated Data

In this section, we compare the reasoning capabilities of the NdFluents ontology with other approaches to annotate statements, namely RDF reification, N-ary relations, and singleton property. The interest is to know what RDFS and OWL entailments are preserved with regards to the original unannotated data. For that, we need to formally define what annotations and entailment preservation mean. We assume that annotated statements can be described as a pair (G, A) where G is the graph corresponding to the statements that are annotated, and A denotes the annotations on G. The structure of A could be arbitrarily complex (e.g., containing dates, creator, provenance) but for the sake of this section and to simplify the presentation, we simply assume that the annotation structure is identified with an IRI. Thus, we approximate the notion of annotated statements with the concept of named graphs, i.e., pairs (n, G) where n is an IRI and G is an RDF graph. However, there is no standard way of reasoning with named graphs [137]. Our objective then is to compare approaches that convert annotated statements into RDF graphs. We name such approaches *RDF representation of annotated statements* and formalize it as follows.

Definition 3.1[f] (RDF representation of annotated statements)**.** *An* RDF representation of annotated statements *is a function f that maps annotated statements (in our simplified model, named graphs) (n, G) to an RDF graph $f(n, G)$.*

For examples of this function, refer to subsection 3.6.1, where we describe four existing models to annotate statements and present their corresponding functions.

We want to assess to what extent each representation is preserving entailment with the notions of entailment preservation (when the entailment preserves also the annotations) and non-contextual entailment preservation (when only the original entailment is preserved) defined as follows.

Definition 3.2 (Entailment preservation)**.** *Let G_1 and G_2 be two RDF graphs such that $G_1 \models G_2$ and f be an RDF representation of annotated statements.*[1] *We say that f* preserves the entailment *between G_1 and G_2 iff for all annotation IRI n, $f(n, G_1) \models f(n, G_2)$.*

[1] This definition can apply to any entailment regime so that it is not necessary to specify what the relation $\models$ exactly is.

[e] Note that this is only a problem if the NdFluents ontology is used to model data "by hand" without using a contextualization function that respects this restriction.

[f] This definition is equivalent to *Contextualization Function* for First Order Logic and Description Logics in Chapters 1 and 2.

Definition 3.3 (Non-Contextual Entailment preservation)**.** *Let G_1 and G_2 be two RDF graphs such that $G_1 \models G_2$ and f be an RDF representation of annotated statements.*[1] *We say that f* non-contextually preserves the entailment *between G_1 and G_2 iff for all annotation IRI n, $f(n, G_1) \models G_2$.*

We generalize these notions to the case of entailment rules of the form $P(\mathbf{x}) \leftarrow Q(\mathbf{x}, \mathbf{y})$, where P and Q are graph patterns and $\mathbf{x}$, $\mathbf{y}$ are tuples of variables used in the patterns.

Definition 3.4 (Rule preservation)**.** *Let $R = P(\mathbf{x}) \leftarrow Q(\mathbf{x}, \mathbf{y})$ be a rule and f an RDF representation of annotated statements. We say that f* preserves the rule *R iff for all mappings μ from variables in $\mathbf{x}$ and $\mathbf{y}$ to RDF terms, $f(n, Q(\mu(x), \mu(y))) \models f(n, P(\mu(x)))$.*

Definition 3.5 (Non-Contextual Rule preservation)**.** *Let $R = P(\mathbf{x}) \leftarrow Q(\mathbf{x}, \mathbf{y})$ be a rule and f an RDF representation of annotated statements. We say that f* non-contextually preserves the rule *R iff for all mappings μ from variables in $\mathbf{x}$ and $\mathbf{y}$ to RDF terms, $f(n, Q(\mu(x), \mu(y))) \models P(\mu(x))$.*

For example, if we have an inference rule that allows us to infer that `(France, hasCapital,Paris)` from the triple `(Paris,capitalOf,France)`, and we have a representation of annotated statements for `(Paris,capitalOf,France),(508,now)`, rule preservation would allow us to infer `(France, hasCapital,Paris),(508,now)`, while non-contextual rule preservation would allow to infer `(France,hasCapital,Paris)` from the annotated triple. This kind of inferences where the triples of the condition are annotated with a context, but the triples in the conclusion are not, are usually not desirable. This will be further explained in subsection 3.6.2.[g]

In the following subsections we first present the *RDF representation of annotated statements* (see Definition 3.1) for the representation approaches, and then proceed to compare the rule preservation for each one of them.

3.6.1 RDF representation approaches

- **Reification**[2] is the standard W3C model to represent information about an statement, proposed in 2004. A triple is represented as an instance of `rdf:Statement`, that relates to the original triple with the properties `rdf:subject`, `rdf:predicate` and `rdf:object`. Then, a triple (s, p, o) is replaced by the following set: $\{(i$,`rdf:type`,`rdf:Statement`$)$, $(i$,`rdf:subject`$, s)$, $(i$,`rdf:predicate`$, p)$, $(i$,`rdf:object`$, o)\}$, and annotations are related to i.

- **N-Ary relations** [110] were proposed in 2006 to represent relations between more than two individuals, or to describe the relations themselves. In this model, an individual is created to represent the relation, which can be used as the subject for new statements. Thus, a triple (s, p, o) is replaced by the following set: $\{(s, p'_1, r), (r, p'_2, o)\}$, and annotations are related to r.

- The **Singleton Property** [108] is a recent proposal to represent information about statements in RDF. A particular instance of the predicate is created for every triple. This instance is related to the original predicate by the `singletonPropertyOf` property. Then, each statement can be unequivocally referenced using its predicate for attaching additional information. Therefore, a

[2] `https://www.w3.org/TR/2004/REC-rdf-primer-20040210/#reification`

[g] We rephrased this sentence to make it more clear.

triple (s, p, o) is replaced by the set: $\{(s, p', o)$, $(p'$,`sp:singletonPropertyOf`$, p)\}$, and annotations are related to p'.

- **NdFluents**, the approach presented in this paper, creates a contextualized individual for both subject and object (in case it is a URI or blank node) of the triple. The triple is the replaced by a new one that uses the contextualized individuals. These two new resources are related to the original individuals and with a Context, where the annotations are attached. Hence, the original triple (s, p, o) is replaced by the following set of triples $\{(s_c, p, o_c)$, $(s_c$,`nd:contextualPartOf`$, s)$, $(o_c$,`nd:contextualPartOf`$, o)$, $(s_c$,`nd:contextualExtent`$, c)$, $(o_c$,`nd:contextualExtent`$, c)\}$, where c is a function of the context. Annotations are related to c.

.

3.6.2 Comparison of rule preservation

For comparing how entailment is preserved in each of the 4 approaches presented in Section 3.6.1, we analyze which rules from the pD* fragment of OWL Horst [81] are preserved. This fragment is a modified subset of RDFS and OWL that can be expressed as a complete set of rules and is computationally feasible. For each rule, we check if it is in accordance with *Rule Preservation* and *Non-Contextual Rule Preservation* (*i.e.*, for the former, if the inference rule holds when we apply the *RDF representation of annotated statements* function to both condition and conclusion; for the latter, if it holds when we apply the function only to the condition). It is important to note that the representation approaches are usually used to annotate data on relations between resources. For this reason, we decide to implement the representations on triples that do not include RDF, RDFS, or OWL vocabularies.

Table 3.1 shows the D* (modified RDFS) entailment rules and rule preservations for each one of the approaches, whereas Table 3.2 presents the same information for P entailments (modified subset of OWL). Note that we remove those rows where both condition and conclusion include only triples with RDF, RDFS, or OWL vocabularies. A P indicates that there is rule preservation for the corresponding approach, while a P_{NC} denotes non-contextual rule preservation. As mentioned in Section 3.6, it is worth noting that not all rule preservations are desirable. When the preserved rule entails new knowledge on the non-annotated graph, and the annotated triples are not universally true, then the inferences can lead to conclusions that do not conform with real-world knowledge. This happens when the *RDF representation of annotated statements* function annotates at least one triple of the condition, and either we have non-contextual rule preservation, or we have rule preservation but the function does not annotate the triple in the conclusion. This is actually what happens with the Singleton Property for the rules `rdfs2`, `rdfs3`, and `rdfs7x` from the D*-entailments ruleset, and rules `rdfp1`, `rdfp2`, `rdfp3`, `rdfp4`, `rdfp8ax`, `rdfp8bx`, `rdfp11`, `rdfp14a`, `rdfp15`, and `rdfp16` (identified in the table with an exclamation mark), due to the RDFS interpretation that considers the singleton property as belonging to the extension of the original property [108, Section 3]. While there is no problem if the annotated fact is universally true (*i.e.*, we just want to provide additional information about a fact), it leads to undesirable conclusions when the context of the annotation is related with the identity of the resources (such as provenance or trust contexts), where we want to express that something is true only according to a source, or with a degree of confidence. For instance, let us suppose a functional property `birthplace` that we want to use in the context of provenance. It can be desirable to model that

Table 3.1: Preserved D* entailments (P = Rule Preservation, P_{NC} = Non-Contextual Rule Preservation, ! = Risk of undesirable inference)

Rule	Condition	Constraint	Conclusion	Reif.	N-Ary	S.P.	NdF
lg	$v\ p\ l$	$l \in L$	$v\ p\ b_l$	P	P	P	P
gl	$v\ p\ b_l$	$l \in L$	$v\ p\ l$	P	P	P	P
rdf1	$v\ p\ w$		p `type Property`			P	P
rdf2-D	$v\ p\ l$	$l = (s,a) \in L_D^+$	b_l `type` a	P	P	P	P
rdfs1	$v\ p\ l$	$l \in L_p$	b_l `type Literal`	P	P	P	P
rdfs2	p `domain` u						
	$v\ p\ w$		v `type` u			P!	
rdfs3	p `range` u						
	$v\ p\ w$	$w \in U \cup B$	w `type` u			P!	
rdfs4a	$v\ p\ w$		v `type Resource`	P	P	P	P
rdfs4b	$v\ p\ w$	$w \in U \cup B$	w `type Resource`	P	P	P	P
rdfs7x	p `subPropertyOf` q						
	$v\ p\ w$	$q \in U \cup B$	$v\ q\ w$			P_{NC}!	P

Table 3.2: Preserved P-Entailments (P = Rule Preservation, P_{NC} = Non-Contextual Rule Preservation, ! = Risk of undesirable inference)

Rule	Condition	Constraint	Conclusion	Reif.	N-Ary	S.P.	NdF
rdfp1	p `type FunctionalProperty`						
	$u\ p\ v$						
	$u\ p\ w$	$v \in U \cup B$	v `sameAs` w			P!	
rdfp2	p `type InverseFunctionalProperty`						
	$u\ p\ w$						
	$v\ p\ w$		v `sameAs` w			P!	
rdfp3	p `type SymmetricProperty`						
	$v\ p\ w$	$w \in U \cup B$	$w\ p\ v$			P_{NC}!	P
rdfp4	p `type TransitiveProperty`						
	$u\ p\ v$						
	$v\ p\ w$		$u\ p\ w$			P_{NC}!	P
rdfp5a	$v\ p\ w$		v `sameAs` v	P	P	P	P
rdfp5b	$v\ p\ w$	$w \in U \cup B$	w `sameAs` w	P	P	P	P
rdfp8ax	p `inverseOf` q						
	$v\ p\ w$	$w,q \in U \cup B$	$w\ q\ v$			P_{NC}!	P
rdfp8bx	p `inverseOf` q						
	$v\ q\ w$	$w \in U \cup B$	$w\ p\ v$			P_{NC}!	P
rdfp11	$u\ p\ v$						
	u `sameAs` u'						
	v `sameAs` v'	$u' \in U \cup B$	$u'\ p\ v'$	P	P	P_{NC}!	
rdfp14a	v `hasValue` w						
	v `onProperty` p						
	$u\ p\ w$		u `type` v			P!	
rdfp14bx	v `hasValue` w						
	v `onProperty` p						
	u `type` v	$p \in U \cup B$	$u\ p\ w$	P_{NC}	P_{NC}	P_{NC}	P_{NC}
rdfp15	v `someValuesFrom` w						
	v `onProperty` p						
	$u\ p\ x$						
	x `type` w		u `type` v			P!	
rdfp16	v `allValuesFrom` w						
	v `onProperty` p						
	u `type` v						
	$u\ p\ x$	$x \in U \cup B$	x `type` w			P!	

Table 3.3: Conclusions for rules with no rule preservation for NdFluents

Rule	Conclusion	Rule	Conclusion	Rule	Conclusion
rdfs2	v_c `type` u	**rdfp1**	v_c `sameAs` w_c	**rdfp14a**	u_c `type` v
rdfs3	w_c `type` u	**rdfp2**	v_c `sameAs` w_c	**rdfp15**	u_c `type` v

Barack Obama was born in the United States according to a source, but in Kenya according to a different source. In this case the rule `rdfp1` would infer that the United States and Kenya are the same place in the non-annotated graph when using the Singleton Property.

It can be seen that Reification and N-Ary relations show poor preservation of rules, where most of those rules could be considered tautologies. The Singleton Property provides a mixture of rule preservation and non-contextual rule preservation for all the rules, that can be useful when we want to annotate universally true facts, but it is not usable when we want to have contextual information that is not universally true. NdFluents, by contrast, has neither non-contextual rule preservation nor rule preservation that can lead to undesirable inferences for any rule. There is only one rule where NdFluents is surpassed by the other approaches. Rule `rdfp11` presents *Rule Preservation* for Reification and N-Ary relations, but no rule preservation at all for NdFluents.

In addition, for the rules where NdFluents has no rule preservation, we observe that different conclusions hold, where we entail contextual knowledge. In Table 3.3 we see the conclusions for that set of rules with their conclusions. We can observe that the individual used in the annotation is entailed in the conclusion. For instance, let us suppose a property `capitalOf` with a domain of `PopulatedPlace`; if we state that Babylon was the capital of the Babylonian empire between 609 BC and 539 BC, instead of inferring that Babylon *is* a populated place (as a universal truth), we entail that Babylon between 609 BC and 539 BC was a populated place.

3.7 Related Work

In the original 4dFluents paper there were some issues not addressed by the authors. Later works have tried to identify and address those issues. Zamborlini and Guizzardi [133] present an alternative work to 4dFluents, where they present two different alternatives to represent temporally changing information in OWL. Both approaches have a similar model to Welty and Fikes's, where the entities are sliced for different times. The main difference is that in the first one, *Individual Concepts and Rigidity*, the original individuals are considered as classes. Thus, they are not described by any property, and a new slice has to be created every time that a property changes. On the other hand the second approach, "Objects and Moments", is based on *Relators* and *Qua-individuals* [98], where the individuals are represented by an entity, and their slices inherit its properties. Then, any time a property changes, it is reflected in the original entity. The first approach is more prone to the proliferation of timeslices, and can only guarantee the immutability of original properties by repetition on every timeslice. The second approach solves those issues at the cost of blurring the details of the changes of individual properties, and it is not clear how inheritance works in OWL. In a later work [134], Zamborlini and Guizzardi focus on solving the issues of the prior approaches for representing events and properties of individuals. They maintain the fluent-like representation for events, but move to an N-ary representation for properties. However, they still do not address the possibility to have more than one domain relation, nor address how inheritance is performed in OWL.

There are also other works that compare the different approaches to represent contextual information. Gangemi and Presutti [48] present and compare a number of design patterns to represent N-Ary relations, including Reification and Context Slices [130], to represent additional information on binary relations. The comparison

is done in four qualitative dimensions (DL reasoning support, polymorphism support, relation footprint, and intuitiveness) and five quantitative dimensions (number of needed axioms, expressivity, consistency checking time, classification time, and amount on newly generated constants). However, they only provide a brief outline of the reasoning power of each approach, while we are interested in more fine-grained comparison of entailment preservations. Scheuermann, Motta, Mulholland, Gangemi, and Presutti [120], on the other side, perform a qualitative research that compares user preferences and ability for using different design patterns. In their study the fluents pattern is regarded as the most complicated and less used to model, while making a temporal slice of the predicate (which could be represented using the Singleton Property in RDF) seems more intuitive. The N-ary pattern is the model most frequently used. The model regarded as the most user-friendly is not representable using OWL, because it requires having a predicate as an argument of another (an approximation in RDF could be using N-Quads, though). Hernández, Hogan, and Krötzsch [72] compare Reification, N-Ary relations, Singleton Properties and Named Graphs to encode Wikidata in practice. They provide space requirements and query performance for each approach in 4store[3], BlazeGraph[4], GraphDB[5], Jena TDB[6] and Virtuoso[7]. They report that Singleton Properties provide the most concise representation on a triple level, while N-Ary predicates is the only model with built-in support for SPARQL property paths. In addition, the Singleton Property usually lacks performance due to the number of predicates, whereas there is no clear winner among the other approaches. Virtuoso exhibits the best performance, while Jena and 4store show the worst results. Later, Hernández, Hogan, Riveros, Rojas, and Zerega [73] extend their previous work to compare Virtuoso, BlazeGraph, Neo4J[8], PostgreSQL[9] with a set of new experiments, based on the idea of performing sets of lookups for atomic patterns with exhaustive combinations of constants and variables, in order to give an idea of the low-level performance of each configuration. In this set of experiments standard reification and named graphs performed best, with N-Ary relations following in third, and singleton properties not being well-supported.

3.8 Conclusions

Representing annotations on multiple dimensions is a current challenge in RDF and OWL. We have proposed the NdFluents ontology, a multi-dimension annotation ontology, based on 4dFluents. To the best of our knowledge, this is the first generic extension of 4dFluents for an arbitrary combinations of context dimensions. This representation is intended to be extended in a modular way for each desired dimension. In addition, we have presented three design patterns and additional considerations to keep in mind when modeling data with NdFluents. We study how many of the original inference rules are preserved when annotating the data with NdFluents and compare with the main approaches to annotate data: Reification, N-Ary Relations, and Singleton Property. The results show that NdFluents preserves

[3] https://github.com/garlik/4store
[4] https://www.blazegraph.com/
[5] http://graphdb.ontotext.com/
[6] https://jena.apache.org/documentation/tdb
[7] https://virtuoso.openlinksw.com/
[8] https://neo4j.com/
[9] https://www.postgresql.org/

more desirable entailments, while omitting undesirable entailments, than any alternative. The Singleton property presents non-contextual rule preservation for many of the rules, and can lead to undesirable entailments when the annotated facts are not universally true. Reification and N-Ary relations preserve the fewest number of entailment rules.

NdFluents is a specific instantiation of a family of contextualization functions we call **Nd***. In Nd* a subset of terms is renamed (that is, for each term, a fresh term is created and used in place of the original one), and NdFluents is the instantiation where the individuals are renamed.[h]

Lines of future work are manifold: First, we want to apply this model to real world datasets.[i] Our goal is to exploit the context of information to make the datasets fit for question answering, as well as determine the most relevant data sources. This includes providing additional information based on the context and helping to find the most trustworthy data for the answer. Second, we intend to look deeper into the entailment preservations for different approaches using bigger subsets of OWL 2, such as OWL DL and OWL 2 RL/RDF, and possible reformulations of the approaches that could improve the results.[j] Third, we plan to perform an experimental evaluation of the different annotation models using different triple stores with regards to different factors, such as size, loading time, query response time, and query formulation complexity.

References

[22] Carothers, G.: RDF 1.1 N-Quads: A line-based syntax for RDF datasets, W3C (2014). URL: https://www.w3.org/TR/n-quads

[48] Gangemi, A., Presutti, V.: A Multi-dimensional Comparison of Ontology Design Patterns for Representing n-ary Relations. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 7741 LNCS, pp. 86–105. Springer, Heidelberg (2013).

[56] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: A Multi-dimensional Contexts Ontology, Université Jean Monnet (2016). URL: http://arxiv.org/abs/1609.07102

[57] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC), pp. 638–654. Springer, Cham (2017).

[58] Giménez-García, J.M., Zimmermann, A., Maret, P.: Representing Contextual Information as Fluents. In: Knowledge Engineering and Knowledge Management - {EKAW} 2016 Satellite Events, {EKM} and Drift-an-LOD, Bologna, Italy, November 19-23, 2016, Revised Selected Papers, pp. 119–122. Springer, Cham (2017).

[69] Hartig, O., Thompson, B.: Foundations of an Alternative Approach to Reification in RDF, pp. 14–14. arXiv (2014). URL: http://arxiv.org/abs/1406.3399

[72] Hernández, D., Hogan, A., Krötzsch, M.: Reifying RDF: What Works Well With Wikidata? In: Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems, pp. 32–47, Bethlehem, PA, USA (2015).

[73] Hernández, D., Hogan, A., Riveros, C., Rojas, C., Zerega, E.: Querying Wikidata: Comparing SPARQL, Relational and Graph Databases. In: Proceedings of the 15th International Semantic Web Conference (ISWC)LNCS, vol. 9982 LNCS, pp. 88–103. Springer, Heidelberg (2016).

[h] This paragraph was added for the dissertation. While the name Nd* was not mentioned in the publications, the concept was referred to in multiple occasions. Note that NdFluents is equivalent to the instantiation in First Order Logic that renames constants.

[i] This work was performed during the thesis and it is shown in Chapter 7.

[j] This has been done during the thesis for First Order Logic (Chapter 1) and Description Logics (Chapter 2).

[80] Horrocks, I., Patel-schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, pp. 1–20. WC3 (2004). URL: https://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

[81] Horst, H.J.t.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics 3(2), 79–115 (2005).

[84] Kifer, M., Boley, H.: RIF Overview (Second Edition) - W3C Working Group Note 5, W3C (2013). URL: http://www.w3.org/TR/rif-overview

[98] Masolo, C., Guizzardi, G., Vieu, L., Bottazzi, E., Ferrario, R.: Relational roles and qua-individuals. In: AAAI Fall Symposium on Roles, an interdisciplinary perspective, pp. 103–112 (2005).

[107] Nardi, D., Brachman, R.J.: An Introduction to Description Logics. In: The Description Logic Handbook: Theory, Implementation, and Applications, pp. 1–40. Cambridge University Press (2003).

[108] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. In: Proceedings of the 23rd International Conference on the World Wide Web (WWW), pp. 759–770. ACM (2014).

[110] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web, pp. 1–1. W3C (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/

[120] Scheuermann, A., Motta, E., Mulholland, P., Gangemi, A., Presutti, V.: An Empirical Perspective on Representing Time. In: 7th International Conference on Knowledge CaptureK-CAP '13, pp. 89–96. ACM, New York, NY, USA (2013).

[130] Welty, C.: Context Slices: Representing Contexts in OWL. In: Workshop on Ontology PatternsCEUR Workshop Proceedings, pp. 59–60. CEUR-WS.org (2010).

[131] Welty, C., Fikes, R.: A Reusable Ontology for Fluents in OWL. In: Bennett, B., Fellbaum, C.D. (eds.) Proceedings of the 2006 conference on Formal Ontology in Information SystemsFrontiers in Artificial Intelligence and Applications, pp. 226–236. IOS Press (2006).

[133] Zamborlini, V., Guizzardi, G.: On the Representation of Temporally Changing Information in OWL. In: Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC, pp. 283–292. IEEE (2010).

[134] Zamborlini, V.C., Guizzardi, G.: An Ontologically-Founded Reification Approach for Representing Temporally Changing Information in OWL. In: 11th International Symposium on Logical Formalizations of Commonsense Reasoning (COMMONSENSE 2013), Centre for Telematics and Information Technology ; Telematica Instituut, Crete (2013).

[137] Zimmermann, A.: RDF 1.1: On Semantics of RDF Datasets, W3C (2014). URL: http://www.w3.org/TR/2014/NOTE-rdf11-datasets-20140225/

Chapter 4

NdProperties: Encoding Contexts in RDF Predicates with Inference Preservation*

In this chapter, we take the final step from First Order Logic to OWL. In previous chapters, for First Order Logic we have presented several possibilities to represent the context of a statement (increasing the arity of the predicates by one to include the context, renaming predicates, and renaming constants). In the case of Description Logics, where only unary and binary predicates are allowed, we have presented NdTerms, that could be seen as a generalization of the renaming predicates and renaming individuals in First Order Logic. In the previous chapter, we have presented NdFluents, an approach and ontology to represent the context of statements in OWL by renaming the individuals of the set of statements.

In this chapter, we decide to explore the possibility of renaming predicates. As with Description Logics, it is not possible to make a link between the predicate and the context in OWL semantics. However, it is possible to use the same IRI that identifies a property to identify an individual. While this does not have any semantic implications, in practice it is possible to query this information.

Finally, we compare NdProperties against existing approaches, including the NdFluents approach presented in the previous chapter.

4.1 Introduction

In the recent years, the necessity of annotating RDF statements with contextual information have been increasing. However, a triple can only represent a binary relation between a subject and an object. To make explicit the relationship between the statement and its context, either the information must be encoded outside of the RDF syntax, or the statement and its context must be split into multiple triples, possibly resulting in the subject and the object being disconnected.

With most common approaches for representing contextual information, inferences that were following from the original statement are not preserved. Recent

*This chapter is based on unpublished work by José M. Giménez-García and Antoine Zimmermann, itself based on the following prior publication:

- Giménez-García, J.M., Zimmermann, A.: NdProperties: Encoding contexts in RDF predicates with inference preservation. In: Joint Proceedings of the International Workshops on Contextualized Knowledge Graphs, and Semantic Statistics co-located with 17th International Semantic Web Conference (ISWC 2018) Monterey, USA, October 8th, 2018, Monterey, USA (2018). [55] The most important differences with regards to this publication, as well as some clarifications, will be mentioned in more footnotes with Latin alphabet.

approaches allow for some inferences to be preserved. However, the result of the inference has no connection with the context. This makes impossible to separate the knowledge in different context, and can lead to data that is inconsistent or that does not correspond with the ground truth, due to some undesirable inference results.

In this work we present *NdProperties*, an OWL-DL ontology to use triple predicates to relate statements to their context. This approach is inspired by Singleton Properties [108] (which also use the predicates to represent the statement) and Nd-Fluents [57] (which is able to better separate the contexts). NdProperties can be seen as a concrete implementation of Nd*, a generic approach where a subset of the terms is contextualized, explored also for all the terms in NdTerms [138]. The NdProperties ontology can be implemented in different ways, in order to allow more or less inferences.

The rest of the document is organized as follows: Section 4.2 presents some definitions, necessary to understand the rest of the work; Section 4.3 present current approaches to annotate statements with additional data; Section 4.4 describes the NdProperties ontology, with Section 4.6 studying its inference capabilities; finally, we provide some discussion possible future work in Section 4.7.

4.2 Preliminaries

In order to go from a set of triples, plus the desired contextual annotations, to a set of contextualized triples (*i.e.*, a set of reified or modified triples, plus additional triples that express the annotations), it is necessary to perform a transformation. In this section we introduce the notation we will use for the rest of the paper, formalize this transformation as a *contextualization function*, and present some desired properties that such a function should have. This section summarizes and updates previous work in the topic [57, 138].

We assume infinite disjoint sets $\mathcal{I}$ (IRIs), $\mathcal{B}$ (blank nodes), and $\mathcal{L}$ (literals). An RDF triple is a tuple $(s, p, o) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$, where s is called the *subject*, p is the *predicate* and o is the *object*. We write $\mathcal{T}$ the infinite set of triples. An RDF graph G is a set of RDF triples. We denote $\mathcal{I}_G$, $\mathcal{B}_G$ and $\mathcal{L}_G$ the sets of all IRIs, blank nodes, and literals that appear in a graph G. RDF, as well as semantic extensions such as RDFS and OWL, provides semantics that allow to entail new statements from an RDF graph. A graph G *entails* another graph G' under the semantics S (S-entails) when G' is a valid conclusion of G (*i.e.*, every interpretation that satisfies G also satisfies G' in S). The S-closure of a graph G is a graph that contains all possible entailments of G under S.

A contextual annotation can be thought of as a set of triples that describe the context of an individual representing the triple. We call this individual the *anchor*.[a]

Definition 4.1 (Connected terms)**.** *Two terms $z_0, z_n \in (\mathcal{I} \cup \mathcal{B})$ are* connected terms *in an RDF graph G iff either z_0 and z_n are the same term, or $(z_{i-1}, p_i, z_i) \in G$, where $p_i \in \mathcal{I}$ and $z_{i-1}, z_i \in (\mathcal{I} \cup \mathcal{B})$, for $1 \leq i \leq n$.*

Definition 4.2 (Contextual annotation)**.** *A* contextual annotation *$\mathcal{A}$ is a pair $(C, \mathbf{a})$, where C is an RDF graph and $\mathbf{a} \in (\mathcal{I}_C \cup \mathcal{B}_C)$ is a distinguished term (called the* anchor*) such that for all $x \in (\mathcal{I}_C \cup \mathcal{B}_C)$, if $(x, p, o) \in C \vee (s, p, x) \in C$, then $\{\mathbf{a}, x\}$ are* connected terms.

[a] In the published paper, we assumed that a contextual annotation was a single IRI, without loss of generality. In this chapter we reintroduce the concepts of *anchor*, *connected terms*, and the definition of *contextual annotation* from Chapter 2.

A contextualized statement is the combination of an RDF statement with a contextual annotation. We assume an infinite set of contextual annotations. $\mathcal{C} \subset 2^{\mathcal{T}}$.

Definition 4.3[b] (Contextualization function)**.** *A contextualization function for a graph G, an RDF statement $t = (s, p, o) \in G$, and a contextual annotation $\mathcal{G} = (C, \mathbf{a})$ is a function $f(\mathcal{G}, G, t) = \mathsf{St}_G(\mathcal{G}, t) \cup \mathsf{Cx}_G(\mathcal{G}, t)$, where $\mathsf{St}_G(\mathcal{G}, t)$ and $\mathsf{Cx}_G(\mathcal{G}, t)$ are graphs such that:*

- $s \in (\mathcal{I}_{\mathsf{St}_G(\mathcal{G},t)} \cup \mathcal{B}_{\mathsf{St}_G(\mathcal{G},t)})$
- $p \in \mathcal{I}_{\mathsf{St}_G(\mathcal{G},t)}$
- $o \in (\mathcal{I}_{\mathsf{St}_G(\mathcal{G},t)} \cup \mathcal{B}_{\mathsf{St}_G(\mathcal{G},t)} \cup \mathcal{L}_{\mathsf{St}_G(\mathcal{G},t)})$
- *there exists two terms $\mathbf{a}_s, \mathbf{a}_c \in ((\mathcal{I}_{\mathsf{St}_G(\mathcal{G},t)} \cup \mathcal{B}_{\mathsf{St}_G(\mathcal{G},t)}) \cap (\mathcal{I}_{\mathsf{Cx}_G(\mathcal{G},t)} \cup \mathcal{B}_{\mathsf{Cx}_G(\mathcal{G},t)}))$ (that we call the* statement anchor *and the* context anchor*) and $r \in \{s, p, o\}$ such that:*
  - *$\mathbf{a}_s$ and r are connected terms in $\mathsf{St}_G(\mathcal{G}, t)$.*
  - *for all $(\mathbf{a}, y, z) \in C$, $(\mathbf{a}_c, y, z) \in \mathsf{Cx}_G(\mathcal{G}, t)$;*
  - *for all $(x, y, \mathbf{a}) \in C$, $(x, y, \mathbf{a}_c) \in \mathsf{Cx}_G(\mathcal{G}, t)$;*
  - *for all other $(x, y, z) \in C$, $(x, y, z) \in \mathsf{Cx}_G(\mathcal{G}, t)$;*
  - *$\mathbf{a}_s$ and $\mathbf{a}_c$ are connected terms in $((\mathcal{I}_{\mathsf{St}_G(\mathcal{G},t)} \cup \mathcal{B}_{\mathsf{St}_G(\mathcal{G},t)}) \cap (\mathcal{I}_{\mathsf{Cx}_G(\mathcal{G},t)} \cup \mathcal{B}_{\mathsf{Cx}_G(\mathcal{G},t)}))$*

We extend the definition of contextualization functions to graphs as $f(\mathcal{G}, G) = \bigcup_{t \in G} f(\mathcal{G}, G, t)$.

Contextualization functions need to create fresh terms, one of which will become the statement anchor. This is often done by renaming elements of the statements using one or more injective functions ren, which depend on the context and the terms in the triple being renamed.

Definition 4.4[c]**.** *Given a graph G, a renaming function with regards to G is an injective function $\mathsf{ren} : \mathcal{C} \times (\mathcal{T} \cup \mathcal{I} \cup \mathcal{B}) \to \mathcal{I} \setminus \mathcal{I}_G \cup \mathcal{B} \setminus \mathcal{B}_G$.*

Contextualization functions often use the fresh terms to "replace" terms in triples of the graph and/or the contextual annotation. This is done by a replacement function.

Definition 4.5. *A replacement function for a triple $t = (s, p, o)$ and two terms x and y is a function $\mathsf{rplc}(t, x, y) = (s', p', o')$ such that:*

- *$s' = y$ iff $s = x$, otherwise $s' = s$*
- *$p' = y$ iff $p = x$, otherwise $p' = p$*
- *$o' = y$ iff $o = x$, otherwise $o' = o$*

For any graph G, we extend the definition of replacement function to graphs as $\mathsf{rplc}(G, x, y) = \bigcup_{t \in G} \mathsf{rplc}(t, x, y)$

The idea of a contextualization function is that it expresses that the knowledge of the original graph holds in the context. However, many meaningless contextualization functions are possible that do not necessarily encode that knowledge. In order to identify useful contextualization functions, we propose the following property:

[b] This definition supersedes the definition in the published paper.

[c] This definition did not exist in the published paper. It has been introduced to clarify the content. It is similar to Definitions 1.8 and 1.9 from Chapter 1 and to the function $\mathsf{ren}_{\mathcal{G}}$ in Section 2.4 of Chapter 2.

Definition 4.6[d] (Inference preservation)**.** *A contextualization function f* preserves inferences *with regards to a set of graphs $\mathcal{G}$ under the semantics S iff for all $@a \in \mathcal{C}$ and all graphs $G, G' \in \mathcal{G}$, $f(@a, G) \models_S f(@a, G')$ iff $G \models_S G'$.*

Inference preservation expresses the idea that knowledge that can be inferred in the original graph should also be inferred within the context. It is also possible that the contextualization function preserves some inferences, but it is not able to encapsulate the inferred knowledge in the context. We call this property *Non-contextual Inference preservation*. Although this property can be useful if the goal of contextualization function is to simply annotate triples that are considered universally true, it can lead to undesirable inferences if that is not the case, as we will see in Section 4.6.

Definition 4.7[e] (Non-contextual Inference preservation)**.** *A contextualization function f non-contextually preserves entailments with regards to a set of graphs $\mathcal{G}$ under the semantics S iff for all $@a \in \mathcal{C}$ and all ontologies $G, G' \in \mathcal{G}$, $f(@a, G) \models_S G'$ iff $G \models_S G'$.*

This definitions will be used to study existing reification approaches, as well as the newly defined *NdProperties*.

4.3 Existing approaches for representing context

In this section we present and describe the most relevant contextualization approaches. Some of these approaches can conform to a contextualization function. They focus on describing how to relate the statement anchor to the original triple, leaving open how to connect the statement anchor and the context anchor. More than one contextualization function can be possible for any approach. While there are a limited number of options for $\mathsf{St}_G(@a, t)$, $\mathsf{Cx}_G(@a, t)$ can be arbitrary as long as it respects the definition of contextualization function. In practice, however, usually they have the context anchor to be the same term as the statement anchor, or they connect them by a single triple. We give a description and a possible contextualization function for each of them down below.

RDF reification [16, Sec. 5.3] is the standard W3C model to represent information about a statement. A contextualization function for reification can be defined as follows.

Definition 4.8 (RDF reification contextualization function)**.** *Let $@a \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s, p, o) \in G$ a triple. We define the contextualization function $f_R(@a, G, t) = \mathsf{St}_G(@a, t) \cup \mathsf{Cx}_G(@a, t)$ such that $\mathsf{St}_G(@a, t) = \{(\mathsf{ren}_G(@a, t), type, Statement), (\mathsf{ren}_G(@a, t), subject, s), (\mathsf{ren}_G(@a, t), predicate, p), (\mathsf{ren}_G(@a, t)), object, o)\}$ and $\mathsf{Cx}_G(@a, t) = \mathsf{rplc}(@a, \mathbf{a}, \mathsf{ren}_G(@a, t))$, where $\mathsf{ren}_G(@a, t) = \mathbf{a}_s = \mathbf{a}_c$.*

Note that if we remove the triple $= (\mathsf{ren}_G(@a, t), type, Statement)$ from $\mathsf{St}_G(@a, t)$, f_R would still be a valid contextualization function for RDF reification. Similar changes can be made in many of the contextualization functions for the next approaches.

Techniques for representing *n-ary relations* in RDF [110] were published in 2006 as a W3C note. A contextualization function for them can be defined as follows.

[d] This definition differs from the one in the published paper in that it includes the semantics used to make the inferences.

[e] This definition differs from the one in the published paper in that it includes the semantics used to make the inferences.

Definition 4.9[f] (N-ary relations contextualization function). *Let $\mathcal{C}a \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s,p,o) \in G$ a triple. We define the contextualization function $f_{nary}(\mathcal{C}a,G,t) = \mathsf{St}_G(\mathcal{C}a,t) \cup \mathsf{Cx}_G(\mathcal{C}a,t)$ such that $\mathsf{St}_G(\mathcal{C}a,t) = \{(s,\mathsf{ren}^s_G(\mathcal{C}a,p),\mathsf{ren}^r_G(\mathcal{C}a,t)),(\mathsf{ren}^r_G(\mathcal{C}a,t),\mathsf{ren}^v_G(\mathcal{C}a,p),o)\}$ and $\mathsf{Cx}_G(\mathcal{C}a,t) = \mathsf{rplc}(\mathcal{C}a,\mathbf{a},\mathsf{ren}_G(\mathcal{C}a,t))$, where $\mathsf{ren}_G(\mathcal{C}a,t) = \mathbf{a}_s = \mathbf{a}_c$, and $\mathsf{ren}^r_G(\mathcal{C}a,t)$, $\mathsf{ren}^s_G(\mathcal{C}a,p)$, and $\mathsf{ren}^v_G(\mathcal{C}a,p)$ are three different renaming functions with disjoint ranges such that $\mathsf{ren}^s_G : \mathcal{C}\times\mathcal{T}\to\mathcal{I}$, $\mathsf{ren}^v_G : \mathcal{C}\times\mathcal{T}\to\mathcal{I}$.*

For n-ary relations, three renaming functions are needed: One to create the individual that represents the relation between the subject and object of the original triple, and two to create the properties that connect the subject and object with this new individual. These two last renaming function have different codomain.

Singleton properties [108] are a recent proposal based on creating a unique property for each triple and using it to reify the triple. It extends the RDF semantics in order to make each singleton property unique and to include its extension in the extension of the original property. However, these semantics can be emulated using OWL.

Definition 4.10[g] (Singleton Property contextualization function). *Let $\mathcal{C}a \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s,p,o) \in G$ a triple. We define the contextualization function $f_{sp}(\mathcal{C}a,G,t) = \mathsf{St}_G(\mathcal{C}a,t) \cup \mathsf{Cx}_G(\mathcal{C}a,t)$ such that $\mathsf{St}_G(\mathcal{C}a,t) = \{(s,\mathsf{ren}_G(\mathcal{C}a,t),o),(\mathsf{ren}_G(\mathcal{C}a,t),type,SingletonProperty),(\mathsf{ren}_G(\mathcal{C}a,t),singletonPropertyOf,p),\} \cup \mathsf{Sp}((s,\mathsf{ren}_G(\mathcal{C}a,t),o)) \cup \{((singletonPropertyOf,subPropertyOf,subPropertyOf))\}$ and $\mathsf{Cx}_G(\mathcal{C}a,t) = \mathsf{rplc}(\mathcal{C}a,\mathbf{a},\mathsf{ren}_G(\mathcal{C}a,t))$, where $\mathsf{ren}_G(\mathcal{C}a,t) = \mathbf{a}_s = \mathbf{a}_c$, and $\mathsf{Sp}((s,\mathsf{ren}_G(\mathcal{C}a,t),o))$ are the OWL statements equivalent to the DL axioms $\{s\} \equiv \exists \mathsf{ren}_G(\mathcal{C}a,t).\{o\}$ and $\{o\} \equiv \exists \mathsf{ren}_G(\mathcal{C}a,t)^-.\{s\}$.*

The *companion property* [47] is an attempt to reduce the number of unique properties that are generated by singleton properties. Its contextualization function can be defined as follows.

Definition 4.11[h] (Companion Property contextualization function). *Let $\mathcal{C}a \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s,p,o) \in G$ a triple. Let $g \subset G$ be a graph such that $t' = (s',p',o') \in g$ iff $t' \in G$ and $s' = s$ and $p' = p$. We define the contextualization function $f_{cp}(\mathcal{C}a,G,t) = \mathsf{St}_G(\mathcal{C}a,t) \cup \mathsf{Cx}_G(\mathcal{C}a,t)$ such that $\mathsf{St}_G(\mathcal{C}a,t) = \{(s,\mathsf{ren}^{cp}_g(\mathcal{C}a,t),o),(\mathsf{ren}^{cp}_g(\mathcal{C}a,t),companionPropertyOf,p),(s,\mathsf{ren}^{sp}_g(\mathcal{C}a,t),\mathsf{ren}_G(\mathcal{C}a,t)),(\mathsf{ren}^{sp}_g(\mathcal{C}a,t),idPropertyOf,\mathsf{ren}^{cp}_g(\mathcal{C}a,t))\} \cup \{(companionPropertyOf,subPropertyOf,subPropertyOf)\}$ and $\mathsf{Cx}_G(\mathcal{C}a,t) = \mathsf{rplc}(\mathcal{C}a,\mathbf{a},\mathsf{ren}_G(\mathcal{C}a,t))$, where $\mathsf{ren}_G(\mathcal{C}a,t) = \mathbf{a}_s = \mathbf{a}_c$, and ren_G, ren^{cp}_g and ren^{sp}_g are three different renaming functions with disjoint ranges such that $\mathsf{ren}^{cp}_g : \mathcal{C}\times\mathcal{T}\to\mathcal{I}$, $\mathsf{ren}^{sp}_g : \mathcal{C}\times\mathcal{T}\to\mathcal{I}$, $\forall t,t' \mathsf{ren}^{cp}_g(\mathcal{C}a,t) \neq \mathsf{ren}^{cp}_g(\mathcal{C}a,t')$.*

The companion property contextualization function uses three renaming functions: ren^{cp}_g creates a companion property of the original property. It has a unique name with regards to the pair <*subject, predicate*> (that is, two triples with different subject but the same property can share companion properties). Then, ren^{sp}_G creates

[f] In the published paper the function ren^r_G took the predicate as its parameter, but it should take the triple. We have updated the definition to correct it. We have also made explicit the contextualization functions are different to make the definition self-contained.

[g] Note that the definition of $\mathsf{ren}_G(p)$ as both functional and inverse functional property emulates the extension of the RDF semantics that make each singleton property unique in their original paper.

[h] The notation has been updated from the original paper for the sake of clarity, to help with the explanations in the following paragraph. We have also made explicit the contextualization functions are different to make the definition self-contained.

a sibling property for the companion property. Finally, ren_G creates the unique term to reify the triple.

Note that, in all previous approaches, using $\mathsf{Cx}_G(\mathcal{a}, t) = \{\mathcal{a} \cup \{(\mathsf{ren}_G\mathcal{a}, t, hasContext, \mathbf{a})\}\}$ instead of $\mathsf{Cx}_G(\mathcal{a}, t) = \mathsf{rplc}(\mathcal{a}, \mathbf{a}, \mathsf{ren}_G(\mathcal{a}, t))$, and having $\mathbf{a}_c = \mathbf{a}$, would leave us with a valid contextualization function for the approach. Similar changes can be made for other contextualization functions in this chapter without affecting its results.

NdFluents [57] was created with the purpose of improving inference preservation. [i] It consists in considering individuals as separate entities that exist according to each context. Its contextualization function can be defined as follows.

Definition 4.12 (NdFluents contextualization function)**.** *Let $\mathcal{a} \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s, p, o) \in G$ a triple. We define the contextualization function $f_{ndf} = \mathsf{St}_G(\mathcal{a}, t) \cup \mathsf{Cx}_G(\mathcal{a}, t)$ such that $\mathsf{St}_G(\mathcal{a}, t) = \{(\mathsf{ren}_G(\mathcal{a}, s), p, \mathsf{ren}_G(\mathcal{a}, o)), (\mathsf{ren}_G(\mathcal{a}, s), type, ContextualPart), (\mathsf{ren}_G(\mathcal{a}, o), type, ContextualPart), (\mathsf{ren}_G(\mathcal{a}, s), contextualPartOf, s), (\mathsf{ren}_G(\mathcal{a}, o), contextualPartOf, o), (\mathsf{ren}_G(\mathcal{a}, s), contextualExtent, \mathsf{ren}_G(\mathcal{a}, \mathbf{a})), (\mathsf{ren}_G(\mathcal{a}, o), contextualExtent, \mathsf{ren}_G(\mathcal{a}, \mathbf{a})), (\mathsf{ren}_G(\mathcal{a}, t), type, Context)\}$ and $\mathsf{Cx}_G(\mathcal{a}, t) = \mathsf{rplc}(\mathcal{a}, \mathbf{a}, \mathsf{ren}_G(\mathcal{a}, \mathbf{a}))$, where $\mathsf{ren}_G(\mathcal{a}, \mathbf{a}) = \mathbf{a}_s = \mathbf{a}_c$.*

NdTerms [138] is a generalization of NdFluents that has been specifically studied in the context of Description Logics.[j] It extends the approach to every term in the ontology instead of just the individuals. If the initial ontology is in OWL, then it can be represented in RDF, and the contextualization could be expressed according to the above Definition 4.3. However, it also introduces several notions that are required to ensure other desirable properties, that are fairly long to give in detail here. So we do not formalize the related contextualization function.

NdFluents and NdTerms can be seen as instantiations of a general approach that we call Nd*, where a chosen set of terms is renamed within context. The other existing approaches and Nd* follow a different principle to represent the context of a statement. RDF reification, n-ary relations, the singleton property, and the companion property follow the idea of reifying the triples in a set of a statements. That means that the statement anchor of each triple is a different term, to which the context is then linked. The idea of NdFluents and NdTerms is, instead, to represent the statement within the context. In these two approaches the statement anchor is not unique for each term, but common for a context or set of contexts. This translates into better properties with regards to how well they preserve the semantics of the original set of statements, as well as the possibility of having the semantics of different contexts separated.

Other approaches do not represent the statement and their context as an RDF graph, but extend the RDF syntax and/or semantics to do it. Named graphs, RDF*[1], and Notation 3 are three such cases.

Named graphs [23] extend the syntax of RDF by adding a fourth term to the triples. This fourth term is used to identify the graph to which the triples belong. This graph can then include the information about the context of the statement. While this term can conceptually be used as the statement anchor or the context

[1] Now renamed to "RDF-star"[68].

[i] This work is presented in Chapter 3.
[j] This work is presented in Chapter 2.

anchor for the triples in the graph, there is no syntactic or semantic connection between them in RDF. In addition, what the validity of a statement in a context means for the statement in other contexts is not defined.

RDF* [67] extends RDF with the possibility to use an RDF* statement in subject and object positions of another RDF* triple. This new RDF* triple can be used to describe the context of the statement. An RDF* triple used as subject or object of another triple is then conceptually equivalent to the statement anchor of the RDF* triple. This makes RDF* a conceptually similar solution to contextual functions in the reification paradigm. It is, however, a less flexible solution, since it is not possible to have two different statement anchors for the same triple, and therefore assert the validity of the statement in different sets of contexts.

Notation-3 [6] extends RDF even further with more constructs, such as rule definitions and variable quantifications. The comparison with contextualization functions is similar to that of RDF*.

These approaches do not conform to the definition of contextualization function, since the result of their transformation is not an RDF graph. Hence, they will be kept out of scope for comparisons in the rest of the paper.

4.4 The NdProperties Ontology and Contextualization Function

NdFluents [57] and NdTerms [138] are based on the idea of creating contextual terms (individuals for the former, all terms for the latter) that mirror the terms of the graph or ontology. They are concrete instantiations of Nd*, a general approach in which a subset of the terms is contextualized. Nd* proposes a new paradigm with regards to existing approaches. The contextualization functions for existing approaches are based on the principle of reifying each statement into a different term, and then linking this term with the context. The contextualization functions for Nd*, by contrast, rename a set of terms for each statement, and it is these terms that are linked with the context. Whenever a term appears in more that one statement under the same context it is renamed into the same term. This makes it possible to

```
Prefix( rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#> )
Prefix( owl:=<http://www.w3.org/2002/07/owl#>)
Prefix( ndp:=<http://w3id.org/nd/properties#> )
Ontology( <http://w3id.org/nd/properties>
        Declaration( Class( ndp:Context ) )
        Declaration( Class( ndp:ContextualProperty ) )
        SubClassOf( ndp:ContextualProperty rdf:Property ) )
        Declaration( Class( ndp:ContextualObjectProperty ) )
        SubClassOf( ndp:ContextualObjectProperty ndp:ContextualProperty ) )
        SubClassOf( ndp:ContextualObjectProperty owl:ObjectProperty ) )
        Declaration( Class( nd:ContextualDatatypeProperty ) )
        SubClassOf( ndp:ContextualDatatypeProperty ndp:ContextualProperty ) )
        SubClassOf( ndp:ContextualDatatypeProperty owl:DatatypeProperty ) )
        Declaration( ObjectProperty( ndp:contextualPropertyOf ) )
        ObjectPropertyDomain( ndp:contextualPropertyOf rdf:ContextualProperty ) )
        ObjectPropertyRange( ndp:contextualPropertyOf ndp:Property )
        Declaration( ObjectProperty( ndp:contextualExtent ) )
        ObjectPropertyDomain( ndp:contextualExtent ndp:ContextualProperty )
        ObjectPropertyRange( ndp:contextualExtent ndp:Context )
)
```

Ontology 4.1: The NdProperties ontology

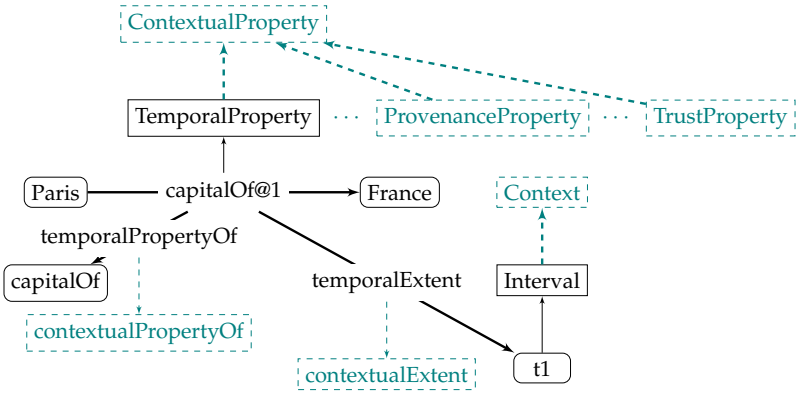


Figure 4.1: Temporal Extension of NdProperties

better preserve the semantics of the statements within the context, as well as clearly assert in which contexts, or combination of contexts, the statement is true.

NdProperties is a new instantiation of Nd* where the relations are contextualized. It creates a different property for each existing property in the graph and each different context we want to use to annotate the data. The ontology can be seen in Ontology 4.1[k], and the contextualization function is shown in Definition 4.13.

Definition 4.13[l] (NdP contextualization function)**.** *Let $\mathcal{C}a \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s, p, o) \in G$ a triple. We define the contextualization function $f_{NdP} = \mathsf{St}_{NdP}(\mathcal{C}a, t) \cup \mathsf{Cx}_{NdP}(\mathcal{C}a, t)$ such that $\mathsf{St}_{NdP}(\mathcal{C}a, t) =$*

$\{(s, \mathsf{ren}_G(\mathcal{C}a, p), o)\} \cup$
$\{(\mathsf{ren}_G(\mathcal{C}a, p), type, ContextualProperty)\} \cup$
$\{(\mathsf{ren}_G(\mathcal{C}a, p), contexualPropertyOf, p)\} \cup$
$\{\mathsf{ren}_G(\mathcal{C}a, p), contextualExtent, \mathsf{ren}_G(\mathcal{C}a, \mathbf{a})\}$
$\{\mathsf{ren}_G(\mathcal{C}a, t), type, Context\} \cup$, *and*
$\mathsf{Cx}_{NdP}(\mathcal{C}a, t) = \mathsf{rplc}(\mathcal{C}a, \mathbf{a}, \mathsf{ren}_G(\mathcal{C}a, \mathbf{a}))$,
where $\mathsf{ren}_G(\mathcal{C}a, \mathbf{a}) = \mathbf{a}_s = \mathbf{a}_c$.

Note that we make use of *punning* [62, Section 2.4.1] for the new property that characterizes the statement. That is, we use the same name for the property and an individual related to the contextual extent. In addition, the individuals of class `nd:ContextualPart` are used as the statement anchor.[m]

While the ontology can be used "as is", in order to represent more than one context it is necessary to be extended for each context (in a similar fashion as NdFluents [57]). In 4.1 we show the representation of a statement with temporal annotations using this ontology. The non-dashed parts belong to the temporal extension, while the dashed parts correspond to the NdProperties ontology plus other possible extensions. Other dimensions, such as provenance, can be modeled similarly to the temporal dimension by replacing the appropriate classes and properties.

[k] The ontology had some mistakes in the published paper. It has been updated to correct them.

[l] This definition differs from the definition in the published paper, since it needs to conform to the new definition of contextualization function.

[m] The published paper did not mention punning.

4.5 Extending the NdProperties Contextualization Function[n]

In the NdProperties contextualization function f_{NdP}, the semantics of the original property p are lost (*e.g.*, its domain and range, or as a functional property, transitive property, etc.). However, it is possible to extend the contextualization functions in order to retain, at least partially, the semantics of the original property. A first approach could be to simply add the triples where p was subject or object. We can see such contextualization function in Definition 4.14. This would fit the intuition of adding the semantics of the original property to the contextualized property. This solution would work for some cases (such as domain and range), but in some other cases (such as subproperty or transitivity) it would lead to non-contextual inference preservation.

Definition 4.14 (NdP$^{\mathrm{f}}$ contextualization function)**.** *Let $\mathcal{C}a \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s,p,o) \in G$ a triple. We define the contextualization function $f_{NdP^f} = \mathsf{St}_{NdP^f}(\mathcal{C}a,t) \cup \mathsf{Cx}_{NdP}(\mathcal{C}a,t)$ such that $\mathsf{St}_{NdP^f}(\mathcal{C}a,t) = \mathsf{St}_{NdP}(\mathcal{C}a,t) \cup$*
$\{(\mathsf{ren}_G(\mathcal{C}a,p),q,r) \mid (s,\mathsf{ren}_G(\mathcal{C}a,p),o) \in \mathsf{St}_G(\mathcal{C}a,t) \wedge (p,q,r) \in G\} \cup$
$\{(m,n,(\mathsf{ren}_G(\mathcal{C}a,p)) \mid (s,\mathsf{ren}_G(\mathcal{C}a,p),o) \in \mathsf{St}_G(\mathcal{C}a,t) \wedge (m,n,p) \in G\}$

NdP$^{\mathrm{f}}$ leads to non-contextual inference preservation because it links the semantics of a contextualized property with those of a non contextualized property. Because of that, triples within the context lead to infer triples outside the context. In the case we desired to have (general) inference preservation, a naive solution to this problem could be applying the renaming function to terms to which the contextualized property is connected. We can see this contextualization function in Definition 4.15. This contextualization function, however, leads to two new problems: First, the inferences of the newly renamed properties are not preserved. That means that, for example, if p_1 is subproperty of p_2, and p_2 is subproperty of p_3, in the contextualized statements $\mathsf{ren}_G(\mathcal{C}a,p_1)$ will be subproperty of $\mathsf{ren}_G(\mathcal{C}a,p_2)$, but the same cannot be said about $\mathsf{ren}_G(\mathcal{C}a,p_2)$ and $\mathsf{ren}_G(\mathcal{C}a,p_3)$. Second, and more important, it contextualizes terms other than properties, leading to bizarre inferences.

Definition 4.15 (NdP$^{\mathrm{n}}$ contextualization function)**.** *Let $\mathcal{C}a \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s,p,o) \in G$ a triple. We define the contextualization function $f_{NdP^n} = \mathsf{St}_{NdP^n}(\mathcal{C}a,t) \cup \mathsf{Cx}_{NdP}(\mathcal{C}a,t)$ such that $\mathsf{St}_{NdP^n}(\mathcal{C}a,t) = \mathsf{St}_{NdP}(\mathcal{C}a,t) \cup$*
$\{(\mathsf{ren}_G(\mathcal{C}a,p),q,\mathsf{ren}_G(\mathcal{C}a,r)) \mid (s,\mathsf{ren}_G(\mathcal{C}a,p),o) \in \mathsf{St}_G(\mathcal{C}a,t) \wedge (p,q,r) \in G\} \cup$
$\{\mathsf{ren}_G(\mathcal{C}a,m),n,(\mathsf{ren}_G(\mathcal{C}a,p)) \mid (s,\mathsf{ren}_G(\mathcal{C}a,p),o) \in \mathsf{St}_G(\mathcal{C}a,t) \wedge (m,n,p) \in G\}$

The first problem of NdP$^{\mathrm{n}}$ can be addressed by using the concept of *connected terms*, renaming terms that are connected to the original property in the original statements. The second problem is more complex in nature. This is due to the fact that, contrary to Description Logics, the syntax of RDF does not allow to know if an arbitrary term is a property or not. In essence, we need to check, for a term x, if a triple `(x,rdf:type,rdf:Property)` exists *or can be inferred* in G. But inferring that triple depends on the concrete semantic extension we are using. Thus, we need to include this new element in the definition of the contextualization function if we want to have general inference preservation. This contextualization function is shown in Definition 4.16.

Definition 4.16 (NdP$^{\mathrm{S}}$ contextualization function)**.** *Let $\mathcal{C}a \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, $t = (s,p,o) \in G$ a triple, and S a semantic extension of*

[n]This section, as well as future mentions to its content, correspond to new work not included in the published paper.

RDF respectively. Let G^* be the S-closure of G. We define the contextualization function $f_{NdP^S} = \mathsf{St}^S(\mathcal{G},t) \cup \mathsf{Cx}_G(\mathcal{G},t)$ such that $\mathsf{St}^S(\mathcal{G},t) = \mathsf{St}_G(\mathcal{G},t) \cup$

$\{(\mathsf{ren}_{NdP}(\mathcal{G},x), y, \mathsf{ren}_G(\mathcal{G},z)) \mid (x,y,z) \in G \wedge (s,\mathsf{ren}_G(\mathcal{G},p),o) \in \mathsf{St}_G(\mathcal{G},t) \wedge$ (x,rdf:type,rdf:Property), (z,rdf:type,rdf:Property) $\in G^* \wedge x$, z and p are connected terms in $G^*\} \cup$

$\{((\mathsf{ren}_G(\mathcal{G},x), y, z) \mid (x,y,z) \in G \wedge (s,\mathsf{ren}_G(\mathcal{G},p),o) \in \mathsf{St}_G(\mathcal{G},t) \wedge$ (x,rdf:type,rdf:Property) $\in G^* \wedge$ (z,rdf:type,rdfs:Class) $\notin G^* \wedge x$, z and p are connected terms in $G^*\} \cup$

$\{(x, y, \mathsf{ren}_G(\mathcal{G},z) \mid (x,y,z) \in G \wedge (s,\mathsf{ren}_G(\mathcal{G},p),o) \in \mathsf{St}_G(\mathcal{G},t) \wedge$ (x,rdf:type,rdfs:Class) $\notin G^*$, (z,rdf:type,rdf:Property) $\in G^* \wedge x$, z and p are connected terms in $G^*\}$

In short, f_{NdP^S} introduces new statements for each triple in which the subject or the object are connected terms to the property that is being contextualized in the original graph. If the term is a property it is renamed; if not, the term remains unchanged. Note that this contextualization function introduces new complexity that can make it computationally expensive or even non-decidable, depending on the semantic extension. In addition, there exist still cases where not even f_{NdP^S} is able to completely preserve the semantics of the original set of statements. Imagine for example the case of triple `(p,owl:equivalentProperty,owl:sameAs)`. In this case `owl:sameAs` would be renamed by the contextualization function, losing entailment preservation. Note that not contextualizing `owl:sameAs` would not provide entailment preservation either, but only non-contextual entailment preservation.

In the following section we will compare the inference preservation of the NdProperties contextualization function and its extensions against the other approaches.

4.6 Reasoning with NdProperties

Following a similar approach as for NdFluents [57], we analyze for which rules from the pD* fragment of OWL [81] the contextualization functions preserve inferences. For each rule, we check if the contextualization function for each approach has *Inference Preservation* or *Non-Contextual Inference Preservation* with regards to a graph that contains the corresponding triples. Following NdFluents evaluation, we apply the contextualization function only on triples that do not include RDF, RDFS, or OWL vocabularies. The intuition behind this is that reification approaches are usually used to annotate data on relations between resources. Table 4.1 shows the D* (modified RDFS) entailment rules and rule preservations for each one of the approaches, whereas Table 4.2 presents the same information for P (modified subset of OWL)

Table 4.1: Preserved D* entailments (P = Rule Preservation, P_{NC} = Non-Contextual Rule Preservation, ! = Risk of undesirable inference)

Rule	Condition	Constraint	Conclusion	Reif.	N-Ary	S.P.	NdF	NdP	NdP^f	NdP^n	NdP^S
lg	$v\ p\ l$	$l \in L$	$v\ p\ b_l$	P	P	P	P	P	P	P	P
gl	$v\ p\ b_l$	$l \in L$	$v\ p\ l$	P	P	P	P	P	P	P	P
rdf1	$v\ p\ w$		p `type Property`			P	P	P	P	P	P
rdf2-D	$v\ p\ l$	$l = (s,a) \in L_D^+$	b_l `type` a	P	P	P	P	P	P	P	P
rdfs1	$v\ p\ l$	$l \in L_p$	b_l `type Literal`	P	P	P	P	P	P	P	P
rdfs2	p `domain` u $v\ p\ w$		v `type` u			P!			P!	*	P!
rdfs3	p `range` u $v\ p\ w$	$w \in U \cup B$	w `type` u			P!			P!	*	P!
rdfs4a	$v\ p\ w$		v `type Resource`	P	P	P	P	P	P	P	P
rdfs4b	$v\ p\ w$	$w \in U \cup B$	w `type Resource`	P	P	P	P	P	P	P	P
rdfs7x	p `subPropertyOf` q $v\ p\ w$	$q \in U \cup B$	$v\ q\ w$			P_{NC}!	P		P_{NC}!	P*	P

Table 4.2: Preserved P-Entailments (pD* = Rule Preservation, P_{NC} = Non-Contextual Rule Preservation, ! = Risk of undesirable inference)

Rule	Condition	Constraint	Conclusion	Reif.	N-Ary	S.P.	NdF	NdP	NdP^f	NdP^n	NdP^S
rdfp1	p `type` `FunctionalProperty` $u\ p\ v$ $u\ p\ w$	$v \in U \cup B$	v `sameAs` w			P!			P!	*	P!
rdfp2	p `type` `InverseFunctionalProperty` $u\ p\ w$ $v\ p\ w$		v `sameAs` w			P!			P!	*	P!
rdfp3	p `type` `SymmetricProperty` $v\ p\ w$	$w \in U \cup B$	$w\ p\ v$			P_{NC}!	P		P	*	P
rdfp4	p `type` `TransitiveProperty` $u\ p\ v$ $v\ p\ w$		$u\ p\ w$			P_{NC}!	P		P	*	P
rdfp5a	$v\ p\ w$		v `sameAs` v	P	P	P	P	P	P	P	P
rdfp5b	$v\ p\ w$	$w \in U \cup B$	w `sameAs` w	P	P	P	P	P	P	P	P
rdfp8ax	p `inverseOf` q $v\ p\ w$	$w, q \in U \cup B$	$w\ q\ v$	P	P	P_{NC}!	P		P_{NC}!		P
rdfp8bx	p `inverseOf` q $v\ q\ w$	$w \in U \cup B$	$w\ p\ v$			P_{NC}!	P		P_{NC}!	P	P
rdfp11	$u\ p\ v$ u `sameAs` u' v `sameAs` v'	$u', v' \in U \cup B$	$u'\ p\ v'$	P	P	P_{NC}!		P	P	P	P
rdfp14a	v `hasValue` w v `onProperty` p $u\ p\ w$		u `type` v			P!			P!	*	P!
rdfp14bx	v `hasValue` w v `onProperty` p u `type` v	$p \in U \cup B$	$u\ p\ w$	P_{NC}	P_{NC}	P_{NC}	P_{NC}	P_{NC}	P	*	P
rdfp15	v `someValuesFrom` w v `onProperty` p $u\ p\ x$ x `type` w		u `type` v			P!			P!	*	P!
rdfp16	v `allValuesFrom` w v `onProperty` p u `type` v $u\ p\ x$	$x \in U \cup B$	x `type` w			P!			P!	*	P!

entailment rules. Note that we remove those rows where both condition and conclusion include only triples with RDF, RDFS, or OWL vocabularies. Because we do not apply the contextualization function to such statements, they trivially have rule preservation. A symbol P indicates that there is rule preservation for the corresponding approach, while a symbol P_{NC} denotes non-contextual rule preservation. A symbol ! means that it is possible to have undesirable inferences. This happens when the contextualization function annotates at least one triple of the condition, and either we have non-contextual inference preservation, or we have inference preservation but the function does not modify the triple in the conclusion.

For other approaches the conclusions of NdFluents [57] hold: reification and n-ary relations show poor preservation of rules, where most of those rules could be considered tautologies. The Singleton Property provides a mixture of inference preservation and non-contextual inference preservation for all the rules, that can be useful when we want to annotate universally true facts, but it is not usable when we want to have contextual information that is not universally true. NdFluents, by contrast, has neither non-contextual rule preservation nor rule preservation that can lead to undesirable inferences for any rule.

For NdProperties, we can observe that NdP behaves slightly better than reification and n-ary relations, having inference preservation for two additional rules: `rdf1` and `rdfp11`. We see that NdP^f behaves in a similar way to the Singleton Property. This is expected, since both contextualization functions link the semantics of the contextualized properties with those of the non-contextualized one. NdP^n shows what happens if we are not careful when extending a contextualization function. It creates bizarre entailments that contextualize classes—including classes of the OWL 2 vocabulary—without providing any increase in reasoning power over the original NdProperties function. Note that, even if our evaluation of rdfs7x gives rule preservation, the transitivity of `subPropertyOf` is not necessarily preserved. This means that some inferences regarding subproperties can be lost after the contextualization. Finally, NdP^S has inference preservation for all the rules in pD*. In seven cases it can lead to undesirable inferences, though.

In general, the extensions of NdProperties show that adding axioms to the contextualization function to describe the contextualized terms in a similar way to the original ones can improve its reasoning capabilities. However, there is a compromise between these capabilities and the complexity of the function. In addition, these axioms can lead to other undesirable inferences.

4.7 Conclusion and Future Work

NdProperties can be seen as a concrete instantiation of a family of contextualization functions that rename a subset of the terms in an ontology. We call this family of contextualizations **Nd***. Other instantiations that follow this approach are NdFluents [57] (where the individuals are renamed) and NdTerms [138] (where all the terms are renamed).[P] In this paper, we have presented the contextualization function for NdProperties and an ontology to represent it in RDF. We also showed three possible extensions that showcase a compromise between complexity and reasoning power, as well as the consequences of not being careful enough when devising a contextualization function. We compare these functions against existing reification approaches in terms of inference preservation and see that they can preserve more desirable entailments, with the additional advantage that they can be tuned to preserve only a subset of desired inferences.

In the future, we plan to continue our research in the Nd* approach, studying their instantiations and properties and how to combine and relate different contexts, and performing a systematic comparison of the different instantiations of the approach.

References

[6] Berners-Lee, T., Connolly, D.: Notation3 (N3): A readable RDF syntax, W3C (2011). URL: https://www.w3.org/TeamSubmission/n3/

[16] Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C (2004). URL: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[23] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. Journal of Web Semantics 3(4), 247–267 (2005).

[47] Frey, J., Müller, K., Hellmann, S., Rahm, E., Vidal, M.-E.: Evaluation of metadata representations in RDF stores. Semantic Web 10(2), 205–229 (2017).

[55] Giménez-García, J.M., Zimmermann, A.: NdProperties: Encoding contexts in RDF predicates with inference preservation. In: Joint Proceedings of the International Workshops on Contextualized Knowledge Graphs, and Semantic Statistics co-located with 17th International Semantic Web Conference (ISWC 2018) Monterey, USA, October 8th, 2018, Monterey, USA (2018).

[57] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC), pp. 638–654. Springer, Cham (2017).

[62] Golbreich, C., Wallace, E.K.: OWL 2 web ontology language, new features and rationale (second edition), W3C (2012). URL: https://www.w3.org/TR/2012/REC-owl2-new-features-20121211/

[67] Hartig, O.: Foundations of RDF* and SPARQL* (An Alternative Approach to Statement-Level Metadata in RDF). In: Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017. (2017).

[P] The name Nd* was not mentioned in the publications, but references to the concept were continually made.

[68] Hartig, O., Champin, P.-A., Kellogg, G., Seaborne, A., Arndt, D., Broekstra, J., DuCharme, B., Lassila, O., Patel-Schneider, P.F., Prud'hommeaux, E., Thibodeau, T., Thompson, B.: RDF-star and SPARQL-star, W3C (2021). URL: https://w3c.github.io/rdf-star/cg-spec

[81] Horst, H.J.t.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics 3(2), 79–115 (2005).

[108] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. In: Proceedings of the 23rd International Conference on the World Wide Web (WWW), pp. 759–770. ACM (2014).

[110] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web, pp. 1–1. W3C (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/

[138] Zimmermann, A., Giménez-García, J.M.: Contextualizing DL Axioms: Formalization, a New Approach, and Its Properties. In: Joint Proceedings of the Web Stream Processing workshop (WSP 2017) and the 2nd International Workshop on Ontology Modularity, Contextuality, and Evolution (WOMoCoE 2017) co-located with 16th International Semantic Web Conference (ISWC 2017), pp. 74–85 (2017).

Conclusions to Formalizing the Context of Statements

Through this part, we have seen our goals fulfilled: we have validated our hypothesis that it is possible to add contextual information to a set of statements without extending the formalism of the system, and we have described formally the process to do so and its properties. We call this formal description *Contextualization Function*. Furthermore, we have presented a set of contextualization functions that preserve the semantics of the original data better than any alternative existing in the Semantic web, and that allow different contexts to coexist.

All in all, the chapters of this part have addressed the following research questions:

- **R1:** *How to express formally the process needed to add contextual information to a statement or set of statements?*
- **R2:** *What are the formal properties of a contextualization process with regards to how it affects the semantics of the resulting data?*
- **R3:** *How to represent contextual information in order to preserve as much semantics as possible from original formalized data?*

We have explored the research questions following a top-down strategy, starting from First Order Logic, going through Description Logic, and arriving to OWL and RDF. In each of the three logic systems, we have defined formally the process to create the set of contextualized statements from a set of statements plus a context in the form of a *Contextualization Function*. This corresponds to contribution **C1** (the definition and formalization of *Contextualization Function*). Then, we have investigated and formally defined the properties a contextualization function can have with regards to the semantics of the logic system. This corresponds to contribution **C2** (the formal definition of properties of a contextualization function). With these tools, we have proposed Nd*, a family of contextualization functions, that can be applicable to the three logical systems. We showed that they better respect the defined properties than any other alternative existing in the Semantic Web. This corresponds to contribution **C3** (the family of contextualization functions *Nd**).

We argue that Nd* better respects these properties because they follow a different paradigm than the existing approaches. Other approaches follow the idea of reifying each axiom in a different individual and linking it to the context. Since a single term is used to assert the validity of each axiom with regards to the context, the reification paradigm struggles to conform to the original semantics of the set of axioms. By comparison, Nd* relies on contextualizing a set of terms of the axiom to represent its validity within the context. An equivalent axiom with the contextualized terms is then used to assert the validity in the context. This allows to better preserve those semantics and better separate the logical implications within different context.

We also showed that there exists a compromise between how well a contextualization function complies with the properties, and the number and complexity of

the statements it generates. This gives us our first glimpse into **R4:** *How does a contextualization function affect the efficiency with regards to how the data and their context are stored and queried?*.

Two questions remain open after this part. Namely **R5:** *How to capture existing contextual information that exists implicitly in non-formalized data when transforming it into formal statements?* and **R6:** *How to efficiently manage contextualized data, independently of the concrete representation used to model it?*. These questions will be addressed on parts II and III respectively.

Part II

Capturing the Context of Statements

Preamble to Capturing the Context of Statements

In the previous part, we studied how to represent contextual information about statements in different logic systems, from First Order Logic to OWL and RDF, going through Description Logic. We provided the definition of Contextualization Function: a function that takes a set of statements in a logic system and a context, and maps them to another set of statements in the same logic system. We studied the properties that a contextualization function can have with regards to how it preserves the semantics of the original set of statements, how different contexts can coexists, and if it is possible to add new data. We presented Nd*, a family of contextualization functions that rename a specific subset of terms. We showed that Nd* better respects the properties that are of interest to us (inference conservation, separability, context independence...), but that there is a compromise between how well each contextualization function does it and the quantity and complexity of the resulting set of statements.

In this part of the thesis we change our focus to look at real-life data in the wild. This data is not necessarily expressed formally, and the context is usually not explicitly stated. The first two chapters (*i.e.*, Chapters 5, 6) are dedicated to making explicit contextual data about openly available datasets. In them, we extract implicit data about how RDF datasets are reused by other datasets and users. This chapters do not have a direct relation with Part I and are based on tasks that were performed in parallel during the thesis. The approaches proposed in these chapters generate contextual data about datasets in the contexts of trust, reuse, and quality. They show the relevance of having formal representations for contextual information, and of having generic approaches to extract contextual information. In the following chapters, we resume from where we left in Part I. We make first use of concepts presented in Part I to represent contextual data from a real-world dataset. Finally, we propose a semi-automatic approach to capture data and its context from Web tables. The data and context are then represented in RDF using a contextualization function among those studied on Part I, including one of our proposals.

In more detail, the contents of this part are the following:

Chapter 5, "Assessing Trust with PageRank in the Web of Data" presents a work where we extract links between datasets that are openly available as data dumps. We use these links to calculate the PageRank value of each dataset. We argue that this value can be seen as contextual information about trust (*i.e.*, how much the creators of a dataset trusted external datasets in order to link to their data).

Chapter 6, "What does Dataset Reuse tell us about Quality?" continues the work in dataset reuse from the previous chapter. Whereas in the previous chapter we studied the reuse of datasets by other datasets, in this chapter we quantify how datasets are discussed by the scientific community. We do so by defining the concept of *dataset mention* and extracting its value for scientific publications and mailing lists. Then, we extract a number of quality metrics for the same datasets and analyze the correlation of mentions and quality of datasets.

Chapter 7, "NELL2RDF: Reading the Web, Tracking the Provenance, and Publishing It as Linked Data" makes use of the contributions of Part I (*i.e.*, the concept of contextualization function and the NdFluents approach), to model contextual data about statements of the NELL dataset. NELL is a system that continually reads the web to create new statements (called beliefs) and assign a confidence score to them. NELL statements and confidence scores are generated through a series of coupled components working in parallel. There are huge quantities of contextual data about what components proposed a statements and their confidence score, as well as the final confidence score and decision about whether the belief can be promoted or have to stay as candidate belief. This makes NELL an ideal candidate to use contextualization functions to represent its data in RDF. For this we use existing approaches from the Semantic Web community, as well as NdFluents, our approach proposed in Chapter 3. This also allows to compare the output of each contextualization function with regards to the number of statements and the size of the generated data.

Chapter 8, "Towards Capturing Contextual Semantic Information About Statements in Web Tables" learns from previous chapter, where an *ad hoc* approach for a specific dataset was developed. This chapter proposes a generic, semi-automatic, approach to capture data and its context from Web tables and represent them in RDF using different contextualization functions. A prototype that demonstrates the feasibility of this approach is developed and made publicly available.

On the whole, this part deals mainly with Research Question **R5:** *How to capture existing contextual information that exists implicitly in non-formalized data when transforming it into formal statements?*. Research Question **R4:** *How does a contextualization function affect the efficiency with regards to how the data and their context are stored and queried?* is also explored in Chapter 7. Its contents correspond to Contributions **C4:** *Generation and publication of large data sets containing explicit contextual information from non-formalized data with implicit context*, and **C5:** *A generic approach to capture contextual information from relational tables and transform it into RDF contextualized statements*.

Chapter 5

Assessing Trust with PageRank in the Web of Data*

In the Semantic Web environment, datasets regularly include terms from other sources, and each of these connections express a degree of trust on that source. However, determining *what* is a dataset in this context is not straightforward. We study the concepts of dataset and dataset link, to finally use the concept of Pay-Level Domain to differentiate datasets, and consider usage of external terms as connections among them. Using these connections we compute the PageRank value for each dataset, and examine the influence of ignoring predicates for computation. This process has been performed for more than 300 datasets, extracted from the LOD Laundromat. The results show that reuse of a dataset is not correlated with its size, and provide some insight on the limitations of the approach and ways to improve its efficacy.

This chapter does not lean on previous content of the thesis, and together with the following chapter, it can be read in a non-sequential order with the rest of the thesis. This chapter provides interesting results on its own, experimenting with PageRank as a measure of trust and studying it in hundreds of datasets. These result are new information about the data itself in the context of trust. It underlines the existence of implicit contextual data, and the importance of having formalisms to represent it and generic approaches to extract it.

5.1 Introduction

The WDAqua project[1] aims at advancing the state of the art in data-driven question answering, with a special focus on the Web of Data. The Web of Data comprises thousands of datasets about varied topics, interrelated among them, which contain large quantities of relevant data to answer a question. Nonetheless, in an environment of information published independently by many different actors, data veracity is usually uncertain [112, 93], and there is always the risk of consuming misleading data. While some quality metrics have been proposed that can help to identify good datasets [29], there is a lack of trust metrics to provide a confidence on the veracity of the data [125].

[1] http://wdaqua.informatik.uni-bonn.de/

*This chapter is based on the following publication:

- Giménez-García, J.M., Thakkar, H., Zimmermann, A.: Assessing Trust with PageRank in the Web of Data. In: The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected PapersCEUR Workshop Proceedings, pp. 293–307. Springer, Cham (2016). [54]

In this context, we argue that actual usage of data can be seen as an act of trust. In this paper we focus on reuse of resources by other datasets as a usage metric. We consider reuse of a resource of a dataset by any other given dataset as an outlink from the later to the former. Under this purview, we can compute the PageRank [111] value of each dataset and rank them according to their reuse. PageRank has been successfully used to obtain trust metrics on individual triples [10]. In order to obtain a good measure of reuse, we perform the process on a large scale. We make use of the tools provided by the LOD Laundromat [117] to go beyond LOD Cloud, and process more than 38 billion triples, distributed in more than 600 thousand documents. The LOD Laundromat provides data from data dumps collected from the Internet, so it is not limited to dereferenceable linked data. However, what is regarded as a dataset is an important issue when dealing with data dumps. We make use of the concept of Pay-Level Domain (or PLD, also known as Top-Private Domain) to draw a distinction between datasets, and consider the influence of ignoring predicates when extracting outlinks. We perform a grouping of the triples in datasets according to their PLD and compute their PageRank values as a first measure of trust. Finally, we discuss the results and limitations of the approach, suggesting improvements for future work.

This document is organized as follows: in Section 5.2, we first discuss the relation of trust and popularity in the Web of Data, what should be considered a dataset in our context in order to clarify the problem we address, and finally present the LOD Laundromat; Section 5.3 describes the experiments and results, which we further discuss; Section 5.4 presents relevant related work; finally, we provide some conclusions and directions for future work in Section 5.5.

5.2 Ranking the Web of Data

5.2.1 PageRank, Reuse, and Trust in the Web of Data

We would like to assess trust in datasets by measuring their popularity based on the reuse of resources from a dataset in another dataset. To do this, we rely on the PageRank algorithm [111]. PageRank is the original algorithm developed by Page, Brin, Motwani, and Winograd that Google uses to rank their search results. It takes advantage of the graph structure of the web, considering each link from one page as a "vote" from the source to the destination. Using the links, the importance of a page is propagated across the graph, dividing the value of a page among its outlinks. This process is repeated until convergence is reached. The final result of PageRank corresponds to a stationary distribution, where each page value amounts to the probability for a random surfer to be at any moment in the page.

PageRank is meant to measure popularity (*i.e.*, "human interest and attention") on web pages. However, we argue that reuse of resources in the Semantic Web has a slightly different meaning. When there is a link from one web page to another, it does not mean necessarily that the author considers the linked page a trustworthy fact (it could be even linking something the author is criticizing). However, resources are reused to express facts in the author's dataset, which implicitly means that the author trusts that the resource is correct. This is supported by the analysis of predicates used for linking datasets by Schmachtenberg, Bizer, and Paulheim [121], where the top used predicates are used to express statements about identity or relatedness (`owl:sameAs`, `rdfs:seeAlso`, `skos:exactMatch`, `skos:closeMatch`), authorship (using Doublin Core vocabulary), and social relations (using `foaf` and `sioc` vocabularies).

To compute PageRank in a set of datasets, it is first necessary to define what is considered a dataset and what is a link between datasets. RDF graphs, although formally defined as a set of triples, can be seen as directed multigraphs in which predicates play the role of arcs. This view suggests that if a triple contains a resource of dataset A as subject, and a resource of dataset B as object, it can be seen as a link from dataset A to dataset B. However, the links formed by arcs in an RDF graph are irrelevant to the notion of dataset linking. In fact, only the presence of hyperlinks suffices to indicate a link between one source and destination, therefore any HTTP IRI in an RDF graph can be seen as a link. So the question is, what it means that a resource belongs to a dataset, and to what dataset a hyperlink "points to". A naïve approach would be to consider that any IRI existing in a dataset belongs to the dataset and thus, that links connect two datasets having one same resource. However, this would imply, for instance, that any triple anywhere that uses a DBpedia IRI is considered to be linked to from the DBpedia dataset. As a result, any dataset that reuses a DBpedia IRI would increase their PageRank according to this definition.

Alternatively, we could take advantage of the linked data principles which stipulate that IRIs should be addresses pointing to a location on the Web. Again, one could naïvely assume that the location that the address points to is what defines the dataset, that is, the document retrieved when one gets the resource using the HTTP protocol. However, this would lead us, for instance, to define each DBpedia article as an individual dataset.

A second possibility would be to use the domain part of the URL, so datasets are grouped by the same publisher. This approach is taken by Ding and Finin [33] to characterize data in the Semantic Web. This way, it would be easy to determine what dataset is being linked to. Such approach would work well if all datasets were accessible from dereferenceable IRIs. However, there are large portions of the Web of Data that provide access to data dumps only [79, 42]. In this case, the domain of the dump does not necessarily match the domain of the individual IRIs found in the dataset. As an example, the DBpedia dumps are found at `http://downloads.dbpedia.org/` while all DBpedia IRIs start with `http://dbpedia.org/`.

The last approach is to use the concept of *PLD*, *i.e.*, the subdomain component of a URL followed by a public suffix, to identify a dataset. Then, datasets are grouped not necessarily by the same publisher, but by the same publisher authority. This approach has already been used by other works [77, 121]. As an example, if a file found at `http://download.dbpedia.org/` contains the following triple:

```
<http://dbpedia.org/wiki/Europe>
        <http://www.w3.org/2002/07/owl#sameAs>
                <http://sws.geonames.org/6255148/>
```

We consider that the dataset having the PLD `dbpedia.org` is linking to the dataset with PLD `geonames.org`. It is important to notice that the source of the link (`dbpedia.org`) is obtained from the URL of the document that contains the triple (`http://download.dbpedia.org/`), not from the subject of the example. This approach enables us to extract outlinks from datasets published in dumps, and therefore access the majority of accessible semantic web data.

Definition 5.1 (Dataset)**.** *A* dataset *is a non empty collection of triples that can be retrieved from sources accessible at a URL having a common Pay-Level Domain. The PLD identifies the dataset.*

In the previous example, we see that the predicate IRI is linking to the standard OWL vocabulary. It is very likely that predicates in general will be linking to vocabularies that are extensively reused. However, our intent is to evaluate trust on actual

data that can be used to answer questions, and not vocabularies used to describe the data. We predict that extracting outlinks from predicates will lead to higher values for datasets containing only vocabularies. For this reason, we perform the same experiment with and without taking predicates into consideration.

Definition 5.2 (Dataset link)**.** *There exists a* link *from a dataset A to a dataset B if and only if there exists a triple in a file at a location having the PLD that identifies A in which the PLD of its subject, its object, or both matches the PLD that identifies B.*

This definition is in line with the PageRank algorithm [111] where the number of links between the same two nodes is irrelevant. Note that since datasets must be non empty, links to PLDs that do not host RDF have to be ignored.

5.2.2 The LOD Laundromat and Frank

The LOD cloud[2], and in general Linked Open Data, contains a wide variety of formats, publishing schemes, errors, that make it difficult to perform a large-scale evaluation. Yet, to be accurate, our study requires to be comprehensive. Fortunately, the LOD Laundromat [5, 118] makes this data available by gathering dataset dumps from the Web, including archived data. LOD Laundromat cleans the data by fixing syntactic errors and removing duplicates, and then makes it available through download (either as gzipped N-Triples or N-Quads, or HDT [44] files), a SPARQL endpoint, and Triple Pattern Fragments [127]. Using the LOD Laundromat is also a better solution than trying to use documents dereferenced by URIs, because most of datasets available online are data dumps [79, 42], thus not accessible by dereferencing.

Frank [117] is a command-line tool which serves as an interface of the LOD Laundromat, and makes it easy to run evaluations against very large numbers of datasets.

5.3 Experiments and Results

The process to compute PageRank involves the following steps, detailed further below and illustrated in Figure 5.1. The code and results are provided online.[3]

1. Extracting the document list from LOD Laundromat.
2. Parsing the content of each document to extract the outlinks.
3. Consolidating the results
4. Computing PageRank

5.3.1 Extracting the document list from LOD Laundromat

We use the Frank command line tool [117] to obtain a snapshot of the contents of the LOD Laundromat. While the output of Frank can be directly pipelined to our process, the next step is performed in parallel in several machines. For this reason, we need that every machine reads the exact same input. An update in the contents of the LOD Laundromat during the next process could have impacted the results in that case. We retrieve the list of documents in the LOD Laundromat with the following command.

[2] http://lod-cloud.net/

[3] https://github.com/jm-gimenez-garcia/LODRank

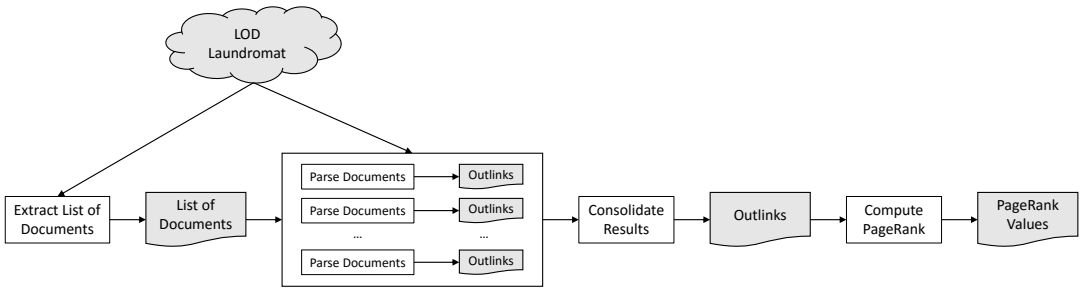


Figure 5.1: Outlink extraction and PageRank computation workflow

```
SELECT ?url
WHERE {<%s> <http://lodlaundromat.org/ontology/url> ?url}
```

Query 5.1: Query to retrieve crawled URL of a non-archived document

```
$ frank documents > documents.dat
```

This command retrieves a list of pairs *(downloadURL-resourceURL)*, where the first is the URL to download the gzipped datasets, and the second the resource identifier in the LOD Laundromat ontology. At the moment of the experiments, it retrieved 649,855 documents.

5.3.2 Parsing the content of each document to extract the outlinks.

A prototype tool[4] has been developed to stream the contents of the documents and extract the outlinks. This tool reads the list of pairs *(downloadURL-resourceURL)* from the standard input, and accepts two optional parameters for partial processing: *Step* and *Start*. The first one tells how many lines the process reads in every iteration, processing the last one, while the second denotes what line to use for the first input. For each line processed, it queries the SPARQL endpoint to retrieve the URL where that datasets was crawled. This information can be found in the LOD Laundromat ontology connected to the resource, in the case the document was crawled as a single file, or connected to the archive that contains the document, if it was crawled compressed in a compressed file, possibly along other documents. In the first case, we retrieve the URL with Query 5.1, in the second case we retrieve the URL using Query 5.2, where %s is substituted by the *resourceURL*. The Pay-Level Domain is then extracted and stored. This will be considered as the identifier of the dataset.

Then, the gzipped file is streamed from the *downloadURL* and the triples are parsed. The Pay-Level Domain is extracted for the subject and object (in case it is

[4] https://github.com/jm-gimenez-garcia/LODRank/tree/master/src/com/chemi2g/lodrank/outlink_extractor

```
SELECT ?url
WHERE {
       ?archive <http://lodlaundromat.org/ontology/containsEntry> <%s> .
       ?archive <http://lodlaundromat.org/ontology/url> ?url
}
```

Query 5.2: Query to retrieve crawled URL of an archived document

Process	Documents	Triples	Datasets (w. p.)	Datasets (w/o. p.)
1	81,220	3,994,446,393	135	121
2	81,226	3,742,870,561	137	118
3	83,422	4,146,249,367	140	127
4	81,225	3,376,784,600	135	120
5	81,225	3,623,413,245	142	120
6	88,198	3,377,773,585	131	116
7	81,226	4,132,960,522	137	115
8	89,781	3,911,917,919	134	123

Table 5.1: Data extracted from the LOD Laundromat by each process

a URI) for each triple and compared against their dataset PLD. If they have a valid PLD and it is different from their dataset's Pay-Level Domain, the pair *(datasetPLD-resourcePLD)* is stored as an outlink for the dataset. The output of each dataset is stored in a different file. This file will be updated if a different document is identified as the same dataset (*i.e.*, it has the same PLD).[a]

This process makes use of Apache Jena[5] v3.0.1 to query the SPARQL endpoint of the LOD Laundromat and Google Guava[6] v19.0 to extract the Pay-Level Domain of the datasets.

In the experiments the process was launched in parallel in 8 virtual machines using Google Cloud Platform[7] free trial resources, each one processing a different subset of the list downloaded in the previous step. A statistical description of the results of each process, with and without considering predicates, is detailed in Table 5.1. "Documents" correspond to the number of dump files in the LOD Laundromat, while "Datasets" are the number of PLDs that the process is dealing with. There can be an overlap in the datasets of several processes, so the total number of datasets is not equal to the sum. We can see that the number of triples processed by each process is not proportional to the number of documents processed.

5.3.3 Consolidating the results

Once the outlinks have been extracted, the different files have to be appended and duplicates removed using a simple tool.[8] In the experiments, the data from each virtual machine was downloaded in a separate folder of a unique machine. Then files with the same name in each folder were concatenated and we removed the duplicates. The total number of datasets after consolidating the results is 412 when considering predicates, and 319 when not. The result was again concatenated in a single file.

5.3.4 Computing PageRank

For PageRank computation we make use of the igraph R package [26]. The ordered PageRank values for all datasets can be seen in Figure 5.2 and Figure 5.3, with a

[5] https://jena.apache.org/

[6] https://github.com/google/guava

[7] https://cloud.google.com/

[8] https://github.com/jm-gimenez-garcia/LODRank/tree/master/src/com/chemi2g/lodrank/duplicate_remover

[a] This paragraph was updated in this chapter for the sake of clarity.

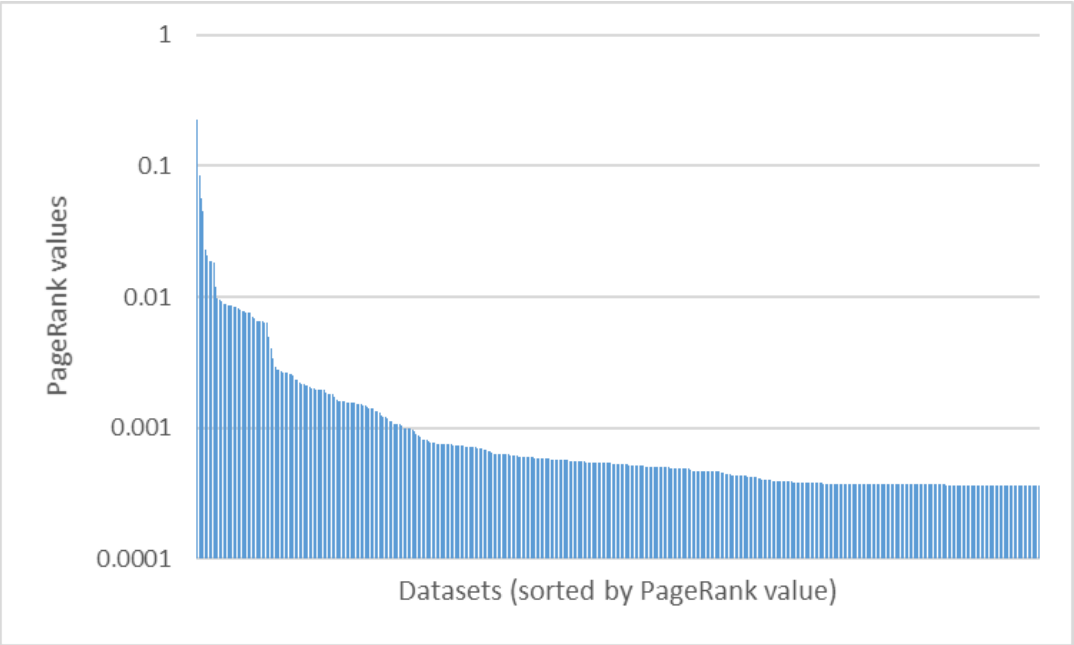


Figure 5.2: PageRank values sorted from higher to lower, with predicates

logarithmic scale. The complete list of results is published online.[9] We can see that in both cases the top-ranked dataset is very much higher than the rest, then the slope becomes more regular until it reaches a plateau at the end, with a minimum value shared by several datasets that have no inlinks at all. Tables 5.2 and 5.3 show the 10 highest ranked datasets.

5.3.5 Discussion

Here we provide additional information about the datasets, especially the top-ranked ones, in order to understand how ranking correlates with other statistical values, such as number of triples, number of documents. We also discuss how our own choices influenced the results.

The datasets appearing on the top 10 list are generally not surprising, with the only exception of `holygoat.co.uk`, the only domain in the top 10 owned by an individual person, Richard Newman, a computer scientist who wrote several ontologies in the early days of the Semantic Web. This is even more remarkable considering that the dataset has only 7 inlinks. The reason is that `rdfs.org`, the domain of `sioc` ontology for instance, includes resources from `holygoat.co.uk`. Because this dataset has only 2 outlinks, half of its PakeRank score is forwarded to `holygoat.co.uk`, which accounts for 96% of its PageRank value.

As predicted, when including predicates the first positions incorporate more datasets about vocabularies. When removing the predicates, `w3.org`, `xmlns.com`, `schema.org`, and `ogp.me` no longer appear in the top positions, and datasets with factual data move upwards. `lodlaundromat.org` seems to appear when considering predicates because the LOD Laundromat adds information about the cleaning process when processing the data. While not an optimum solution (considering that `purl.org` and `rdfs.org`, hosts of well known ontologies, are still in the top positions), ignoring the predicates proves to be a simple but useful technique.

[9] https://github.com/jm-gimenez-garcia/LODRank/tree/master/results

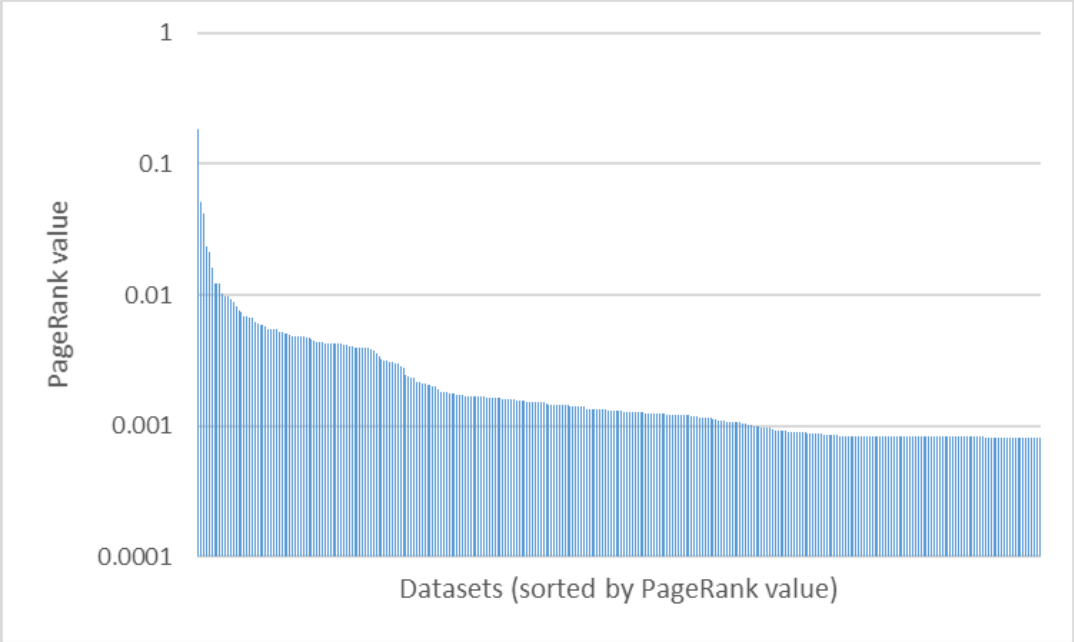


Figure 5.3: PageRank values sorted from higher to lower, without predicates

Rank	Dataset	PageRank value	Inlinks	Outlinks
1	w3.org	0.224806691	411	32
2	purl.org	0.085548846	278	64
3	lodlaundromat.org	0.056963209	188	1
4	xmlns.com	0.045452453	219	3
5	schema.org	0.023239532	32	1
6	creativecommons.org	0.020496922	106	2
7	dbpedia.org	0.018894825	118	160
8	rdfs.org	0.018738995	108	5
9	ogp.me	0.018442606	37	4
10	usefulinc.com	0.012066847	26	4

Table 5.2: PageRank values for the top 10 datasets, with predicates

Rank	Dataset	PageRank value	Inlinks	Outlinks
1	purl.org	0.185304616	181	50
2	creativecommons.org	0.051625742	93	1
3	dbpedia.org	0.04234706	104	119
4	rdfs.org	0.023497322	73	2
5	geonames.org	0.02127494	59	6
6	loc.gov	0.016137225	33	8
7	fao.org	0.012392539	27	8
8	europa.eu	0.012182709	30	13
9	holygoat.co.uk	0.012179038	7	1
10	data.gov.uk	0.010364034	19	11

Table 5.3: PageRank values for the top 10 datasets, without predicates

```
PREFIX llo: <http://lodlaundromat.org/ontology/>
PREFIX ll: <http://lodlaundromat.org/resource/>
SELECT (COUNT(DISTINCT ?resource) AS ?count)
WHERE {
       {
              ?resource llo:url ?url
              FILTER regex(?url, "[^/\\.]*\\.?%s/", "")
       }
       UNION
       {
              ?archive llo:containsEntry ?resource ;
                     llo:url ?url
              FILTER regex(?url, "[^/\\.]*\\.?%s/", "")
       }
}
```

Query 5.3: Query to retrieve the number of documents per dataset

```
PREFIX llo: <http://lodlaundromat.org/ontology/>
PREFIX ll: <http://lodlaundromat.org/resource/>
SELECT (COUNT(DISTINCT ?resource) AS ?count) (SUM(?triples) as ?sum)
WHERE {
       {
              ?resource llo:url ?url ;
                     llo:triples ?triples
       FILTER (?triples > 0)
              FILTER regex(?url, "[^/\\.]*\\.?%s/", "")
       }
       UNION
       {
              ?archive llo:containsEntry ?resource ;
                     llo:url ?url .
              ?resource llo:triples ?triples
       FILTER (?triples > 0)
              FILTER regex(?url, "[^/\\.]*\\.?%s/", "")
       }
}
```

Query 5.4: Query to retrieve the number of documents with triples and number of triples

We used two queries, (Query 5.3 and Query 5.4), to obtain the number of documents and triples for each PLD, from the LOD Laundromat.

The result of the queries are given in Table 5.4 for all the datasets that appear in the top 10 of both experiments.[b]

As we can see, popularity is not at all correlated with the size of the datasets. Indeed, a number of the top 10 datasets have less that 200 triples, while `dbpedia.org` and `europa.eu` both have billions of triples.

The enormously high page rank of `purl.org` should be mitigated by the fact that `purl.org` does not actually host any data. It is a redirecting service that many data publishers are using. This result highlights a drawback in our heuristic for identifying datasets: the PLD is not always referring to a single dataset. To overcome this particular case, we could consider the PLD of the URL of the document obtained

[b]Note that some numbers in the table may seem surprising, such as `schema.org` contains only one triple, but remember that these are the contents of the LOD Laundromat as of 2016.

Dataset	Rank	Documents	Documents with triples	Triples
w3.org	1 / -	413	256	1,973,715
purl.org	2 / 1	9,166	9,073	254,548,441
lodlaundromat.org	3 / -	68	1	4
xmlns.com	4 / -	4	4	1895
schema.org	5 / -	1	1	1
creativecommons.org	6 / 2	1	1	117
dbpedia.org	7 / 3	1,888	1,752	1,257,930,891
rdfs.org	8 / 4	6	6	1,808
ogp.me	9 / 274	68	1	231
usefulinc.com	10 / -	2	2	1398
geonames.org	20 / 5	6	4	9,762
loc.gov	29 / 6	16	12	263,653,979
fao.org	36 / 7	17	11	48366
europa.eu	12 / 8	7734	7,705	3,414,066,228
holygoat.co.uk	37 / 9	1	1	95
data.gov.uk	42 / 10	157	88	51,401,490

Table 5.4: Documents and triples per dataset in LOD Laundromat

after dereferencing the IRI, in the same way as Hogan, Harth, Umbrich, Kinsella, Polleres, and Decker [77] do for the general case.

Another possible drawback of the approach is that triples with `rdf:type` in predicate position have their object pointing to a class in an ontology. This is in contradiction with our remark in Section 5.2 where we say that we want to rank instance data rather than terminological knowledge. This can have a major impact on the results since `purl.org` is most often used to redirect to vocabularies more than datasets, and `rdfs.org` only hosts ontologies.

5.4 Related work

The authors of Semantic Web Search Engine (SWSE [77]) strongly advocate that the use of a ranking mechanism is very crucial for prioritizing data elements in the search process. Their work is inspired by the Google PageRank algorithm, which treats hyperlinks to other pages as a positive score. The PageRank algorithm is targeted for hyperlink documents and its adaptation to the LOD is however non-trivial, as we have seen. They point out that the primary reason for this is that LOD datasets may not have direct hyperlinks to other datasets but rather in most cases make use of implicit links to other web pages via the re-use of dereferenceable URIs. In their work the unit of search becomes the entity and not the document itself. The authors briefly re-introduce the concept of naming authority, from their previous work [66] in order to rank structured data from an open distributed environment. They assume that the naming authority should match the Pay-level domain such that computing PageRank is performed on a naming authority graph where the nodes are PLDs. Their intuition therefore is in accordance with our reasoning from Section 5.2. They have discussed and contrasted the interpretation of naming authorities on a document level (e.g. `http://www.danbri.org/foaf.rdf`) and a PLD level (`danbri.org`). Also, they make use of a generalization for the method discussed in the paper [35] for ranking entities and carry out links analysis on the PLD abstraction layer.

The authors of Swoogle [34] develop OntoRank algorithm in order to rank documents. OntoRank, a variation of Google PageRank, is an iterative algorithm for calculating the ranks for documents built on references to terms (i.e., classes and properties) which are defined in other documents.

In the paper [24], the authors calculate the rank of entities (or as they call them "objects") based on the logarithm of the number of documents where that particular object is mentioned.

In their work [63] present LinkQA, an extensible data quality assessment framework for assessing the quality of linked data mappings using the network measures. For this, they assess the degree of interlinking of datasets using five network measures, out of which two network measures are specifically designed for Linked Data (namely, Open Same-As chains and Description Richness) and the other three standard network measures (namely, degree, centrality, and the clustering coefficient) in order to assess variation in the quality of the overall linked data with respect to a certain set of links.

In [10], PageRank is used to compute a measure that is in turn associated to individual statements in datasets for the purpose of incorporating trust in reasoning. Therefore, as in our own approach, they consider that PageRank is an indication of trustworthiness. However, they only compute PageRank on a per document basis, and report on the PageRank values of the top 10 documents obtained from their web crawl.

5.5 Conclusion & Future work

Data-driven question answering, the aim of project WDAqua mentioned in the introduction of this paper, requires quality data in which one can trust. Our aim has been to provide insight on how a trust measure can be based on dataset interlinking. To that end, we consider Pay-Level Domains as identifiers of unique datasets and compute PageRank on them. Our results show that the design choices greatly affect the results. Whether taking into account or not predicates for outlink extraction impacts how vocabularies are ranked, and the choice of PLD as definition of dataset is arguable, as some PLDs group many data dumps. In order to improve this, we could associate well known datasets to IRI patterns, such as `it.dbpedia.org` for the Italian version of DBpedia.

In addition, we also intend to explore further applications of PageRank that may be useful for question answering. User interaction that provides trust values in a number of dataset could be used to compute PageRank values with those datasets as a teleport set, as suggested by Gyöngyi, Garcia-Molina, and Pedersen [65]. Also, Topic-Sensitive PageRank [70] could help a question-answering system to select different datasets when a question is identified to belong to a specific topic.

Finally, this work is part of a broader objective that we want to pursue: to ascertain the relationship between the perceived trust on a dataset and its objective quality. We will explore this area in a future work where other data reuse metrics will be considered and compared against different quality metrics.[c]

References

[5] Beek, W., Rietveld, L., Bazoobandi, H.R., Wielemaker, J., Schlobach, S.: LOD Laundromat: A Uniform Way of Publishing Other People's Dirty Data. In: Lecture Notes in Computer Science

[c] This work is presented in Chapter 6.

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 8796, pp. 213–228. Springer, Heidelberg (2014).

[10] Bonatti, P.A., Hogan, A., Polleres, A., Sauro, L.: Robust and scalable Linked Data reasoning incorporating provenance and trust annotations. Web Semantics: Science, Services and Agents on the World Wide Web 9(2), 165–201 (2011).

[24] Cheng, G., Qu, Y.: Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. International Journal on Semantic Web and Information Systems 5(3), 49–70 (2009).

[26] Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJournal, Complex Systems 1695(5), 1–9 (2006).

[29] Debattista, J., Londoño, S., Lange, C., Auer, S.: Quality Assessment of Linked Datasets Using Probabilistic Approximation. In: The Semantic Web. Latest Advances and New Domains - ESWC 2015 - 12th Extended Semantic Web ConferenceLNCS, vol. 9088, pp. 221–236. Springer, Heidelberg (2015).

[33] Ding, L., Finin, T.: Characterizing the Semantic Web on the Web. In: The Semantic Web - ISWC 2006 - 5th International Semantic Web ConferenceLNCS, vol. 4273, pp. 242–257. Springer, Heidelberg (2006).

[34] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management, pp. 652–659. ACM (2004).

[35] Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P.: Finding and Ranking Knowledge on the Semantic Web. In: The Semantic Web - ISWC 2005 - 4th International Semantic Web ConferenceLNCS, vol. 3729, pp. 156–170. Springer, Heidelberg (2005).

[42] Ermilov, I., Martin, M., Lehmann, J., Auer, S.: Linked Open Data Statistics: Collection and Exploitation. In: Knowledge Engineering and the Semantic Web - KESW 2013 - 4th International ConferenceCommunications in Computer and Information Science, pp. 242–249. Springer Berlin Heidelberg, St. Petersburg, Russia (2013).

[44] Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary RDF representation for publication and exchange (HDT). Journal of Web Semantics 19, 22–41 (2013).

[54] Giménez-García, J.M., Thakkar, H., Zimmermann, A.: Assessing Trust with PageRank in the Web of Data. In: The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected PapersCEUR Workshop Proceedings, pp. 293–307. Springer, Cham (2016).

[63] Guéret, C., Groth, P., Stadler, C., Lehmann, J.: Assessing linked data mappings using network measures. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 7295 LNCS, pp. 87–102. Springer, Heidelberg (2012).

[65] Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with TrustRank. In: Proceedings of the Thirtieth international conference on Very large data bases, pp. 576–587. VLDB Endowment (2004).

[66] Harth, A., Kinsella, S., Decker, S.: Using naming authority to rank data and ontologies for web search. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 5823 LNCS, pp. 277–292. Springer, Heidelberg (2009).

[70] Haveliwala, T.H.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE Transactions on Knowledge and Data Engineering 15(4), 784–796 (2003).

[77] Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A., Decker, S.: 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 (2011).

[79] Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., Decker, S.: An empirical survey of Linked Data conformance. Journal of Web Semantics 14, 14–44 (2012).

[93] Liu, S., D'Aquin, M., Motta, E.: Towards Linked Data Fact Validation through Measuring Consensus. In: Proceedings of the 2nd Workshop on Linked Data Quality co-located with 12th Extended Semantic Web Conference (ESWC 2015)CEUR Workshop Proceedings, CEUR-WS.org, Portorož, Slovenia (2015).

[111] Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. World Wide Web Internet And Web Information Systems 54(1999), 1–17 (1998).

[112] Paulheim, H., Bizer, C.: Improving the Quality of Linked Data Using Statistical Distributions. International Journal on Semantic Web and Information Systems 10(2), 63–86 (2014).

[117] Rietveld, L., Beek, W., Schlobach, S.: LOD lab: Experiments at LOD scale. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 9367, pp. 339–355. Springer, Heidelberg (2015).

[118] Rietveld, L., Verborgh, R., Beek, W., Sande, M.V., Schlobach, S.: Linked Data-as-a-Service: The Semantic Web Redeployed. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 9088, pp. 471–487. Springer, Heidelberg (2015).

[121] Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 8796, pp. 245–260. Springer, Heidelberg (2014).

[125] Thakkar, H., Endris, K.M., Gimenez-Garcia, J.J.M., Debattista, J., Lange, C., Auer, S., Thakkar, H., Endris, K.M., Gimenez-Garcia, J.J.M., Debattista, J., Lange, C., Auer, S.: Are Linked Datasets fit for Open-domain Question Answering? A Quality Assessment. In: Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics, WIMS 2016, pp. 1–12. ACM Press, Nîmes, France (2016).

[127] Verborgh, R., Vander Sande, M., Colpaert, P., Coppens, S., Mannens, E., deWalle, R.: Web-Scale Querying through Linked Data Fragments. In: Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014)CEUR Workshop Proceedings, Seoul, Korea (2014).

Chapter 6

What does Dataset Reuse tell us about Quality?*

In the previous chapter, we extracted how datasets are reused by other datasets. In this chapter we continue this line of work by extracting how datasets are used and cited by users, presenting metrics to quantify dataset reuse in a scientific community. Considering the broad definition of data *quality* as "fitness for use", we compare this data against different quality metrics, trying to glean any correlation between them.

The contents of this chapter do not make use of precedent content. They arise from work performed in parallel during the thesis, and similarly to the content of precedent chapter, they can be read non-sequentially with the rest of the thesis. It provides results in the contexts of reuse and quality about existing data. In conjunction with the previous chapter, it highlights the importance of having generic approaches to extract existing contextual data, as well as formalisms to represent them. Thus, these two chapters show the motivation to formulate the challenges and research questions of this thesis.

6.1 Introduction

The number of datasets publicly available within the Linked Open Data (LOD) cloud is rapidly growing and has increased by more than three times in just three years from 295 datasets in 2011 to over 1,000 datasets in 2014 [121]. Despite this growth, the interlinking, and, speaking more generally, *reuse* of the Web of Data remains limited, and is often focused on few well-known reference datasets, such as DBpedia [3], YAGO [123] or Freebase [9]. However, are those datasets that are being published, actually useful for the applications, or, vice versa, which characteristics of the datasets can be optimized to facilitate their broader reuse? Considering the broad definition of data *quality* as "fitness for use" [129], this research question can be rephrased as follows: Which dataset characteristics or quality metrics correlate with high reuse? Intuitively, one would expect a strong correlation, at least with respect to some quality dimensions and forms of reuse. A better understanding of this relationship can guide the data publishing process, support dataset generation and

*This chapter is based on unpublished content by José M. Giménez-García, Kemele M. Endris, Harsh Thakkar, Elena Demidova, Elena Simperl, Christoph Lange, and Antoine Zimmermann in 2016. This content was later adapted for the following publication:

- Endris, K.M., Giménez-García, J.M., Thakkar, H., Demidova, E., Zimmermann, A., Lange, C., Simperl, E.: Dataset Reuse: An Analysis of References in Community Discussions, Publications and Data. In: Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017, 5:1–5:4 (2017). [41]

maintenance and, more generally speaking, facilitate a more efficient take up of the Web of Data, broadening the spectrum of the dataset reuse.

Recently, an increasing number of quality dimensions and corresponding metrics has been discussed in the literature and a growing number of tools has become available to evaluate these metrics in the context of LOD (see for example a recent dataset profiling survey by Ellefi et al. [40]). Existing quality metrics reflect various quality dimensions such as accessibility, availability, licensing, interlinking, security and performance [135]. Given this variety, we can phrase our question more precisely: Which of the existing quality metrics correlate with the actual reuse of the datasets in different use cases?

The aim of the paper is to quantify the correlation between the reuse of a dataset and its quality metrics at the example of the Semantic Web community. We investigate dataset reuse aspects including mentions and reuse of datasets in publications that appear in the proceedings of the key Semantic Web and Web conferences such as ISWC, ESWC and WWW in the time frame from 2007 to 2015, mailing list discussions about the datasets on the *public-lod@w3.org* and the *semantic-web@w3.org* mailing lists and mutual reuse of datasets in terms of interlinking. We define and compute reuse metrics and build correlations between selected quality metrics and observed reuse behavior.

Our results provide first insights in the correlations with respect to dataset reuse across different communication channels and reuse forms in the Semantic Web community, such as publications, mailing lists, dataset paper citations and interlinking. They also indicate interesting patterns with respect to the correlations of the selected quality and reuse metrics.

6.2 Related Work

In recent years a lot of research has focused on different aspects of quality evaluation for Linked Data [135], conformance of published datasets to the Linked Data best practices [79, 121], dataset profiling [40] and scientific impact of published datasets [78]. In this section we discuss these related areas in more detail.

Dataset quality. Dataset quality can be defined as "fitness for use" [129]. This rather broad definition has resulted in a variety of quality dimensions and metrics that have been identified as relevant for different applications. A recent survey of Linked Data quality assessment methods [135] provides insights in this space, including 18 quality dimensions that can be grouped in accessibility, intrinsic, contextual and representational categories as well as 69 metrics to precisely measure quality w.r.t. these dimensions. Some of these quality metrics also build an important part of dataset *profiles* – descriptive metadata to facilitate dataset discovery and reuse in applications [40]. Despite this variety, quality metrics discussed in the literature only reflect certain forms of reuse, such as interlinking and reuse of vocabularies. Our study focuses on a broader analysis of different reuse forms and their correlation with further quality dimensions.

Best practices conformance. The aim to facilitate better discovery and an efficient integration of Linked Data by applications has inspired best practices for Linked Data publishing. These best practices include interlinking, vocabulary reuse and

metadata provision [71].[1] Recently, several works analyzed the adoption of best practices: In 2011, the state of the LOD cloud was analyzed using metadata from the `datahub.io` portal [83]. In 2014, Schmachtenberg et al. continued this analysis by conducting a crawl of the Web of Linked Data that collected over 1,000 datasets containing above 8 million resources [121]. In particular, this study provides statistics related to linking, provenance, and metadata computed using the crawl [121].

Hogan et al. provided results for many metrics that relate to Linked Data best practices and discussed how these metrics relate to quality [79]. Most importantly, they mention the correlation (or lack thereof) between best practice conformance and PageRank [111] – a metric of measuring the importance of resources based on their interlinking. Their conclusion was that highly ranked datasets do not necessarily conform well to best practices.

Dataset impact through scientific publications. The importance of publishing datasets is increasingly recognized in the scientific community (see e.g. [114, 78]). On the one hand, it has been observed that papers providing open datasets as a part of their contribution make a greater impact than those without [114]. For example, the DBpedia paper by Lehmann et al. [92] has received 334 citations since 2015 [78]. On the other hand, new channels for publishing dedicated dataset papers in high impact venues have emerged. For example, the Linked Dataset Description track of the Semantic Web Journal has published 38 dataset papers in the period 2012–2015.[2] By the end of 2015, each of these papers has already been cited 7.35 times on average [78]. Although the number of pure dataset papers is still limited, we would like to obtain a first impression if these datasets expose differences in reuse compared to other datasets in our collection.

Dataset reuse and quality. Judging dataset usefulness, e.g., in terms of coverage of domain-specific aspects, can be difficult without domain-specific knowledge [78]. In this context, *evidence of actual dataset reuse* gains increasing attention in the context of dataset publishing and, in particular, evaluation of dataset papers. The current call for submissions to the SWJ dataset track even requires such evidence [78]. In this work we analyze if a correlation analysis between reuse and quality metrics can support such judgements.

6.3 Problem Statement and Methodology

The aim of this paper is to better understand reuse of datasets by the community and to quantify the possible correlation between dataset reuse and its quality metrics. Dataset reuse leaves diverse traces, most prominently including dataset references in scientific publications, discussions on community mailing lists and citations of dedicated dataset papers. We quantify dataset reuse by defining metrics for each of these sources.

Evaluating dataset quality over the LOD cloud poses computational challenges due to the large scale of the data and a wide variety of metrics. We aim at covering various quality dimensions discussed in the literature, such as availability, licensing, interpretability and others while focusing on those quality metrics that can be

[1] According to the W3C Data on the Web Best Practices Working Group, best practices for publishing data on the Web even include providing metadata about the quality of the data to guide selection and reuse [43].

[2] `http://www.semantic-web-journal.net/accepted-datasets`

computed efficiently. This approach can enable efficient evaluation and, if necessary, adaptation of datasets.

In summary, to facilitate this analysis we perform the following steps:

1. We define dataset reuse metrics relying on different information sources such as: 1) Publications and communication channels within a scientific community, and 2) Data interlinking.
2. We select efficiently computable quality metrics covering a broad range of dimensions.
3. We compute these reuse and quality metrics on datasets from the LOD cloud and analyze the correlation between these metrics.

The following sections cover these steps in more detail.

6.3.1 Dataset Reuse Metrics

Dataset reuse leaves traces in different communication channels within scientific communities: by mentions in publications and mailing lists, and by means of citations of dataset papers. Further indications of reuse include references to data instances and vocabulary reuse that can be observed within other linked datasets. We propose metrics that quantify dataset reuse using these indications.

Dataset mentions in publications and mailing lists.

First we define the concept of a "dataset mention" in publications and mailing lists. Then we present our reuse metric for these channels, i.e. popularity. Finally, we discuss the procedure of dataset mention extraction.

Dataset mention. In the context of scientific publications and mailing list discussions, a dataset can be referenced by its metadata including its name, URI, etc. We call a reference to a dataset *ds* in a document *d* using (some of) its metadata a *mention*. Here, a document can be a scientific publication or a mailing list thread. We model a dataset mention in a document as a binary relation R_M where $(ds, d) \in R_M$ if and only if the dataset is mentioned in the document. We consider mailing list posts at the granularity of threads, i.e. define $(ds, d) \in R_M$ if there exists an email within the thread mentioning the dataset. To determine dataset mentions, we build a dataset dictionary including dataset metadata (obtained from `datahub.io`) and, match and disambiguate dataset metadata identified in documents against this dictionary, as discussed below.

Dataset reuse. We measure dataset reuse in publications and mailing lists using the notion of popularity. Given a collection of documents D, e.g., a proceedings volume or a mailing list archive containing a number of threads, we compute the popularity of a dataset *ds* as the ratio of the documents (i.e. publications or mailing list threads) that mention this dataset, respectively:

$$popularity(ds, D) = \frac{\#\{(ds, d) \in R_M | d \in D\}}{\#D}. \tag{6.1}$$

Extraction and disambiguation of dataset mentions. Datasets are not referenced in publications in a standardized way [8]; therefore, extracting and disambiguating dataset mentions can be challenging. A manual inspection of our document collection (cf. Section 6.4.1) has shown that datasets can be referred to using various metadata fields, such as dataset full name, short name, homepage URL, link to a dataset entry on a data portal (such as `datahub.io`), data dump or download URL, SPARQL endpoint, service API URL, or by citing the dataset description paper. Furthermore, we observed that dataset properties as they appear in the full-text of publications or mailing lists can be ambiguous, e.g., different notions of the "name" of a dataset used in different contexts. For example, in our document collection, the references to the dataset `semanticweb.org` listed on `datahub.io` cannot be distinguished easily from the references to the website with the same name.

In our work we choose a pragmatic approach to identifying and disambiguating dataset mentions: The dataset mention must be uniquely identified in the document using either: a) One of the dataset unique attributes, such as the name (if non-ambiguous) or the URL; or b) A citation of the dataset description paper in the reference section of the document. For instance, `datahub.io` has 8 dataset entries that contain the term "WordNet" in their full name ("WordNet", "WordNet 3.0 (VU Amsterdam)", etc.). Therefore, in case the paper includes the term "WordNet", we use other metadata fields to disambiguate the mention. A manual evaluation of this approach on a random sample of 25 articles indicates high precision of 0.93.

Dataset references in linked datasets.

It is a best practice to reuse resources from external datasets where possible. We consider a dataset *ds* to be reusing a resource from an external dataset ds_e if in the original dataset *ds* there is a triple that contains an IRI from the namespace of ds_e. In short, we speak of the original dataset *ds referencing* the external dataset ds_e. To estimate dataset popularity based on dataset references, we model the dataset collection as a directed graph whose nodes are datasets and whose edges are dataset references. On such a graph one can compute a PageRank [111] value for each dataset. PageRank value represents the steady-state probability of the random walk in a node; we can also say that it represents the popularity of the dataset based on the link structure in the graph.

Dataset paper citations.

In case a dedicated dataset paper is available, reuse can also be measured in terms of the citations of this paper, as specified in [78].

6.3.2 Data Quality Metrics

To correlate reuse and quality metrics in a representative way, we intended to cover a broad range of quality dimensions, while at the same time taking into account efficiency of the computation to enable evaluation on a large scale. Our selection of metrics is based on [135], where 69 metrics are listed, categorized in 17 dimensions. Some metrics in [135] cannot be computed efficiently on large scale datasets (e.g., high throughput is defined in [135] as the "(maximum) no. of answered HTTP-requests per second"), and there are others that can only be computed in presence of a particular task (e.g., schema completeness, described in [102] requires knowledge of the task-relevant attributes).

Our work in this paper focuses on the quality metrics that can either be obtained from the dataset description in third-party sources, or by parsing each triple of a dataset at most once. In particular, we used the LOD Laundromat[3] (LLm), SPARQL endpoint status[4], and `datahub.io`. To perform the computation of the remaining metrics, we streamed the datasets from the LOD Laundromat. Based on these considerations, overall we cover 8 of the 17 dimensions, obtaining results for 10 metrics. Each metric is normalized to the range $[0, 1]$, where a higher value indicates better quality. Table 6.1 lists our chosen metrics.

For the reasons discussed above, we do not cover the following dimensions from [135] but leave their investigation to future research: Security, Semantic Accuracy, Consistency, Conciseness, Completeness, Relevancy, Trustworthiness, Timeliness, and Interoperability.

6.3.3 Correlation Analysis

Finally, we analyze the correlation between the reuse metrics described in Section 6.3.1 and quality metrics described in Section 6.3.2 using the Pearson Correlation Coefficient (PCC). This computation requires our dataset collection to be represented as a vector space, where each dataset represents a dimension, and each metric is a variable. $PCC \in [-1, 1]$ is a measure of the linear correlation between two variables, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation.

6.4 Data Collection

After presenting our document collection, which includes publications and mailing lists threads, as well as our dataset dictionary containing dataset metadata, we present extraction results for dataset mentions from our document collection.

6.4.1 Publications

We consider publications from three key conferences in the fields of Semantic Web and Web, namely ISWC, ESWC and WWW. As the earliest dataset entry on `datahub.io` appeared in 2007, we consider publications from 2007 to 2015. We include all papers published in the proceedings of these conferences except the Part II of the ESWC 2010 and 2011 proceedings, which we did not have access to at the time of writing. In total our collection contains 2,162 papers. We extracted dataset mentions appearing in the main content, the evaluation section and the reference part of each paper. To this end we used pdfminer[5] to extract full-text from the publications, regular expressions to match relevant sections of the papers and the mention extraction and disambiguation procedure described in Section 6.3.1.

In the context of scientific publications, a dataset mention in the evaluation section can be a particularly strong indicator of dataset reuse, e.g., for an algorithm evaluation or data analysis. Therefore, we initially extracted dataset mentions appearing in these sections separately. Nevertheless, as we observed in our test collection, there is a strong correlation (PCC of 0.955) between the mentions of datasets in the evaluation section and the rest of the full-text in a publication. In other words,

[3] http://lodlaundromat.org/
[4] http://sparqles.ai.wu.ac.at/
[5] https://euske.github.io/pdfminer/

Table 6.1: Overview of computed quality metrics (definitions by [135]).

Dimension /Metric	Definition	How
Availability	The extent to which data (or some portion of it) is present, obtainable and ready for use.	
Endpoint availability	Proportion of time when the SPARQL endpoint is available (1 if no downtime detected, 0 if never available or there is no endpoint for the dataset).	SPARQLES[a]
Licensing	The granting of permission for a consumer to reuse a dataset under defined conditions.	
License	Presence of open license information (1), or either closed license or no presence of license information (0).	`datahub.io`
Interpretability	If data is in a machine-readable representation.	
Blank Nodes	Proportion of unique IRIs relative to the sum of unique IRIs plus blank nodes (1 if there are no blank nodes in the dataset, 0 if there are no IRIs).	LLm[b] metadata
Performance	The efficiency in accessing data of the dataset.	
Low latency	Inverse of the average response time to 8 different queries, normalized between [0,1) (1 if zero seconds, 0 if greater than 10 seconds).	SPARQLES
Syntactic Validity	The degree to which an RDF document conforms to the specification of the serialization format.	
Serialization Warnings	The inverse of the proportion of serialization warning relative to the number of triples in the dataset.	LLm metadata
Serialization Errors	The inverse of the proportion of serialization errors relative to the number of triples in the dataset.	LLm metadata
Understandability	The extent to which data can be used to provide unambiguous information to a human consumer.	
Labels	Proportion of labels relative to the number of unique IRIs in the dataset.	LLm processing
Conciseness	The extent to which the dataset does not contain redundant information.	
Query Parameters	Proportion of unique IRIs that include query parameters relative to the total number of unique IRIs in the dataset.	LLm processing
RDF Primitives	Proportion of RDF primitives (aggregate of RDF reification, RDF containers, and RDF collections statements) relative to the number of triples in the dataset (1 if there are no such statements, 0 in the case there would be no other statements).	
Versatility	The extent to which data is available in different representations and in an internationalized way.	
Different Languages	Proportion of languages relative to the maximum number of languages in the set of datasets.	LLm processing

[a] SPARQL Endpoint Status
[b] LOD Laundromat

for the majority of the publications analyzed, if a dataset was mentioned in any part of the paper, it was also mentioned in the evaluation section. Therefore, we do not further differentiate between the mentions in different sections of the publications.

6.4.2 Mailing Lists

We downloaded a crawl containing JSON encodings of the discussions on the *public-lod@w3.org* (available from 2008 to 2015) and *semantic-web@w3.org* (available from 2007 to 2015). From the mailing lists, we extracted the title, date, body and answer(s) body of each thread. For extracting and disambiguating dataset mentions from publications and mailing lists, we used the approach discussed in Section 6.3.1. In total, we extracted 9,046 discussion threads from *semantic-web@w3.org* and 4,661 discussion threads from *public-lod@w3.org*.

6.4.3 Dataset Dictionary

We extracted metadata of 1,131 datasets tagged as Linked Open Data ("lod") from `datahub.io`. 818 of these are available under an open license according to the open definition [6] ("isopen" tag) and 313 other datasets (194 of them do not specify any license information and 119 explicitly specify other licenses, e.g., a commercial license). We refer to this group of 313 datasets as "non-open" in the following. From the metadata we used the name (unique in the catalogue) of a dataset, the title (long name), the homepage URL and the resources URL (downloadable resources and/or service APIs such as endpoints) to construct our dictionary. We also recorded the year the dataset entry was added to `datahub.io`. Furthermore, we identified publications describing 142 of the datasets in the dictionary by manually inspecting homepages of datasets and searching Google Scholar for the dataset title. To provide a comprehensive metadata collection, during our manual inspection we also added alternative dataset names used by the dataset authors on their homepages to the dictionary.

6.4.4 Dataset Dumps

To enable computation of quality metrics, we retrieved metadata and a total of 430 data dumps from the LOD Laundromat. We used LOD Laundromat because it offers uniform access and format for all datasets. This reduces the cost of preparing the data by many orders of magnitude. The loss in number of datasets retrieved is acceptable as we still get a significantly large number of them (about 40%). Metadata includes descriptive information about the dataset (e.g., such as number of triples or different IRIs), and processing reports (e.g., number of syntactical warnings and errors when parsing the dump).

We used the resources URL from our dataset dictionary to identify the documents for each dataset in the LOD Laundromat. Those documents where used to either extract metadata by querying the LOD Laundromat SPARQL endpoint, or to stream the cleaned N-Triples or N-Quads dumps they provide.

6.4.5 Resulting Collection: An Overview

Figure 6.1 shows an overview of data collected for our evaluation (the y axis is logarithmic). Figure 6.1.a) illustrates the number of datasets published on `datahub.io`

[6] `http://opende{f}{i}nition.org`

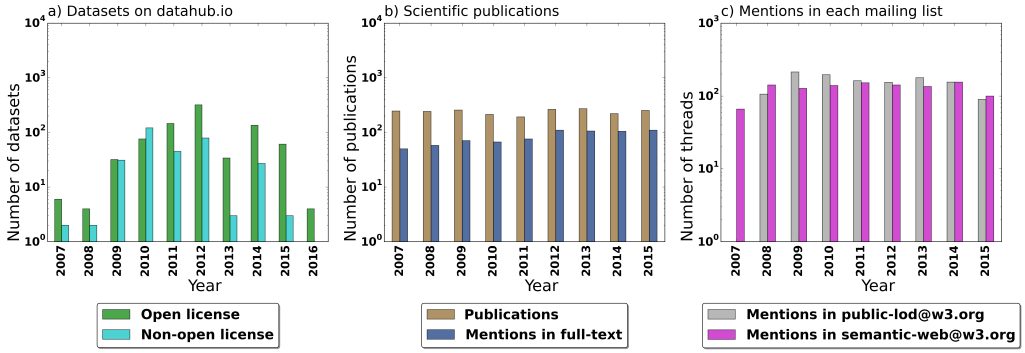


Figure 6.1: Data collection overview

in each year and the proportion between open and non-open datasets. The number of new entries on `datahub.io` was particularly high in 2009–2012 and 2014, with a total of 818 datasets added in these years (i.e. 72% of the overall 1131 datasets in our collection). While the majority of the datasets in our collection are open, most non-open datasets were added in 2010–2012. A total of 245 out of 313 non-open datasets, i.e., 78.2%, was added in this period. Figure 6.1.b) shows the aggregated number of papers from three scientific conferences—ISWC, ESWC and WWW—in each year and the number of dataset mentions in the full-text. The number of publications remained stable over the whole time interval (a slight decrease in 2010–2011 is due to Part II of the ESWC proceedings missing in our collection), whereas the number of dataset mentions in the publications has grown from 50 in 2007 to 110 in 2015. Also, we observed that more recently the number of publications without an evaluation section decreased, from 61 publications in 2012 to 39 in 2015, probably reflecting an increasing importance of the evaluation within the community. Finally, Figure 6.1.c) shows the number of dataset mentions extracted from the *semantic-web@w3.org* and *public-lod@w3.org* mailing lists. The overall number of threads in both mailing lists remained rather stable over time with an average of 1,005 per year for the *semantic-web@w3.org* and 583 per year for *public-lod@w3.org* (starting from 2008). As we can observe, the average number of dataset mentions is slightly higher in the *public-lod@w3.org* mailing list with 156.8 threads per year, vs. 128.6 threads on average in the *semantic-web@w3.org* mailing list.

6.5 Evaluation

The goal of our evaluation is to provide insights in the dataset reuse in the communication channels we observed, and to analyze the correlation of the reuse metrics defined in Section 6.3.1 and the selected quality metrics described in Section 6.3.2 on our dataset collection. [7]

6.5.1 Reuse Evaluation Results

We first discuss dataset reuse as reflected in different channels.

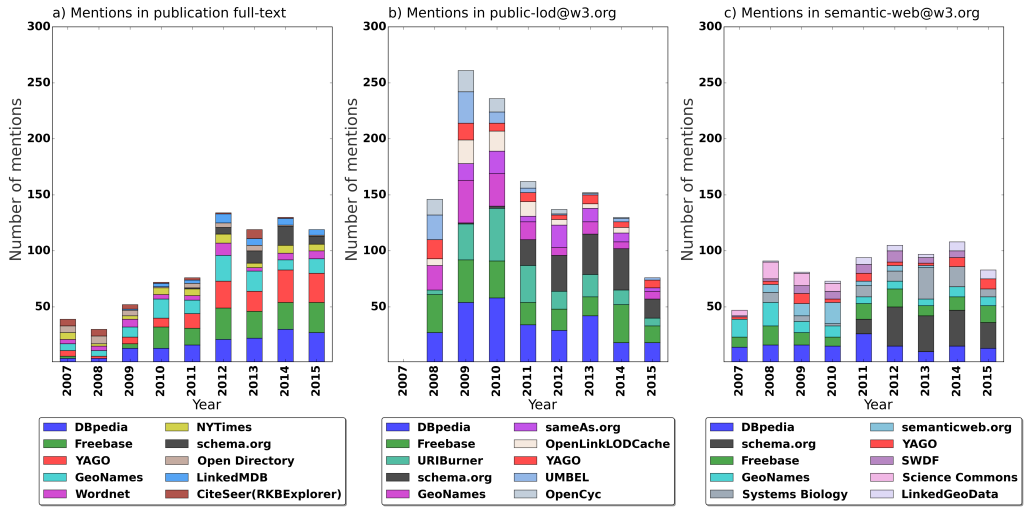


Figure 6.2: Top-10 datasets mentioned from 2007 to 2015.

Dataset mention results.

Figure 6.2 shows the top datasets mentioned in publications and mailing lists. Part a) shows the top-10 datasets mentioned in the full-text of scientific publications. The number of mentioned datasets has grown over the years with a particular increase starting from 2012—the same year for which we observed an increased number of `datahub.io` entries; cf. Figure 6.1.a). Overall, DBpedia, Freebase, YAGO and GeoNames are the most widely mentioned datasets across all channels. Overall each of them were mentioned more than 100 times in publications. Figure 6.2.b) shows the top-10 datasets mentioned in the *public-lod@w3.org* mailing list. Fig. 6.2.c) shows the top-10 datasets mentioned in the *semantic-web@w3.org* mailing list. There was a particularly high number of dataset discussions in the *public-lod@w3.org* mailing list between 2009 and 2010. We can observe, that *public-lod@w3.org* mailing list has been overall more popular for the dataset discussions than the *semantic-web@w3.org* mailing list.

Correlation between reuse metrics in publications and mailing lists.

Table 6.2 presents an overview of the correlation between the reuse metrics computed for different communication channels using PCC. There is a strong correlation between the dataset mentions in the overall full-text of the publications and the mentions in the evaluation section and other sections. Other communication channels are also strongly correlated, with $PCC = 0.86$ between the two mailing lists, and a moderate positive relationship ($PCC > 0.65$) between the mailing lists and the publications.

Reuse of resources in datasets.

In our collection, most of the datasets are associated with a namespace specified in `datahub.io`. To estimate reuse of resources, we consider the datasets for which we

[7]Statistics collected during our evaluation can be viewed in: https://drive.google.com/folderview?id=0B6dJh1dMpPRScUp0UkV2WWY4YXM

	Full-text	*semantic-web*	*public-lod*	Non-eval.	Evaluation
Full-text	1.	0.659814	0.682735	0.837973	0.838151
semantic-web	0.659814	1.	0.860982	0.762222	0.720168
public-lod	0.682735	0.860982	1.	0.766012	0.745704
Non-eval.	0.837973	0.762222	0.766012	1.	0.955192
Evaluation	0.838151	0.720168	0.745704	0.955192	1.

Table 6.2: Correlation between reuse metrics.

have both—namespace definition and dumps—available from the LOD Laundromat. This sub-collection contains 393 datasets in total. To compute reuse of the resources, we extracted references to other datasets in our sub-collection by streaming and parsing these datasets from the LOD Laundromat. After the extraction we removed duplicate references and computed PageRank on the resulting dataset graph, obtaining overall 261 datasets that were referenced at least once.

Table 6.3 presents the correlation results between PageRank and other reuse metrics for the 261 datasets in our collection that have $PageRank > 0$. For these datasets we observe a moderate positive correlation between the PageRank values and reuse in publications, and a strong positive correlation between PageRank and reuse in mailing lists.

Evolution of reuse over time.

	Full-text	Non-eval.	Evaluation	*semantic-web*	*public-lod*
PageRank	0.643109	0.559696	0.622233	0.734210	0.732594595

Table 6.3: Correlation between PageRank and reuse metrics for 261 datasets that have $PageRank > 0$.

We compute the correlation between the creation year of a dataset with its popularity in each channel. The results are shown in Table 6.4. First, we use the year the dataset entry was created in `datahub.io`. The result shows that there is no correlation between the metadata creation year with the popularity of the dataset in other channels. We then calculate an estimate of the dataset creation year by looking into the year the dataset was mentioned for the first time in a publication or a mailing list, and select the earliest date. The correlation result shows that there is a weak positive correlation between the estimated creation year with the dataset popularity in the full text of the publications. This illustrates that popularity of datasets in publications can grow with their age. However, we did not observe any significant correlation between the age of the dataset and its popularity in the mailing lists. For example, OpenCyc was often discussed in the public-lod mailing list, but become less popular with time. DBpedia had its discussion peak in the public-lod mailing list in 2009–2010; later it was mainly mentioned in the publications.

Dataset papers in SWJ dataset track.

The most cited dataset published in the dedicated track of Semantic Web Journal (SWJ) to date is `AGROVOC` with 39 citations of its SWJ paper [79]. Among all

	Full-text	*public-lod*	*semantic-web*
Metadata Created	0.128290	0.191551	0.143659
Dataset created (est.)	0.422133	0.225004	0.230189

Table 6.4: Correlation between the popularity of a dataset and its age in `datahub.io` or an estimated dataset creation date.

SWJ datasets under consideration, this dataset has also the highest values with respect to the reuse indications in publications (13 mentions) and mailing lists (24 `public-lod@w3.org`). Although the absolute numbers of mentions are not very high, that can be explained by the relatively young age of the SWJ dataset track.

	Full-text	*public-lod*	*semantic-web*
Paper Citations	0.270327	0.555587	0.403551

Table 6.5: Dataset published with dataset description in SWJ.

Table 6.5 presents correlation results between the citations of the dataset papers published in the SWJ track and reuse of these datasets as indicated in publications and mailing lists. The correlation results show that there is a positive correlation between the number of citations of the dataset papers and mentions in publications, that is currently most prominent in the `public-lod@w3.org` mailing list.

6.5.2 Dataset Quality Evaluation Results

We computed the quality metrics presented in Section 6.3.2 for the 430 datasets from our dictionary whose data dumps are in the LOD Laundromat. Table 6.6 displays the quality metrics for the top-10 datasets with regard to the dataset reuse in publications (note that other reuse metrics are positively correlated with dataset reuse and their top-10 list includes a similar set of datasets). Low values of the Availability and Latency metrics observed for several popular datasets, such as Freebase, YAGO and GeoNames, illustrate that these datasets either have no SPARQL endpoint, or it lacks performance. The Labels and Different Languages metrics show that most of the datasets presented in the table contain only a small proportion of their resources labelled in one or more languages. The rest of the metrics, on the other hand, yield good results. For most of the datasets there were no issues during parsing in Laundromat leading to high scores for Warnings and Error metrics, and they seem to contain either a small number of blank nodes and RDF primitives, or none at all. Of some datasets, such as Freebase, YAGO, LinkedMDB, and LinkedGeoData, the LOD Laundromat contains only a small subset. Therefore, we could not obtain reliable values for the quality metrics that require content analytic for these datasets. These values are marked with "X".

Overall, we observed that whereas some of the quality metrics under consideration, such as Availability, Latency, Labels, Different Languages and Licence provide interesting insights in the datasets, the values computed for some of the other quality metrics are always very similar (with the majority being near 1) for every dataset.[a] This includes blank nodes (avg. 0.936, st.dev. 0.146), query parameters (avg. 0.981, st.dev. 0.068) and RDF primitives (avg. 0.992, st.dev. 0.049).

[a] This sentence has been reworded to avoid confusion.

	DBpedia	Freebase	YAGO	GeoNames	NYTimes	LinkedMDB	LinkedGeoData	CiteSeer	OpenCyC	DBpedia-Live
Availability	0.991183601	0	0	0	0	0.988277032	0.38765754	0.88558153	0	0.972898166
B. Nodes	1	X	X	1	1	X	X	0.99995951	0.983601755	1
Latency	0.9865375	0	0	0	0	0.7912	0	0	0	0.9237125
Warnings	1	0	1	1	1	1	1	1	1	1
Errors	1	0	1	1	1	1	1	1	1	1
Labels	0.001388346	X	X	0.029925187	0	X	X	0.000047355	0.393847162	0.00462963
Q. Param.	0.99990084	X	X	1	1	X	X	0.999999314	1	0.99537037
RDF Prim.	1	X	X	1	1	X	X	1	0.98888558	1
Diff. Lang.	0.640625	X	X	0.015625	0	X	X	0.0078125	0.015625	0.015625

Table 6.6: Quality metrics for the top-10 highly reused datasets.

Metric	Publications	*public-lod*	*semantic-web*	PageRank
Availability	-0.008	0.030	0.003	0.043
B. Nodes	0.034	0.047	0.045	0.073
Latency	0.039	0.033	0.024	0.133
Warnings	-0.087	-0.084	-0.092	-0.017
Errors	-0.174	-0.160	-0.160	-0.047
Labels	-0.069	-0.075	-0.077	-0.052
Q. Param.	-0.231	-0.056	-0.078	0.023
RDF Prim.	0.022	0.029	0.034	0.026
Diff. Lang.	0.264	0.322	0.304	0.526

Table 6.7: Correlation between reuse metrics and quality metrics.

6.5.3 Results of Correlation between Reuse and Quality

Table 6.7 presents the correlation results between quality metrics and different forms of reuse using PCC for the full set of 430 datasets. As we can observe, the only metric showing a weak positive correlation with reuse in the mailing lists as well as a moderate positive relationship with reuse in datasets (PageRank) is the "Different Languages" quality metric. This metric represents the proportion of languages used in the dataset. Although for the top-10 datasets highly reused in publications, this metric did not indicate high values, except for DBpedia (as illustrated in Table 6.6), it can be seen as a good reuse indicator in general if the whole collection of 430 datasets is taken into account. Interestingly, the "Errors" and "Warnings" metrics, which are inversely proportional to the number of errors and warnings in a dataset, show a tendency towards weak negative correlation with reuse, meaning that datasets with warnings and errors tend to be more discussed and reused. As License is a categorical metric, we analyze the correlation between this metric and reuse using average values for the reuse metrics in the open and non-open license categories. In general, we observed the trend that datasets with open licenses tend to have higher absolute reuse values in all reuse channels compared to the non-open license datasets. The lack of correlation between reuse and some other quality metrics appears surprising at the first glance. The reason lies in the nearly-equal[b] values for some of the quality metrics, as described in Section 6.5.2. Given our dataset collection, the metrics such as blank nodes, query parameters and RDF primitives tend to have high scores in general and are not informative enough to predict reuse of the datasets.

[b]Changed from "nearly-uniform" to "nearly-equal", since the original word was not accurate and could lead to misunderstanding.

6.6 Discussion

Contributions. In this paper, we made several contributions: 1) We defined metrics to quantify reuse of datasets by a scientific community; 2) We analyzed dataset reuse behavior within the Semantic Web community over nine years applying these metrics to a large scale collection of 1131 linked datasets; 3) We analyzed the quality of the subset containing 430 datasets for which the dumps were available in the LOD Laundromat using a number of quality metrics discussed in the literature; and 4) We presented the results of our correlation analysis between the dataset reuse reflected in different channels and the selected quality metrics. In this section we discuss our method and results in more detail.

Reuse metrics. First of all, our proposed reuse metrics enable us to quantify reuse of datasets and compare reuse traces in different communication channels within a scientific community, in particular including publications, mailing lists, dataset references and citations of dataset papers. Although some datasets like DBpedia are generally popular in all channels under consideration, other datasets can be mainly referred in the specific channels at a given point in time. Our method enables us to track and quantify these aspects for a dataset collection, while also taking into account the temporal dimension (i.e. tracking changes in reuse behavior over time). These methods also enable insights in the behavior of certain communication channels with respect to dataset referencing. Our analysis results obtained by applying the proposed reuse metrics to a large scale dataset collection indicated that different channels of reuse, such as publications, mailing lists, dataset references and citations of dataset papers were well-correlated in these settings.

Quality metrics. In the next step, we selected several quality metrics defined in the literature, guiding our selection by the efficiency of the computation and coverage of various quality dimensions. We then computed these metrics for the large scale dataset collection containing 430 datasets for which we had both, reuse statistics and data dumps available in the LOD Laundromat. Our first observation was that the datasets for which we have observed the highest reuse according to our reuse metrics, varied a lot with respect to some of the quality metrics. For example, while four quality metrics presented very high scores (warnings, errors, blank nodes and query parameters), four other metrics indicated low scores: availability, latency, labels, and languages.

Correlation results. Finally, we built correlations between reuse and quality metrics for all 430 datasets in our collection. As it could be expected, we observed that open licence datasets tend to have higher reuse values in our collection. Interestingly, warnings and errors indicate a trend towards negative correlation with the mailing list usage, suggesting that mailing list discussions of datasets can also reflect lack of quality, sometimes. In this group of datasets, we also observed a positive correlation between the "different languages" quality metric and reuse in the mailing lists and datasets, indicating that multilingual labels can be an important factor for dataset reuse. We could not observe any correlation between reuse and several other quality metrics we evaluated on our dataset collection, such as blank nodes, query parameters and RDF primitives. The values of these metrics are generally high for the majority of the datasets in our dataset collection, so that these metrics are not informative enough to further distinguish between the datasets in these settings.

Limitations and future research. First of all, the analysis performed in this paper is focused on the reuse channels within the scientific community and does not take into account reuse channels that can be available in a different context, such as reuse of datasets in software applications, or reuse by data journalists for online newspaper articles. These channels can provide an interesting direction for future research. Second, the results of the evaluation performed in this paper are based on, and also limited by, our selection of quality metrics and the availability of data dumps in the LOD Laundromat. In future research, we would like to analyze a broader spectrum of quality metrics to better understand their information content in different settings and their suitability for reuse prediction. This work can also be a starting point to perform a more general investigation of the practical applicability of the existing quality metrics to a broader range of scenarios.

References

[3] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Proceedings of the 6th International Semantic Web Conference (ISWC) and the 2nd Asian Semantic Web Conference (ASWC), pp. 722–735. Springer, Busan, Korea (2007).

[8] Boland, K., Ritze, D., Eckert, K., Mathiak, B.: Identifying references to datasets in publications. In: TPDL, (2012).

[9] Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the ACM SIGMOD 2008, (2008).

[40] Ellefi, M.B., Bellahsene, Z., Breslin, J.G., Demidova, E., Dietze, S., Szymanski, J., Todorov, K.: Dataset profiling – a guide to features, methods, applications and vocabularies. (2016).

[41] Endris, K.M., Giménez-García, J.M., Thakkar, H., Demidova, E., Zimmermann, A., Lange, C., Simperl, E.: Dataset Reuse: An Analysis of References in Community Discussions, Publications and Data. In: Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017, 5:1–5:4 (2017).

[43] Farias Lóscio, B., Burle, C., Calegari, N.: Data on the web best practices, W3C (2016). URL: http://www.w3.org/TR/dwbp/

[71] Heath, T., Bizer, C.: *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool (2011).

[78] Hogan, A., Hitzler, P., Janowicz, K.: Linked dataset description papers at the semantic web journal: A critical assessment. Semantic Web 7 (2016).

[79] Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., Decker, S.: An empirical survey of Linked Data conformance. Journal of Web Semantics 14, 14–44 (2012).

[83] Jentzsch, A., Cyganiak, R., Bizer, C.: State of the LOD cloud, Mannheim (2011)

[92] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., vanKleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. SWJ 6(2) (2015).

[102] Mendes, P.N., Mühleisen, H., Bizer, C.: Sieve: Linked data quality assessment and fusion. In: EDBT/ICDT Workshops, (2012).

[111] Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. World Wide Web Internet And Web Information Systems 54(1999), 1–17 (1998).

[114] Piwowar, H.A., Vision, T.J.: Data reuse and the open data citation advantage. PeerJ (2013).

[121] Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 8796, pp. 245–260. Springer, Heidelberg (2014).

[123] Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web, pp. 697–706. ACM (2007).

[129] Wang, R.Y., Strong, D.M.: Beyond Accuracy: What Data Quality Means to Data Consumers. Journal of Management Information Systems 12(4), 5–33 (1996).

[135] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality Assessment for Linked Open Data: A Survey. Semantic Web Journal (by IOS Press) 1(1), 1–31 (2014).

Chapter 7

NELL2RDF: Reading the Web, Tracking the Provenance, and Publishing It as Linked Data*

In Part I we studied how contextual data about statements can be represented in first order logic, description logic, and OWL/RDF. We provided a formal definition of contextualization function—a function that maps a set of statements and a context to a new set of statements in the same logic system—and studied what properties such a function could have. Then, we proceeded to propose Nd*, a family of contextualization functions that can be used in the three logic systems. We compared several of these functions against other approaches existing in the Semantic Web community. We showed that our proposal preserved semantics better, and was able to differentiate different contexts, better than any other existing approach.

In this chapter, we make practical use of this knowledge. We extract a lot of contextual data about the provenance of the statements of NELL. NELL is a system that continuously reads the Web to extract knowledge in form of entities and relations between them. It has been running since January 2010 and extracted over 50,000,000 candidate statements. NELL's generated data comprises all the candidate statements together with detailed information about how it was generated. This information includes how each component of the system contributed to the extraction of the statement, as well as when that happened and how confident the system is in the veracity of the statement. This is a real-world dataset with a lot of contextual information (bigger, in fact, than the statements themselves). However, this data is only available in an *ad hoc* CSV format that makes it difficult to exploit out of the context of NELL. In order to make it more usable for other communities, we adopt Linked Data principles to publish a more standardized, self-describing dataset with rich provenance metadata. The results will be useful later in this dissertation to evaluate our proposals on managing contextual data.

*This chapter is based on the following publication:

- Giménez-García, J.M., Duarte, M.C., Zimmermann, A., Gravier, C., Jr., E.R.H., Maret, P.: NELL2RDF: Reading the web, and publishing it as linked data, Université Jean Monnet (2018). URL: http://arxiv.org/abs/1804.05639 [53]

Part of this content has been published in:

- Giménez-García, J.M., Duarte, M., Zimmermann, A., Gravier, C., Hruschka Jr., E.R., Maret, P.: NELL2RDF: Reading the web, tracking the provenance, and publishing it as linked data. In: Joint Proceedings of the International Workshops on Contextualized Knowledge Graphs, and Semantic Statistics co-located with 17th International Semantic Web Conference (ISWC 2018) Monterey, USA, October 8th, 2018ceurwp, CEUR, Monterey, USA (2018). [52]

7.1 Introduction

Never-Ending Language Learning (NELL) [20, 103] is an autonomous computational system that aims at continually and incrementally learning. NELL has been running for more than 10 years in Carnegie Mellon University (US). Currently, NELL has collected over 50 million of candidate beliefs, from which about 3.6 million have been promoted as trustworthy statements. NELL learns from the web and uses an ontology previously created to guide the learning. One of the most significant resource contributions of NELL, in addition to the millions of beliefs learned from the Web, is NELL's internal representation (or metadata) for categories, relations and concepts. Such internal representation grows in every iteration, and is used by NELL as a set of different (and constantly updated) *feature vectors* to continuously retrain NELL's learning components and build its own way to understand what is read from the Web. Zimmermann, Gravier, Subercaze, and Cruzille [140] published in 2013 a solution to convert NELL's beliefs and ontology into RDF and OWL. However, NELL's internal metadata is not modeled in their work. Thus, the main contribution of this work is to extend the approach to include all the provenance metadata (NELL's internal representation) for each belief. We publish this data using five different representation models: RDF reification [15, Sec. 5.3], N-Ary relations [110], Named Graphs [23], Singleton Properties [108], and NdFluents [57]. In addition, we publish not only the promoted beliefs, but also the candidates. As far as we know, this dataset contains more metadata about the statements than any other available dataset in the linked data cloud. This in itself can also be interesting for researchers that seek to manage and exploit meta-knowledge.

Our intention is to keep this information updated and integrate it on NELL's web page[1].[a]

The rest of the paper is organized as follows: Section 7.2 presents NELL and the components it comprises; in Section 7.3 describes the transformation of NELL data and metadata to RDF; Section 7.4 presents the dataset generated in this paper and how it is published; finally, Section 7.5 provides final remarks and future work.

7.2 The Never-Ending Language Learning System

NELL [20, 103] was built based on a new Machine Learning (ML) paradigm, the Never-Ending Learning (NEL). NEL paradigm is a semi-supervised learning [7] approach focused on giving the ability to a machine learning system to autonomously use what it has previously learned to continuously become a better learner. NELL is based on a number of coupled components working in parallel. These components read the web and use different approaches to not only infer new knowledge in the form of beliefs, but also to infer new ways of internally representing the learned beliefs and their properties. Beliefs are divided into candidates and promoted beliefs. In order to be promoted a belief needs to have a confidence score of at least 0.9.

1. **AliasMatcher** finds relations between entities and their Wikipedia URL on Freebase. It was run only once and is currently not active.

2. **CML** *(Coupled Morphologic Learner)* [21] is responsible for identifying morphological regularities (such as that words finished in `burg` could be cities). It

[1] `http://rtw.ml.cmu.edu/`

[a] Regrettably, it was not possible to achieve this goal.

makes use of orthographic features of noun phrases (*e.g.*, length and number of words, capitalization, prefixes and suffixes). *CMC* is the previous version of this component.

3. **CPL** *(Coupled Pattern Learner)* [21] is the component that learns Named Entities (NE) and Textual Patterns (TP) from text in the web pages. Internally, a different implementation was used between 2010 and 2013 that could learn categories and relations together. After that, CPL was splitted in CPL1 and CPL2, the former learning categories and the latter relations, but the distinction is not made in the knowledge base. All the knowledge from CPL1 is promoted only if CPL2 agrees. *i.e.*, CPL will extract TPs for categories (`_ is a city`, `city such as _`, *etc.*) and for relations (`arg1 is a city located in arg2`, `arg1 is the capital of arg2`, *etc.*). Then, using those TPs, CPL will extract NEs for categories (e.g. `city(Paris)`, `city(Annecy)`, *etc.*) and NE pairs for relations (`locatedIn(Paris, France)`, `locatedIn(Annecy, France)`, *etc.*).

4. **KbManipulation** is used to correct some old bugs from NELL's internal indexing knowledge. Several of these bugs should be removed automatically, but NELL has not one automated process for this task yet.

5. **LatLong** matches the literal string of Named Entities against a fixed geolocation database.

6. **LE (Learned Embeddings)** [132] predicts new categories or relations of entities based on Event and Named Entity extraction. It creates a feature space where each dimension is a single NELL predicate, and NELL's learned NE (or NE pairs for relations) is used as training examples. LE's process predicts category or relation for NE (or NE pairs) that were not related in the training set.

7. **MBL**, also known as *ErrorBasedIntegrator* and *Knowledge Integrator*, is the component responsible for taking the decision of promotion based on the contributions of the other components. *EntityResolverCleanup* is the name used for the same MBL process applied during a big alteration in NELL's knowledge base. In 2010 a big change was made in the NELL's KB structure to make possible for two words to have different meanings (e.g apple the fruit and Apple the company) and, conversely, for a concept to use different words (e.g Google and Google Inc.).

8. **OE** *(Open Eval)* [119] queries the web and extract small text using predicate instances. OE calculates the score based on the text distance between the instances in a relation.

9. **OntologyModifier** is used for any ontology alteration. This component appears in the Knowledge base when a new seed or an ontology extension is manually introduced.

10. **PRA** *(Path Ranking Algorithm)* [49] is based on Random Walk Inference. PRA analyzes the connections between two categories instances which are the arguments for a relation. This component replaced the old *Rule Learner* component.

11. **RL** *(Rule Learner)* [88] extracts new knowledge using Horn Clauses based on the ontology. Its implementation was based on FOIL [116]. It can be found in NELL's KB, but its execution stopped when NELL started to deal with polysemy resolution.

12. **SEAL** *(Coupled Set Expander for Any Language)* [128] is the component responsible for extracting knowledge from HTML patterns. It works in a similar way to CPL, but using HTML patterns instead of textual patterns. In the past it was called *CSEAL*, but after some improvements in its performance it changed the name for SEAL.

13. **Semparse** [86] combines syntactic parsing from CCGbank (a conversion of the corpus of trees Penn Treebank [94]) and distant supervision.

14. **SpreadsheetEdits** provides modifications in the NELL's Knowledge base using human feedback.

Each of of these components, with the exception of `LE`, output provenance information regarding theirs execution. In the next sections we present how this metadata is modeled in RDF.

7.3 Converting NELL to RDF

In this section we describe how NELL data and metadata are transformed into RDF. The first subsection presents how NELL's ontology and beliefs are converted, following the work by Zimmermann, Gravier, Subercaze, and Cruzille [140]; the second subsection describes how we convert the provenance metadata associated with each belief. NELL's Knowledge bases used in this paper for the promoted and candidates beliefs are respectively corresponding to the iterations 1075[2] and 1070[3]. The code is publicly available in GitHub[4].

7.3.1 Converting NELL's beliefs to RDF

NELL's ontology is published as a file with three tab-separated values per line, where each line expresses a relationship between categories and other categories, relations, or values used by NELL processes. In order to convert NELL's ontology to RDF each line is transformed into a triple as per Zimmermann, Gravier, Subercaze, and Cruzille [140]. In short, the first and the third values are a pair of categories or relations, or either a category or relation in the first field and a value in the third. The second field is a predicate that indicates the relationship between the two elements. The transformations can be seen in Table 7.1.

NELL's beliefs are also published in tab-separated format, where each line contains a number of fields to express the belief and the associated metadata, such as iteration of promotion, confidence score, or the activity of the components that inferred the belief. All the fields except 4, 5, 6, and 13 are used to convert the beliefs into RDF statements. Table 7.2 shows the meaning of each field. Fields 1, 2, and 3 are converted into the subject, predicate, and object of an RDF statement; the content of fields 7 and 8 create new statements using `rdf:label` properties; fields 9 and 10 create new triples with the property `skos:prefLabel`; finally, fields 11 and 12 are used to create triples indicating the types of the subject and the object. For a more detailed description of this step, refer to Zimmermann, Gravier, Subercaze, and Cruzille [140].

[2] http://rtw.ml.cmu.edu/resources/results/08m/NELL.08m.1075.esv.csv.gz
[3] http://rtw.ml.cmu.edu/resources/results/08m/NELL.08m.1070.cesv.csv.gz
[4] https://github.com/WDAqua/nell2rdf

Table 7.1: NELL's ontology predicates and their translation in RDFS / OWL (from [140])

NELL predicate	Translation to RDFS / OWL
antireflexive	rdf:type owl:IrreflexiveProperty
antisymmetric	antisymmetric Literal(?object,xsd:boolean)
description	rdfs:comment Literal(?object,@en)
domain	rdfs:domain Class(?object)
domainwithinrange	domainWithinRange Literal(?object,xsd:boolean)
generalizations	rdfs:subClassOf Class(?object)
humanformat	humanFormat Literal(?object,xsd:string)
instancetype	instanceType IRI(?object)
inverse	owl:inverseOf ?object
memberofsets	*if* ?object *is* rtwcategory *then* rdf:type rdfs:Class *else* ?object *is* rtwrelation *then* rdf:type rdf:Property
mutexpredicates	*if* ?subject *is a* class *then* owl:disjointWith ?object *else* ?subject *is a* property *then* owl:propertyDisjointWith ?object
nrofvalues	*if* ?object *is* 1 *then* rdf:type owl:FunctionalProperty
populate	populate Literal(?object,xsd:boolean)
range	rdfs:range ?object
rangewithindomain	rangeWithinDomain Literal(?object,xsd:boolean)
visible	visible Literal(?object,xsd:boolean)

7.3.2 Converting NELL metadata to RDF

Fields 4, 5, 6, and 13 of each NELL's belief are used to extract the metadata. Each belief is represented by a resource, to which we attach the provenance information. In the promoted beliefs process, field 4 is used to extract the iteration when the belief was promoted, while field 5 gives a confidence score about it. On the other hand, in the candidate beliefs process, fields 4 and 5 contains the iterations when each component generated information about the belief, and the confidence score provided by each of them. Field 6 contains a summary information about the activity of MBL when processing the promoted belief. The complete information from field 6 is a summary of field 13. For that reason, we only process field 13. Finally, in field 13 every activity that took part in generating the statement is parsed.

Table 7.2: Description of NELL's beliefs fields

#	Field	Description
1	Entity	Subject of the belief
2	Relation	Predicate of the belief
3	Value	Object of the belief
4	Iteration	Iteration when the belief was promoted, or a list of iterations when the components generated the belief
5	Probability	Confidence score of the belief
6	Source	MBL activity to promote the belief
7	Entity literalStrings	Labels of the subject
8	Value literalStrings	Labels of the object
9	Best Entity literalString	Preferred label of the subject
10	Best Value literalString	Preferred label of the object
11	Categories for Entity	Classes of the subject
12	Categories for Value	Classes of the object
13	Candidate Source	Activity of the components that generated the belief

Table 7.3: Description of NELL metadata classes

Class	rdfs:subClassOf	Description
`Belief`	`prov:Entity`	A belief
`PromotedBelief`	`Belief`	A promoted belief
`CandidateBelief`	`Belief`	A candidate belief
`ComponentExecution`	`prov:Activity`	The activity of a component in an iteration
`Component`	`prov:SoftwareAgent`	A component
`Token`	`owl:Thing`	The tuple that was inferred by the activity
`RelationToken`	`Token`	The tuple <Entity,Entity> that was inferred for a relation
`GeneralizationToken`	`Token`	The tuple <Entity,Category> that was inferred for a generalization
`GeoToken`	`Token`	The tuple <Entity,Longitude,Latitude> that was inferred for a geografical belief

The ontology can be seen in Figure 7.1. We make use of the PROV-O ontology [90] to describe the provenance. Each `Belief` can be related with one or more `ComponentExecution` that, in turn, are performed by a `Component`. If the belief is a `PromotedBelief`, it has attached its `iterationOfPromotion` and `probabilityOfBelief`. The `ComponentExecution` is related to information about the process: the `iteration`, `probabilityOfBelief`, `Token`, `source` and `atTime` (the date and time it was processed). The `Token` expresses the concepts that the `Component` is relating. Those concepts can be a pair of entities for a `RelationToken`, and entity and a class for a `GeneralizationToken` (note that `LatLong` component has a different token `GeoToken`, further described later). Finaly, each component has a `source` string describing its process for the belief. This string is then further analyzed and translated into a different set if IRIs for each type of component in the subsections below.

The classes of the ontology are described in Table 7.3 and properties of the ontology are described in Table 7.4. The classes and properties of each component are described down below.

AliasMatcher execution is denoted by a resource of class `AliasMatcherExecution`, and includes the date when the data was extracted from Freebase using the property `freebaseDate`. The added ontology can be seen in Figure 7.2.

CMC execution is denoted by a resource of class `CMCExecution`. A number of morphological patterns `MorphologicalPatternScoreTriple` are attached to it, each one containing a name, a value, and a confidence score. The properties used can be seen in Table 7.5, while the ontology diagram is shown in Figure 7.3.

CPL execution is denoted by a resource of class `CPLExecution`. It contains a series of textual patterns `patternOccurrences`, each one with a literal that describes the pattern, and the number of times it has occurred in the NELL's data source. The properties used are described in Table 7.6, and the diagram for the ontology is shown in Figure 7.4.

KbManipulation execution is denoted by a resource of class `KbManipulationExecution`. It contains the bug `oldBug` that was manually fixed. Its shown in Figure 7.5.

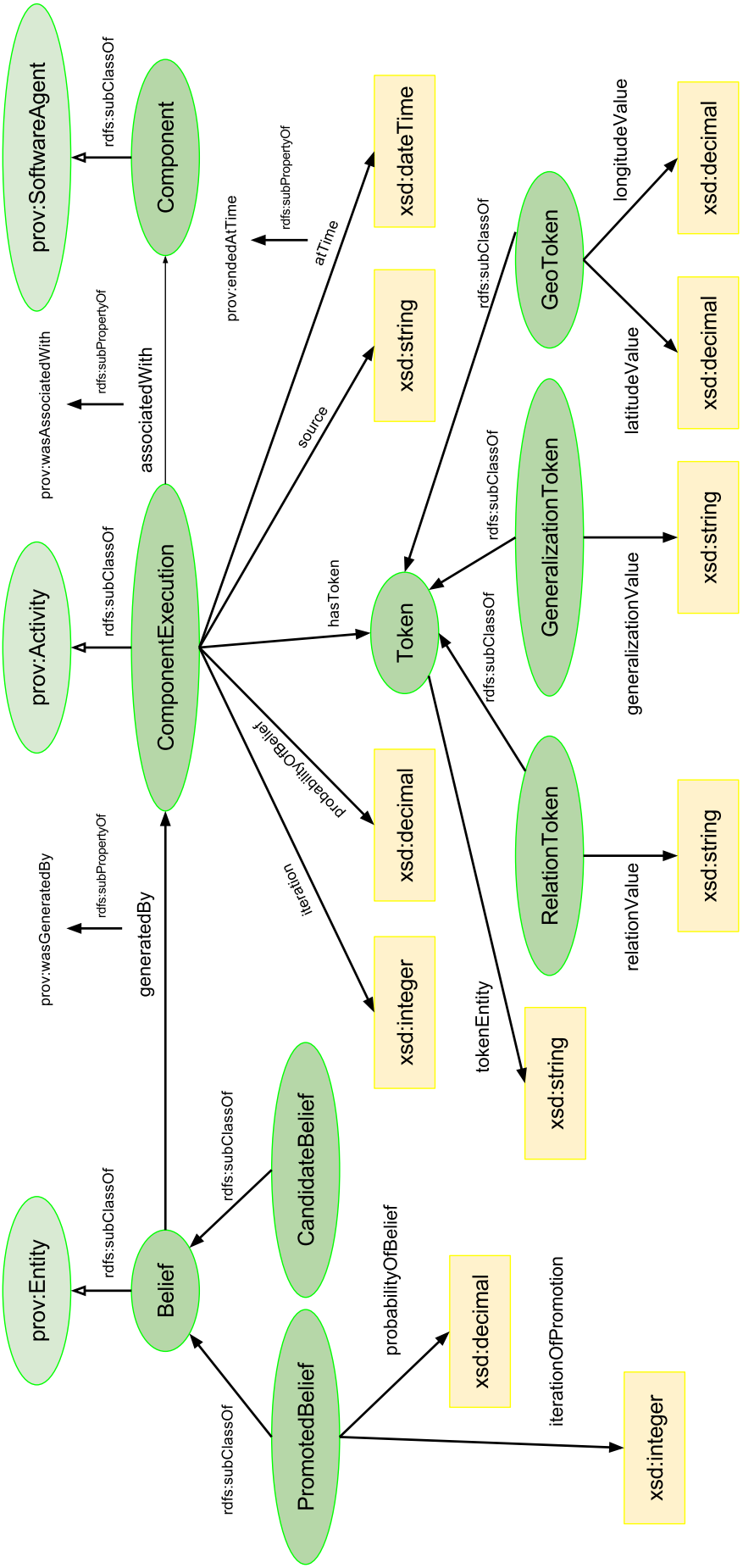


Figure 7.1: NELL2RDF metadata ontology

Table 7.4: Description of NELL metadata properties

Property	rdfs:subPropertyOf	rdfs:domain	rdfs:range
	Description		
generatedBy	prov:wasGeneratedBy	Belief	ComponentExecution
	The Belief was generated by the execution of the component		
associatedWith	prov:wasAssociatedWith	ComponentExecution	Component
	The execution was performed by the component		
iterationOfPromotion	owl:DatatypeProperty	PromotedBelief	xsd:integer
	Iteration in which the component was promoted		
probabilityOfBelief	owl:DatatypeProperty	PromotedBelief	xsd:decimal
	Confidence score of the Belief		
iteration	owl:DatatypeProperty	ComponentExecution	xsd:integer
	Iteration in which a component performed the activity		
probability	owl:DatatypeProperty	ComponentExecution	xsd:decimal
	Confidence score given by the component		
hasToken	owl:ObjectProperty	ComponentExecution	Token
	The concepts that the component is relating		
source	owl:DatatypeProperty	ComponentExecution	xsd:string
	Data that was used by the component in the activity		
atTime	owl:DatatypeProperty	ComponentExecution	xsd:dateTime
	Date and time when the component execution was performed		
tokenEntity	owl:DatatypeProperty	Token	xsd:string
	Entity on which the data was inferred		
relationValue	owl:DatatypeProperty	RelationToken	xsd:string
	Entity related the entity appointed by tokenEntity		
generalizationValue	owl:DatatypeProperty	GeneralizationToken	xsd:string
	Class of the entity appointed by tokenEntity		

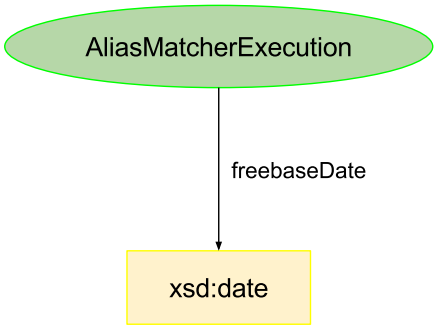


Figure 7.2: AliasMatcherExecution metadata ontology

Table 7.5: Description of CMC metadata properties

Property	rdfs:domain	rdfs:range
	Description	
morphologicalPattern	CMCExecution	MorphologicalPatternScoreTriple
	One of the morphological patterns used by CMC	
morphologicalPatternName	MorphologicalPatternScoreTriple	xsd:string
	Name of the morphological pattern (*i.e.*, prefix, suffix, etc.)	
morphologicalPatternValue	MorphologicalPatternScoreTriple	xsd:string
	Value of the morphological pattern (*i.e.*, prefix = Saint and suffix = burgh)	
morphologicalPatternScore	MorphologicalPatternScoreTriple	xsd:decimal
	Score of the morphological pattern	

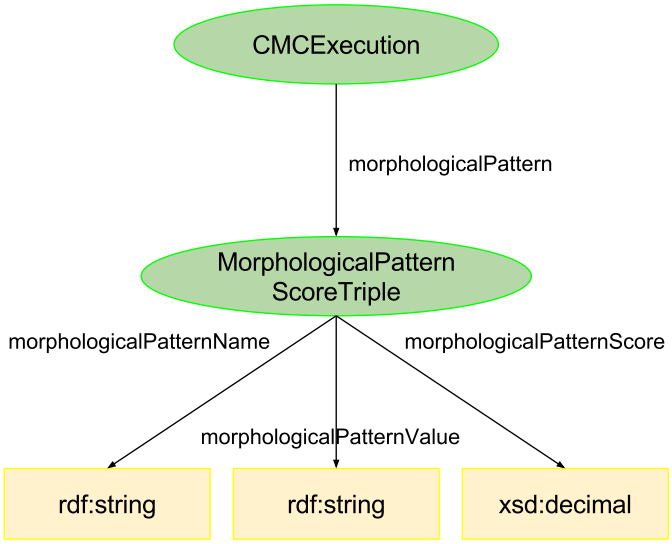


Figure 7.3: CMC metadata ontology

Table 7.6: Description of CPL metadata properties

Property	**rdfs:domain**	**rdfs:range**
	Description	
`patternOccurrences`	`CPLExecution`	`PatternNbOfOccurrencesPair`
	One of the textual patterns used by `CPL`	
`textualPattern`	`PatternNbOfOccurrencesPair`	`xsd:string`
	Textual pattern in the form of a sentence	
`nbOfOccurrences`	`PatternNbOfOccurrencesPair`	`xsd:nonNegativeInteger`
	Number of times it has occurred in the NELL's source data	

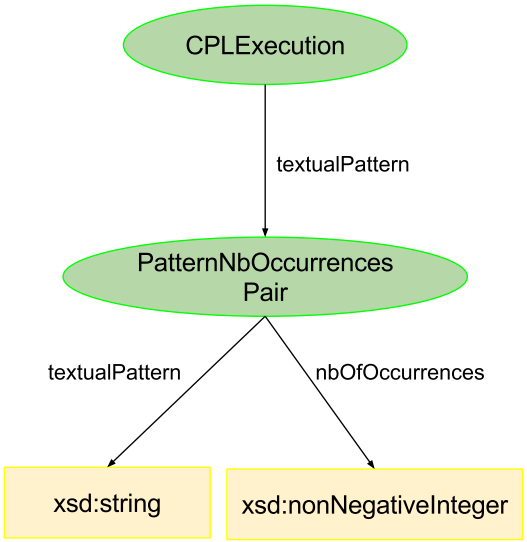


Figure 7.4: CPL metadata ontology

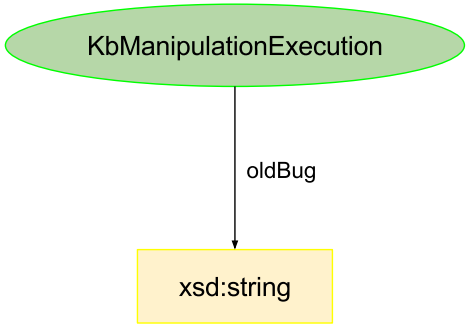


Figure 7.5: KbManipulation metadata ontology

Table 7.7: Description of LatLong metadata properties

Property	rdfs:domain	rdfs:range
	Description	
`location`	`LatLongExecution`	`NameLatLongTriple`
	One of the locations used by `Latlong`	
`name`	`NameLatLongTriple`	`rdf:langString`
	Name of the location	
`latitudeValue`	`NameLatLongTriple`	`xsd:decimal`
	Latitude of the location	
`longitudeValue`	`NameLatLongTriple`	`xsd:decimal`
	Longitude of the location	

LatLong execution is denoted by a resource of class `LatLongExecution`. It contains a list of locations `NameLatLongTriple` that were used to infer the belief. Each one containing the `name` and the latitude and longitude values. This execution has also its own token `GeoToken` with the latitude and longitude values reusing the same properties. The properties are detailed in Table 7.7, and the ontology diagram is shown in Figure 7.6.

LE execution is denoted by a resource of class `LEExecution`. It does not contain any additional triples.

MBL execution is denoted by a resource of class `MBLExecution`. It contains the entities and the categories of the other belief that was used to promote this one. The properties used are described in Table 7.8, and the ontology diagram is shown in Figure 7.7.

OE execution is denoted by a resource of class `OEExecution`. It contains a set of pairs `TextUrlPair`, each one including the sentence that was used to infer the belief, and the URL from where it was extracted. The properties used can be found in Table 7.9, and the ontology diagram in Figure 7.8.

OntologyModifier execution is denoted by a resource of class `OntologyModifierExecution`. It contains the `ontologyModification`, which can be either a modification of a category or a modification of a relation. The ontology diagram can be seen in Figure 7.9.

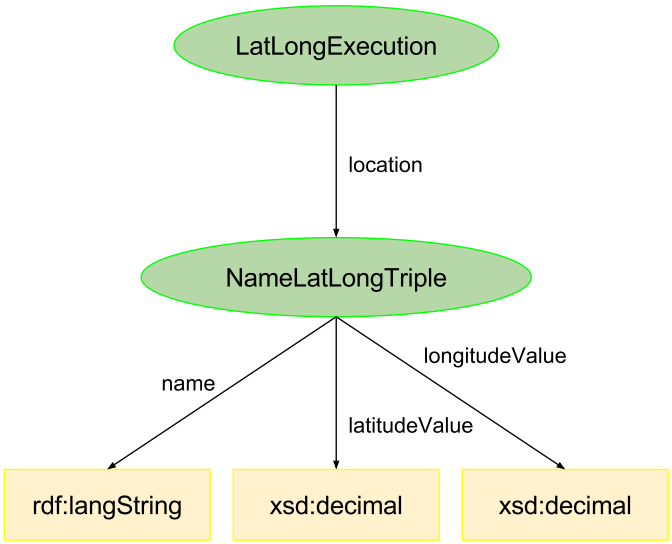


Figure 7.6: LatLong metadata ontology

Table 7.8: Description of MBL metadata properties

Property	**rdfs:domain** **rdfs:range** **Description**
`promotedEntity`	`MBLExecution` `xsd:string` Entity of a belief previously promoted
`promotedEntityCategory`	`MBLExecution` `xsd:string` Category of the entity of the promoted belief
`promotedRelation`	`MBLExecution` `xsd:string` Relation of the promoted belief
`promotedValue`	`MBLExecution` `xsd:string` Value of the promoted belief
`promotedValueCategory`	`MBLExecution` `xsd:string` Category of the promoted belief, if applicable

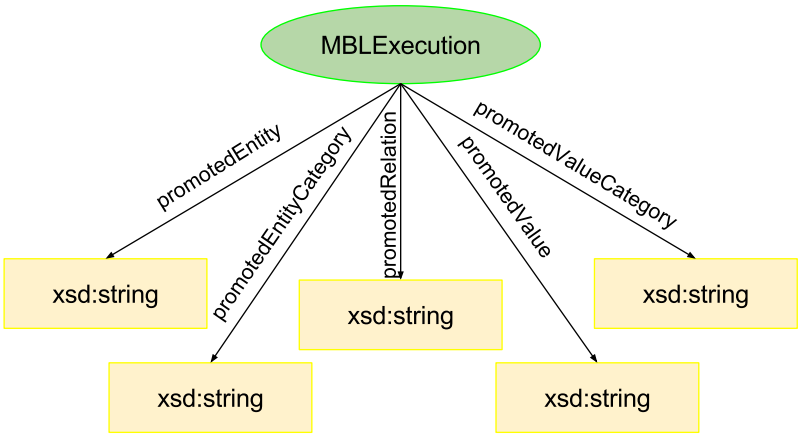


Figure 7.7: MBL metadata ontology

Table 7.9: Description of OE metadata properties

Property	rdfs:domain rdfs:range Description
`textUrl`	`OEExecution` `TextUrlPair` One of the pairs <text, url> used by `OE`
`text`	`TextUrlPair` `rdf:langString` Text extracted from the web
`url`	`xsd:anyURI` Web page where the text was extracted

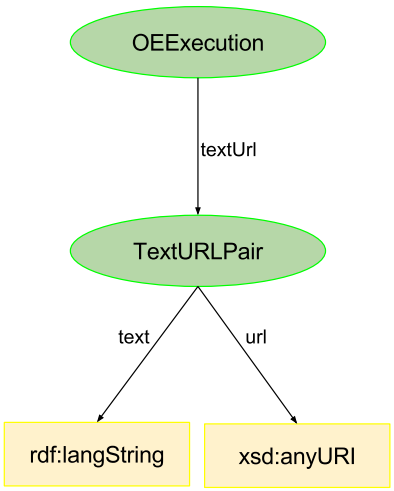


Figure 7.8: OE metadata ontology

PRA execution is denoted by a resource of class `PRAExecution`. It includes a series of `Path` resources describing the path followed in NELL dataset to infer the belief. Each `Path` includes its direction and a confidence score, along with a list of relations followed. The properties used can be seen in Table 7.10, while the ontology diagram is shown in Figure 7.10.

RL execution is denoted by a resource of class `RLExecution`. It contains a resource `RuleScoresTuple` that contains the `Rule` and a set of scores indicating the confidence, and the number of beliefs that are estimated to be correctly and incorrectly inferred (and the number of inferred beliefs for which it is not known if they are correct or

Table 7.10: Description of PRA metadata properties

Property	rdfs:domain rdfs:range Description
`relationPath`	`PRAExecution` `Path` Relation path that entails the belief
`direction`	`Path` `DirectionOfPath` Direction of the path
`score`	`Path` `xsd:decimal` Score assigned to the entailment
`listOfRelations`	`Path` `rdf:List` Ordered list of relations in the path

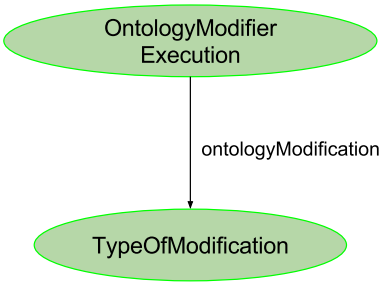


Figure 7.9: OntologyModifier metadata ontology

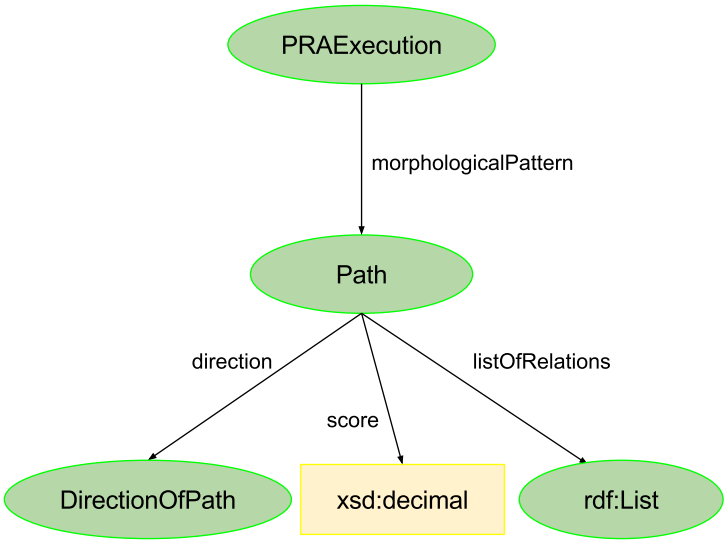


Figure 7.10: PRA metadata ontology

not) with that rule. The rule itself contains the variables and their values, and the predicates that are part of it. Each `Predicate` includes the name of the predicate and the two variables it uses. The complete list of properties can be found in table 7.11. The ontology diagram is presented in Figure 7.11.

SEAL execution is denoted by a resource of class `SEALExecution`. It includes the URL it used with the property `url`. The ontology diagram can be seen in Figure 7.12.

Semparse execution is denoted by a resource of class `SemparseExecution`. It includes a literal with the sentence used during it, using the property `sentence`. The ontology diagram can be seen in Figure 7.13.

SpreadsheetEdits execution is denoted by a resource of class `SpreadsheetEditsExecution`. It contains a set of literals describing the user who made the modification, the file used as input, the action made, and the modified entity, relation, and value. The list of properties can be seen in Table 7.12, while the ontology diagram is shown in Figure 7.14.

7.4 The NELL2RDF Dataset

The current version of NELL2RDF updates the promoted beliefs to the last version, adding the provenance triples about them. It also adds the candidate beliefs and

Table 7.11: Description of RL metadata properties

Property	rdfs:domain	rdfs:range
	Description	
`ruleScores`	`RLExecution`	`RuleScoresTuple`
	The rule and set of scores used by `RL`	
`rule`	`RuleScoresTuple`	`Rule`
	The rule `RL` used to infer the belief, in the form of horn clauses	
`accuracy`	`RuleScoresTuple`	`xsd:decimal`
	Estimated accuracy of the rule in NELL	
`nbCorrect`	`RuleScoresTuple`	`xsd:nonNegativeInteger`
	Estimated number of correct beliefs created by the rule	
`nbIncorrect`	`RuleScoresTuple`	`xsd:nonNegativeInteger`
	Estimated number of incorrect beliefs created by the rule	
`nbUnknown`	`RuleScoresTuple`	`xsd:nonNegativeInteger`
	Number of rules created by the rules with no known correctness	
`variable`	`Rule`	`xsd:string`
	One of the variables that appear in the rule	
`valueOfVariable`	`Rule`	`xsd:string`
	Value of the variable inferred by the rule	
`predicate`	`Rule`	`Predicate`
	One of the predicates that appear in the rule	
`predicateName`	`Predicate`	`xsd:string`
	Name of the predicate	
`firstVariable`	`Predicate`	`xsd:string`
	First variable of the predicate	
`secondVariable`	`Predicate`	`xsd:string`
	Second variable of the predicate	

Table 7.12: Description of SpreadsheetEdits metadata properties

Property	rdfs:domain	rdfs:range
	Description	
`user`	`SpreadsheetEditsExecution`	`xsd:string`
	User that made the modification	
`entity`	`SpreadsheetEditsExecution`	`xsd:string`
	Entity of the belief affected by the modification	
`relation`	`SpreadsheetEditsExecution`	`xsd:string`
	Relation of the belief affected by the modification	
`value`	`SpreadsheetEditsExecution`	`xsd:string`
	Value of the belief affected by the modification	
`action`	`SpreadsheetEditsExecution`	`xsd:string`
	Action made in the modification	
`file`	`SpreadsheetEditsExecution`	`xsd:string`
	File where the modification was saved and then read by `SpreadsheetEdits`	

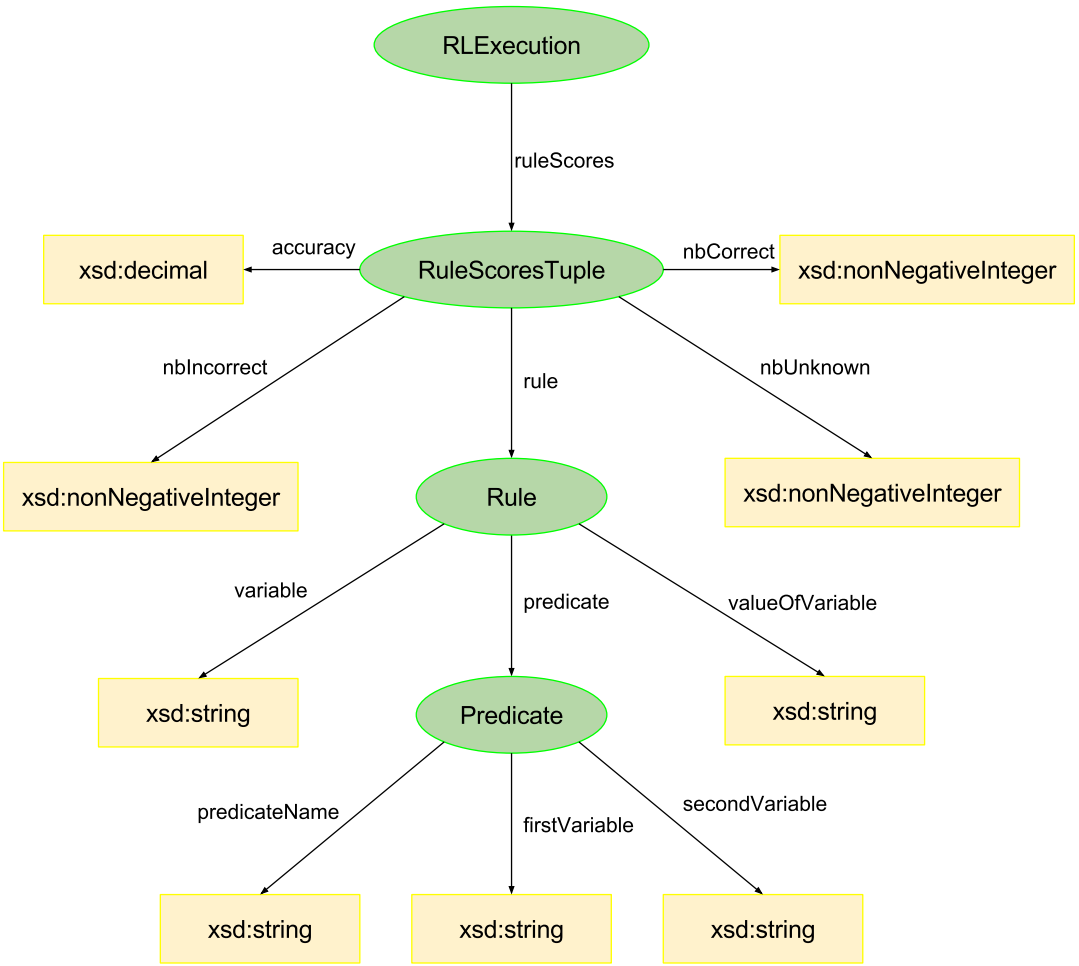


Figure 7.11: RL metadata ontology

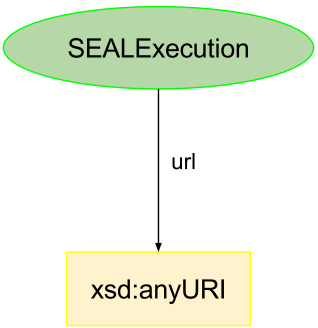


Figure 7.12: SEAL metadata ontology

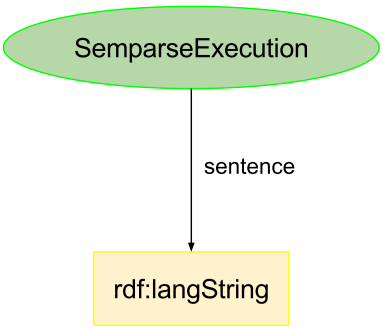


Figure 7.13: Semparse metadata ontology

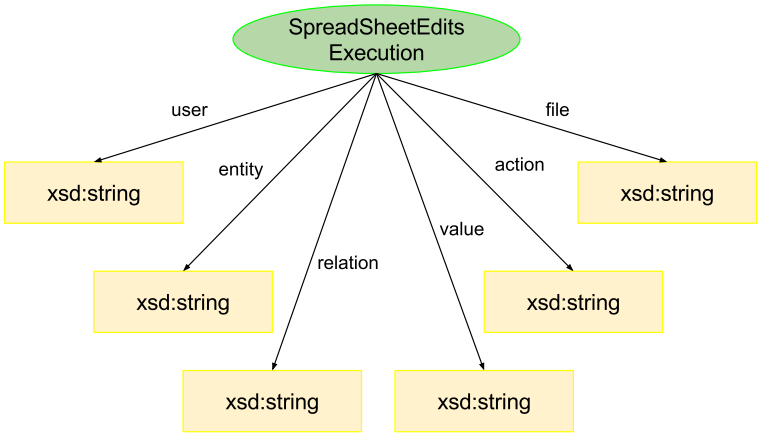


Figure 7.14: SpreadsheetEdits metadata ontology

their corresponding provenance triples. We provide the dumps for the promoted beliefs[5] and the candidate beliefs[6]. The ontologies for the beliefs[7] and the provenance metadata[8] is common for both dumps. Metadata about the dataset[9] is modeled using VoID and DCAT vocabularies.

In order to attach the metadata to each belief, we need to reify the statement into a resource. We follow five different models, described down below. A graphical representation of the models is shown in Figure 7.15. A summary of the triples and resources of each model can be seen in Table 7.13.

- *RDF Reification* [15, Sec. 5.3] represents the statement using a resource, and then creates triples to indicate the subject, predicate and object of the statement.
- *N-Ary relations* [110]: This model creates a new resource that identifies the relation and connects subject and object using different design patterns. Wikidata[10] makes use of this model of annotation.
- *Named Graphs* [23]: A forth element is added to each triple, that can be used to identify a triple or set of triples later on. This model is used by Nanopublications [104].
- *The Singleton Property* [108] creates a unique property for each triple, related to the original one. It defines its own semantics that extend RDF, RDFS.
- *NdFluents* [57] creates a unique version of the subject and the object (in the case it is not a literal) of the triple, and attaches them to the original resources and the context of the statement.

7.5 Discussion and Future Work

In this work we present the conversion of both data and metadata from NELL into RDF. It presents a thesaurus of entities and binary relations between them, as well

[5] https://w3id.org/nellrdf/nellrdf.promoted.n3.gz
[6] https://w3id.org/nellrdf/nellrdf.candidates.n3.gz
[7] https://w3id.org/nellrdf/ontology/nellrdf.ontology.n3
[8] https://w3id.org/nellrdf/provenance/ontology/nellrdf.ontology.n3
[9] https://w3id.org/nellrdf/metadata/nellrdf.metadata.n3
[10] https://www.wikidata.org

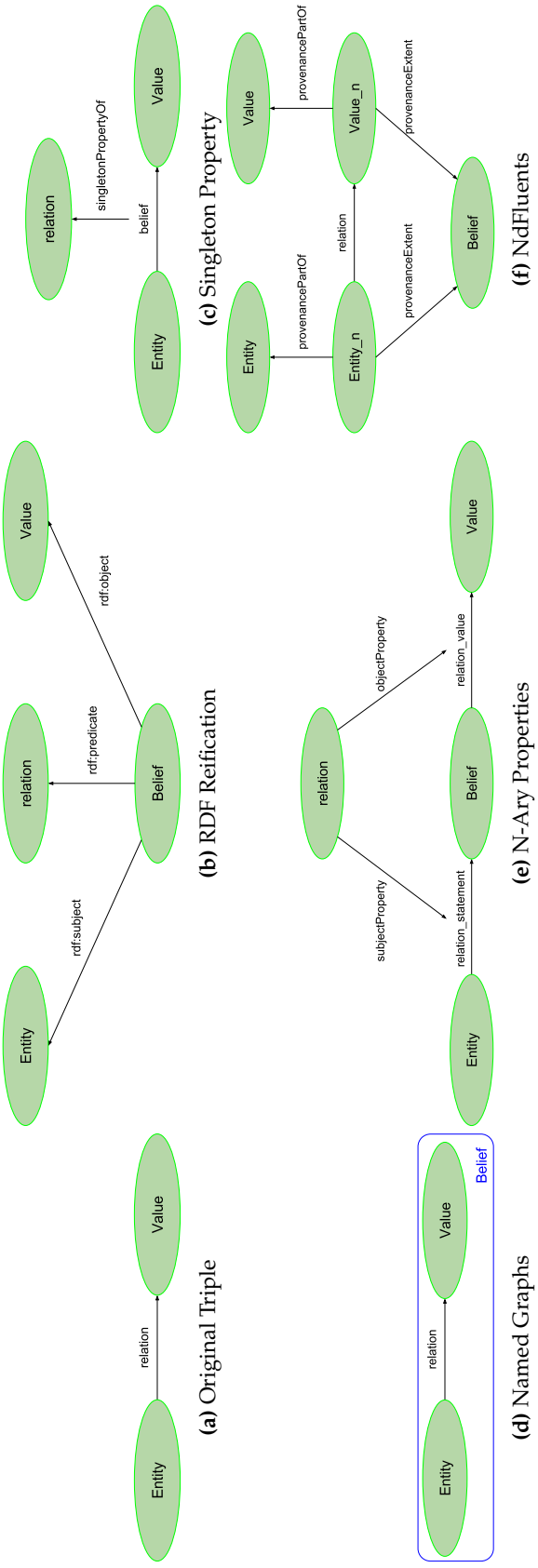


Figure 7.15: Reification models

Table 7.13: Summary of dataset stats for each model

Model	Promoted		Candidates		Total	
	Size	Triples	Size	Triples	Size	Triples
W/O metadata	1.13GB	0.01B	22.3GB	0.14B	22.3GB	0.14B
RDF Reification	47.5GB	0.22B	447GB	2.17B	457GB	2.22B
N-Ary Relations	46.4GB	0.21B	400GB	1.92B	410GB	1.97B
Named Graphs	46.3GB	0.21B	398GB	1.92B	409GB	1.97B
Singleton Property	46.4GB	0.21B	399GB	1.92B	409GB	1.97B
NdFluents	47.9GB	0.23B	466GB	2.43B	476GB	2.48B

as a number of lexicalizations for each entity. It also includes detailed provenance metadata along with confidence scores, encoded using five different reification approaches.

Our goals for this dataset are twofold: First, we want to improve WDAqua-core0 [31] query answering system, providing it with more relations and lexicalizations, along with confidence scores that can help to give hints about how trustworthy is the answer. Second, given that it contains a big proportion of metadata statements, we want to use it as a testbed to compare how the different metadata representations behave in current triplestores.

While currently we only publish the dumps of the datasets, we plan to provide SPARQL endpoint and full dereferenceable URLs. In addition, NELL is starting to be explored in languages different than English, such as Portuguese [82, 38] and French [39]. Our intention is to convert those datasets to RDF as they become available to the public, since the system and knowledge base are exactly the same used in the English one.

References

[7] Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100. ACM (1998).

[15] Brickley, D., Guha, R.: RDF Schema 1.1, pp. 91–122. W3C (2014). DOI: 10.1016/B978-0-12-373556-0.00006-X URL: http://www.w3.org/TR/rdf-schema/

[20] Carlson, A., Betteridge, J., Hruschka, E.R., Mitchell, T.M.: Coupling Semi-Supervised Learning of Categories and Relations. In: Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing, (2009).

[21] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an Architecture for Never-Ending Language Learning. In: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI), (2010).

[23] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. Journal of Web Semantics 3(4), 247–267 (2005).

[31] Diefenbach, D., Singh, K.D., Maret, P.: WDAqua-Core0: A Question Answering Component for the Research Community. In: Semantic Web Challenges - 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28 - June 1, 2017, Revised Selected Papers, pp. 84–89 (2017).

[38] Duarte, M.C., Hruschka, E.R.: How to Read The Web In Portuguese Using the Never-Ending Language Learner's Principles. In: Proceedings of the 14th International Conference on Intelligent Systems Design and Applications, (2014).

[39] Duarte, M.C., Maret, P.: Vers une instance française de NELL : chaîne TLN multilingue et modélisation d'ontologie. Revue des Nouvelles Technologies de l'Information Extraction et Gestion des Connaissances, RNTI-E-33, 469–472 (2017).

[49] Gardner, M., Talukdar, P.P., Krishnamurthy, J., Mitchell, T.M.: Incorporating Vector Space Similarity in Random Walk Inference over Knowledge Bases. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014).

[52] Giménez-García, J.M., Duarte, M., Zimmermann, A., Gravier, C., Hruschka Jr., E.R., Maret, P.: NELL2RDF: Reading the web, tracking the provenance, and publishing it as linked data. In: Joint Proceedings of the International Workshops on Contextualized Knowledge Graphs, and Semantic Statistics co-located with 17th International Semantic Web Conference (ISWC 2018) Monterey, USA, October 8th, 2018ceurwp, CEUR, Monterey, USA (2018).

[53] Giménez-García, J.M., Duarte, M.C., Zimmermann, A., Gravier, C., Jr., E.R.H., Maret, P.: NELL2RDF: Reading the web, and publishing it as linked data, Université Jean Monnet (2018). URL: http://arxiv.org/abs/1804.05639

[57] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC), pp. 638–654. Springer, Cham (2017).

[82] Hruschka, E.R., Duarte, M.C., Nicoletti, M.C.: Coupling as Strategy for Reducing Concept-Drift in Never-Ending Learning Environments. Fundamenta Informaticae (2013).

[86] Krishnamurthy, J., Mitchell, T.M.: Joint Syntactic and Semantic Parsing with Combinatory Categorial Grammar. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1188–1198 (2014).

[88] Lao, N., Mitchell, T., Cohen, W.W.: Random Walk Inference and Learning in A Large Scale Knowledge Base. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 529–539. Association for Computational Linguistics (2011).

[90] Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., Zhao, J.: PROV-O: The PROV Ontology, pp. 1–80. W3C (2013). URL: https://www.w3.org/TR/prov-o/

[94] Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn Treebank: Annotating Predicate Argument Structure. In: Proceedings of the Workshop on Human Language Technology, (1994).

[103] Mitchell, T.M., Cohen, W.W., Hruschka, E.R., Talukdar, P.P., Betteridge, J., Carlson, A., Mishra, B.D., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E.A., Ritter, A., Samadi, M., Settles, B., Wang, R.C., Wijaya, D.T., Gupta, A., Chen, X., Saparov, A., Greaves, M., Welling, J.: Never-Ending Learning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 2302–2310 (2015).

[104] Mons, B., Velterop, J.: Nano-Publication in the e-science era. In: Workshop on Semantic Web Applications in Scientific Discourse (SWASD), pp. 14–15 (2009).

[108] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. In: Proceedings of the 23rd International Conference on the World Wide Web (WWW), pp. 759–770. ACM (2014).

[110] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web, pp. 1–1. W3C (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/

[116] Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A Midterm Report. In: Proceedings of the European Conference on Machine Learning, (1993).

[119] Samadi, M., Veloso, M.M., Blum, M.: OpenEval: Web Information Query Evaluation. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA. Citeseer (2013).

[128] Wang, R.C., Cohen, W.W.: Language-Independent Set Expansion of Named Entities Using the Web. In: Proceedings of the 7th IEEE International Conference on Data Mining, pp. 342–350. IEEE (2007).

[132] Yang, B., Mitchell, T.M.: Joint Extraction of Events and Entities within a Document Context. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pp. 289–299 (2016).

[140] Zimmermann, A., Gravier, C., Subercaze, J., Cruzille, Q.: Nell2RDF: Read the Web, and Turn it into RDF. In: Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data, pp. 1–7 (2013).

Chapter 8

Towards Capturing Contextual Semantic Information About Statements in Web Tables*

In Part I we studied how contextual data about statements can be represented in first order logic, description logic, and OWL/RDF. We provided a formal definition of contextualization function—a function that maps a set of statements and a context to a new set of statements in the same logic system—and studied what properties such a function could have. Then, we proceeded to propose a family of contextualization functions, that can be used in the three logic systems, and compared against other approaches that exist in the Semantic Web community. We showed that our proposal preserved semantics better, and was able to differentiate different contexts, better than any other existing approach.

In previous chapters of this part, we have seen that contextual information exists on data in the wild, either implicitly, or explicitly but without any formalization. We have devised *ad hoc* tools to extract that data and, in one case, representing it using a number of contextualization functions in RDF.

In this chapter, our goal is to make a step forward in having generic approaches to extract contextual data and represent it using a contextualization function. We focus on web tables as an example of structured data on the Web that has no clear semantics. We see that there is an emerging research effort in lifting tabular data into semantic web formats, however, most of the work is focused around entity recognition in tables with simple structure. In this work we explore how to capture the semantics of complex tables and transform them to knowledge graph. These complex tables include contextual information about statements, such as time or provenance. Hence, we need to use contextualized knowledge graphs to represent the information of the tables. We explore how this contextual information is represented in tables, and relate it to previous classifications of web tables, and how to encode it in RDF using different

*This chapter is based on the following publication:

- Quécole, F., Martines, R., Giménez-García, J.M., Thakkar, H.: Towards capturing contextual semantic information about statements in web tables. In: Capadisli, S., Cotton, F., Giménez-García, J.M., Haller, A., Kalampokis, E., Nguyen, V., Sheth, A.P., Troncy, R. (eds.) Joint proceedings of the international workshops on contextualized knowledge graphs, and semantic statistics co-located with 17th international semantic web conference (ISWC 2018)CEUR workshop proceedings, CEUR-WS.org (2018). [115]

approaches. Finally, we present a prototype tool that converts web tables from Wikipedia into RDF, covering most existing approaches.

8.1 Introduction

Data is being published in the web at an ever-increasing speed. However, most of this data lacks semantics. This makes it difficult to use it to generate value. Knowledge-graphs are a well-known representation to encode data semantics. The Semantic Web provides standards to represent inter-operable knowledge graphs were each resource can be unequivocally referenced. Tools to generate semantic data from structured web data (specially tables) are gaining traction in the recent years. Most approaches focus on entity recognition and disambiguation, in order to automatically extract the information and transform it to RDF. However, to the best of our knowledge, existing approaches tackle only simple tables with no additional information about the statements that can be extracted. More complex tables exists that provide statements in different contexts (*e.g.*, according to different sources, or valid at different time periods). In order to encode this contextual information (or statement metadata), we need to identify those contexts and represent the information accordingly using contextualized knowledge graphs. In this work we focus on transforming tables into RDF, where contexts are represented by means of reifying the statements using the main existing approaches.

The rest of the paper is organized as follows: in Section 8.2 is discussed some background information; Section 8.3 presents an overview of how data is usually represented in web tables, challenges to represent this data in RDF, and how recent research is dealing with them; Section 8.4 discusses the proposed approach to transform data from web tables to RDF; finally, Section 8.5 draw some conclusions and possible lines of future work.

8.2 Background

In this section we introduce the necessary background information about RDF, existing reification approaches, and tools to convert automatically structured data to RDF.

8.2.1 RDF

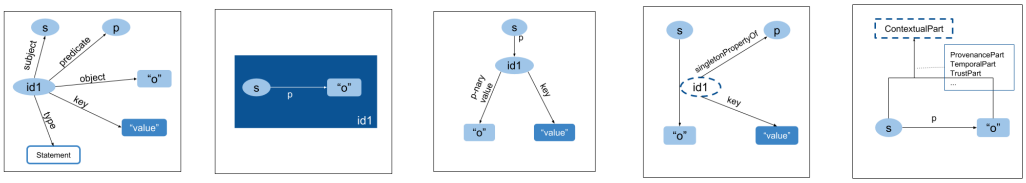
RDF is the data model used in the Semantic Web. It represents statements as triples `<Subject, Predicate, Object>`. The subject identifies the resource being described, the predicate is the property applied to it, and the object is the concrete value for this property. Triples can share subject and/or object, hence creating a interconnected graph of (possibly heterogeneous) statements. Formal definitions of RDF triple and RDF graph can be seen in Definitions 8.1 and 8.2.

Definition 8.1 (RDF triple)**.** *Assume infinite, mutually disjoint sets I (IRI references), B (Blank nodes), and L (Literals). An RDF triple is a tuple $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$, where "s" is the subject, "p" is the predicate and "o" is the object.*

Definition 8.2 (RDF graph)**.** *An RDF graph G is a set of RDF triples $\{(s, p, o)\}$. It can be represented as a directed labeled graph $s \xrightarrow{p} o$.*

8.2.2 Annotating RDF with contextual information

As seen in the previous section, RDF statements represent binary relations between to resources (the subject and the object). This model is not well suited to represent additional contextual information about the statement themselves (such as date of validity, provenance, or confidence). Current approaches to represent this kind of information reify the statement into a new resource, that can be then used as subject or object of new statements that represent the context. Down below we describe the five main existing approaches. In Figure 1, we illustrate each of them.



(a) Standard Reification **(b)** Named Graphs **(c)** n-ary Relations **(d)** Singleton Properties **(e)** NdFluents

Figure 8.1: RDF Approaches

In the following definitions, we assume (s, p, o) is an RDF triple and i is a term (either an IRI or a blank node) that reifies (*i.e.*, unequivocally identifies) the triple.[a]

In RDF Reification [89, Section 4], a resource can be used as a statement, and additional information can be added as follows: a statement can be described by the triples $(i, \texttt{r:subject}, s)$, $(i, \texttt{r:predicate}, p)$ and $(i, \texttt{r:object}, o)$.

Named Graphs [23] considers a sets of pairs in the form (G, n) where G is a RDF graph and n is an URI (Uniform Resource Identifier). Then, we have N-Quads directly describing an (s, p, o, i) quad.

In N-ary Relations [110], a resource is used to describe a relationship, considering that a subject is involved in a relationship, which in turn has its own identifiers and qualifiers. Here, a triple can be decomposed in (s, p_s, i) and (i, p_v, o), $(p_v, \texttt{:value}, p)$, $(p_s, \texttt{:statement}, s)$.

Singleton Properties [108] creates a property that is only used for a unique statement. To represent a triple we need of the triples (s, i, o) and $(i, \texttt{:singlePropertyOf}, p)$.

NdFluents [57]: creates contextual versions of subject and object and links them to the original and the context using the triples $(s_i, contextualPartOf, s)$, $(o_i, contextualPartOf, o)$, $(s_i, contextualExtent, c)$, $(o_i, contextualExtent, c)$.

8.2.3 RDF generation tools

In order to transform a data source into RDF, a common approach is to use a mapping language to represent how the data from one source has to be transformed into triples. Several tools exist to transform heterogeneous data formats into RDF, most of them tackling a single data model or format. In this section we focus on the two most prominent mapping languages: RML [32] and SPARQL-Generate [91]. Our approach will make use of both in different steps of the process.

[a] This paragraph has been added and the definitions updated in this chapter to avoid ambiguity. Note that this definitions are given using the concept of contextualization function in Chapter 4.

RML [32] stands for "RDF Mapping Language". It is an extension of R2RML (Relational to RDF Mapping Language)[1]. While R2RML can be used to express customized mappings from relational databases to RDF datasets, RML also supports other structured formats, such as CSV, TSV, XML and JSON. R2RML's mapping references relational tables' column by name, and uses predicates such as SubjectMap, PredicateObjectMap, PredicateMap and ObjectMap. Each of the above mentioned predicates have as object a column or a URI and the triples are created according to the predicates and their respective referenced column(s). RML extends the R2RML vocabulary to include more general clauses (in which the R2RML's clauses are included—as a subset or sub-property), i.e., *rr:logicalTable* and *rr:tableName* become a sub-property of *rml:logicalSource* and *rml:sourceName*. In our work, we further extend RML to include additional vocabulary about how to extract statements and their contextual information from a Web table. With this vocabulary it is possible to indicate the column that corresponds to the subject, the type of table (see Section 8.3), or how to locate the table in the HTML DOM.[b]

SPARQL-Generate [91] extends SPARQL 1.1 to be able to extract information from heterogeneous data sources. SPARQL-Generate includes three new clauses:

- **source** clause: used to bind variables to documents
- **iterator** clause: used to extract bits of information from the documents
- **generate** clause: extends the existing *construct* clause of SPARQL 1.1, allowing modularization of queries and factorization of the RDF generation.

The first two clauses (source–and its binding functions– and iterator) allow SPARQL-Generate to support various data formats and navigate through them.

8.3 Tables on the Web

According to Crestan and Pantel [25] web tables can be categorized as *layout tables* (used for presentation purposes, not really structuring any knowledge), and *relational tables*. Relational tables encode implicit semantics of the data, and can be further divided according to their structure in *vertical listing*: tables that list in each row one or more attributes for a series of similar entities located in one column (the subject column); *horizontal listing*: similar to vertical listing, horizontal listings present their subjects in one row; *attribute/value*: these tables are a specific case of vertical listings and horizontal listings, but they do not contain the subjects in the table; *matrix*: tables that have the same value type for each cell at the junction of a row and a column; *calendar*: a specific case of the matrix type, differing only in its semantics; and *enumeration*: tables that list a series of objects that have the same ontological relation.

Muñoz, Hogan, and Mileo [105] identify three types of tables in Wikipedia: *toc*, *infobox*, and *wikitable*. The first corresponds to layout tables. In these tables (and here "toc" stands for: "table of content") the topics of the article are presented. The second and the third correspond to relational tables. Infoboxes have a clear horizontal listing structure where the subject is the same for all rows, and the predicate

[1] https://www.w3.org/TR/r2rml

[b] This paragraph has been rewritten in this chapter for the sake of clarity.

Table 8.1: Subset of World population estimates table from Wikipedia

Year	United States Census Bureau(2017)	Maddison(2008)
1950	2,557,628,654	2,544,000,000
1951	2,594,939,877	2,571,663,000
1952	2,636,772,306	2,617,949,000
1953	2,682,053,389	2,665,959,000

and object can be identified in each row.[c] They form the basis of extracted data to create DBPedia [3]. Wikitables are used to embed tables with semantic content in a Wikipedia article, but their structure is highly variable.

While solutions for transforming data in tables to RDF have been proposed, most of them focus on challenges such as identifying the subject column, interpret the implicit structure of table, entity recognition and disambiguation, and mapping values in the table with classes and properties in a knowledge base [97]. In addition, they only tackle vertical and horizontal listings with simple structure. In this work, we tackle more complex tables, where contextual information needs to be expressed about the extracted triples (such as date or provenance). This contextual information is usually encoded in the tables in one of the following two ways: (1) In horizontal and vertical listings, by grouping columns by the context.[2] (2) In matrix tables, by using row and column headers as identifiers of the context.[3]

8.4 Approach

The transformation from tables to Knowledge Graphs needs to consider the different typologies of tables presented in the previous section. For tables without contextual metadata about the statements the process is relatively simple: each cell in the subject column is mapped to a subject in a triple and each cell of the same row to an object, using a property that depends on the column of the object. However, for tables that contain contextual information it is necessary to capture the context of the triples. RDF, as mentioned in Section 8.2.1, only supports binary relations. In order to capture the context of the triples it will be necessary to resort to a reification approach (see Section 8.2.2). Take as an example Table 8.1[4]. We want to extract information not only about the population estimates, but also about the corresponding year and the agency responsible for that estimation. This table is an example of a matrix table, where contexts are indicated by the headers of rows and columns. Listing 8.1 exemplifies an expected output for the value for the cell of row 1 and column 2, including all the contextual metadata.

In addition, the approach needs to read the webpage and extract the information. However, the HTML structure of the table can be arbitrary, and this is one of the challenges to face in this approach. Hence, it is necessary to include a preliminary step to pre-process the table. For this prototype, we decide to get some of

[2]See https://en.wikipedia.org/wiki/List_of_sovereign_states_and_dependent_territories_by_mortality_rate, where the same data is given twice but with different sources.

[3]See Table 8.1.

[4]Taken from https://en.wikipedia.org/wiki/World_population_estimates

[c]The original sentence in the paper stated that "subject, predicate, and object can be identified in each row. The subject does not really appear in the table, and is related to the topic of the page. The sentence has been changed for the sake of clarity.

Listing 8.1: Expected output example

```
wp:year1950   a   time:DateTimeDescription , time:Interval ;
                  time:year   "1950"^^xsd:gYear .

wp:Maddison   a   ex:Provenance ;
        prov:wasGeneratedBy [
                  a event:Event , prov:Activity ;
                  event:time [
                           a time:Interval ;
                           time:hasDateTimeDescription [
                                    a time:DateTimeDescription ;
                                    time:year
"2008"^^xsd:gYear ] ] ].

<http://purl.org/az/worldpop#earth:year1950:Maddison>
        rdf:object              2544000000 ;
        rdf:predicate           dbo:populationTotal ;
        rdf:subject             dbr:Earth ;
        time:intervalDuring     wp:year1950 ;
        prov:agent              wp:Maddison .
```

the necessary information from the user. The preprocessing step produces as output a modified version of the table with additional information: indexes for column and row, the datatype for the value in each cell, category of the table and groups of columns. This information is then used by an RDF conversion module. Note that this approach could be extended to include other kinds of knowledge graphs, such as property graphs, by adding a new conversion module. A schema of this process is shown in Figure 8.2.

The input taken by the preprocessing module is written in RDF using RML [32]. We extend the vocabulary with the following terms:

- *CSSselector*: indicates the CSS selector for the target table in the web page;
- *TablePosition*: index for the target table, given the CSS selector;
- *Reification*: indicates to which category the table belongs;
- *SubjectIndex*: indicates the column that helds the subject for the triple;
- *HeaderRow*: (when columns are grouped by context) indicates in which row the headers (that will be used as predicates) are;
- *ColumnPredicate*: index of the column that is part of the predicate.

The RDF conversion module makes use of SPARQL-Generate [91], using its XPath function to iterate over the elements of the table, and the above mentioned input from the user, except for the first three that are used in the preprocessing step, are used to compose the SPARQL-Generate query. The values inserted by the user dictate the role for each column from the HTML table, that is, which column is the subject, part of the predicate or just the object of the triples (with the header being the predicate).

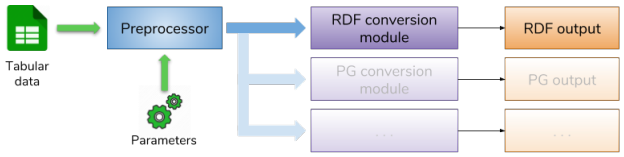


Figure 8.2: Table to KG transformation workflow

The prototype tool is publicly available[5] under Apache-2.0 license.

8.5 Conclusions

Transforming web tables into knowledge graphs while capturing their semantics and contextual information is a challenging task for various reasons: On the side of the knowledge graph representation, it can be necessary to use reification techniques in order to encode the context. On the side of the table, the HTML structure can be arbitrary, and the contents of the table can be difficult to identify. We propose a two-step process. The first step takes additional information and pre-processes the table, generating an enriched version of the table with the information needed by the second step, such as the category of the table or how to extract the contextual metadata about the statements. The second step reads the output of the preprocessor and transforms the data in a knowledge graph. We have implemented a tool that gets part of the necessary information from the user (falling back to default values in case some information is not given) in the first step, and a RDF conversion module as second step. Note that other approaches focusing on different challenges, such as entity disambiguation or subject column identification, could be incorporated in the preprocessing step. Conversely, new modules can be added to substitute the RDF transformation to another kind of knowledge graph, such as property graphs.

References

[3] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Proceedings of the 6th International Semantic Web Conference (ISWC) and the 2nd Asian Semantic Web Conference (ASWC), pp. 722–735. Springer, Busan, Korea (2007).

[23] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. Journal of Web Semantics 3(4), 247–267 (2005).

[25] Crestan, E., Pantel, P.: Web-scale table census and classification. In: Proceedings of the fourth ACM international conference on Web search and data mining, pp. 545–554 (2011).

[32] Dimou, A., Sande, M.V., Colpaert, P., Verborgh, R., Mannens, E., deWalle, R.V.: RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In: Proceedings of the Workshop on Linked Data on the Web, co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, (2014).

[57] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC), pp. 638–654. Springer, Cham (2017).

[89] Lassila, O., Swick, R.R.: Resource Description Framework (RDF) Model and Syntax Specification, W3C (1999). URL: https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[91] Lefrançois, M., Zimmermann, A., Bakerally, N.: A SPARQL Extension for Generating RDF from Heterogeneous Formats. In: The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I, pp. 35–50 (2017).

[5] https://github.com/felipequecole/table2rdf

[97] Martinez-Rodriguez, J., Hogan, A., Lopez-Arevalo, I.: Information Extraction meets the Semantic Web: A Survey. Semantic Web journal (2018).

[105] Muñoz, E., Hogan, A., Mileo, A.: Using linked data to mine RDF from wikipedia's tables. In: Proceedings of the 7th ACM international conference on Web search and data mining, pp. 533–542 (2014).

[108] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. In: Proceedings of the 23rd International Conference on the World Wide Web (WWW), pp. 759–770. ACM (2014).

[110] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web, pp. 1–1. W3C (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/

[115] Quécole, F., Martines, R., Giménez-García, J.M., Thakkar, H.: Towards capturing contextual semantic information about statements in web tables. In: Capadisli, S., Cotton, F., Giménez-García, J.M., Haller, A., Kalampokis, E., Nguyen, V., Sheth, A.P., Troncy, R. (eds.) Joint proceedings of the international workshops on contextualized knowledge graphs, and semantic statistics co-located with 17th international semantic web conference (ISWC 2018)CEUR workshop proceedings, CEUR-WS.org (2018).

Conclusions to Capturing the Context of Statements

In Part I we explored how contextual information can be added to a set of statements within the same logic system and preserve their semantics. Following a top-down path from First Order Logic, through Description Logics, and arriving at OWL and RDFS, we validated that it was indeed possible, and proposed a formalization of the process in the form of the *contextualization function*. We then formalized the properties that a contextualization function can have. We compared how existing approaches in the Semantic Web community behaved according to them. Finally, we proposed a family of contextualization functions, *Nd**, and showed that they better preserve the semantics of the original set of statements when adding a context.

In this part, we have focused our work in answering the research question **R5:** *How to capture existing contextual information that exists implicitly in non-formalized data when transforming it into formal statements?* In order to answer this question, we have explored the implicit contextual data found on existing data in the wild; we have extracted and transformed data and their context into RDF using different contextualization functions; and finally we have proposed a generic approach to capture and transform the context about facts in Web tables, together with the data, into RDF. Transforming data into RDF using different contextualization functions has allowed us to delve into research question **R4:** *How does a contextualization function affect the efficiency with regards to how the data and their context are stored and queried?*

The contributions of this part include:

- **C4:** Generation and publication of large data sets containing explicit contextual information from non-formalized data with implicit context. This contribution corresponds to the research questions R4 and R5.
- **C5:** A generic approach to capture contextual information from relational tables and transform it into RDF contextualized statements. This contribution corresponds to the research questions R5.

From here, only research question **R6:** *How to efficiently manage contextualized data, independently of the concrete representation used to model it?* remains open. This questions is addressed on the next part: III: Managing the Context of Statements.

Part III

Managing the Context of Statements

Preamble to Managing the Context of Statements

In Part I we studied how the context about a set of statements in a logic system could be added to the set, using the same formal system, and preserving the original semantics within the context. This was achieved by what we call *Contextualization Function*, a function that maps a set of statements and a context to a new set of statements. We formalized existing approaches to model context in the Semantic Web into contextualization functions in Chapter 4. We saw that every contextualization function for existing approaches creates, for each statement, a fresh term that reifies it and connects it with the context. By the means of formal properties that a contextualization function can have with regards to how it preserves their semantics, we proposed Nd*, a family of contextualization functions that better preserves them, and compared against the existing approaches.

In Part II, we explored contextual data existing in real-world data. We first extracted contextual data about the reuse of linked open datasets by other datasets and by users in chapters 5 and 6. Then, we transformed data and contextual provenance data into RDF, using a number of contextualization functions in Chapter 7. Finally, we proposed a generic approach to capture data and their context from Web tables into RDF, using an arbitrary contextualization function, in Chapter 8.

However, this variety in contextualization functions hampers the use in practice of contextual data. They are not interoperable, and choosing one of them is not an easy task: previous work shows that there is a compromise between how well semantics are preserved and the number and complexity of generated statements. This part deals with these problems of heterogeneity of contextualization functions and their varying degrees of complexity.

The goal of this part is to evaluate how the selection of contextualization approach affects the storage and retrieval of data, as well as to explore solutions to manage contextualized RDF statements with independence of the contextualization approach used to generate them.

This part contains one chapter: Chapter 9, "HDTr: Managing Reified Triples in Compressed Space" presents HDTr, a binary serialization that extends HDT for contextualized RDF statements. It assumes that each triple is reified in a term, which will be used as the anchor for the contextual annotation. This assumption allows to encode contextual information with improved compression and query performance. HDTr allows to manage contextualized statements from the most used subset of contextualization functions, as well as other existing approaches not based on contextualization functions.

Chapter 9

HDTr: Managing Reified Triples in Compressed Space*

In this chapter we make contribution with regards to how to manage contextual statements in the Semantic Web. We have seen in previous chapters that the context of an RDF statement can be represented by either extending the syntax of RDF or using a contextualization function. Approaches that extend the syntax have no clear semantics with regards to how statements and the contexts are related. Different contextualization functions have different properties relative to both preservation and separation of their semantics within a context. There is usually a compromise between the properties of the contextualization function and the number and complexity of the statements it generates. Currently, there is no generic tool that allows to efficiently manage contextualized statements regardless of the concrete contextualization approach used to represent them. Thus, the approach to model contextualized statements and the tool to manage them needs to be chosen on a case-by-case basis by the user who wants to publish or make use of contextual data. Then, to query this data, it will be always necessary to know what approach was chosen to represent the data, in order to construct the query and interpret its results.

In this chapter we propose HDTr *(HDT for reification)*, the first step towards a model-agnostic tool that can manage contextualized RDF statements with independence of the actual contextualization approach used. HDTr is a compressed binary serialization of contextualized statements that allows for triple retrieval without the need to decompress the data. RDF currently targets the subset of contextualization approaches where a statement is reified into a term. This subset is the most commonly used in current real-world datasets, and we predict that HDTr can be generalized to a solution that includes other contextualization approaches.

9.1 Introduction

In the previous chapters of this dissertation, we have shown how the context of a set of RDF statements can be expressed by either extending the syntax of RDF, or by rewriting and introducing additional RDF statements. We have formalized this second process in what we name *contextualization function*. We have described existing contextualization approaches and provided contextualization functions when

*This chapter is based on unpublished content by José M. Giménez-García in 2022 about work made in collaboration with Thomas Gautrais, Javier D. Fernández, and Miguel A. Martínez-Prieto.

possible. We have as well proposed two new contextualization functions, namely *NdFluents* and *NdProperties*, that preserve RDFS and OWL entailments better than any other existing proposal.

During this process, we have seen that approaches that extend the syntax do not have clear semantics about what being true in a context means. For contextualization functions, we note that each of them has different properties with regards to how well they preserve the semantics of the original statements and how well they can separate information of different contexts. In addition, we have observed that there can be a compromise between these properties and the increase in number and complexity of the contextualized statements. For these reasons, different users can choose a different contextualization approaches to encode their data.

But this comes at a cost in terms of interoperability and efficient management of the data. If different datasets contextualize their statements using different contextualization approach, it will be harder to work with them together. If the contextualization of a set of statements increases its size and complexity, it will increase the efforts to efficiently store and query them.

This work makes a first step to address the problem of efficiently managing contextualized statements independently from the contextualization approach used to generate them. It extends HDT, a well known binary data format that uses compact data structures to store RDF data in compressed space, while allowing efficient access to the triples without the need for decompression.

The rest of the chapter is organized as follows: First, we recall the necessary background to understand the rest of the content in Section 9.2. Section 9.3 presents HDTr, the extension of HDT for contextualized triples. Section 9.4 presents empirical evaluations of HDTr against HDT and a selection to triplestores.Finally, Section 9.5 presents some conclusions about the work and lines of future work.

9.2 Preliminaries

This section provides a summary and discussion of the background knowledge necessary to understand the contribution of this chapter. We start by describing RDF. Then, we present HDT, a state-of-the-art RDF binary serialization that compresses RDF while allowing for triple pattern query functionality. Next, we recall what a contextualization approach on RDF is from Part I. Finally, we discuss how contextualized RDF is handled in standard solutions (including HDT) and existing proposals, and why another solution is needed.

9.2.1 RDF

RDF [28] is the data model, standardized by the W3C, to represent statements in the Semantic Web. An RDF statement is composed of 2 nodes (subject and object) and a relation. Nodes are either IRIs, or Blank nodes, or Literals (but subject nodes cannot be literals), while relations are IRIs. The set of RDF statements composes an RDF graph. Their formal definitions are as follows:

Definition 9.1 (RDF Triple)**.** *We assume infinite disjoint sets $\mathcal{I}$ (IRIs), $\mathcal{B}$ (blank nodes), and $\mathcal{L}$ (literals). An RDF triple is a tuple $(s, p, o) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$, where s is called the* subject*, p is the* predicate *and o is the* object*. We write $\mathcal{T}$ the infinite set of triples.*

Definition 9.2 (RDF graph)**.** *An RDF graph $G \subset 2^{\mathcal{T}}$ is a set of RDF triples.*

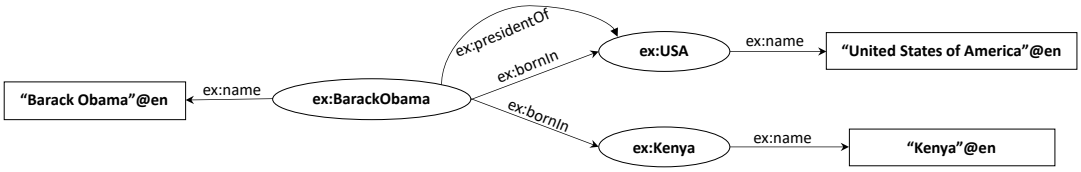


Figure 9.1: Example of RDF Graph

As an example, Figure 9.1 shows an RDF graph with statements about Barack Obama being the president of United States and birth places. The two birth places correspond to those given by two different information sources. Note that RDF does not provide a way to give information about statements themselves, and something to add this contextual information is needed.

9.2.2 HDT

HDT (**H**eader-**D**ictionary-**T**riples) [44] is a binary serialization format designed to optimize storage and transmission of RDF data, while enabling querying of triple patterns. An HDT file encodes an RDF graph in three components:

1. **Header:** It contains metadata to discover and parse the HDT file.
2. **Dictionary:** It organizes the set of terms used in the set of triples and assigns a unique identifier to each term.
3. **Triples:** It substitutes the terms by the identifiers from the Dictionary part and encodes the triples using compact data structures.

The Dictionary and Triples components make use of compact data structures to compress a set of RDF triples while allowing triple-pattern retrieval.

The HDT Header

The *Header* provides metadata about the RDF graph and the HDT file itself. It makes use of existing vocabularies for dataset description, such as VoiD [1] and OWL annotations [75, Sec. 5.3]. It contains four basic types of metadata:

- *Publication information:* Site of publication, dates of creation and/or modification, language(s), namespaces, etc.
- *Dataset statistics:* Statistical data about the RDF graph, such as number of triples; number of different subjects, properties, and objects; etc.
- *Format information:* Metadata about the format of the HDT file, such as the dictionary and triples implementations.
- *Other information:* Any other information that the creator of the dataset or the HDT file wants to provide.

Contrary to the dictionary and the triples, the header is always written in plain text, so it is always readable by any user.

The HDT Dictionary

The *Dictionary* component presents an interface where the terms are organized in three categories, corresponding to the three roles in which terms can appear in the RDF graph, respectively: *subject*, *predicate*, and *object*. In each category each term is mapped to an integer sequential ID. Note that the same ID can correspond to different terms in different sections.

HDT allows for different Dictionary implementations. The most efficient and used in practice is the *Four Section Dictionary*. Following the structure of the interface, it organizes the terms in separate sections according to their role. There is a separate section for each role and an additional section for terms that appear at the same time as subject and object. The number of elements in each section is stored in the Header component. Each section is sorted lexicographically, with the position of the terms implicitly encoding their ID, and compressed independently using prefix-based encoding [96]. This allows to save space and improves access time. The internal IDs of each section range from 1 to $|X|$, where $|X|$ is the number of elements of the section. However, to allow queries using the interface, these IDs are mapped to external IDs according to their role. In detail, the dictionary is comprised of four disjoint sections:

- **SO:** contains the terms that appear as both subject and object in the set of triples. The IDs in this section are mapped to $[1, |SO|]$, where $|SO|$ is the number of unique terms that appear as subject and object in the triples.
- **S:** contains the terms that appear as subject (but not as object) in the set of triples. The IDs in the section are mapped to $[|SO| + 1, |SO| + |S|]$, where $|S|$ is the number of unique terms that appear as subject but not as objects in the triples.
- **O:** contains the terms that appear as object (but not as subject) in the set of triples. The IDs in the section are mapped to $[|SO| + 1, |SO| + |O|]$, where $|O|$ is the number of unique terms that appear as objects but not as subjects in the triples.
- **P:** contains the terms that appear as predicates in the set of triples. The IDs in this section are mapped to $[1, |P|]$, where $|P|$ is the number of unique terms that appear as predicate in the triples. Note that there is no sections for "shared predicates". A term that appears at the same time as predicate and subject and/or object in the set of triples will be included in this section as well as the corresponding sections for subjects and/or objects. The reasoning behind this decision is that IRIs in the predicate position do not often appear as subject or object in other triples in the same graph.

Figure 9.2 shows the Dictionary component for the example given in Figure 9.1 with the internal and external IDs of each section, as well as the interface view of its contents.

Storing terms that appear at the same time in the subject and object positions of a statement has an additional benefit: Just by knowing the IDs of the terms in a set of triples, it is possible to know if the subject of a triple is the same term or not as the object of another triple, without the need to perform all the necessary operations to retrieve the term in the dictionary. If their ID is the same and it is equal or lower than $|SO|$, then it corresponds to the same term. If their ID is either different, or equal but higher than $|SO|$, then it corresponds to a different term. This is particularly

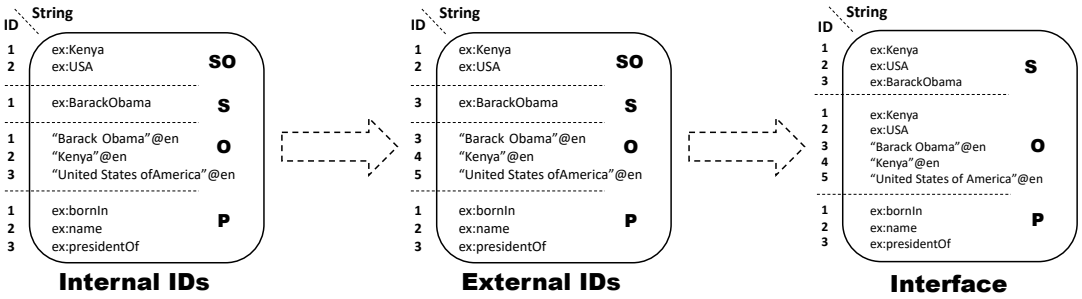


Figure 9.2: HDT Dictionary

useful when different sets of triples are retrieved and it is necessary to perform a join operation on them, which is a common operation in SPARQL queries.

The C++ implementation of HDT provides an additional mapping that allows to not keep track of $|SO|$ at the cost of reduced compression in the Triples component. Sections **SO**, **S**, and **P** have the same mapping as previously described, but section **O** is mapped to $[|SO| + |S| + 1, |SO| + |S| + |O|]$. That means that terms that appear in sections **S** and **O** are never mapped to the same ID, which allows to directly compare the IDs in subject and object positions.

The Four Section Dictionary allows for two basic operations to retrieve its contents:

- $locate(element, role)$: returns a unique ID for the given term and role, if it appears in the dictionary.
- $extract(id, role)$: returns the term for a given ID and role in the dictionary, if it exists.

The HDT Triples

The *Triples* component presents an interface that allows to query for triples. HDT allows for different implementations of the Triples component, but the most efficient and used in practice is the *Bitmap Triples*.

The Bitmap Triples implementation encodes the triples of the RDF graph, replacing their terms by their IDs in the Dictionary. That is, each triple is transformed in a tuple of 3 IDs (ID-triples from now on): <id_s,id_p,id_o>; where id_s, id_p, and id_o are the IDs of the subject, predicate, and object in the dictionary, respectively.

The ID-triples are then sorted according to their IDs in subject, predicate, and object positions, and organized as a forest of trees, with a tree for each distinct term that appears as subject in the set of triples. Each tree has the subject as the root; in the next level there are the terms that appear as predicates in the triples with that subject; and the leaves contain, for each predicate, the terms that appear as object in the triples with the subject-predicate pair.

The forest of trees is then codified with four coordinate sequences: two ID sequences and two bit sequences. These sequences are organized in pairs of bit sequence and ID sequence: one for the predicates (B_p and S_p) and one or the objects (B_o and S_o). The bit sequences contain a 1 for the last term that belongs to the same term in the precedent level (that is, same subject for B_p, and same predicate for B_o), and a 0 for the rest. The ID sequences contain the sequence IDs for each level of the forest of trees. Both types of sequences are encoded using compact data structures that allow to perform search operations in an efficient manner with a minimal space increase. Each bit sequence supports the following two basic operations:

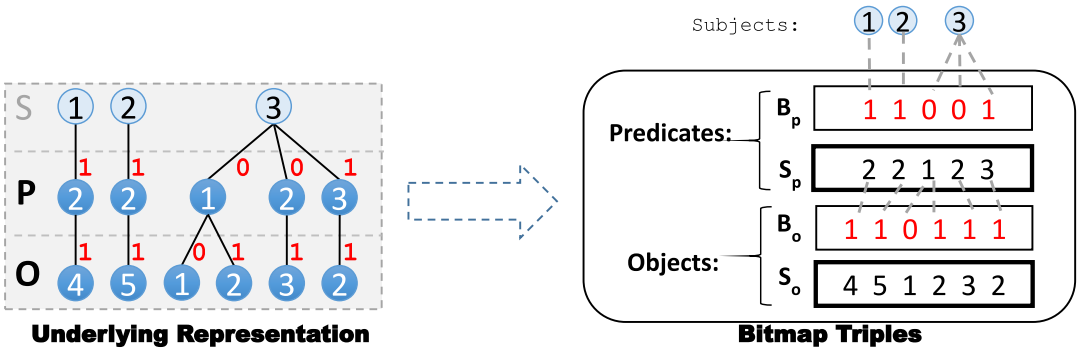


Figure 9.3: HDT Triples

- $rank_a(B,i)$ counts the occurrences of a symbol $a \in \{0,1\}$ in $B[1,i]$.
- $select_a(B,i)$ finds the i^{th} occurrence of symbol $a \in \{0,1\}$ in B. In practice, $select_a(B,0) = 0$;

Figure 9.3 shows the Triples component for the example given in Figure 9.1 with its underlying forest-of-trees representation and the actual implementation.

Querying HDT

The basic operations of the Dictionary and Triples components allow to retrieve a set of statements using triple patterns where the subject is known (*i.e.*, (s,p,o), (s,p,?o), (s,?p,o), and (s,?p,?o)). This is achieved through the following operations:

1. Obtain the corresponding IDs <id_s,id_p,id_o> for the known terms in the triple pattern using the $locate(element, role)$ operation.
2. Obtain the position for the predicates of a given subject using the following steps:
   (a) Retrieve the positions where the predicates start and finish using $begin_p = select_1(B_p, id_s - 1) + 1$ and $end_p = select_1(B_p, id_s)$.
   (b) Retrieve the list of predicate IDs $\overline{id_p}$ between $begin_p$ and end_p.
   (c.1) If an ID id_p was used, find its position $pos_{\overline{id_p}}(id_p)$ inside $\overline{id_p}$ using binary search. Then the position of id_p in B_p is $pos_{B_p}(id_p) = begin_p + pos_{\overline{id_p}}(id_p) - 1$
   (c.2) If a variable was used in the predicate position of triple pattern, the result is the list of positions $\{begin_p \ldots end_p\}$.
3. Obtain the position for the objects of a given pair <id_s,id_p> using the following steps:
   (a) Retrieve the positions where the objects start and finish using $begin_o = select_1(B_o, pos_{B_p} - 1) + 1$ and $end_o = select_1(pos_{B_p}, id_s)$.
   (b) Retrieve the list of object IDs $\overline{id_o}$ between $begin_o$ and end_o.
   (c.1) If an ID id_o was used, find its position $pos_{\overline{id_o}}(id_o)$ inside $\overline{id_o}$ using binary search. Then the position of id_o in S_o is $pos_{B_o}(id_o) = begin_o + pos_{\overline{id_o}}(id_o) - 1$

(c.2) If a variable was used in the object position of triple pattern, the result is the list of positions $\{begin_o \dots end_o\}$.

4. Obtain the corresponding terms for the IDs in the retrieved triples using the $extract(id, role)$ operation.

In order to retrieve triples with different triple patterns, it is necessary to introduce additional indexes for the predicates and the objects. This is proposed in *HDT-FoQ* (HDT Focused on Querying) [95]. In HDT-FoQ two additional structures are added:

- A pair of bit and ID sequences (B_{op} and S_{op}) which stores, for each object id, a sorted list of references to the predicate positions related to it. This allows to locate what predicates and pairs predicate-subject are related to any given object.
- A pair of bit and ID sequences (B_{ps} and S_{ps}) which stores, for each predicate position, a sorted list of references to the subjects related to it. This allows to locate what subjects are related to any given predicate or predicate position. Together with the sequences for the objects, it enables retrieving triples using triple patterns in the form (?s,p,o) and (?s,?p,o).[1]

Using these two additional adjacency lists it is possible to retrieve triples where the subject is not known, but the predicate and/or the object is (*i.e.*, (?s,p,o), (?s,p,?o), (?s,p,o), and (?s,?p,o)). Note that the triples for the triple pattern (?s,?p,?o) are all existing triples in the HDT file, and can be retrieved by sequentially traversing the triples.

With the result of all these triple patterns it is possible to resolve basic SELECT, ASK and CONSTRUCT SPARQL queries.

9.2.3 Representing the context of RDF statements

There are several approaches to represent the context of an RDF statement or set of statements. Some of them represent both the statements and the context in a single RDF graph. These approaches conform to what we call a *contextualization function*. Some other approaches extend the RDF syntax and/or semantics to represent the context of a statement.

Contextualization Functions

Among the existing approaches that conform to a contextualization function we find RDF reification, n-ary relations, the singleton property, and the companion properties. In Part I we have explored what contextualization functions are and what are their properties, formalized these approaches to contextualize statements in contextualization functions and studied their properties, and proposed Nd*, a family of contextualization functions that better preserve the semantics of the original statements. In RDF we have proposed NdFluents and NdProperties in Chapters 3 and 4, that are particular instances of Nd*. Here we recall the important concepts about contextualization functions in RDF.

[1] Note that in the original HDT-FoQ publication [95] it is reported that B_p and S_p are replaced by a wavelet tree, but in practice that change was reversed and the adjacency list was implemented, as reported by Hernández-Illera, Martínez-Prieto, Fernández, and Fariña [74].

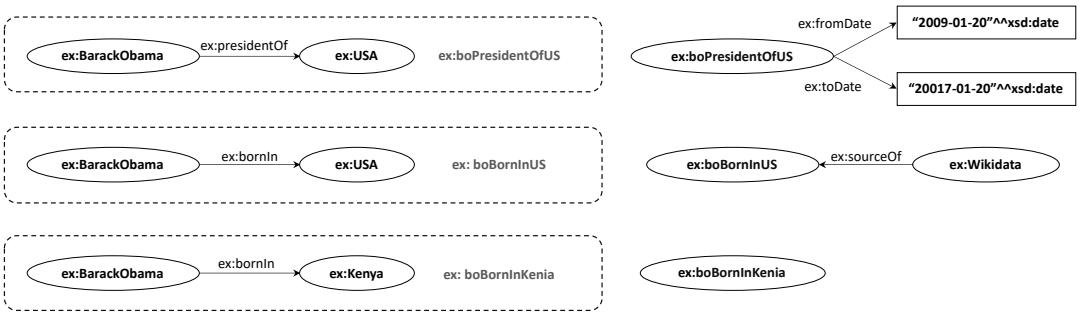


Figure 9.4: Example of Abstract Contextualization Function

A *contextual annotation* can be thought of as a graph C that describes the context of an individual representing the triple. This graph has a distinguished term that is connected to every term that appears in C. We call this individual the *anchor*. Formally, a contextual annotation $\mathcal{a}$ is the pair $(C, \mathbf{a})$.

A contextualization function is, then, a function that maps an RDF statement and a contextual annotation to another RDF graph that (ideally) encodes the same information within the context of the contextual annotation. This graph is the union of two graphs: The first one ($\mathsf{St}(\mathcal{a}, t)$) creates new triples to describe the original statement and has a term (the *statement anchor*) related to at least one of the terms in the statement. The second ($\mathsf{Cx}(\mathcal{a}, t)$) is equivalent to the graph in the contextual annotation, but can have the anchor replaced by another term (the *context anchor*). These two anchors are connected individuals in the union graph (*i.e.*, they are either the same term or are connected using one or more triples).

In practice, a contextualization function needs introduce new terms in the resulting graph. This is done using what we call *renaming function*: a injective function that creates a fresh term that reifies a term or a statement within the context. The renaming function introduces, among others, the terms used for the statement anchor and the context anchor.

Figure 9.4 shows the abstract representation of the results of a contextualization function on some triples of Figure 9.1, where each contextualized statement has a statement anchor associated in the left side, and a contextual annotation for each context in the right side. In this example the context anchors are the same as the statement anchors. Note that for the statement of Barack Obama being born in Kenya the contextual annotation is only context anchor.

As example, we provide the $\mathsf{St}(\mathcal{a}, t)$ functions for *RDF reification* [16, Sec. 5.3] and *NdFluents* [57] ($\mathsf{Cx}(\mathcal{a}, t)$ adds the triples in the right side of Figure 9.4). RDF reification is the standard W3C model to represent information about a statement, where a term that reifies the triple is created and connected to the terms appearing in the triple. NdFluents is a contribution of this dissertation, and was presented in Chapter 3. It introduces contextual versions of the subject and object and links the to the context. Figures 9.5 and 9.6 show the concrete result of the RDF reification and NdFluents contextualization functions for the example given in Figure 9.4 (note that only the statements created by the function $\mathsf{St}(\mathcal{a}, t)$ are included).

Definition 9.3 (RDF reification contextualization function)**.** *Let $\mathcal{a} \in \mathcal{C}$ be a contextual annotation, $G \in 2^{\mathcal{T}}$ a graph, and $t = (s, p, o) \in G$ a triple. We define the contextualization function $f_R(\mathcal{a}, G, t) = \mathsf{St}_G(\mathcal{a}, t) \cup \mathsf{Cx}_G(\mathcal{a}, t)$ such that $\mathsf{St}_G(\mathcal{a}, t) = \{(\mathsf{ren}_G(\mathcal{a}, t), type, Statement), (\mathsf{ren}_G(\mathcal{a}, t), subject, s), (\mathsf{ren}_G(\mathcal{a}, t), predicate, p), (\mathsf{ren}_G(\mathcal{a}, t)), object, o)\}$ and $\mathsf{Cx}_G(\mathcal{a}, t) = \mathsf{rplc}(\mathcal{a}, \mathbf{a}, \mathsf{ren}_G(\mathcal{a}, t))$, where $\mathsf{ren}_G(\mathcal{a}, t) = \mathbf{a}_s = \mathbf{a}_c$.*

Definition 9.4 (NdFluents contextualization function)**.** *Let* $\mathcal{C}a \in \mathcal{C}$ *be a contextual annotation,* $G \in 2^{\mathcal{T}}$ *a graph, and* $t = (s, p, o) \in G$ *a triple. We define the contextualization function* $f_{ndf} = \mathsf{St}_G(\mathcal{C}a, t) \cup \mathsf{Cx}_G(\mathcal{C}a, t)$ *such that* $\mathsf{St}_G(\mathcal{C}a, t) = \{(\mathsf{ren}_G(\mathcal{C}a, s), p, \mathsf{ren}_G(\mathcal{C}a, o)), (\mathsf{ren}_G(\mathcal{C}a, s), type, ContextualPart), (\mathsf{ren}_G(\mathcal{C}a, o), type, ContextualPart), (\mathsf{ren}_G(\mathcal{C}a, s), contextualPartOf, s), (\mathsf{ren}_G(\mathcal{C}a, o), contextualPartOf, o), (\mathsf{ren}_G(\mathcal{C}a, s), contextualExtent, \mathsf{ren}_G(\mathcal{C}a, \mathbf{a})), (\mathsf{ren}_G(\mathcal{C}a, o), contextualExtent, \mathsf{ren}_G(\mathcal{C}a, \mathbf{a})), (\mathsf{ren}_G(\mathcal{C}a, t), type, Context)\}$ *and* $\mathsf{Cx}_G(\mathcal{C}a, t) = \mathsf{rplc}(\mathcal{C}a, \mathbf{a}, \mathsf{ren}_G(\mathcal{C}a, \mathbf{a}))$, *where* $\mathsf{ren}_G(\mathcal{C}a, \mathbf{a}) = \mathbf{a}_s = \mathbf{a}_c$.

We see that these two approaches follow a different principle to represent the context of a statement. RDF reification follows the idea of reifying the triples in a set of a statements. That means that the statement anchor of each triple is a different term, to which the context is then linked. Other existing approaches studied in Part I follow this principle too. The paradigm of Nd* (from which NdFluents is part of) is, instead, to represent the statement within the context. In Nd* the statement anchor is not unique for each statement, but shared by the statements within the same context. This translates into better properties with regards to how well they preserve the semantics of the original set of statements, as well as the ability of modeling different contexts to have their semantics separated.

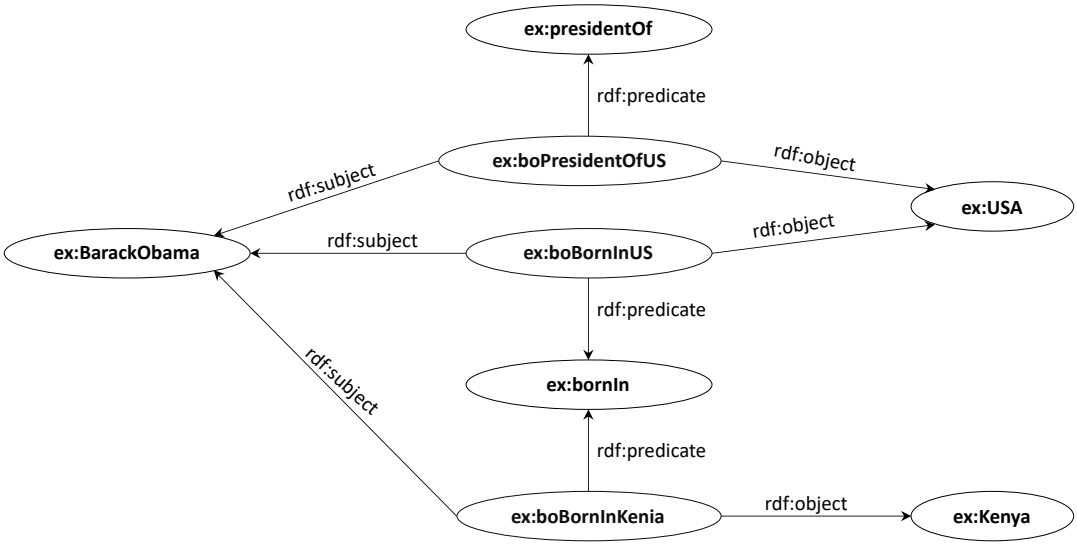


Figure 9.5: Example of RDF Reification Contextualization Function

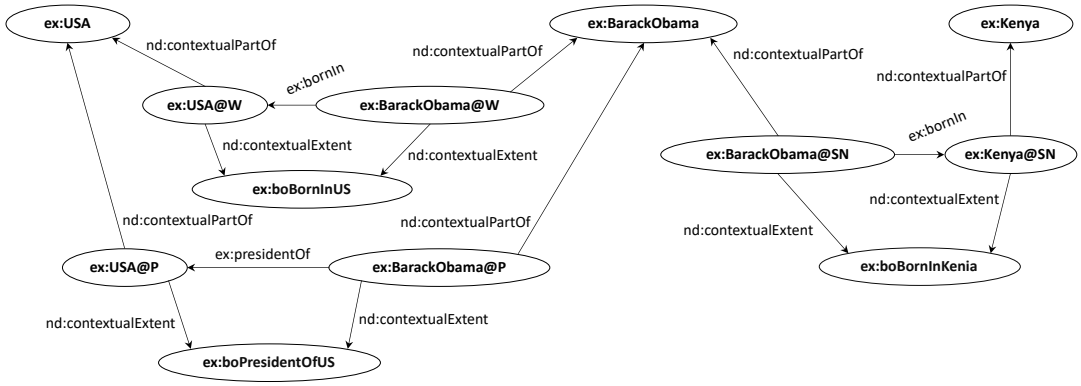


Figure 9.6: Example of NdFluents Contextualization Function

Other approaches

Other approaches do not represent the statement and their context as an RDF graph, but extend the RDF syntax and/or semantics to do it. Named graphs, RDF*[2], and Notation 3 are three such cases.

Named graphs [23] extend the syntax of RDF by adding a fourth term to the triples. This fourth term is used to identify the graph to which the triples belong. This graph can then include the information about the context of the statement. **RDF*** [67] extends RDF with the possibility to use an RDF* statement in subject and object positions of another RDF* triple. This new RDF* triple can be then used to describe the context of the statement. **Notation-3** [6] extends RDF even further with more constructs, such as rule definitions and variable quantifications.

These approaches, however, do not conform to the syntax of RDF. This makes it difficult to manage their results using standard RDF tools. They either put sets of triples in different labeled buckets, or give a label to a triple that can be later used in another statements. In the case of named graphs, this label could be used for the purposes as the statement or context anchor of a contextualization function. RDF* and Notation3 instead reify the statement into a new term, to which the add the contextual information. This term is then conceptually equivalent to the statement anchor in contextualization functions in the reification paradigm, with the exception that there can be only a unique term for triple (*i.e.*, their equivalent to the St function does not depend on the context). This makes them arguably less flexible that these contextualization functions. In addition, none of the approaches have clear semantics about reasoning within a context or what the validity of a statement in a context means for the statement in other contexts. In that sense, all of them are comparable solutions to the reification paradigm in contextualization functions.

9.2.4 Managing the context of RDF statements

Managing contextualized RDF statements that are the result of a contextualization function is straightforward in standard tools to manage RDF triples, such as triplestores or HDT: These statements are regular RDF triples and can be managed as that. However, it is important to note that the result of different contextualization functions are different RDF graphs, with varying number of triples and complexity, and different properties with regards to how well they preserve the semantics of the original set of statements and how well they separate contexts (for more on this and other approaches, refer to Chapter 4). For this reason, it is then necessary to know which contextualization function has been used to generate a set of statements in order to query and to manage it efficiently.

Managing statements that are modeled using an approach that extends the RDF syntax depends on the specific model and tool. Current triplestores are usually able to manage named graph, making it possible to manage sets of statements where the contexts is modeled using exclusively that approach. A number of triplestores (such as AnzoGraph[3], Blazegraph[4], GraphDB[5], or Stardog[6]) are able to manage RDF* statements. To the best of our knowledge, there is no triplestore that allows to manage Notation-3. Hence, in order to use these approaches, it is not only necessary to know which one was used, but to choose a tool that specifically support it, if it exists.

[2] Now renamed to "RDF-star"[68].
[3] https://cambridgesemantics.com/anzograph/
[4] https://www.blazegraph.com/
[5] http://graphdb.ontotext.com/
[6] https://www.stardog.com/

We argue that it is necessary have a model-agnostic management tool for contextualized statements that can be used to store and query the data without the need to know which approach was use to generate the RDF graph.

The issue of representing different approaches using a model-agnostic approach has been addressed by MaSQue [46]. They propose *Meta-RDF*: an intermediate JSON representation for triples and their contextual information, and *Meta-SPARQL*: a tool that automatically rewrites SPARQL queries that use a specific reification model into Meta-RDF. This is a short paper with no evaluation. While the authors make step forward in using a model-agnostic representation of contextualized statements in RDF, the efficient management of this representation is not yet addressed.

TrieDF [113] is a recent work that proposes an in-memory tuple store architecture that can handle RDF data augmented with any type of metadata. Similarly to HDT, it is comprised of a dictionary that assigns an ID to a term. It makes use of tries [14] (prefix-based trees for string storage) to store both the dictionary and the triples. To store contextual data, it extends triples to tuples with and additional position for each context. The publication is too short in details to allow us to properly evaluate it.

HDTQ [45] is an extension of HDT to represent RDF datasets with multiple named graphs by encoding quads instead of triples. It extends the dictionary with an additional section for the IRIs of the named graphs, and uses a bit matrix to note, for each pair (triple,graph), if the triple appears in the named graph or not. HDTQ can arguably be used to encode the context of a set of statements by having the named graph represent the context. However, since the goal of HDTQ is to manage named graphs it makes some assumptions that hinder its use for managing contextualized statements:

1. That the IRI of a graph is not often used as subject or object in another triples. For this reason the dictionary does not include a shared sections for these cases. This reasoning is similar to that about properties in HDT. However, in contextualized statements, it is usually the case that there are a number of statements to describe the triple (equivalent to the contextual annotation), where this IRI can serve as a reference to the context (equivalent to the anchor). Hence, this term often appears as subject or object of other triples.[7] This design, when using HDTQ to serialize contextualized triples, leads to worse compression rates and the impossibility of comparing terms that appear as the graph name and either subject and object of a triple by using their IDs.

2. That the number of graphs is much less than both the number of distinct term in the triples and the number of triples. While the bit matrix is a good representation if that is the case, it is not optimal for a high number of contexts. However, in real-world datasets with contextual information about statements, the number of contexts is usually comparable to the number of statements.[8] This is especially true for contextualization approaches based in the reification paradigm. This design leads to worse compression and query time in such cases.

3. That knowing in which graphs a triple appears in enough information. However, when representing the different contexts in which a statement is true, it is often necessary to be more explicit. For example, the statement could be true

[7] As an example, every belief in NELL2RDF (presented in Chapter 7) has a corresponding contextual annotation.

[8] As an example, every belief in NELL2RDF (presented in Chapter 7) has a different context.

in both contexts independently or true in the combination of the contexts. This is not possible to represent using the aforementioned bit matrix.

While HDTQ can be an ideal serialization to manage RDF datasets with different named graphs, it is not enough to manage contextualized RDF statements. The next section describes HDTr, our proposal to extend HDT to manage contextualized statements.

9.3 Extending HDT for contextualized triples: HDTr

As seen in Section 9.2.3, every contextualization approach introduces a term that represents the context (what we call the *context anchor*). This new term is then connected to a statement or a set of statements, either by extending the syntax of RDF, or via a *statement anchor*, created using a contextualization function. For approaches that extend the syntax, a new tool that deals with their particular syntax is needed. In the case of contextualization functions, each one creates a different RDF graph, with a different number of triples and a different structure. This can have a series of undesirable effects when managing the resulting data, such as the overhead with regards to size of the original data, the performance of the queries needed to retrieve the data, or the interoperability problems between the results of two or more contextualizations.

While HDT could potentially be a solution to the size overhead due to its compression, the interoperability issues and the increased query complexity would still need to be addressed. A different HDT file would be needed for each contextualization function, and all the triples created by it would need to be retrieved at query time. HDTQ could be used to include all the statements that are true in a context under the same named graph. Then, HDTQ could serve as the core of a tool that maps each approach to the original statements and contexts, and vice versa. This tool would allow to encode and query contextualized data with a model-agnostic solution. However, as we have seen in the previous section, its design assumptions make it suboptimal to manage contextualized triples.

In this section we describe our first step towards such a solution. We focus on the subset of contextualization approaches based on the reification paradigm and propose **HDTr** *(HDT for reification)* to serve as the core of such model-agnostic solution. This subset is the most common in current real-world datasets, and we predict that HDTr can be generalized to a solution that includes the Nd* paradigm.

HDTr takes advantage of the structure of the Triples component in HDT. In this structure there is a one-to-one relation between the RDF statements and the IDs in the object sequence S_o. This allows us to easily refer to a triple and connect it with its statement anchor. Leveraging this relation, we extend HDT to manage the result of a contextualization approach by representing the statement anchor without the need to store the specific triples that the function $\mathsf{St}(\mathcal{G}, t)$ generates, or the concrete syntax extension that other approach can have. The triples that connect the statement anchor with the context anchor (or their equivalents in other approaches) and the triples that describe the context of the statement (the result of $\mathsf{Cx}(\mathcal{G}, t)$ for contextualization functions) are stored as regular triples. In addition, we explore alternative ID mappings for the anchors, each one with a different tradeoff with regards to compression and data retrieval.

HDTr is comprised of the same components as HDT: **H**eader, **D**ictionary, and **T**riples. We extend each component to contain the additional information we need in order to make the connection between a statement and its anchor. For that, we

need additional structures in the Triples component to indicate which statements are reified and their unique anchors. These IDs are then used in the Triples component when storing the statements of the contextual annotation. In addition we need to extend the Dictionary component to include new sections for the fresh terms that reify statements. All the modification are made with the additional goal of being retro-compatible with HDT.

9.3.1 The HDTr Header

The HDTr Header contains the same information as in HDT, plus additional statistical and format information about the terms and triples that are used for the contextual information (see Subsections 9.3.2 and 9.3.3).

9.3.2 The HDTr Dictionary

The HDTr Dictionary interface is extended to include and additional category for the anchors. For the implementation, we extend the Four Section Dictionary. It is important to note that these terms are used as subject and/or object in the triples that describe the contextual annotation. Hence, it is necessary to have sections that account for all the possible combinations of role and anchor. As in a regular Four Section Dictionary, internal IDs range from 1 to $|X|$, where $|X|$ is the number of elements of the section. Similarly, the contents of each section are sorted lexicographically and encoded using prefix-based encoding [96]. We divide the HDTr dictionary in two sub-dictionaries:

1. *Triples Dictionary:* It stores the terms that are not anchors. It contains the same sections as the HDT Dictionary:
   - **SO_T:** contains the terms that appear as both subject and object in the set of triples.
   - **S_T:** contains the terms that appear as subject (but not as object) in the set of triples.
   - **O_T:** contains the terms that appear as object (but not as subject) in the set of triples.
   - **P_T:** contains the terms that appear as predicates in the set of triples.
2. *Anchors Dictionary:* It stores the anchor terms. It has a similar structure to the HDT Dictionary, with two exceptions: it contains an additional section for "unused" anchors (that is, the anchor is never used as subject or object in any triple); and it does not contain a section for predicates (since the anchors are only used as subject or object in the contextual annotation).
   - **SO_A:** contains the anchors that appear as both subject and object in the set of triples.
   - **S_A:** contains the anchors that appear as subject (but not as object) in the set of triples.
   - **O_A:** contains the anchors that appear as object (but not as subject) in the set of triples.
   - **U_A:** contains the anchors that appear neither as subject nor object in any triple.

Figure 9.7(A) shows the Triples Dictionary and the Anchors Dictionary for the running example. It contains the triples from Figures 9.1 and 9.4. We see that the Anchors Dictionary contains the terms for the statement anchors of Figure 9.4 in their corresponding section in relation with the positions they are in the triples of the contextual annotation. The Triples Dictionary, in turn, includes the same terms as the HDT Dictionary had for the non-contextualized statements (see Section 9.2.2) plus the terms that are part of the contextual annotation but are not the statement anchor.

The HDTr dictionary can present itself as a *"Five Section Dictionary"*, where each section is the combination of one or more sections of each sub-dictionary. It is retro-compatible with a regular Four Section Dictionary dictionary, providing the usual *SO*, *S*, *O*, and *P* sections, plus an additional *A* section for the anchors:

- $\mathbf{SO} = SO_T \cup SO_A$, with IDs in the range $[1, |SO|]$, where $SO = |SO_T| + |SO_A|$.
- $\mathbf{S} = S_T \cup S_A$, with IDs in the range $[1, |S|]$, where $|S| = |S_T| + |S_A|$.
- $\mathbf{O} = O_T \cup O_A$, with IDs in the range $[1, |O|]$, where $|O| = |O_T| + |O_A|$.
- $\mathbf{P} = P_T$, with IDs in the range $[1, |P|]$, where $|P| = |P_T|$.
- $\mathbf{A} = SO_A \cup S_A \cup O_A \cup U_A$, with IDs in the range $[1, |A|]$, where $|A| = |SO_A| + |S_A| + |O_A| + |U_A|]$

Figure 9.2(B) shows the Five Section Dictionary for the running example. It presents a unified view of the two sub-dictionaries in Figure 9.2(A).

This five sections can then be mapped to the four categories of the interface (subject, predicate, object, and anchor). In order to map the IDs of each section to the interface categories, we explore three different alternatives.

Naive Mapping. This first alternative maps the IDs of subjects, predicates and objects in an identical way as HDT. That is, **SO** is mapped to $[1, |SO|]$, **S** is mapped to $[|SO| + 1, |SO| + |S|]$, **O** is mapped to $[|SO| + 1, |SO| + |O|]$, and **P** is mapped to $[1, |P|]$. The anchor section **A** is mapped to $[1, |A|]$.

The naive mapping is a straightforward function that provides the same IDs as HDT for the terms that appear in subject, predicate, and object position. For the anchors it maps directly the IDs of **A** without any modification. While this provides the smallest possible IDs for the anchors, it leads to the same problematic as HDTQ: The IDs of the anchors are not comparable to the IDs of subjects and objects.

Same-ID Mapping. This second mapping is created with the goal of having comparable IDs for subjects, objects, and anchors. It follows the idea of the alternative mapping provided by the C++ implementation of HDT: different terms are always mapped to different IDs. Hence, terms in **SO** are mapped to $[1, |SO|]$, terms in **S** are mapped to $[|SO| + 1, |SO| + |S|]$, terms in **O** are mapped to $[|SO| + |S| + 1, |SO| + |S| + |O|]$, and terms in **P** is mapped to $[1, |P|]$. The mappings for terms to the anchor category are less straightforward, since IDs that appear in $\mathbf{SO_A}$, $\mathbf{S_A}$, and $\mathbf{O_A}$ need to have the same ID in **S**, **O**, and **A**. The mapping for this sections is as follows: $\mathbf{SO_A}$ is mapped to $[|SO_T| + 1, |SO|]$, $\mathbf{S_A}$ is mapped to $[|SO| + |S_T| + 1, |SO| + |S|]$, $\mathbf{O_A}$ is mapped to $[|SO| + |S| + |O_T| + 1, |SO| + |S| + |O|]$, and $\mathbf{U_A}$ is mapped to $[|SO| + |S| + |O| + 1, |SO| + |S| + |O| + |U_A|]$.

The same-id mapping provides the same IDs as the alternative mapping in the C++ implementation of HDT for the terms that appear in subject, predicate, and

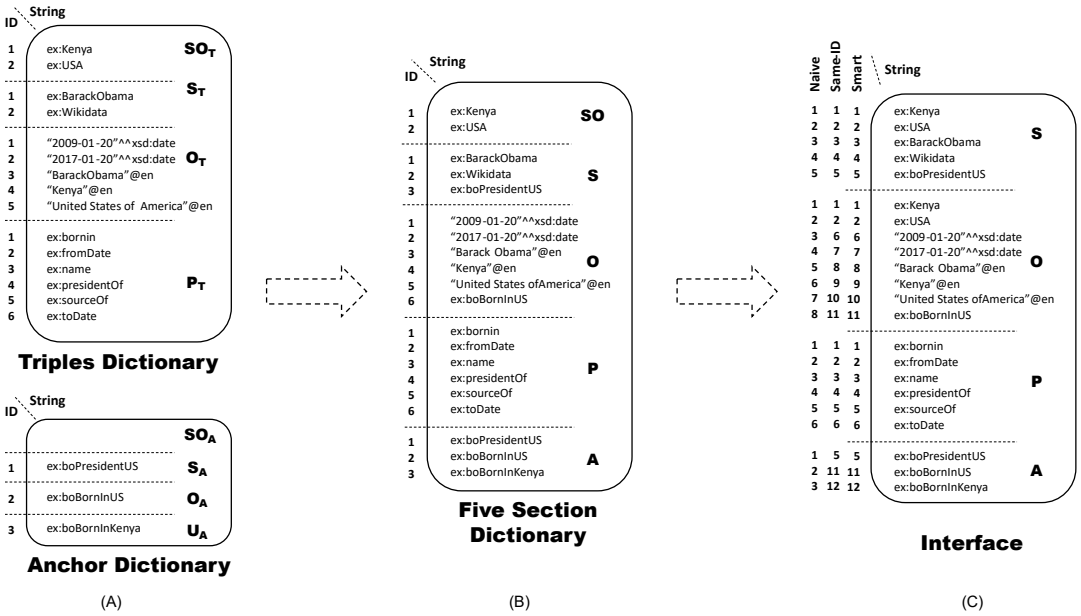


Figure 9.7: HDTr Dictionary

object position. For the anchors it provides the IDs that are comparable to those of subjects and objects. This allows to compare terms in triples by ID without the need to retrieve the actual term from the dictionary, at the cost of having higher IDs and thus, worse compression rates in the Triples component.

Smart Mapping. This final alternative intends to take the best from the two previous options. It accomplishes the goal of having comparable IDs while avoiding the size overload of the same-id mapping. This is achieved by providing the following operation:

- $mapping_m(id, m')$ returns the ID of a term using mapping m, provided that id is the ID of the same term in mapping m'.

Then, the Dictionary component uses the same-id mapping, but the Triples component encodes the triples with the naive mapping. Whenever triples are retrieved, the IDs in the Triples component are mapped to the IDs in the dictionary using $mapping_{same\text{-}id}(id, naive)$. Note that the inverse mapping $mapping_{naive}(id, same\text{-}id)$ is necessary when serializing RDF into HDTr, in order to transform the IDs in the dictionary to those that need to be stored in the triples.

The smart mapping provides an alternative solution, where the IDs of subjects, objects, and anchors from retrieved triples are comparable. It avoids the space overloads of the same-id mapping, but introduces additional operations that need to be called for each triple at serialization and query time.

The three mappings present a tradeoff between compression, serialization time, and query time. Note that it is not possible to have a mapping where IDs of subjects and objects were mapped to **S** and **O** in the same way as the standard HDT and the IDs of the anchors were equal to them for the same terms. The reason is that there could be different anchors that appear as subject and objects with the same ID. Figure 9.7(C) shows the interface for the HDTr Dictionary component for the running example with the three different mappings.

All sections and categories support the same $locate(element, role)$ and $extract(id, role)$ as the Four Section Dictionary. The size of each individual and combined section are stored in the Header component.

9.3.3 The HDTr Triples

The HDTr Triples component is extended to identify what statements are contextualized and relate them their anchor. Conceptually, it relates the position of each contextualized triple with its position in the Dictionary component. It has the same structure of HDT plus two additional aligned data structures, a bit sequence B_a and a permutation [106] P_a:

- **B_a:** The same data structure used for the bit sequences for predicates and objects in the HDT Triples. It contains a 1 if the triple is contextualized, and a 0 if not.
- **P_a:** A data structure similar to ID sequences. It can only contain unique sequential IDs in the range $[1, |P_a|]$. P_a contains, for each contextualized triple, the position in A of the anchor in the Dictionary component. It provides the following operations in an efficient manner:
  - $\pi(i)$: returns the ID for a given position. This allows to retrieve in the Dictionary the anchor for the statement in that position.
  - $\pi^{-1}(i)$: return the position for a given ID. This allows to retrieve the position of the triple from the ID of its anchor.

Figure 9.8 shows the Triples component of HDTr for the running example with the underlying forest-of-trees representation and the actual implementation. Note that the IDs stored for the objects depends on the mapping used. If either the naive mapping or the same-id mapping are used, the IDs in the Triples component are the same as those in the Dictionary component for the same terms. If the smart mapping is used, the IDs in the Triples component are always the naive mapping IDs, but will be converted into same-ID mapping IDs when retrieving triples.

Conceptually, the permutation P_a does not store IDs, but makes a relation between the position of the triples and the position of the term in the anchor section. However, this is equivalent to storing the naive ID and using the operation $mapping_m(id, m')$ to transform to the same-ID mapping if necessary. While the concrete implementation of this transformation can vary, we will make use of the operation during the rest of the chapter for convenience.

9.3.4 Querying HDTr

HDTr supports all HDT querying operations (that is, it is possible to load a set of triples to HDTr and use it as if it was an HDT file). In order to allow querying for reified triples, the interface and internal operations are extended to support querying for quads. The element in the fourth position is used for the anchor.

Since anchors are unique and have one-to-one relation to triples, triple patterns with a known anchor[9] can be resolved by locating the triple and retrieving its values. This is done with the following operations:

[9]Triple patterns with known anchor are the following: (s,p,o,a), (?s,p,o,a), (s,?p,o,a), (s,p,?o,a), (?s,?p,o,a), (?s,p,?o,a), (s,?p,o,a), (s,?p,?o,a), and (?s,?p,?o,a).

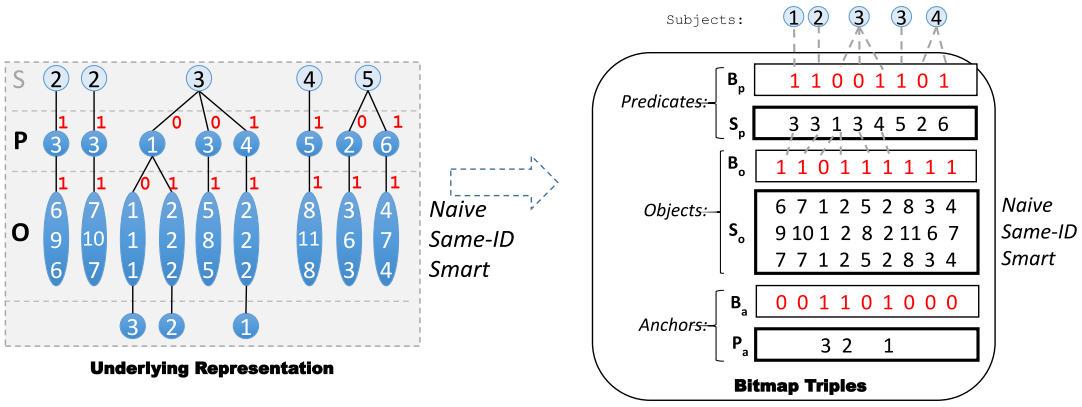


Figure 9.8: HDTr Triples

1. Obtain the corresponding IDs <id_s,id_p,id_o^d,id_a^d> for the known terms in the quad pattern using the $locate(element, role)$ operation.
2. If smart mapping is used, obtain the IDs for the object in the Triples component using $id_o^t = mapping_{same-id}(id_o^d, naive)$. If not, $id_o^t = id_o^d$.
3. If same-ID or smart mapping is used, obtain the IDs in the permutation using $id_a^t = mapping_{same-id}(id_a^d, naive)$. If not, $id_a^t = id_a^d$.
4. Retrieve the position of id_a^t in P_a of using the operation $pos_{P_a}(id_a^t) = \pi^{-1}(id_a^t)$.
5. Locate the position of the object using $pos_{S_o}(o) = select_1(B_r, pos_{P_a}(id_a^t))$.
6. Retrieve the position of the predicate for the given object position. This is done using $pos_{S_p}(p) = rank_1(B_o, pos_{S_o}(o))$.
7. Obtain the subject ID for the given predicate position: $id_s = rank_1(B_p, pos_{S_p}(p))$
8. Obtain the corresponding terms for the retrieved IDs using the $extract(id, role)$ operation.

If a variable is given for the position of the anchor, then it is necessary to retrieve the triples that match the first three positions of the quad pattern using regular HDT operations[10] and then obtain the anchor using the following operations:

- Obtain the position of the anchor within P_a using $pos_{P_a}(a) = rank_1(B_a, pos_{B_o}(id_o^t))$.
- Obtain position of the anchor in the Dictionary component using $id_a^t = \pi(pos_{P_a}(a))$.
- If smart or same-ID mapping is used, obtain the IDs for anchor in the Dictionary component using $id_a^d = mapping_{naive}(id_a^t, same - id)$. In not, $id_a^d = id_a^t$.
- Obtain the corresponding term for id_a^d using the $extract(id, role)$ operation.

[10]See Subsection 9.2.2. Note that the operation $mapping_{naive}(id_o, same\text{-}id)$ is necessary after step 3b if the smart mapping is used.

9.4 Evaluation

We developed prototypes of HDTr in Java[11] and C++[12]. In this section we evaluate the performance of the C++ version against HDT and four triplestores that support loading and querying quads: Virtuoso, Blazegraph, GraphDB, and Fuseki.

For the data we use the results of Chapter 7 "NELL2RDF: Reading the Web, Tracking the Provenance, and Publishing It as Linked Data", where we presented a tool to extract the beliefs of NELL and their metadata into RDF using different contextualization functions. NELL2RDF creates an RDF statement for each of NELL's beliefs, plus an optional contextual annotation for the statement including all its provenance information. To represent together the statements and their context, NELL to RDF uses a number of contextualization functions (namely, RDF Reification, N-ary relations, Singleton Properties, and NdFluents), as well as named graphs using n-quads. The number of statements in the contextual annotations is an order of magnitude bigger than the number of contextualized statements. Since we are interested in measuring the efficiency HDTr and the three different mappings managing the contextualized triples (*i.e.*, the results of the function $\mathsf{St}(\mathcal{G}, t)$ in a contextualization function, or its equivalent for other approaches), we generate versions of the datasets where the contextualized triples are created but the contextual annotations are not attached to them. In addition, we use the version of NELL2RDF with non-contextualized statements (that we call *vanilla*). This version will be used for comparisons against HDT.

We generate the RDF statements for 3 slices of NELL's beliefs (from iteration 1100): The full dataset, half the statements, and a quarter of the statements. The beliefs are randomized before taking a slice and creating the RDF versions. Table 9.1 shows the number of beliefs, triples, and quads for every slice, as well as their respective size in gigabytes (in n-triples and n-quads format).

Using this data, we explore the performance of HDTr with regards to loading time, data compression, and query time. We compare the effect of the three mappings in the three dimensions, as well as against the original HDT and the four selected triplestores when possible. This will allow to gauge the impact perspectives of HDTr and to gain insights about future improvements of both HDTr and its implementation.

All experiments are performed in a cluster. Each experiment is assigned 2 cores and 320 GB of RAM in a random machine with similar characteristics. To avoid as much skew as possible we repeat each experiment ten times.

9.4.1 Loading time

In this subsection, we compare loading and indexing time against HDT and the four selected triplestores: Virtuoso, Blazegraph, GraphDB, and Fuseki.

We perform two comparisons against HDT. In both cases, we serialize each RDF file ten times and calculate the average loading time.

First, we compare loading and indexing times of HDTr using n-quads files against load times of HDT using different contextualization approaches. The goal of this comparison is to measure the possible reduction of loading and indexing time of HDTr with regards to HDT when serializing contextualized statements. Table 9.2 shows the times to load each file into RDF, as well as the indexing time for each file (that is, the time required to create the additional structures of HDT-FoQ for each

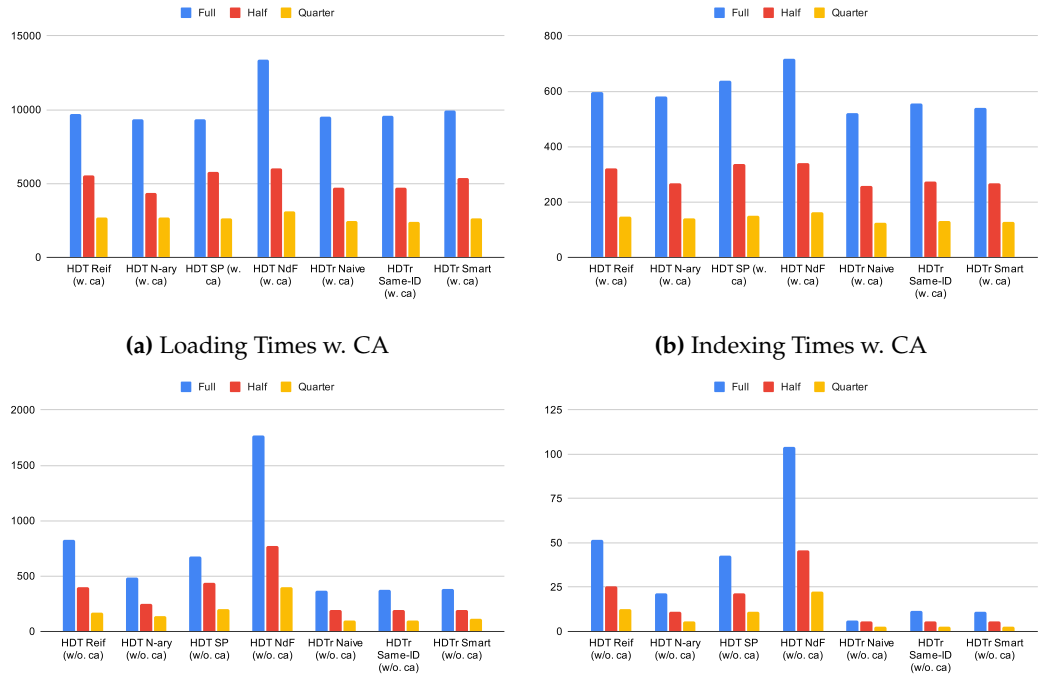
[11] https://github.com/jm-gimenez-garcia/hdt-2-java
[12] https://github.com/jm-gimenez-garcia/hdt-2-cpp

Table 9.1: Summary of NELL's slices (in million lines and GB)

File		Full Data		Half Data		Quarter Data	
		#	Size	#	Size	#	Size
Beliefs		30.7	25.6	15.4	12.9	7.8	6.5
Vanilla		40.7	6.0	23.5	3.4	13.4	1.9
N-Ary	*w. ca*	1,012.7	229.1	521.8	117.8	268.7	60.5
	w/o ca	56.8	11.4	29.9	6.0	15.5	3.1
NdF	*w. ca*	1173.3	254.0	600.7	130.0	308.3	66.6
	w/o. ca	217.5	36.3	108.7	18.1	55.1	9.2
Reif	*w. ca*	1,069.6	239.2	551.7	123.1	284.2	63.3
	w/o. ca	113.7	21.5	59.7	11.3	31.0	5.8
SP	*w. ca*	1,041.1	234.4	536.7	120.6	276.4	62.0
	w/o. ca	85.3	16.7	44.8	8.7	23.2	4.5
Quads	*w. ca*	984.0	224.6	506.6	115.4	260.8	59.3
	w/o. ca	28.1	6.9	14.7	3.6	7.6	1.8

Table 9.2: HDT and HDTr serialization times of contextualized statements (in seconds)

File		Full Data		Half Data		Quarter Data	
		HDT(r)	Index	HDT(r)	Index	HDT(r)	Index
HDT Reif	*w. ca*	9,723.6	596.7	5,578.5	320.8	2,683.1	148.6
	w/o. ca	830.1	51.9	399.4	25.4	172.1	12.5
HDT N-Ary	*w. ca*	9,366.2	582.6	4,341.7	267.3	2,709.7	140.9
	w/o. ca	488.2	21.6	247.5	10.9	135.1	5.6
HDT SP	*w. ca*	9,355.1	638.9	5,780.4	337.7	2,612.2	150.8
	w/o. ca	677.2	43.0	441.3	21.3	204.0	11.3
HDT NdF	*w. ca*	13,398.3	718.7	6,001.0	341.3	3,129.4	161.6
	w/o. ca	1,771.3	104.0	768.6	45.9	402.9	22.5
HDTr Naive	*w. ca*	9,506.8	521.3	4,693.2	258.4	2,439.9	124.0
	w/o. ca	369.6	6.0	190.6	5.6	99.1	2.7
HDTr Same-ID	*w. ca*	9,595.2	556.4	4,692.3	274.2	2,394.9	132.8
	w/o. ca	376.0	11.5	190.7	5.6	99.0	2.8
HDTr Smart	*w. ca*	9,961.7	540.5	5,370.6	267.7	2,663.8	127.8
	w/o. ca	382.5	11.3	196.7	5.5	111.5	2.8



(a) Loading Times w. CA

(b) Indexing Times w. CA

(c) Loading Times w/o CA

(d) Indexing Times w/o CA

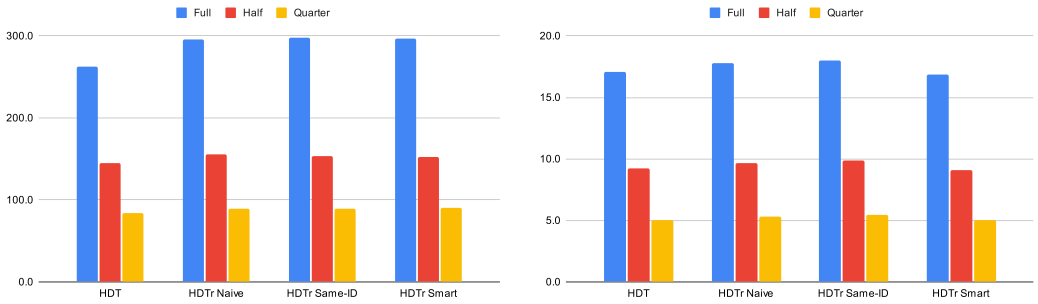
Figure 9.9: Serialization times of HDT and HDTr for contextualized statements (in seconds)

file). We see that loading times are similar when looking at files with contextual annotation (referred as *w. ca*), with the exception of NdFluents, that reports loading times around 40% longer than the rest. Indexing times, however, show that HDTr is faster than HDT for all contextualization approaches when creating the HDT-FoQ structures for their respective serializations. When looking at the files without contextual annotations (referred as *w/o. ca*) we see that HDTr shows significative improvements in both loading and indexing times. Figure 9.9 shows the comparison of loading and indexing times for all files.

Second, we compare loading time of HDTr against HDT when loading the vanilla files. This evaluation aims to measure the impact of the additional structures and loading operations of HDTr and identify possible improvements. We create HDT and HDTr files, as well as their indices, for the three NELL slices. Table 9.3 shows the loading and indexing time for HDT and the three versions of HDTr. We see

Table 9.3: HDT and HDTr serialization times of non-contextualized statements (in seconds)

	Full Data		Half Data		Quarter Data	
File	**HDT(r)**	**Index**	**HDT(r)**	**Index**	**HDT(r)**	**Index**
HDT	262.4	17.1	144.3	9.3	84.1	5.1
HDTr Naive	296.0	17.8	155.9	9.7	89.4	5.3
HDTr Same-ID	298.0	18.0	153.3	9.9	89.0	5.4
HDTr Smart	297.0	16.9	152.5	9.1	90.3	5.0



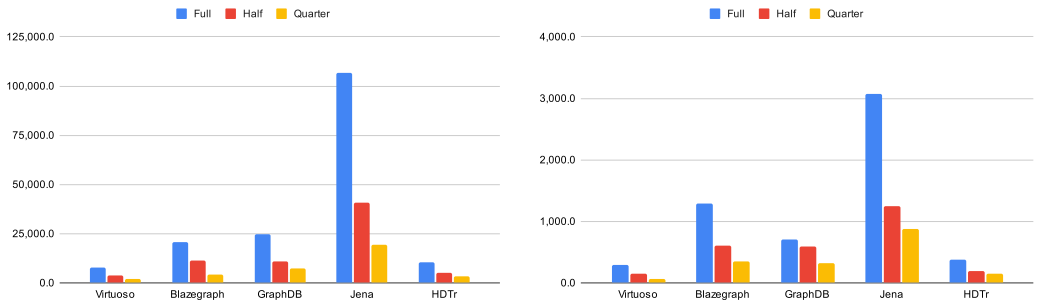
(a) Loading Times **(b)** Indexing Times

Figure 9.10: Serialization times of HDT and HDTr for non-contextualized statements (in seconds)

Table 9.4: Loading Time in Triplestores (in seconds)

		Full Data	Half Data	Quarter Data
Virtuoso	*w. ca*	7,927.8	3,752.9	1,829.3
	w/o. ca	294.8	143.8	67.9
Blazegraph	*w. ca*	20,659.2	11,400.7	4,081.8
	w/o. ca	1,288.8	608.7	345.1
GraphDB	*w. ca*	24,628.8	11,115.8	7,271.2
	w/o. ca	699.0	586.1	320.5
Fuseki	*w. ca*	106,833.0	40,887.4	19,213.5
	w/o. ca	3,070.2	1,244.8	878.0

that loading times of all versions of HDTr are increased by around 10% with regards to HDT, while indexing times remain similar. This matches the expectations, since loading data requires additional operations to check whether IRIs in subject or object positions are anchors of other statements. We predict, however, that these operations can be optimized to reduce the increment in loading time. We see that loading and indexing times are similar in all HDTr versions too. Figure 9.10 shows the comparison of loading and indexing times for all files.



(a) Data w. Contextual Annotations **(b)** Data w/o. Contextual Annotations

Figure 9.11: Loading time of HDTr and Triplestores Data (in seconds)

Finally, we compare HDTr against the four chosen triplestores when loading quads. Again, we load each file ten times and calculate the average load time in seconds. We compare this time against the aggregated time that HDTr takes to serialize the data and create the HDT-FoQ indexes. Table 9.4 shows the loading times for the triplestores for each file. We see that HDTr is slightly slower than Virtuoso, the fastest triplestore. Figure 9.11 shows the comparison of loading times between the four triplestores and the average of the three versions of HDTr.

9.4.2 Data compression

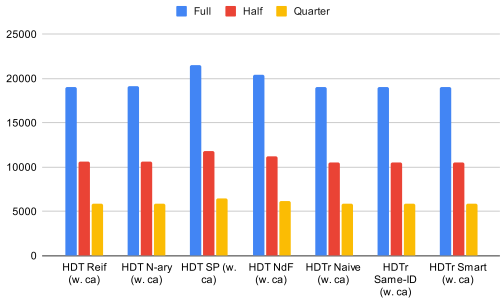
In this subsection we compare the space consumption of HDTr against HDT and the four triplestores when encoding reified triples.

HDTr allows to implicitly identify a triple by its position, which in turn allows for space savings in both the Dictionary and Triples component of HDTr with regards to HDT. This savings should be proportional to the number of triples that are reified and the number of triples each contextualization function introduces. The comparison against HDT allows us to quantify these space savings, while the comparison against the two triplestores show how HDTr compare against traditional solutions, and allows us to put into context the rest of the experiments that compare HDTr against them.

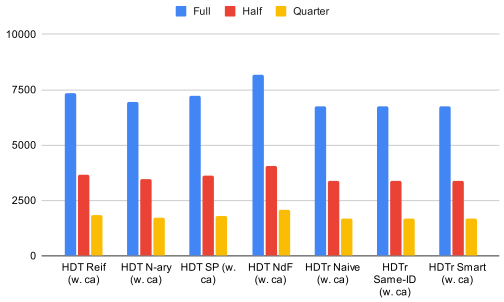
We create the HDT and HDTr serialization for each one of the different files, as well as their HDT-FoQ indexes. Table 9.5 shows the sizes for each HDT and HDTr file. We can observe that the size of the HDTr files with no contextual annotation

Table 9.5: HDT and HDTr sizes (in MB)

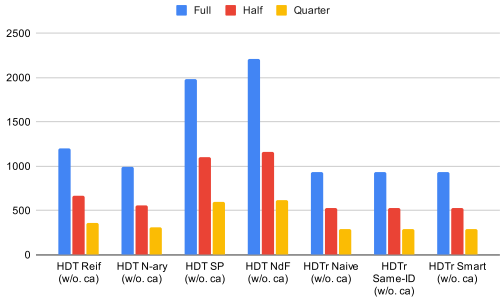
File		Full Data		Half Data		Quarter Data	
		HDT(r)	Index	HDT(r)	Index	HDT(r)	Index
HDT Reif	*w. ca*	19,072	7,351	10,649	3,675	5,870	1,831
	w/o. ca	1,198	759	664	385	363	192
HDT N-Ary	*w. ca*	19,127	6,959	10,615	3,480	5,887	1,735
	w/o. ca	995	299	563	155	314	79
HDT SP	*w. ca*	21,521	7,236	11,829	3,617	6,501	1,801
	w/o. ca	1,982	654	1,100	331	603	165
HDT NdF	*w. ca*	20,413	8,193	11,248	4,065	6,173	2,087
	w/o. ca	2,216	1,476	1,166	712	616	348
HDTr Naive	*w. ca*	19,083	6,745	10,587	3,370	5,840	1,680
	w/o. ca	936	101	526	56	292	30
HDTr Same-ID	*w. ca*	19,083	6,745	10,587	3,370	5,840	1,680
	w/o. ca	936	101	526	56	292	30
HDTr Smart	*w. ca*	19,083	6,745	10,587	3,370	5,840	1,680
	w/o. ca	936	101	526	56	292	30



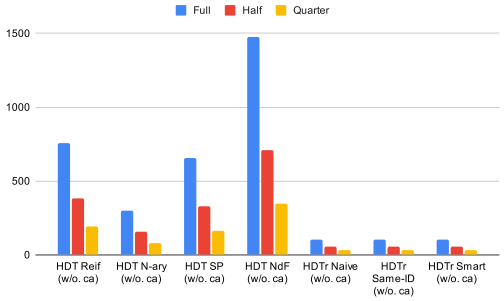
(a) HDT/HDTr sizes (w. CA)



(b) Index Sizes (w. CA)



(c) HDT/HDTr sizes (w/o. CA)

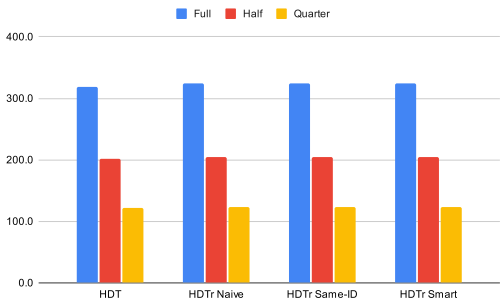


(d) Index Sizes (w/o. CA)

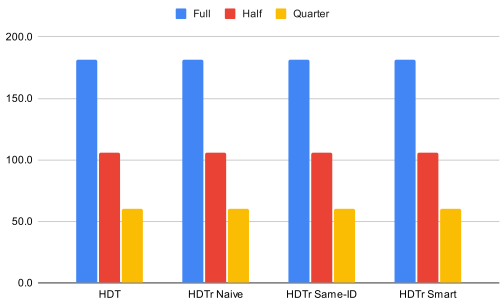
Figure 9.12: Size of HDT and HDTr files and their indexes (in MB)

Table 9.6: Size of HDT and HDTr files for non-contextualized statements (in MB)

File	Full Data		Half Data		Quarter Data	
	HDT(r)	Index	HDT(r)	Index	HDT(r)	Index
HDT	319.1	181.6	202.1	105.7	122.2	60.1
HDTr Naive	323.9	181.6	204.9	105.7	123.8	60.1
HDTr Same-ID	323.9	181.6	204.9	105.7	123.8	60.1
HDTr Smart	323.9	181.6	204.9	105.7	123.8	60.1



(a) HDT files

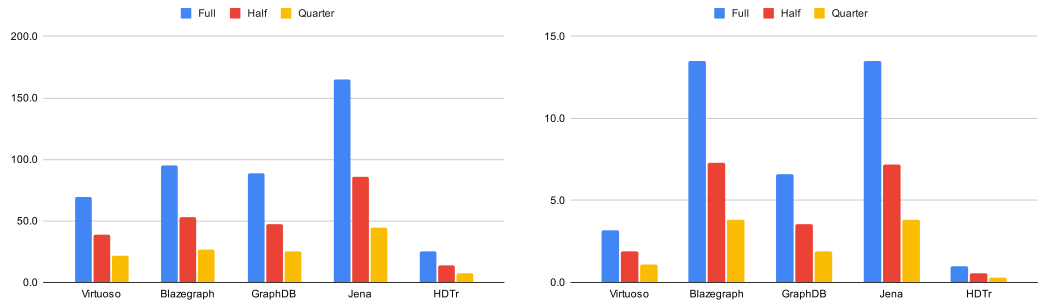


(b) Index files

Figure 9.13: Sizes of HDT and HDTr files for non-contextualized statements (in MB)

Table 9.7: Data size in Triplestores (in GB)

		Full Data	Half Data	Quarter Data
Virtuoso	*w. ca*	69.4	38.5	21.4
	w/o. ca	3.2	1.9	1.1
Blazegraph	*w. ca*	95.1	53.4	27.0
	w/o. ca	13.5	7.3	3.8
GraphDB	*w. ca*	88.7	47.2	25.2
	w/o. ca	6.6	3.6	1.9
Fuseki	*w. ca*	165.2	85.8	44.6
	w/o. ca	13.5	7.2	3.8



(a) Data w. Contextual Annotations

(b) Data w/o. Contextual Annotations

Figure 9.14: Size of HDTr files and Triplestores Data (in GB)

is lower than the HDT files with any contextualization approach, ranging 6% for n-ary relations and 237% for NdFluents. The difference for the indexes is even more pronounced, going from 263% to 1461% for the same approaches. However, if we look at the files with contextual annotations we see that the HDTr and the HDT files for RDF reification and N-Ary relations have similar size, and the space savings for the singleton property and NdFluents have been reduced. Figure 9.12 shows a comparison of these sizes.

In the second experiment we compare space requirements for HDTr and HDT when serializing the vanilla files. We expect to see a slight increase in space due to the additional structures that HDTr incorporates. Table 9.6 shows the sizes of the HDTr and HDT files, along with their indexes, for the three NELL slices. We see that the size of all three HDTr files are almost equal, the difference being a matter of bytes. When compared with the HDT files for the same data, the size increment is between 1.3% and 1.5%. The size of the indexes is identical for all four files of the same slice. Figure 9.13 shows the size comparison of all files.

Finally, we compare space requirements of HDTr against the four chosen triplestores when storing quads. We compare the space the triplestores use to store the data against the aggregated size of the HDTr files plus their HDT-FoQ indexes. Table 9.7 shows the storage size for the triplestores for each file. We see that HDTr provides space savings between 3.5 and 5 times, depending on the triplestore. Figure 9.14 shows the comparison of storage size between HDTr and the four triplestores.

9.4.3 Query time

In this subsection we compare the time that HDTr takes to retrieve triple and quad patterns against HDT and the four selected triplestores. For this, we use the HDTr files generated and the data loaded in the triplestores in the previous subsections.

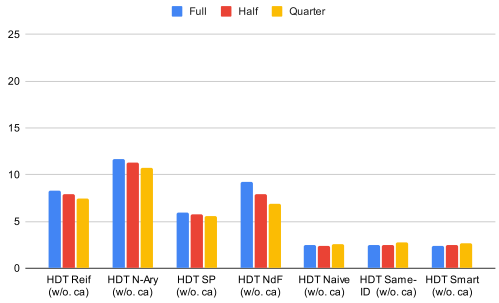
We first compare the retrieval times of statements and their anchors by HDTr and the HDT files using different contextualization approaches. In the case of HDTr that means using a quad pattern, while in the HDT files a different set of patterns is needed, depending on the contextualization approach that they encode. For example, to retrieve the the statements and their anchor that have a subject **s**, we need to use the quad pattern `(s???)` in HDTr, while in the HDT file for the data using the RDF reification approach we need use the triple pattern `(? rdf:subject s)` to get the anchors of the statements and then the patterns `(a rdf:predicate ?)` and `(a rdf:object ?)` for each anchor **a**. Table 9.8 shows the types patterns necessary to retrieve a statement and its anchor for each contextualization approach.

Since we do not have a SPARQL engine to perform joins and query optimization for the HDT files, we decide to estimate a lower bound for the time needed to retrieve the necessary triples. To do that, we calculate the average retrieval time per triple for each triple pattern, and add together the times for all the needed triple patterns. Note that this is an optimistic estimation that does not take into account the cost of the joint operations. We then compare this time against the average retrieval time of a quad pattern type for the same statement in HDTr. To calculate the average retrieval time for triple and quad patterns, we extract all possible triple patterns that fit each pattern type for each HDT or HDTr file, query the corresponding file ten times, and calculate the average. If for any given pattern type we extract less that 10,000 patterns, we query the patterns again until we obtain at least 10,000 time measures. In Figure 9.15 we show the average retrieval times for the quad patterns and sets of triple patterns. We see that for five out of eight cases HDTr has lower retrieval time than any estimation for the HDT files, whereas for the three remaining cases it shows comparable results to the best estimations for HDT.

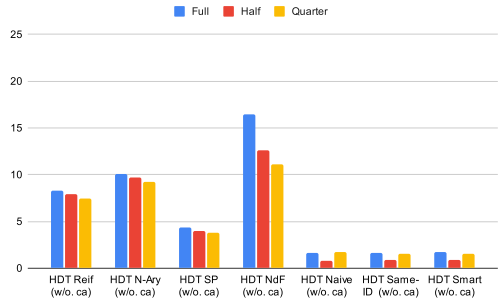
We then compare triple pattern retrieval of single triples for HDTr against HDT

Table 9.8: Types of patterns needed to obtain a triple and its context in each contextualization approach

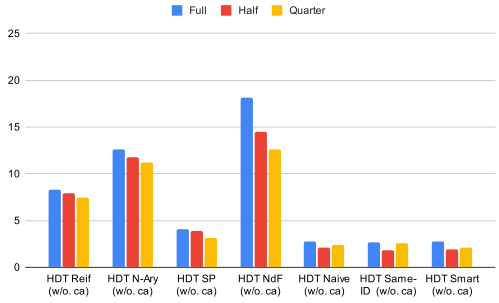
Quad	s???	?p??	??o?	sp??	s?o?	?po?	spo?	???a
Reif	?po	?po	?po	?po	?po	?po	?po	sp?
	sp?	sp?	sp?	spo	sp?	spo	spo	sp?
	sp?	sp?	sp?	sp?	sp?	sp?	spo	sp?
N-Ary	s??	sp?	??o	sp?	s??	sp?	sp?	??o
	?po	sp?	?po	sp?	s??	sp?	sp?	?po
	sp?	?p?	sp?	sp?	sp?	?po	sp?	sp?
	sp?	sp?	?po	sp?	sp?	?po	spo	sp?
SP	s??	?po	??o	sp?	s?o	?po	?po	?p?
	sp?	?p?	?po	?po	sp?	?po	spo	sp?
NdF	?po	?p?	sp?	?po	?po	spo	?po	?po
	s??	sp?	??o	sp?	?po	?po	sp?	s?o
	sp?	sp?	sp?	sp?	s?o	sp?	spo	
		sp?	sp?	sp?	sp?	sp?	sp?	



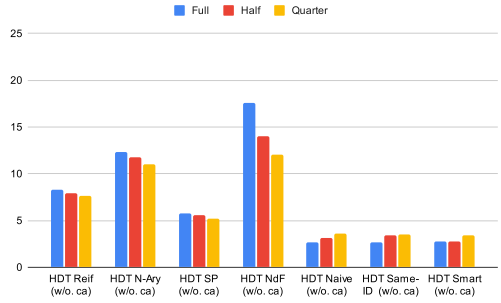
(a) s???



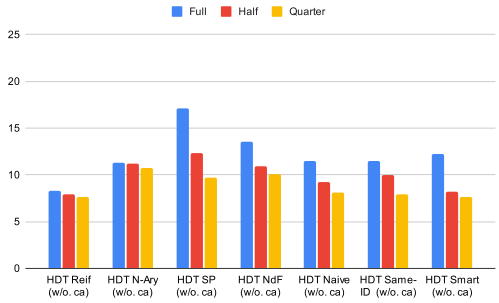
(b) ?p??



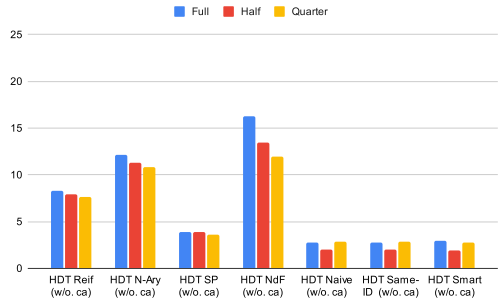
(c) ??o?



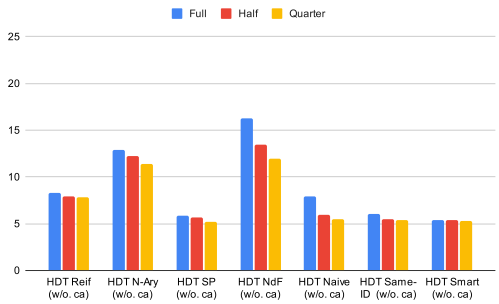
(d) sp??



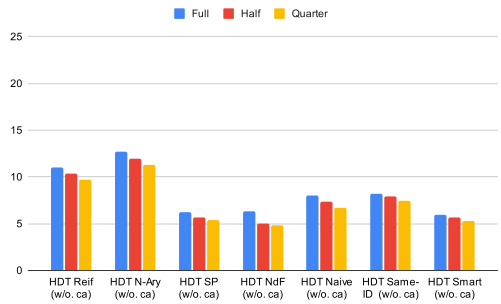
(e) s?o?



(f) ?po?

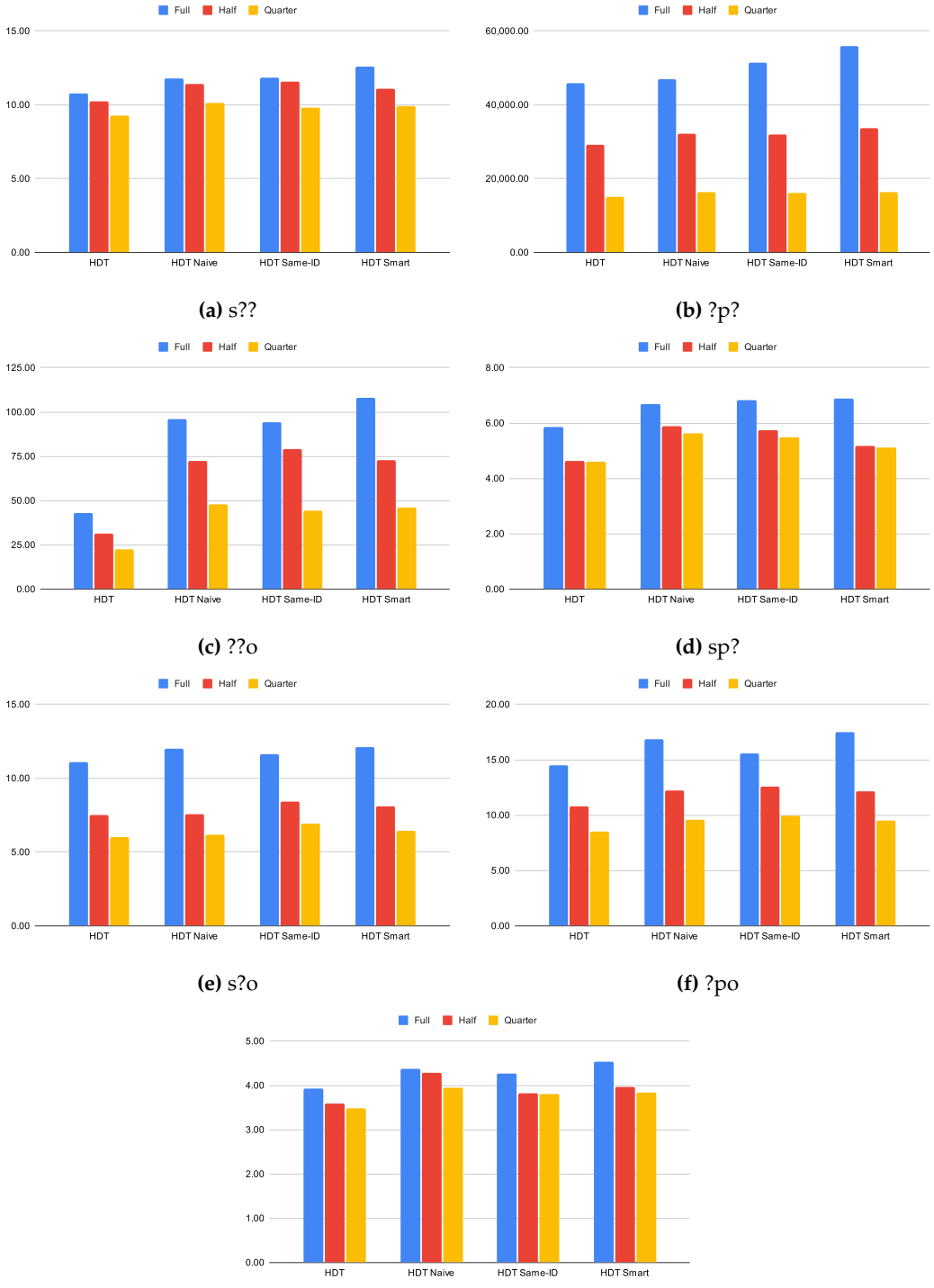


(g) spo?



(h) ???a

Figure 9.15: HDTr vs HDT estimated triple retrieval time for contextualized statements (in microseconds)



(a) s??

(b) ?p?

(c) ??o

(d) sp?

(e) s?o

(f) ?po

(g) spo

Figure 9.16: HDTr vs HDT estimated triple retrieval time for non-contextualized statements (in microseconds)

using non-contextualized statements. HDTr adds some complexity in the HDT structures. This is specially relevant in the Dictionary component, since any query operation needs to check whether the term is an anchor or not, go the appropriate subdictionary and, depending on the mapping, calculate the external ID. Hence, we expect to see that the time to retrieve a single triple is increased in HDTr with regards to HDT. Our first experiment intends to measure this impact. Figure 9.16 shows the retrieval times of all possible triple patterns on HDT and HDTr. We see that for all triple patterns except `??o` HDTr is slightly slower, which fits the expectations. For the patter `??o`, HDTr is almost twice as slow. The reason for that is that HDT does not need to retrieve the position of the object in S_o, but HDTr needs to find this position in order to check if the triple is reified and obtain its anchor in that case. This requires to find the object associated with the predicate position in S_p using binary search.

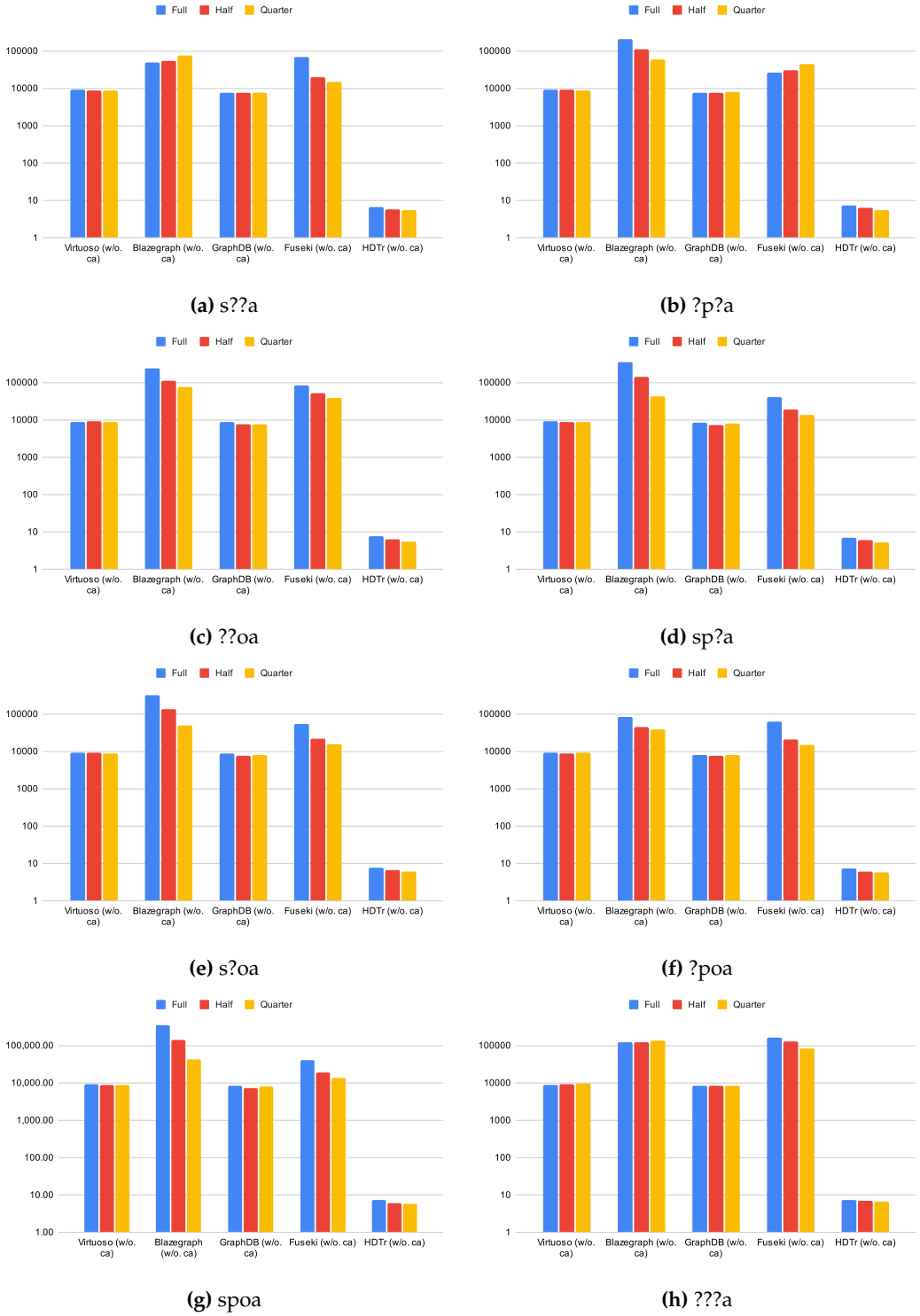
We finally perform an experiment to compare HDTr performance when retrieving quad patterns against the four chosen triplestores. In this case, we load the quad files of the NELL slices in the triplestores and compare the times for quad-pattern retrieval. For each pattern type, we select 10,000 quad patterns to query them ten times against each triplestore. Figure 9.17 shows the query retrieval time for the four chosen triplestores: Virtuoso, Blazegraph, GraphDB, and Jena, as well as the average retrieval time of the three versions of HDTr, for the data without contextual annotation. We see that HDTr is between three and four orders of magnitude faster, depending on the pattern and the triplestore. The comparison using the data with contextual annotations yields similar results. This results are consistent with other evaluations of triple pattern retrieval for HDT against triplestores.

9.4.4 Summary and Discussion

Here we summarize and discuss the results of the evaluations described in this section.

When comparing the serialization of contextualized triples using HDTr against the HDT files of different contextualization approaches, we see similar results in both loading time and rate of compression. When loading the slices without contextual annotations we see moderate gains in both time and compression. However, when adding the statements of the contextual annotation (which increases the number of triples tenfold), both loading time and compression rate are similar in HDTr and the HDT files. This suggests that the current approach yields benefits in these areas when there is a big proportion of contextualized statements in the dataset; but when the number of contextualized triples is small with regards to the total number of triples, the overhead of the additional structures and operations of HDTr can offset the savings per contextualized triple that it achieves. The evaluation for non-contextualized triples confirms this conclusion: both loading time and size are slightly increased in HDTr with regards to HDT. We hypothesize, though, that these overheads can be reduced in order to achieve similar results for datasets with a low number of contextualized triples. We describe how this can be achieved down below.

During the serialization process, HDTr creates the dictionary in two phases. First, it reads all the statements, assigning terms to sections S_T, P_T, and O_T for the subject, predicate and object positions, and to section U_A for the anchors. Then, in a second phase, every term in triples dictionary is searched in section U_A, to move it to the appropriate section in the anchor dictionary if necessary. Changing this second phase so it is the terms in U_A that are searched in the triples dictionary would reduce



(a) s??a **(b)** ?p?a

(c) ??oa **(d)** sp?a

(e) s?oa **(f)** ?poa

(g) spoa **(h)** ???a

Figure 9.17: HDTr vs Triplestores Quad Retrieval Time (in microseconds)

serialization time when the number of non-contextualized triples is large, without affecting the serialization time when all the statements are contextualized.

To encode the Triples component, the current approach uses a bit sequence with a length equal to the number of triples to identify those that are reified. Another approach we are considering is to separate the triples component in two sections: One with the contextualized triples, and a second one for the non-contextualized triples. This would allow to remove the bit sequence, saving space and loading time.

When comparing triple pattern retrieval, we see that HDTr outperforms the HDT for any contextualization approach for five out of the eight patterns, while achieving comparable results to the better-performing contextualization approaches in HDT for the other three patterns. It is important to remember that the times for the HDT files are optimistic lower bounds that do not take into account the cost of joins, and we expect the improvements in HDTr to be higher in practice.

However, the evaluation of triple pattern retrieval for non contextualized triples against HDT reveals that there is still room for improvement. HDTr shows higher retrieval times for all patterns. For the majority of patterns the difference is small, but for the pattern "`??o`" the retrieval time is almost double for HDTr. The reason is that HDT does not need to retrieve the position of the object to return the triple, but HDTr needs to do it in order to check if the statement is reified, and retrieve the anchor in that case.

The time to retrieve triples with this pattern could be improved by two means. First, separating the Triples component into contextualized and non-contextualized statements would allow to skip this operation for the non-contextualized triples. Second, when the operation is needed, adding an additional structure to HDT-FoQ to link object values with their positions would reduce its response time. More research is necessary to design and implement an optimal structure for this purpose.

The last comparison of HDTr against HDT, the creation of the HDT-FoQ structures, shows equivalent times ans space requirements for non-contextualized triples, and better times and space requirements for all cases when comparing against the HDT files for contextualized statements.

Finally, the evaluation of HDTr against triplestores for loading time, space requirements, and triple pattern retrieval shows compelling results. Loading times are comparable to those of Virtuoso, the fastest triplestore in the evaluation. Space requirements shows that HDTr provides space savings of at least 350%. Pattern retrieval times shows that HDTr is between three and four orders of magnitude faster than all triplestores when querying for contextualized triples.

9.5 Conclusions and Future Work

This work presents HDTr, an extension of HDT to store reified triples with a model-agnostic binary representation that is compact and queryable. It provides two prototype implementations in Java and C++ of HDTr and evaluates the C++ version against HDT and four triplestores: Virtuso, Blazegraph, GraphDB, and Jena Fuseki.

HDTr assumes that the contextualization of the statements follows the reification paradigm. This assumption conforms with the majority of existing real-world datasets (such as Wikidata) and contextualization approaches (such as RDF*). This makes HDTr a suitable solution to manage contextualized statements in existing data using the commonly used contextualization approaches. Using this assumption HDTr is able to make a connection between the contextualized statements in

the Triples component and the term for their anchor in the Dictionary component. This allows to efficiently store reified triples and their contexts without the need to deal with the additional triples introduced by a contextualization function or the syntax extensions of other approaches.

The evaluation of HDTr has shown that HDTr outclasses HDT and existing triplestores when retrieving contextualized triples. HDTr is also competitive with regards to loading time and compression rate, providing similar or better results than the best alternative. The evaluation against HDT for non-contextualized statements shows slightly worse results, but suggest some improvements to HDTr that can close the gap. These improvements include a faster approach to create the Dictionary component, separating the Triples component in separate sections for contextualized and non-contextualized statements, and a new index structure to relate object values with their position in the object sequences.

HDTr is a first step towards managing contextualized triples in practice. Future work remains in order to achieve such a tool. Down below we describe the most relevant lines of future work.

Notwithstanding the usefulness of HDTr with current data and approaches, we have shown in Part I that other solutions exist that better preserve the semantics of a set of statements when these are modeled within a context. These solutions do not follow the reification paradigm, and thus, HDTr is not able to efficiently serialize their results. The next step in achieving a general tool to manage contextualized triples is extending HDTr to support Nd*.

HDTr is able to manage contextualized statements independently of the specific approach used to represent them. However, currently HDTr is only able to read and retrieve statements in the form of quads. An intermediate tool that allows to read and retrieve statements represented using different contextualization approaches needs to be addressed. Preliminary work has already been done in the form of two tools that transform a set of triples (respectively, a basic graph pattern) in a contextualization approach to another set or triples (respectively, another basic graph pattern) using another contextualization approach.

Additionally, we have identified a number of improvements or added functionality that can be added to HDTr. We present the most relevant down below.

Currently, statements of the graph that is being contextualized and those of the contextual annotations are serialized together in the Triples component. Triples of the contextual annotation are usually not contextualized. Thus, separating the two sets of triples would eliminate the bit sequence and reduce the size of the permutation used to relate the statements with their anchor.

When representing a contextualized statement in RDF, it is necessary to introduce a fresh term that serves as statement anchor (or an equivalent elements for approaches that extend the syntax of RDF). In practice, these terms are usually just placeholders for the link between the statement and its contextual annotation, and there is no interest in knowing the specific term for each statement. Extending HDTr with the option of not storing the anchor terms, and using a function to generate a unique term for each anchor at query time, would increase space savings. This would be specially useful for approaches where this function is given by the approach (such as RDF*).

The final step to develop a tool that can read, manage, and query contextualized statements in a model-agnostic manner would be to integrate HDTr in a tool that can serialize contextualized statements in any supported contextualization approach, as well as being able to answer SPARQL queries that use one of those contextualization approaches. We have started work on converting contextualized statements and

basic graph patterns from one contextualization approach to another, but it is still in early stages of research.

Finally, preliminary work has been done to extend HDTcat [30] to merge HDTr files. HDTcat is a tool that allows to merge two existing HDT files without the need to decompress them beforehand. HDTcat allows to create larger HDT files with the same resources by splitting the data into smaller chunks, serializing them into HDT, and progressively merging them. We have created a prototype version of HDTrCat that can merge earlier versions of HDTr, but it needs to be adapted to the last version.

References

[1] Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets with the VoID vocabulary, W3C (2011). URL: https://www.w3.org/TR/void/

[6] Berners-Lee, T., Connolly, D.: Notation3 (N3): A readable RDF syntax, W3C (2011). URL: https://www.w3.org/TeamSubmission/n3/

[14] Briandais, R.D.L.: File searching using variable length keys. In: Johnson, R.R. (ed.) Papers presented at the the 1959 western joint computer conference, IRE-AIEE-ACM 1959 (Western), San Francisco, California, USA, March 3-5, 1959, pp. 295–298. ACM (1959).

[16] Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C (2004). URL: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[23] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. Journal of Web Semantics 3(4), 247–267 (2005).

[28] Cyganiak, R., Wood, D., Lanthaler, M.: RDF 1.1 Concepts and Abstract Syntax, pp. 263–270. W3C (2014). URL: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[30] Diefenbach, D., Giménez-García, J.M.: HDTCat: Let's make HDT generation scale. In: Pan, J.Z., Tamma, V.A.M., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) The semantic web - ISWC 2020 - 19th international semantic web conference, athens, greece, november 2-6, 2020, proceedings, part IILecture notes in computer science, pp. 18–33. Springer (2020).

[44] Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary RDF representation for publication and exchange (HDT). Journal of Web Semantics 19, 22–41 (2013).

[45] Fernández, J.D., Martínez-Prieto, M.A., Polleres, A., Reindorf, J.: HDTQ: Managing RDF Datasets in Compressed Space. In: The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings, pp. 191–208 (2018).

[46] Frey, J., Hellmann, S.: MaSQue: An Approach for Flexible Metadata Storage and Querying in RDF. In: 13th International Conference on Semantic Systems Proceedings (SEMANTiCS 2017) - Posters & Demonstrations TrackSEMANTiCS '17, (2017).

[57] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC), pp. 638–654. Springer, Cham (2017).

[67] Hartig, O.: Foundations of RDF* and SPARQL* (An Alternative Approach to Statement-Level Metadata in RDF). In: Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017. (2017).

[68] Hartig, O., Champin, P.-A., Kellogg, G., Seaborne, A., Arndt, D., Broekstra, J., DuCharme, B., Lassila, O., Patel-Schneider, P.F., Prud'hommeaux, E., Thibodeau, T., Thompson, B.: RDF-star and SPARQL-star, W3C (2021). URL: https://w3c.github.io/rdf-star/cg-spec

[74] Hernández-Illera, A., Martínez-Prieto, M.A., Fernández, J.D., Fariña, A.: i*HDT*++: improving HDT for SPARQL triple pattern resolution. Journal of Intelligent & Fuzzy Systems 39(2), 2249–2261 (2020).

[75] Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 web ontology language primer, W3C (2012). URL: https://www.w3.org/TR/owl2-primer/

[95] Martínez-Prieto, M., Arias, M., Fernández, J.: Exchange and Consumption of Huge RDF Data. In: Proc. of ESWC, pp. 437–452 (2012).

[96] Martínez-Prieto, M.A., Brisaboa, N.R., Cánovas, R., Claude, F., Navarro, G.: Practical compressed string dictionaries. Information Systems 56, 73–108 (2016).

[106] Munro, J.I., Raman, R., Raman, V., Rao, S.S.: Succinct representations of permutations and functions. Theor. Comput. Sci. 438, 74–88 (2012).

[113] Pelgrin, O.P., Hose, K., Galárraga, L.: TrieDF: Efficient in-memory indexing for metadata-augmented RDF. In: Proceedings of the 7th Workshop on Managing the Evolution and Preservation of the Data Web, CEUR Workshop Proceedings (2021).

Conclusions to Managing the Context of Statements

In Part I we have seen different approaches to contextualize statements in formal systems. Some of them extend the syntax and/or semantics of the language, while others aim to represent the context within the constraints of the language itself. In both cases, however, existing approaches fail to preserve the semantics of the original statements when putting them in a context. We focused on how contextual information can be added to statements without extending the logic system. We formalized the process in what we call *contextualization function*, defined properties that a contextualization function can have with regards to how it preserves the semantics of the original statements, study existing approaches that conform to a contextualization function under these properties, and proposed *Nd**, a family of contextualization functions that better respect these properties than any existing alternative.

In Part II we have explored how contextual information, whether implicit or explicit, about existing data can be identified and captured. We extracted implicit contextual information about trust, reuse, and quality of publicly available RDF datasets and made it explicit. This work underscores the importance of being able to properly represent and manage contextual information.

In this part, we have focused on managing contextualized data in practice. We have explored research questions **R6** *(How to efficiently manage contextualized data, independently of the concrete representation used to model it?)*. We have made a first step in this direction by proposing HDTr, a binary serialization format that extends HDT to store and query contextualized RDF statements compatible with the most commonly used contextualization approaches. We have evaluated HDTr against HDT and four triplestores using a number of contextualization approaches. The results of this evaluation have also contributed in examining research question **R4** *(How does a contextualization function affect the efficiency with regards to how the data and their context are stored and queried?)*. This work is materialized in two contributions:

- **C6:** An evaluation of how different contextualization approaches affect the data compression and data retrieval using HDT, a well-known binary serialization format with query capabilities for RDF. This contribution corresponds to the research question **R4**.
- **C7:** HDTr, a binary serialization of RDF that extends HDT to allow to manage contextual data in an efficient way, compatible with the majority of the existing contextualization approaches. This contribution corresponds to the research question **R6**.

HDTr is a step towards a general tool that allows to store, manage, and query contextualized RDF statements in a model-agnostic manner. HDTr is intended to be the core of such tool, storing and retrieving the triples. Additional components to be able to read, export, and query contextualized statements are other essential parts

of this tool. While preliminary work in these areas has been done during this thesis, the results are not mature enough to be presented in this dissertation.

Part IV

Conclusions and Afterword

Conclusions and Future Work

Summary of the Contributions

During this thesis, we have explored the notion of context in knowledge representation: How to represent the context of a set of statements while preserving their semantics, how to capture existing contextual information about data into contextualized statements, and how to efficiently manage the statements and their context together. Along this journey, we have worked on several research questions and have proposed corresponding contributions throughout the different parts of the thesis as follows:

In *Part I: Formalizing the Context of Statements* we have focused on the formal representation of statements within a context. We have seen that, in the Semantic Web, existing approaches reify the statement in a fresh term and link it with the statement and with the context. How they link the statement with the term falls in two different solutions: They either extend the syntax of the logical language, or they represent the statement with a new set of statements, in which all the terms of the original statement and the new fresh term that reifies it appear. However, most solutions disregard the semantics of the original set of statements, and those that do not, are not able to properly isolate the validity of a statement within its context. In this part we have focused on representing the statements and their context without extending the syntax of the logical language, exploring the following research questions:

- **R1:** *How to express formally the process needed to add contextual information to a statement or set of statements?* While a number of approaches exist to contextualize statements in a logic system, there was no formal definition of what contextualizing a statement means. Throughout the chapters, we have progressively built and refined our first contribution:
  - ***C1**: The definition and formalization of Contextualization Function*, a functions that maps a set of statements and a context to another set of statements. This definition is provided in several formal systems and is progressively refined along the chapters. We see that a contextualization function introduces two terms called the *statement anchor* and the *context anchor* that are connected in the result of the function. Informally speaking, the first one represents the statement within the context, while the second represents the context itself. For existing solutions, the statement anchor is the reification of the statement, which means that it has a one-to-one relation with the statement it represents.
- **R2:** *What are the formal properties of a contextualization process with regards to how it affects the semantics of the resulting data?* Once we have a formal definition of contextualization function, we need to be able to objectively evaluate how one such function behaves with regards to the semantics of its resulting statements. We have addressed this question by proposing our second contribution:

  - ***C2:*** *The formal definition of properties of a contextualization function with regards to how it preserves the semantics of the original set of statements.* These properties describe how the inferences of the original statements are preserved, and how statements in a context affect the validity of statements in other contexts.

- **R3:** *How to represent contextual information in order to preserve as much semantics as possible from original formalized data?* With the results of contributions *C1* and *C2*, now we are equipped with the tools to analyze existing contextualization functions, evaluate them with regards to how well they preserve the semantics of the original data and how they are able to separate their contexts, and propose new contextualization functions that better follow theses properties. This work has come to fruition in our third contribution:

  - ***C3:*** *The family of contextualization functions* Nd*. Nd* is based on the principle of, instead of reifying statements into terms, contextualizing the terms that appear in the original statement (conceptually, taking a slice of the term that exists within the context). That means that the statement anchor of a statement is not unique, and in general is related with a context or a combination of contexts. Nd* can be instantiated in different contextualization functions where a different subset of terms is contextualized (individuals, relations, concepts, or a combination of them). We have proposed and evaluated a number of these instantiations in First Order Logic (in Chapter 1), Description Logics (in Chapter 2), and Semantic Web formalisms (in Chapters 3 and 4).

In *Part II: Capturing the Context of Statements* we have put our attention on existing real-world data, exploring the following research question:

- **R5:** *How to capture existing contextual information that exists implicitly in non-formalized data when transforming it into formal statements?*. We have examined this question thorough three different perspectives: extracting contextual information about trust, reuse, and quality of existing LOD datasets (in Chapters 5 and 6; transforming a dataset with huge quantities of provenance information into RDF using different contextualization approaches (in Chapter 7; and proposing a first step towards capturing contextual information in web tables and transforming it into RDF (in Chapter 8). All in all, this work has led to the contributions:

  - ***C4:*** *Generation and publication of large data sets containing explicit contextual information from non-formalized data with implicit context*. These include 2 kinds of datasets: (1) structured datasets about the trust, reuse, and quality contexts of real-world openly available datasets; and (2) NELL2RDF, that includes the contents of NELL's[13] beliefs and their provenance expressed as contextualized RDF statements using different contextualization approaches.
  - ***C5:*** *A generic approach to capture contextual information from relational tables and transform it into RDF contextualized statements*. This is a semi-automated approach to read Web tables and capture the facts it contains

[13]NELL (Never-Ending Language Learner) that continually reads the Web and learns new facts, called beliefs. It includes a lot of contextual meta-data about the provenance of the beliefs.

and their context and represent them in using a number of contextualization approaches. This contribution shows that capturing and representing the context of structured data can be generalized.

In *Part III: Managing the Context of Statements* we set our goal on managing contextualized RDF statements in practice. We envision a tool that can read contextualized statements in any given contextualization approach, stores and manages the statements in a model-agnostic manner, and is able to query them using the same or other arbitrary contextualization approach. This part contains a unique chapter (Chapter 9) that makes a first step towards such tool, exploring the following research questions:

- **R4:** *How does a contextualization function affect the efficiency with regards to how the data and their context are stored and queried?* In order to investigate this question, we have used NELL2RDF's contextualized statements (part of the results of contribution *C4*) to serialize the result of different contextualization approaches into HDT. With this we have produced the following contribution:
  - ***C6:*** *An evaluation of how different contextualization approaches affect the data compression and data retrieval using HDT, a well-known binary serialization format with query capabilities for RDF*. In this evaluation we measure serialization time and compression rate for each HDT serialization of different slices of NELL contextualized with different approaches, and we estimate a lower bound for the response time of each of such serializations.
- **R6:** *How to efficiently manage contextualized data, independently of the concrete representation used to model it?* To address this research question, we have focused on the most generally used contextualization approaches: those based on reifying the statements. This allows us to make assumptions to extend HDT in order to implicitly relate the statements with their statement anchor. With this we have proposed our final contribution:
  - ***C7:*** *HDTr, a binary serialization of RDF that allows to manage contextual data in an efficient way independently of the representation chosen to model the context*. HDTr extends HDT with additional structures to identify the statement anchors and relate them to their statements. We suggest three different implementation alternatives and compare them among themselves, against the HDT serializations of the contextualization functions, and a selection of triplestores. HDTr shows the best results in terms of compression, serialization time, and query retrieval time, against all other alternatives.

Limits and Perspectives

While this work represents some advances in the field of context in knowledge representation, we are also well aware of limits it contains and we can draw some perspectives that remain or that arise from our work. While detailed limits and perspectives have been suggested in each chapter, here we give a bird's eye view of the limits and perspectives of the overall work presented in this dissertation.

In Part I we have studied how the semantics of a set of statements can be preserved and isolated within one context. We have scratched the surface of combining statements in multiple contexts, but more work is needed to identify the requirements of the data and the contextual annotations to conform to the desired properties, as well as to be able to formalize relations among contexts (such as contexts inside contexts, contexts that are the intersection or disjunction of other contexts, etc.).

We presented Nd*, a family of contextualization based on the paradigm of contextualizing terms of the statements, as well as some of its instantiations, each of which contextualizes a different subset of terms. We showed that these contextualization functions better preserve and separate the semantics of the original statements, but there are some differences between different members of this family. Future work is needed to perform a systematic comparison of each instantation of Nd* in each formal language.

In Part II we extracted implicit contextual information in existing datasets in the LOD cloud. We also presented some results about trust, reuse, quality, and their relations, in openly available datasets. These contributions were limited by the available data and selected quality metrics. This work can be a starting point on the research about trust and reuse of openly LOD datasets and the relevance of the quality metrics that can be measured on them.

We also presented NELL2RDF, a big dataset of contextualized statements based on a real-world dataset of beliefs and their provenance, as well as the process to generate it. This process is currently semi-automated and requires human intervention, which has led to it not been updated. Updating the results and making them available, as well as fully automatizing the approach are two future tasks that would be useful for the contribution.

Finally, we presented a contribution to semi-automatically read facts and their context from web tables and represent them in contextualized RDF statements. This is a proof of concept to showcase the possibility of a general solution to capture and represent contextual facts. It is again a starting point to a line of research to study this proposal in much deeper detail.

In Part III we have presented HDTr. HDTr is but a step in the direction of a general tool to manage contextualized statements with independence of the contextualization approach used to model them. More work needs to be done in order to extend HDTr to support other contextualization approaches, as well as in other components needed for such a tool to read statements and queries that conform to different contextualization approaches.

In addition, we have identified a number of areas of improvement in HDTr: data compression can be improved by organizing the Triples component in two sections, one for contextualized statements and other for non contextualized statements; serialization time can be improved by optimizing the process to identify if terms in triples are anchors of another statement; and create an additional index to link object values with positions in the object sequences.

Other possible future lines of work with HDTr include integrating it within a SPARQL query engine, extending it to allow updates, or extending HDTcat (a tool to merge HDT files without decompressing them that allows serialize big HDT files with reduced computing resources) to be able to merge HDTr files.

Among all these possible lines of future work, some has already been initiated during the thesis but didn't reach the necessary maturity to be part of this dissertation:

- *Compare contextualization approaches in different triplestores.* We initiated work in this topic using the same data and triplestores used for the evaluation of HDTr. We performed the evaluation of a set of fourteen abstract queries transformed to fit the contextualization approaches. Jena and GraphDB failed to load the datasets contextualized using Singleton Properties, whereas most queries for NdFluents timed out in any triplestore. The preliminary results show that, in conclusion, named graphs usually perform better due to the lower number of joints needed to resolve the queries.

- *Transform statements and queries from one contextualization approach to a different one.* We have explored how to address this issue and created proof of concept prototypes to transform contextualized statements between different contextualization approaches, as well as basic graph patterns that make use of a contextualization approach to basic graph patterns that use a different approach.[14]

- *Extend HDTcat to be able to merge HDTr files.* We created HDTrCat, a version of HDTcat that is able to merge files generated with a previous version of HDTr.[15] HDTrCat has shown similar performance to HDTcat on files of similar size. Some work would need to be done in order to update it to the current or future versions of HDTr.

[14]Prototype CLI applications for transforming statements and SPARQL queries can be found at https://github.com/raiden70/web_semantic_file_transform and https://github.com/raiden70/SparqlTransformCLI, respectively.

[15]HDTrCat is publicly available at https://github.com/jm-gimenez-garcia/hdt-2-java/tree/hdt-r-cat

Acknowledgements

This dissertation never would have came into existence without many people that contributed to it directly or indirectly.

I would like to thank Pierre-Antoine Champin, Manolis Koubarakis, Nathalie Hernandez, and Frederique Laforest for showing interest in this work and being jury in my Defense. Special thanks go for Pierre-Antoine Champin and Manolis Koubarakis for their review of the dissertation. Their comments were full of insight and I think this became a better document.

To my parents and my sister, for their endless love and support, before and during the thesis, no matter what.

To many friends that were at my side during this journey. Dennis Diefenbach and Hady Elsahar, who started in the WDAqua project with me. Maísa Duarte, who taught me about NELL. Omar Alqawasmeh, who share my office during his PhD and tried (futilely) to teach me to pronounce the first letter of his name. Christian Moritz, a person with whom I can spend hours debating if pain is good or bad. Oudom Khem, who survived being shot many times in the dangerous streets of Cambodia. Radha Krishna Ayyasomayajula, a worthy chess opponent. Maria Mislene, who I'm pretty sure will go directly to heaven if there is one. Carlos Arango, who can cook the best cakes ever. Mohammed Elawady, whose chewable coffee I owe a lon sleepless night. Ali Haidar and Felipe Quécole, two amazing interns whose work is tangible in this document. Adrian Villalba and Alexandra Herczku, with whom I share very good memories in Saint-Étienne. Maria Galvan, the only person who could understand how difficult French is. Emilia Kacprzak, with whom I first tasted Żubrówka. Ashot Aleksian, the only person with whom I can speak about the intricacies of language. Priyanka Garg, to whom I prepared coffee every day during one week but I can't still remember how she likes it. Venki and Shiva, who can prepare the hottest spicy food. Lyuba and Matthieu, who are ready to go for fun or having philosophical or political discussions. Beatriz Miguelena and Sam, who can prepare the best Fondues in France. To Kushagra, always up to discuss about computer science, music, or the complexities of gender identity. Sri Kalidindi, we hate each other but we are still friends.

To all the people in the now-defunct Collected Intelligence group. It was an amazing group full of smart and good people.

To Thomas Gautrais, for his amazing C++ coding skills and a great trip to Vienna to watch football matches.

To the people to brought the interest for research to me. They lit the embers that became a burning desire to start a PhD. Miguel A. Martínex Prieto and Javier D. Fernández, my supervisorst during my bacherlor and master theses, where I started dealing with MapReduce, RDF, and HDT. And Yannis Dimitriadis, who thought me research methodology.

To Guillermo Vega Gorgojo and all the people in the GSIC-EMIC and the Cross-Forest teams, who welcomed me when I needed a break. I think we did a great job together and I look forward to the possibility of continuing it.

And my most important thanks go for my supervisors, Pierre Maret and Antoine Zimmerman. I would like to thank them, first, for their guidance and advice during the thesis. But their help and commitment went far beyond the normal role of a supervisor. When I was struggling due to health problems, they were most understanding and gave me the support and help. I honestly think this thesis would not have come to an end if had had other supervisors. So, allow me to repeat myself: thank you Pierre, thank you Antoine, from the deepest of my soul.

And probably many others that I forgot. Maybe you are reading this and are not in these acknowledgements even it you deserved it. Know that I thank you in my heart even if my brain forgot when writing these words.

Bibliography

[1] Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets with the VoID vocabulary, W3C (2011). URL: https://www.w3.org/TR/void/

[2] Attardi, G., Simi, M.: A formalization of viewpoints. Fundamenta Informaticae 23(2), 149–173 (1995).

[3] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Proceedings of the 6th International Semantic Web Conference (ISWC) and the 2nd Asian Semantic Web Conference (ASWC), pp. 722–735. Springer, Busan, Korea (2007).

[4] Beckert, B.: Semantic tableaux with equality. Journal on Logic and Computation 7(1), 39–58 (1997).

[5] Beek, W., Rietveld, L., Bazoobandi, H.R., Wielemaker, J., Schlobach, S.: LOD Laundromat: A Uniform Way of Publishing Other People's Dirty Data. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 8796, pp. 213–228. Springer, Heidelberg (2014).

[6] Berners-Lee, T., Connolly, D.: Notation3 (N3): A readable RDF syntax, W3C (2011). URL: https://www.w3.org/TeamSubmission/n3/

[7] Blum, A., Mitchell, T.: Combining Labeled and Unlabeled Data with Co-Training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100. ACM (1998).

[8] Boland, K., Ritze, D., Eckert, K., Mathiak, B.: Identifying references to datasets in publications. In: TPDL, (2012).

[9] Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the ACM SIGMOD 2008, (2008).

[10] Bonatti, P.A., Hogan, A., Polleres, A., Sauro, L.: Robust and scalable Linked Data reasoning incorporating provenance and trust annotations. Web Semantics: Science, Services and Agents on the World Wide Web 9(2), 165–201 (2011).

[11] Borgida, A., Serafini, L.: Distributed Description Logics: Assimilating Information from Peer Sources. Journal on Data Semantics 1, 153–184 (2003).

[12] Bouquet, P., Ghidini, C., Giunchiglia, F., Blanzieri, E.: Theories and Uses of Context in Knowledge Representation and Reasoning. Journal of Pragmatics 35(3), 455–484 (2003).

[13] Bouquet, P., Giunchiglia, F., Van Harmelen, F., Serafini, L., Stuckenschmidt, H.: C-OWL: Contextualizing Ontologies. In: Fensel, D., Sycara, K.P., Mylopoulos, J. (eds.) The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, ProceedingsLNCS, vol. 2870, pp. 164–179. Springer, Heidelberg (2003).

[14] Briandais, R.D.L.: File searching using variable length keys. In: Johnson, R.R. (ed.) Papers presented at the the 1959 western joint computer conference, IRE-AIEE-ACM 1959 (Western), San Francisco, California, USA, March 3-5, 1959, pp. 295–298. ACM (1959).

[15] Brickley, D., Guha, R.: RDF Schema 1.1, pp. 91–122. W3C (2014). DOI: 10.1016/B978-0-12-373556-0.00006-X URL: http://www.w3.org/TR/rdf-schema/

[16] Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema, W3C (2004). URL: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

[17] Buvač, S.: Quantificational Logic of Context. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, AAAI 96, IAAI 96, Portland, Oregon, August 4-8, 1996, Volume 1. Pp. 600–606 (1996).

[18] Buvač, S., Buvač, V., Mason, I.A.: Metamathematics of Contexts. Fundamenta Informaticae 23(2), 263–301 (1995).

[19] Buvač, S., Mason, I.A.: Propositional Logic of Context. In: Proceedings of the 11th National Conference on Artificial Intelligence. Washington, DC, USA, July 11-15, 1993. Pp. 412–419 (1993).

[20] Carlson, A., Betteridge, J., Hruschka, E.R., Mitchell, T.M.: Coupling Semi-Supervised Learning of Categories and Relations. In: Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing, (2009).

[21] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an Architecture for Never-Ending Language Learning. In: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI), (2010).

[22] Carothers, G.: RDF 1.1 N-Quads: A line-based syntax for RDF datasets, W3C (2014). URL: https://www.w3.org/TR/n-quads

[23] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. Journal of Web Semantics 3(4), 247–267 (2005).

[24] Cheng, G., Qu, Y.: Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. International Journal on Semantic Web and Information Systems 5(3), 49–70 (2009).

[25] Crestan, E., Pantel, P.: Web-scale table census and classification. In: Proceedings of the fourth ACM international conference on Web search and data mining, pp. 545–554 (2011).

[26] Csardi, G., Nepusz, T.: The igraph software package for complex network research. InterJournal, Complex Systems 1695(5), 1–9 (2006).

[27] Cuenca-Grau, B., Kutz, O.: Modular Ontology Languages Revisisted. In: Honavar, V.G., Finin, T.W., Caragea, D., Mladenic, D., Sure, Y. (eds.) SWeCKa 2007: Proceedings of the IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition, Hyderabad, India, January 7, 2007, (2007).

[28] Cyganiak, R., Wood, D., Lanthaler, M.: RDF 1.1 Concepts and Abstract Syntax, pp. 263–270. W3C (2014). URL: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[29] Debattista, J., Londoño, S., Lange, C., Auer, S.: Quality Assessment of Linked Datasets Using Probabilistic Approximation. In: The Semantic Web. Latest Advances and New Domains - ESWC 2015 - 12th Extended Semantic Web ConferenceLNCS, vol. 9088, pp. 221–236. Springer, Heidelberg (2015).

[30] Diefenbach, D., Giménez-García, J.M.: HDTCat: Let's make HDT generation scale. In: Pan, J.Z., Tamma, V.A.M., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) The semantic web - ISWC 2020 - 19th international semantic web conference, athens, greece, november 2-6, 2020, proceedings, part IILecture notes in computer science, pp. 18–33. Springer (2020).

[31] Diefenbach, D., Singh, K.D., Maret, P.: WDAqua-Core0: A Question Answering Component for the Research Community. In: Semantic Web Challenges - 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28 - June 1, 2017, Revised Selected Papers, pp. 84–89 (2017).

[32] Dimou, A., Sande, M.V., Colpaert, P., Verborgh, R., Mannens, E., deWalle, R.V.: RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In: Proceedings of the Workshop on Linked Data on the Web, co-located with the 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, (2014).

[33] Ding, L., Finin, T.: Characterizing the Semantic Web on the Web. In: The Semantic Web - ISWC 2006 - 5th International Semantic Web ConferenceLNCS, vol. 4273, pp. 242–257. Springer, Heidelberg (2006).

[34] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management, pp. 652–659. ACM (2004).

[35] Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P.: Finding and Ranking Knowledge on the Semantic Web. In: The Semantic Web - ISWC 2005 - 4th International Semantic Web ConferenceLNCS, vol. 3729, pp. 156–170. Springer, Heidelberg (2005).

[36] Dinsmore, J.: *Partitioned representations*. Springer (1991).

[37] Dividino, R., Sizov, S., Staab, S., Schueler, B.: Querying for provenance, trust, uncertainty and other meta knowledge in RDF. Journal of Web Semantics 7(3), 204–219 (2009).

[38] Duarte, M.C., Hruschka, E.R.: How to Read The Web In Portuguese Using the Never-Ending Language Learner's Principles. In: Proceedings of the 14th International Conference on Intelligent Systems Design and Applications, (2014).

[39] Duarte, M.C., Maret, P.: Vers une instance française de NELL : chaîne TLN multilingue et modélisation d'ontologie. Revue des Nouvelles Technologies de l'Information Extraction et Gestion des Connaissances, RNTI-E-33, 469–472 (2017).

[40] Ellefi, M.B., Bellahsene, Z., Breslin, J.G., Demidova, E., Dietze, S., Szymanski, J., Todorov, K.: Dataset profiling – a guide to features, methods, applications and vocabularies. (2016).

[41] Endris, K.M., Giménez-García, J.M., Thakkar, H., Demidova, E., Zimmermann, A., Lange, C., Simperl, E.: Dataset Reuse: An Analysis of References in Community Discussions, Publications and Data. In: Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017, 5:1–5:4 (2017).

[42] Ermilov, I., Martin, M., Lehmann, J., Auer, S.: Linked Open Data Statistics: Collection and Exploitation. In: Knowledge Engineering and the Semantic Web - KESW 2013 - 4th International ConferenceCommunications in Computer and Information Science, pp. 242–249. Springer Berlin Heidelberg, St. Petersburg, Russia (2013).

[43] Farias Lóscio, B., Burle, C., Calegari, N.: Data on the web best practices, W3C (2016). URL: http://www.w3.org/TR/dwbp/

[44] Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary RDF representation for publication and exchange (HDT). Journal of Web Semantics 19, 22–41 (2013).

[45] Fernández, J.D., Martínez-Prieto, M.A., Polleres, A., Reindorf, J.: HDTQ: Managing RDF Datasets in Compressed Space. In: The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings, pp. 191–208 (2018).

[46] Frey, J., Hellmann, S.: MaSQue: An Approach for Flexible Metadata Storage and Querying in RDF. In: 13th International Conference on Semantic Systems Proceedings (SEMANTiCS 2017) - Posters & Demonstrations TrackSEMANTiCS '17, (2017).

[47] Frey, J., Müller, K., Hellmann, S., Rahm, E., Vidal, M.-E.: Evaluation of metadata representations in RDF stores. Semantic Web 10(2), 205–229 (2017).

[48] Gangemi, A., Presutti, V.: A Multi-dimensional Comparison of Ontology Design Patterns for Representing n-ary Relations. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 7741 LNCS, pp. 86–105. Springer, Heidelberg (2013).

[49] Gardner, M., Talukdar, P.P., Krishnamurthy, J., Mitchell, T.M.: Incorporating Vector Space Similarity in Random Walk Inference over Knowledge Bases. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014).

[50] Ghidini, C., Giunchiglia, F.: Local Models Semantics, or contextual reasoning=Locality+Compatibility. Artificial Intelligence 127(2), 221–259 (2001).

[51] Ghidini, C., Serafini, L.: Distributed First Order Logics. In: Frontiers Of Combining Systems 2, Studies in Logic and Computation, pp. 121–140. Research Studies Press (1998).

[52] Giménez-García, J.M., Duarte, M., Zimmermann, A., Gravier, C., Hruschka Jr., E.R., Maret, P.: NELL2RDF: Reading the web, tracking the provenance, and publishing it as linked data. In: Joint Proceedings of the International Workshops on Contextualized Knowledge Graphs, and Semantic Statistics co-located with 17th International Semantic Web Conference (ISWC 2018) Monterey, USA, October 8th, 2018ceurwp, CEUR, Monterey, USA (2018).

[53] Giménez-García, J.M., Duarte, M.C., Zimmermann, A., Gravier, C., Jr., E.R.H., Maret, P.: NELL2RDF: Reading the web, and publishing it as linked data, Université Jean Monnet (2018). URL: http://arxiv.org/abs/1804.05639

[54] Giménez-García, J.M., Thakkar, H., Zimmermann, A.: Assessing Trust with PageRank in the Web of Data. In: The Semantic Web - ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected PapersCEUR Workshop Proceedings, pp. 293–307. Springer, Cham (2016).

[55] Giménez-García, J.M., Zimmermann, A.: NdProperties: Encoding contexts in RDF predicates with inference preservation. In: Joint Proceedings of the International Workshops on Contextualized Knowledge Graphs, and Semantic Statistics co-located with 17th International Semantic Web Conference (ISWC 2018) Monterey, USA, October 8th, 2018, Monterey, USA (2018).

[56] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: A Multi-dimensional Contexts Ontology, Université Jean Monnet (2016). URL: http://arxiv.org/abs/1609.07102

[57] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. In: Proceedings of the 14th Extended Semantic Web Conference (ESWC), pp. 638–654. Springer, Cham (2017).

[58] Giménez-García, J.M., Zimmermann, A., Maret, P.: Representing Contextual Information as Fluents. In: Knowledge Engineering and Knowledge Management - {EKAW} 2016 Satellite Events, {EKM} and Drift-an-LOD, Bologna, Italy, November 19-23, 2016, Revised Selected Papers, pp. 119–122. Springer, Cham (2017).

[59] Giunchiglia, F.: Contextual Reasoning. Epistemologica 16, 345–364 (1993).

[60] Giunchiglia, F., Serafini, L.: Multilanguage hierarchical logics or: How we can do without modal logics. Artificial Intelligence 65(1), 29–70 (1994).

[61] Goguen, J.A., Burstall, R.M.: Institutions: Abstract Model Theory for Specification and Programming. Journal of the Association for Computing Machinery 39(1), 95–146 (1992).

[62] Golbreich, C., Wallace, E.K.: OWL 2 web ontology language, new features and rationale (second edition), W3C (2012). URL: https://www.w3.org/TR/2012/REC-owl2-new-features-20121211/

[63] Guéret, C., Groth, P., Stadler, C., Lehmann, J.: Assessing linked data mappings using network measures. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 7295 LNCS, pp. 87–102. Springer, Heidelberg (2012).

[64] THESIS

[65] Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with TrustRank. In: Proceedings of the Thirtieth international conference on Very large data bases, pp. 576–587. VLDB Endowment (2004).

[66] Harth, A., Kinsella, S., Decker, S.: Using naming authority to rank data and ontologies for web search. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 5823 LNCS, pp. 277–292. Springer, Heidelberg (2009).

[67] Hartig, O.: Foundations of RDF* and SPARQL* (An Alternative Approach to Statement-Level Metadata in RDF). In: Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017. (2017).

[68] Hartig, O., Champin, P.-A., Kellogg, G., Seaborne, A., Arndt, D., Broekstra, J., DuCharme, B., Lassila, O., Patel-Schneider, P.F., Prud'hommeaux, E., Thibodeau, T., Thompson, B.: RDF-star and SPARQL-star, W3C (2021). URL: https://w3c.github.io/rdf-star/cg-spec

[69] Hartig, O., Thompson, B.: Foundations of an Alternative Approach to Reification in RDF, pp. 14–14. arXiv (2014). URL: http://arxiv.org/abs/1406.3399

[70] Haveliwala, T.H.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE Transactions on Knowledge and Data Engineering 15(4), 784–796 (2003).

[71] Heath, T., Bizer, C.: *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool (2011).

[72] Hernández, D., Hogan, A., Krötzsch, M.: Reifying RDF: What Works Well With Wikidata? In: Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems, pp. 32–47, Bethlehem, PA, USA (2015).

[73] Hernández, D., Hogan, A., Riveros, C., Rojas, C., Zerega, E.: Querying Wikidata: Comparing SPARQL, Relational and Graph Databases. In: Proceedings of the 15th International Semantic Web Conference (ISWC)LNCS, vol. 9982 LNCS, pp. 88–103. Springer, Heidelberg (2016).

[74] Hernández-Illera, A., Martínez-Prieto, M.A., Fernández, J.D., Fariña, A.: i*HDT*++: improving HDT for SPARQL triple pattern resolution. Journal of Intelligent & Fuzzy Systems 39(2), 2249–2261 (2020).

[75] Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 web ontology language primer, W3C (2012). URL: https://www.w3.org/TR/owl2-primer/

[76] Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artificial Intelligence 194, 28–61 (2013).

[77] Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A., Decker, S.: 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 (2011).

[78] Hogan, A., Hitzler, P., Janowicz, K.: Linked dataset description papers at the semantic web journal: A critical assessment. Semantic Web 7 (2016).

[79] Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., Decker, S.: An empirical survey of Linked Data conformance. Journal of Web Semantics 14, 14–44 (2012).

[80] Horrocks, I., Patel-schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, pp. 1–20. WC3 (2004). URL: https://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

[81] Horst, H.J.t.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. Journal of Web Semantics 3(2), 79–115 (2005).

[82] Hruschka, E.R., Duarte, M.C., Nicoletti, M.C.: Coupling as Strategy for Reducing Concept-Drift in Never-Ending Learning Environments. Fundamenta Informaticae (2013).

[83] Jentzsch, A., Cyganiak, R., Bizer, C.: State of the LOD cloud, Mannheim (2011)

[84] Kifer, M., Boley, H.: RIF Overview (Second Edition) - W3C Working Group Note 5, W3C (2013). URL: http://www.w3.org/TR/rif-overview

[85] Kifer, M., Subrahmanian, V.S.: Theory of Generalized Annotated Logic Programming and its Applications. Journal of Logic Programming 12(3), 335–367 (1992).

[86] Krishnamurthy, J., Mitchell, T.M.: Joint Syntactic and Semantic Parsing with Combinatory Categorial Grammar. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1188–1198 (2014).

[87] Kutz, O., Lutz, C., Wolter, F., Zakharyaschev, M.: E-connections of abstract description systems. Artificial Intelligence 156(1), 1–73 (2004).

[88] Lao, N., Mitchell, T., Cohen, W.W.: Random Walk Inference and Learning in A Large Scale Knowledge Base. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 529–539. Association for Computational Linguistics (2011).

[89] Lassila, O., Swick, R.R.: Resource Description Framework (RDF) Model and Syntax Specification, W3C (1999). URL: https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[90] Lebo, T., Sahoo, S., McGuinness, D., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednik, S., Zhao, J.: PROV-O: The PROV Ontology, pp. 1–80. W3C (2013). URL: https://www.w3.org/TR/prov-o/

[91] Lefrançois, M., Zimmermann, A., Bakerally, N.: A SPARQL Extension for Generating RDF from Heterogeneous Formats. In: The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I, pp. 35–50 (2017).

[92] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., vanKleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. SWJ 6(2) (2015).

[93] Liu, S., D'Aquin, M., Motta, E.: Towards Linked Data Fact Validation through Measuring Consensus. In: Proceedings of the 2nd Workshop on Linked Data Quality co-located with 12th Extended Semantic Web Conference (ESWC 2015)CEUR Workshop Proceedings, CEUR-WS.org, Portorož, Slovenia (2015).

[94] Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., Schasberger, B.: The Penn Treebank: Annotating Predicate Argument Structure. In: Proceedings of the Workshop on Human Language Technology, (1994).

[95] Martínez-Prieto, M., Arias, M., Fernández, J.: Exchange and Consumption of Huge RDF Data. In: Proc. of ESWC, pp. 437–452 (2012).

[96] Martínez-Prieto, M.A., Brisaboa, N.R., Cánovas, R., Claude, F., Navarro, G.: Practical compressed string dictionaries. Information Systems 56, 73–108 (2016).

[97] Martinez-Rodriguez, J., Hogan, A., Lopez-Arevalo, I.: Information Extraction meets the Semantic Web: A Survey. Semantic Web journal (2018).

[98] Masolo, C., Guizzardi, G., Vieu, L., Bottazzi, E., Ferrario, R.: Relational roles and qua-individuals. In: AAAI Fall Symposium on Roles, an interdisciplinary perspective, pp. 103–112 (2005).

[99] McCarthy, J.: Generality in Artificial Intelligence. Communications of the ACM 30(12), 1030–1035 (1987).

[100] McCarthy, J.: Notes on formalizing context. Science 13, 555–560 (1993).

[101] McCarthy, J., Buvač, S.: Formalizing Context (Expanded Notes). Theory & Psychology 15(1), 128–131 (1997).

[102] Mendes, P.N., Mühleisen, H., Bizer, C.: Sieve: Linked data quality assessment and fusion. In: EDBT/ICDT Workshops, (2012).

[103] Mitchell, T.M., Cohen, W.W., Hruschka, E.R., Talukdar, P.P., Betteridge, J., Carlson, A., Mishra, B.D., Gardner, M., Kisiel, B., Krishnamurthy, J., Lao, N., Mazaitis, K., Mohamed, T., Nakashole, N., Platanios, E.A., Ritter, A., Samadi, M., Settles, B., Wang, R.C., Wijaya, D.T., Gupta, A., Chen, X., Saparov, A., Greaves, M., Welling, J.: Never-Ending Learning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 2302–2310 (2015).

[104] Mons, B., Velterop, J.: Nano-Publication in the e-science era. In: Workshop on Semantic Web Applications in Scientific Discourse (SWASD), pp. 14–15 (2009).

[105] Muñoz, E., Hogan, A., Mileo, A.: Using linked data to mine RDF from wikipedia's tables. In: Proceedings of the 7th ACM international conference on Web search and data mining, pp. 533–542 (2014).

[106] Munro, J.I., Raman, R., Raman, V., Rao, S.S.: Succinct representations of permutations and functions. Theor. Comput. Sci. 438, 74–88 (2012).

[107] Nardi, D., Brachman, R.J.: An Introduction to Description Logics. In: The Description Logic Handbook: Theory, Implementation, and Applications, pp. 1–40. Cambridge University Press (2003).

[108] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. In: Proceedings of the 23rd International Conference on the World Wide Web (WWW), pp. 759–770. ACM (2014).

[109] Nossum, R.: A decidable multi-modal logic of context. Journal of Applied Logic 1(1), 119–133 (2003).

[110] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web, pp. 1–1. W3C (2006). URL: https://www.w3.org/TR/swbp-n-aryRelations/

[111] Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. World Wide Web Internet And Web Information Systems 54(1999), 1–17 (1998).

[112] Paulheim, H., Bizer, C.: Improving the Quality of Linked Data Using Statistical Distributions. International Journal on Semantic Web and Information Systems 10(2), 63–86 (2014).

[113] Pelgrin, O.P., Hose, K., Galárraga, L.: TrieDF: Efficient in-memory indexing for metadata-augmented RDF. In: Proceedings of the 7th Workshop on Managing the Evolution and Preservation of the Data Web, CEUR Workshop Proceedings (2021).

[114] Piwowar, H.A., Vision, T.J.: Data reuse and the open data citation advantage. PeerJ (2013).

[115] Quécole, F., Martines, R., Giménez-García, J.M., Thakkar, H.: Towards capturing contextual semantic information about statements in web tables. In: Capadisli, S., Cotton, F., Giménez-García, J.M., Haller, A., Kalampokis, E., Nguyen, V., Sheth, A.P., Troncy, R. (eds.) Joint proceedings of the international workshops on contextualized knowledge graphs, and semantic statistics co-located with 17th international semantic web conference (ISWC 2018)CEUR workshop proceedings, CEUR-WS.org (2018).

[116] Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A Midterm Report. In: Proceedings of the European Conference on Machine Learning, (1993).

[117] Rietveld, L., Beek, W., Schlobach, S.: LOD lab: Experiments at LOD scale. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 9367, pp. 339–355. Springer, Heidelberg (2015).

[118] Rietveld, L., Verborgh, R., Beek, W., Sande, M.V., Schlobach, S.: Linked Data-as-a-Service: The Semantic Web Redeployed. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 9088, pp. 471–487. Springer, Heidelberg (2015).

[119] Samadi, M., Veloso, M.M., Blum, M.: OpenEval: Web Information Query Evaluation. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA. Citeseer (2013).

[120] Scheuermann, A., Motta, E., Mulholland, P., Gangemi, A., Presutti, V.: An Empirical Perspective on Representing Time. In: 7th International Conference on Knowledge CaptureK-CAP '13, pp. 89–96. ACM, New York, NY, USA (2013).

[121] Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)LNCS, vol. 8796, pp. 245–260. Springer, Heidelberg (2014).

[122] Scott, D.: Identity and existence in intuitionistic logic. In: Fourman, M., Mulvay, C., Scott, D.S. (eds.) Applications of Sheaf Theory to Algebra, Analysis and Topology, Lecture Notes in Mathematics, Vol. 753Lecture Notes in Mathematics, pp. 660–696. Springer-Verlag (1979).

[123] Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web, pp. 697–706. ACM (2007).

[124] Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet. J. Web Sem. 6(3), 203–217 (2008).

[125] Thakkar, H., Endris, K.M., Gimenez-Garcia, J.J.M., Debattista, J., Lange, C., Auer, S., Thakkar, H., Endris, K.M., Gimenez-Garcia, J.J.M., Debattista, J., Lange, C., Auer, S.: Are Linked Datasets fit for Open-domain Question Answering? A Quality Assessment. In: Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics, WIMS 2016, pp. 1–12. ACM Press, Nîmes, France (2016).

[126] Udrea, O., Recupero, D.R., Subrahmanian, V.S.: Annotated RDF. ACM Transaction on Computational Logics 11(2), 10:1–10:41 (2010).

[127] Verborgh, R., Vander Sande, M., Colpaert, P., Coppens, S., Mannens, E., deWalle, R.: Web-Scale Querying through Linked Data Fragments. In: Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW 2014)CEUR Workshop Proceedings, Seoul, Korea (2014).

[128] Wang, R.C., Cohen, W.W.: Language-Independent Set Expansion of Named Entities Using the Web. In: Proceedings of the 7th IEEE International Conference on Data Mining, pp. 342–350. IEEE (2007).

[129] Wang, R.Y., Strong, D.M.: Beyond Accuracy: What Data Quality Means to Data Consumers. Journal of Management Information Systems 12(4), 5–33 (1996).

[130] Welty, C.: Context Slices: Representing Contexts in OWL. In: Workshop on Ontology PatternsCEUR Workshop Proceedings, pp. 59–60. CEUR-WS.org (2010).

[131] Welty, C., Fikes, R.: A Reusable Ontology for Fluents in OWL. In: Bennett, B., Fellbaum, C.D. (eds.) Proceedings of the 2006 conference on Formal Ontology in Information SystemsFrontiers in Artificial Intelligence and Applications, pp. 226–236. IOS Press (2006).

[132] Yang, B., Mitchell, T.M.: Joint Extraction of Events and Entities within a Document Context. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pp. 289–299 (2016).

[133] Zamborlini, V., Guizzardi, G.: On the Representation of Temporally Changing Information in OWL. In: Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC, pp. 283–292. IEEE (2010).

[134] Zamborlini, V.C., Guizzardi, G.: An Ontologically-Founded Reification Approach for Representing Temporally Changing Information in OWL. In: 11th International Symposium on Logical Formalizations of Commonsense Reasoning (COMMONSENSE 2013), Centre for Telematics and Information Technology ; Telematica Instituut, Crete (2013).

[135] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality Assessment for Linked Open Data: A Survey. Semantic Web Journal (by IOS Press) 1(1), 1–31 (2014).

[136] Zimmermann, A.: Logical formalisms for Agreement Technologies. In: Ossowski, S. (ed.) Agreement Technologies, pp. 69–82. Springer-Verlag (2013).

[137] Zimmermann, A.: RDF 1.1: On Semantics of RDF Datasets, W3C (2014). URL: http://www.w3.org/TR/2014/NOTE-rdf11-datasets-20140225/

[138] Zimmermann, A., Giménez-García, J.M.: Contextualizing DL Axioms: Formalization, a New Approach, and Its Properties. In: Joint Proceedings of the Web Stream Processing workshop (WSP 2017) and the 2nd International Workshop on Ontology Modularity, Contextuality, and Evolution (WOMoCoE 2017) co-located with 16th International Semantic Web Conference (ISWC 2017), pp. 74–85 (2017).

[139] Zimmermann, A., Giménez-García, J.M.: Integrating Context of Statements within Description Logics, Université Jean Monnet (2017). URL: http://arxiv.org/abs/1709.04970

[140] Zimmermann, A., Gravier, C., Subercaze, J., Cruzille, Q.: Nell2RDF: Read the Web, and Turn it into RDF. In: Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data, pp. 1–7 (2013).

[141] Zimmermann, A., Lopes, N., Polleres, A., Straccia, U.: A general framework for representing, reasoning and querying with annotated Semantic Web data. Journal of Web Semantics 11, 72–95 (2012).