# Uncertainty Management for Linked Data Reliability on the Semantic Web

Ahmed El Amine Djebri

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

# THÈSE DE DOCTORAT

## GESTION DE L'INCERTITUDE POUR LA FIABILITÉ DES DONNÉES LIÉES DANS LE WEB SÉMANTIQUE


**AHMED EL AMINE DJEBRI**

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis


**Présentée en vue de l'obtention du grade de docteur en** Informatique **de l'**Université Côte d'Azur

**Dirigée par** : Andrea TETTAMANZI, Fabien GANDON

**Soutenue le** : 24/02/2022

**Devant le jury, composé de** :
**Président:** Michel BUFFA, Professeur, Université Côte d'Azur
**Rapporteurs:**
Hassina Seridi, Professeur,
Université Badji Mokhtar - Annaba, Algérie
Pierre-Antoine Champin, Maître de conférences
LIRIS, CNRS, Université Claude Bernard Lyon 1
Fernando Bobillo, Associate Professor
Université de Zaragoza, Espagne

Inria

i3S sophia antipolis

# Gestion de l'Incertitude pour la fiabilité des Données Liées dans le Web Sémantique

## (Uncertainty Management for Linked Data Reliability on the Semantic Web)

Devant le jury composé de :

| | |
|---|---|
| **Michel BUFFA** | **Président** |
| Professeur, | |
| Université Côte dAzur, France | |
| **Hassina Seridi** | **Rapporteure** |
| Professeur, | |
| Université Badji-Mokhtar Annaba, Algérie | |
| **Pierre-Antoine Champin** | **Rapporteur** |
| Maître de conférences, | |
| LIRIS, CNRS, Université Claude Bernard Lyon 1, France | |
| **Fernando Bobillo** | **Rapporteur** |
| Associate Professor, | |
| Université de Zaragoza, Espagne | |

## Gestion de l'Incertitude pour la fiabilité des Données Liées dans le Web Sémantique

**Résumé** — Le Web sémantique a évolué pour atteindre différentes applications. Il a été conçu pour permettre aux machines de comprendre les ressources disponibles sur le World Wide Web et d'utiliser les informations extraites dans le processus de prise de décision et de raisonnement. Le Web est donc un monde ouvert où les gens peuvent dire ce qu'ils veulent et où les utilisateurs - dans ce cas, les humains et les machines - peuvent s'y retrouver. L'un des principaux défis actuels consiste à traiter des informations provenant de sources multiples et le plus souvent peu fiables, et les données liées sont une sélection du contenu hétérogène du Web. Qu'elles soient présentées dans un format lisible par une machine, extraites automatiquement de documents Web ou déduites par des processus de raisonnement, les données liées peuvent être périmées, incorrectes, incomplètes, vagues, ambiguës ou, plus généralement, incertaines. Le traitement des données liées incertaines est représenté par de multiples défis tels que la qualification (ou quantification) de l'incertitude, le calcul, la déduction et, dans le cas du Web sémantique, la publication et la réutilisation.

Cette thèse de doctorat se concentre sur plusieurs points :

- comment l'incertitude peut être formalisée et intégrée dans le Web Sémantique,
- comment l'incertitude peut être extraite et accessible,
- comment réconcilier et fusionner des données incertaines distribuées et liées,
- comment l'incertitude peut être évaluée sur la base de sources de données distribuées, et si possible de propager l'incertitude avec les liens.

**Mots clés :** Incertitude, Web Sémantique, Données Liées, Ontologies.

# Uncertainty Management for Linked Data Reliability on the Semantic Web

**Abstract** — The Semantic Web has evolved to reach different applications. It was designed to enable machines to understand available resources on the World Wide Web and use the extracted information in the decision making and reasoning process. Hence, the Web is an open world where people can say whatever they want and users -in this case, people and machines- can relate to it. One of the main challenges nowadays is to deal with information from multiple and mostly unreliable sources and Linked Data is a screening of the heterogeneous content on the Web. Either presented in a machine-readable format, automatically extracted from web documents or inferred by reasoning processes, Linked Data might be outdated, incorrect, incomplete, vague, ambiguous, or more generally, uncertain. Dealing with Uncertain Linked Data is represented by multiple challenges like uncertainty qualification (or quantification), calculus, deduction, and in the case of the Semantic Web: publishing and reusability.

This Ph.D. focuses on several points:

- how uncertainty can be formalized and integrated into the Semantic Web,
- how uncertainty can be extracted and accessed,
- how to reconcile and fuse distributed uncertain linked data,
- how uncertainty can be evaluated based on distributed data sources, and if possible to propagate uncertainty along with links.

**Keywords:** Uncertainty, Semantic Web, Linked Data, Ontologies.

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis

# *Acknowledgements*

Eventually, everything comes to an end. But every end announces new beginnings. As I constantly repeat in Spanish: *"De la nada sale el todo"*, or : *"From nothing, comes everything"*.

From my perspective, this is not just an acknowledgment. It is a statement of gratitude and thankfulness towards everyone who helped me stand my path towards accomplishing this work.

I am grateful for the persistence and curiosity of our kind. Such dedication, engagement, and perseverance allowing this knowledge to reach us and be in our hands, for us to add a small brick on top of it. I am grateful for the question marks that I wrote and still writing.

The idea behind any work is to use what previous ones offered for a starter. I'm grateful for each of the authors I cited and their work and vision. I'm thankful to Sir Tim Berners-Lee for envisioning the World Wide Web and the Semantic Web.

I am grateful for the professors dedicating their time to report on or stand as members to discuss this work. Their remarks, questions, and suggestions are the ones to help the person behind work flourish in his future.

I am grateful to those who accepted me as a member, those I grew up beside, those who answered my questions about research works passionately as I was wandering between their offices. Those who inspired me with their ambition and dedication, those who I consider friends and fellows. I am grateful for the members of WIMMICS and SPARKS, all without exception (maybe a bit more for those with whom I shared coffee breaks).

I am grateful, for eternity, to the people who welcomed me and offered me this opportunity. The people who guided, helped, and comforted me in difficult times. The people who were responsible directly for my growth and progress. To the most dedicated persons I have ever encountered and from whom I am inspired as a researcher

and a person. I am grateful for your kind souls, Andrea and Fabien.

Happiness is not about reaching the goal, but all the satisfaction is in progress. For the days I lived among you and the moments we shared, I am grateful. To every single one of my friends. To my office mates Santiago, Vorakit, Tobias, and Michael. To my peers who used to visit the 416 frequently and with whom I shared memorable moments.

To my parents, whose sacrifice to raise me and whose patience for my absence during these challenging times no words will be enough to describe.

I will not make part of the people denying the role of luck in all of this. I am, until now, considering myself a lucky man in this journey with all of you.

All of this, and for uncertainty. For as much as I don't like being undetermined, I ended up doing an entire thesis about it. I learned through this thesis and journey, to face my doubts in research and life. As I'm still striving to be a better version of myself.

For all of you, I am grateful. To the One Who should be praised, through you and for you I am.

"*He who controls metadata controls the Web.*"

Fabien Gandon

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| **W3C** | World Wide Web Consortium |
| **RDF** | Resource Description Framework |
| **RDFS** | *RDF* Schema |
| **OWL** | Web Ontology Language |
| **RIF** | Rule Interchange Format |
| **SPARQL** | *SPARQL* Protocol and RDF Query Language (Recursive acronym) |
| **URL** | Uniform Resource Locator |
| **URI** | Universal Resource Identifier |
| **IRI** | Internationalized Resource Identifier |
| **LOD** | Linked Open Data |

# Part I

# General Introduction

# Chapter 1

# Introduction

## 1.1 Context of the thesis: Uncertainty on the Semantic Web

**Web and Data** Due to the Internet, we are more linked now than ever before. The expanding infrastructure of the network allowed to connect machines via cables and protocols, and more than that: it connected people. With it being the basis and the transparent proxy, a refrigerator signals a vending machine to deliver a pack of eggs. The same network is the backbone for the content that is daily updated and augmented by the different agents, *i.e.* The Web [1].

The 2020 Dagstuhl Manifesto [2] described the Web as perceived as the "nervous system of the planet". The document selected that year to discuss the Web's future and affirm its size, impact, and stakes. The report issued by DataReportal[1] in January 2021 [3] confirms the aforementioned fact by accounting for the growth of the connected population. The same report mentioned statistically ordered reasons for the use of the Web. The reasons vary from information search and retrieval to education, entertainment, education, finances, and being inspired by the content. The Web helps lives too: governments are digitizing and decentralizing their services, making it easy to issue documents, pay taxes, switch properties, and even get travel passes.

[1]DataReportal is an online reference library offering statistical reports for online activity over the world. Info available here: https://datareportal.com/

The report estimated the current connected population to be 4.66 billion users (about 59.5% of the total population of the world).[2]

**How wide is the Web?** According to the statistics compiled by Domo[3] from several sources, In every 60 seconds of the year 2020, an average of:

- 500 hours of videos are uploaded to Youtube,
- 150,000 messages and 147,000 photos are shared on Facebook,
- 64,499 job applications are submitted on LinkedIn,
- $240,000 are exchanged on Venmo,
- $1M is spent on E-commerce websites,
- 28 music tracks are added to Spotify.

These statistics show the significant quantity of data added to or exchanged via the Web. Other websites provide new information daily, such as news websites (*e.g. bbc.com*, *aljazeera.net*, *francetvinfo.fr*), financial websites (*e.g. tradingview.com*), or inspiration for artists and designers (*e.g. usepanda.com*). As for the "wideness" of the Web, the number of published websites on the Web exceeded alone one billion websites.[4]

The emergence of the Internet of Things **IoT** opened the door for more machine-to-machine autonomous communication. For instance, sensor networks connect to their main stations and communicate data about humidity and temperature. Home assistants link and control the different intelligent devices of a smart home: light bulbs, cameras, appliances, etc. In short, the World Wide Web evolved from the simple network linking documents with hyperlinks to a crucial part of our daily life. It is perceived as a "non-ending project" [1] in constant development for a better reach and inclusion.

[2] This data does not include social media statistics as multiple accounts may refer to a unique person.

[3] https://www.domo.com/learn/data-never-sleeps-8

[4] https://www.internetlivestats.com/total-number-of-websites/

**Semantic Web and Artificial Intelligence** The Web is around us, accessed from different terminals and used to link existing files hosted by interconnected servers across the globe. With one click, a student in *Karachi* can download a textbook from their teacher in *Melbourne*. Despite its usefulness, the Web used to represent only the rigid form of content (such as documents, audio, video, images, etc.) and the links referring to that content. Nowadays, the Web took much more advanced steps towards understanding deeper connections, from linking bits of zeros and ones hosted on machines interconnected via the Internet to linking physical and conceptual entities in real life.

The abstraction of reality revolutionized our perception of the Web. We no longer refer to documents but to resources that might be anything: people, cars, animals, theories, fruits with weird shapes, the concept of "weirdness", etc. Some of these resources do exist in Web documents: a web page talking about lions, an audiobook promoting *grit*, or a TED talk explaining procrastination.

As we stated before, the Web is in constant change. The "Web that makes sense" became more than a set of related files hosted on the Internet. It is a network of interconnected entities that provides some abstraction of reality. Within a context, this network offers a focused perception of our beliefs on a particular matter. This abstraction of reality opened the doors for powerful usages for a network that used to link documents and allowed it to link pieces of data, making the support for many exciting applications. We *believe* that we know all the other resources related to it and the types of relationships between them from one resource. The links between these resources allow us to go on an exploratory journey, from *Albert Einstein*[5] to getting the phone number of *Princeton University* from their Facebook page, where he spent 22 years working (between 1933 and 1955).

Discovering and exploring are amongst the essential applications of this evolution. Nevertheless, we focus here on reasoning as one of the intelligent behaviors studied and supported by the Semantic Web [4]. First, reasoning covers both previous applications

[5]https://www.wikidata.org/wiki/Q937

as it requires resources to discover their entourage and relies on path exploration to jump from one resource to the other. Second, reasoning is the natural application to be used with such abstraction of the world: if we can think with abstract data in our minds, machines should do the same with data explicitly abstracted. Reasoning, in this case, is about letting machines infer new information. That is conditioned by allowing them first to discover, explore, and understand the links, patterns, and semantics of resources. Consequently, AI can create new connections and observe patterns, enabling a better understanding of the sense of resources and relationships between them.

*Strong Artificial Intelligence* refers to the fact that machines will be able to think for themselves, and perform complex reasoning processes and decide on different matters, the same way the human brain does. It is in contrast with **weak AI** limited to specific applications and without access to significant amounts of data, which is the case of machine learning applications nowadays: limited domain of application (task-specific AI), limited data (or access to data, due to size or non-readiness), and limited resources. We think it is early to talk about **strong AI**, assuming the conditions are not met yet for the vision to be true. Nevertheless, we see the Web as a raw material for AI to feed on, as it represents an exceedingly rich source for human contributions to knowledge. We envision for AI to access the Web as any other human agent, do their research, understand, decide, contribute, and communicate the results of its reasoning process to other AIs [5]. We find no better support for such a vision than the Semantic Web: a network of interconnected intelligence, human and artificial ones, helping one another in the production and maintenance of consistent, reliable information that can be consumed and verified by anyone.

**What is wrong with our (Semantic) Web ?** Data reliability is not related only to the Web but linked to the creators and curators of data. The rationality of agents in decision-making is often assumed. That way, any interpretation related to the existing pieces of data is assumed to be accurate. Once uncertainty is introduced, it should be considered, assuming that the agents act upon the data they are served

but still risk bias or mistake. One of the examples of such imperfections that we can present in daily life would be the *cognitive biases*. Even the lines that you are reading here can be subject of the *Naive realism* as an example of the previous biases, where we believe that what we are stating is the complete truth about a subject, and those who contradict must be ignorant of the subject. Here are some other known biases:

- *Confirmation* bias: some people tend to seek information that confirms their opinions or previous beliefs. Significant importance is given to sources favoring the existing idea while ignoring (on purpose or by other means) and avoiding other sources that contradict it.
- *Dunning-Kruger* effect: due to a lack of self-awareness, some people are unable to measure their ineptitude. Hence, they wrongly believe in their superiority in some subjects. In other words, people that are unaware of their ignorance tend to think they know everything about a certain subject.
- *Survivorship* bias: the stories that living people tell about a certain fact might deflect the reality, as the absentees (dead) are unable to present their opinion. The bias usually reflects the selectivity in sampling information by relying on convenient, easily accessible, or simply available resources. That undermines the opinion or the role of information provided by the other resources.

Naturally, we can lie, act wrongly, be lazy, and not abide by the standards, and there is no difference in dealing with the Web. The *Utopian* Web with agreeable and reliable metadata does not exist (Yet) [6]. The "yet" here is yet to be answered in this thesis. But we attempt to shed some light on some of these problems and offer some solutions to specific questions.

**In the Web, we *un*-trust** The Web is an enormous source of information. We can be more specific about the reliability of the latter, judging that they might be *true* or *false*, but even that does not reflect the reality yet. For such open information, we have to aspire to the exactitude of our judgment while expecting its falsehood, similarly to the theory of knowledge *decay* [7].

We may also see the Web as a "selective" abstraction of reality. If that is not enough as a difference, our behavior itself changes when we are online. Some researchers argue that we might not behave the same way while on the Web [8]. And when anonymity is preserved, more freedom is allowed. Our public image is always a subject of polishing by being selective with our communications, and the Web offers more control over that.

The Web offers leverage to some entities over others. When misused, news websites can act as tools to control the flow of information to populations of entire countries. One solution to defend against such abuse is to create new websites or social media accounts to publish what is believed to be the correct information. However, the leverage will always favor more prominent websites with a broader reach and a better ranking on search engines. Media can smear anyone using their large follower base, and the victim will not have been given a chance nor the stand to defend themselves. The uncertainty of information–in this case– is harmful, and the existence of a way to equalize the effect of certain websites with small websites is crucial. We can project such facts to a Semantic Web format, which would help build a bridge between the different narratives but still raises problems. One is the absence of a formalization of (un)certainty and a comparison metric between such information to distinguish which narrative is more accurate. Moreover, data in its raw format might be incomparable. Its translation to a Semantic Web format with explicit uncertainty measures would distinguish and select what information we value to be true.

**Our Doubts are Abstracted** When we get an email with a calendar event, we are often presented with three choices: whether to accept, deny or state that we are unsure of our attendance. The latter opens the door for many interpretations of our schedule: we are yet to decide, we have two or more events with the same priority, we are waiting for an important event to happen, etc. Those interpretations go beyond a simple yes/no answer, but the consequence of their existence is that we cannot determine our schedule in a certain period. The "maybe" varies from one person to another but is eventually linked between all the subjects of interest. If we think

in general about the instruments we use, the uncertainty is by design included in large labels such as “In case of”. Random events, outlier observations, unguaranteed futures, unknown or inaccessible previous information are all sources of doubt.

“Humans are humans”. It should not be perceived as an undermining statement for what humans can do or of their achievements. After all, humans conceived these systems. Still, parameter miscalculation may lead to the explosion of a rocket, and job applications may be lost because the applicant mixed the month and the day in their birth date, so the system could not fetch their profile. Uncertainty opens the door for ongoing trials for information validation. It requires us to stay attentive to any new information to re-evaluate the coherence of our beliefs. The previous discussion also points that it is better to get familiar with the unknown and try to describe its boundaries when possible, instead of ignoring the existence of its pebbles and therefore risk the accumulated consequences. Uncertainty is also related to the growth mindset, as described by Professor Carol Dweck [9]: in contrast with the fixed mindset, a growth mindset is always eager for challenges and is never stopped by the limitations. It does not thrive in the safe area where it always feels sound and intelligent but looks up to challenges as means for progress. We think that this is the same case for a Web running way ahead of its time and run by billions of people worldwide. The Web (and the Semantic Web) is made of changes and should never perceive uncertainty as a burden but live up to it.

The vision of interconnected intelligent entities, humans, and machines, requires a consensus in data representation, but that is not everything. The consensus should also be about the sense of data and the transformation that might happen to the data afterward. Such data have to be reusable, no matter who is using it. All of the previous assumes the sanity of users, producers, and data. When uncertainty is on the line, going underline about it is never a solution. Instead, we may think of having dialogues about the leaks, the impurities, and the different angles from which we perceive our data. The consensus here is not only about agreeing on the truth and how to deal with it. It is also about the abstraction of the considered reality to evaluating that truth.

The authors of the 2020 Dagstuhl Manifesto thoroughly discussed information personalization, privacy, quality, and freedom on the Web. All of the previous problems also concern the specific case of the Semantic Web and raise lots of questions, such as the necessity to make everything machine-readable or give machines access to more/less information. We believe that the first step of remedy is to get familiar with the limitations that our biases present to us by understanding and formalizing uncertainty.

**The goal of this Ph.D. thesis is to investigate the dimension of uncertainty on the Semantic Web. From representing the uncertainty in a standard format, understanding, manipulating, extracting, to propagating it to other sources.**

## 1.2 Problem and Research Challenges

We study three main challenges in supporting uncertainty handling on the Semantic Web.

- Challenge I is **representing uncertainty on the Semantic Web**: this can help to formalize the lack of information, give an idea about data quality, and be a base for other transformations of the knowledge graphs on the Semantic Web (*e.g.*, update, merge, or refute the information.).
- Challenge II is **making uncertainty explicit in knowledge graphs**: this ensures that the data providers are transparent about their data.
- Challenge III is **propagating uncertainty on the Semantic Web**: this allows shipping uncertainty information alongside data and helps to understand the decisions made to select specific pieces of data.

## 1.3 Research Questions and Contributions

In this thesis, the general research question that we study is: **How to qualify the reliability of data on the Semantic Web?**. To be specific, the scope of this thesis focuses on **uncertainty representation**, **manipulation**, **extraction**, and **propagation**.

To go further into details, we present here the research questions we investigated.

**RQ I:** *How can we represent uncertainty on Semantic Web while respecting the existing standards and technologies ?*

**Hypothesis I:** *Offering an ontological representation of uncertainty, allowing to annotate elements of knowledge graphs (and knowledge graphs themselves) with uncertainty values following various theories of uncertainty.*

**Contribution I:** *We propose an ontology for uncertainty metadata (mUnc) that we aligned with an existing query scripting language (LDScript) to create a framework for uncertainty representation and manipulation. We use this framework to offer an uncertainty dictionary for some current uncertainty theories in the wild. The manipulation of uncertainty values is done using remote functions linked to each of the approaches. We demonstrate the possibility of declaring custom uncertainty theories to fit with the requirements of each user.*

The representation of uncertainty in a standard way is the first step toward publishing and reusing it. Some of the questions we treated under this challenge are:

- is the Semantic Web stack compatible with a representation of uncertainty? Do we need to extend the existing standards?
- what layers should uncertainty be included within?

- how to deal with the heterogeneity of uncertain data following different uncertainty theories?

**RQ II:** *How can we access uncertainty data on Semantic Web while respecting its context and exchange it ?*

**Hypothesis II:** *Offering an contextual representation of uncertainty, allowing to represent multi-level uncertainty of knowledge graphs, and transfer negotiated heterogeneous uncertainty values to enable their composition.*

**Contribution II:** *We proposed methods to access and manipulate uncertainty values depending on their context. We offered different readings for uncertainty depending on the level of granularity it includes. We provided translatability extensions for mUnc to enable transforming and negotiating uncertainty.*

The representation of uncertainty in a standard way is the first step toward publishing and reusing it. Some of the questions we treated under this challenge are:

- how to access and interpret uncertainty information encapsulated in different levels in the same source?
- how to combine uncertainty information from several levels in the same source?
- in the case of context-related uncertain data, how to deal with uncertainty in nested contexts?
- how to translate and negotiate uncertainty when presented under a different formalism ?

**RQ III:** *How can uncertainty be extracted from knowledge graphs that do not provide explicit uncertainty information?*

**Hypothesis III:** *Finding a reference point to compare the knowledge graph may help. The projection should be made in a specific context, assessing how good the answers provided by the target knowledge graph can be.*

**Contribution III:** *We propose a framework (Archer) to extract uncertainty from existing knowledge bases with missing uncertainty annotations. We use a task-specific comparison to evaluate the uncertainty of focused corpora of data around selected resources representing anchor points of this evaluation. The extracted uncertainty follows the representation and manipulation principles provided by the first contribution.*

One of the main issues we faced during the thesis was the lack of data with explicit uncertain information. To cope with that, *Archer* allows extracting and annotating data with uncertainty metadata. Some of the questions we treated under this challenge are:

- how to formalize tasks to relate to for uncertainty extraction?
- what resources to focus the analysis of uncertainty around? what data to select around these resources?
- how to use the outcome of the analysis for further exposition and updates?
- how to annotate data with uncertainty information? and what indication would that uncertainty provide?

It is crucial to establish a "dialogue" mechanism to reach some consensus between several uncertain data providers on the Semantic Web. Therefore, the conversation must follow a unified standard. For that, we discussed in our perspectives a view for consensual dialogue between data sources based on the existing data in the LOD cloud and the passage from local beliefs to common knowledge, all with respect to the ambivalence of the Web. Some of the questions we openly discuss are:

- if all sources abide by the dialogue rules, how can it be achieved practically?
- should all sources implement uncertainty ?

- how to transform local beliefs into common knowledge using uncertainty as a dimension?

To build upon previous research work, we analyze the existing results treating uncertainty in the Semantic Web. The work approach is reflected in the organization of this manuscript. After this introduction, we describe in *Chapter 2* the preliminary notions related to uncertainty and the Semantic Web. We provide an overview of the technologies used in the Semantic Web stack and the limits of uncertain data. Afterward, we deliver in *Chapter 3* a representation for uncertainty on the Semantic Web. We discuss our contribution of the uncertainty ontology *mUnc* and the methods for annotating statements with uncertainty. *Chapter 4* discusses uncertainty management and access in a contextualized view and the reading of uncertainty inside contexts. For sources without explicit uncertainty information, we present a framework in *Chapter 5* enabling the evaluation of uncertainty, based on both syntactical and semantic similarities with entities from a reference source and within a specific use case. We conclude with a discussion about dialogue between sources and how we can reach a sure and consensual universe, following that with a view this work with our perception of the reality and perspectives of this research.

## Scientific Production

Here is an exhaustive list of the productions realized during and with relation to the current thesis:

### International Conferences

- **Ahmed El Amine Djebri**, Andrea G.B. Tettamanzi, Fabien Gandon. Publishing Uncertainty on the Semantic Web: Blurring the LOD bubbles. ICCS 2019 - International Conference on Conceptual Structures, July 2019, Marburg, Germany. <https://hal.inria.fr/hal-02167174>

- **Ahmed El Amine Djebri**, Andrea G.B. Tettamanzi, Fabien Gandon. Linking and Negotiating Uncertainty Theories Over Linked Data. WWW 2019 - LDOW/LDDL Workshop of the World Wide Web Conference, May 2019, San Francisco, United States. <https://hal.inria.fr/hal-02064075>

- **Ahmed El Amine Djebri**, Andrea G.B. Tettamanzi, Fabien Gandon. Task-Oriented Uncertainty Evaluation for Linked Data Based on Graph Interlinks. International Conference on Knowledge Engineering and Knowledge Management EKAW 2020. October 2020. Online Venue. <https://hal.inria.fr/hal-02933190>

- **Ahmed El Amine Djebri**, Antonia Ettorre, Johann Mortara. Towards a Linked Open Code. The 18th Extended Semantic Web Conference ESWC 2021. Online Venue. <https://hal.inria.fr/hal-03190617>

### Miscellaneous

- **Ahmed El Amine Djebri**, Andrea G.B. Tettamanzi, Fabien Gandon. mUnc Vocabulary Specification. <http://ns.inria.fr/munc/>
- Mehwish Alam, Tayeb Abderrahmani Ghorfi, **Ahmed El Amine Djebri**, Omar Alqawasmeh, Amina Annane, *et al.* Linked Open Data Validity – A Technical Report from ISWS 2018. 2019. <https://hal.inria.fr/hal-02087112>

# Part II

# Uncertainty integration in the Semantic Web

# Chapter 2

# Introduction to Uncertainty and the Semantic Web

Previously, we presented the general context of our problem and the aspirations and goals of this thesis. The representation of Uncertainty on the Semantic Web requires an understanding of the notions of the Semantic Web and a deep understanding of the technical choices and the theoretical background of uncertainty.

This chapter is structured as follows: Section 2.1 presents an overview of the Semantic Web and the different technicalities linked to the notion. Section 2.2 offers a glimpse over the notion of uncertainty from different backgrounds and focuses on the aspects concerning the context of this work. We bridge between the two concepts of uncertainty and Semantic Web in Section 2.3. Then, the conclusion of this chapter paves the way to further discussion about the needed steps for the integration of uncertainty in the Semantic Web.

## 2.1 Semantic Web: the Web that makes sense for machines

Since 1989, starting from the publication of the first report authored by Tim Berners-Lee describing his vision of the *World Wide Web* [10], this latter continues the evolution towards an open universe of information. The same report initiated a foundation for what is known now as *Linked Information Systems*, which were further discussed in another article of his, focusing on the *semantic* aspect of the Web. This vision targeted the accessibility of information on the Web to different types of agents, helping to move towards ubiquitous artificial intelligence [11].

The Semantic Web nurtures the processes allowing the Web to become a "machine-friendly" space. This starts with the transition from a *document-based* view of resources on the Web to a *data-based* one, as shown in figure 2.1.[1] Allowing more granularity in data representation and taking advantage of structured data already encapsulated in the "Web of Documents" helped to turn it into a Web with interlinked pieces of data (i.e., *Linked Data*). These small interlinked atoms allowed machines to explore, read, understand, and use such pieces for other purposes such as reasoning. The transition was possible due to the introduction of new technologies, standardized by the World Wide Web Consortium *W3C*.

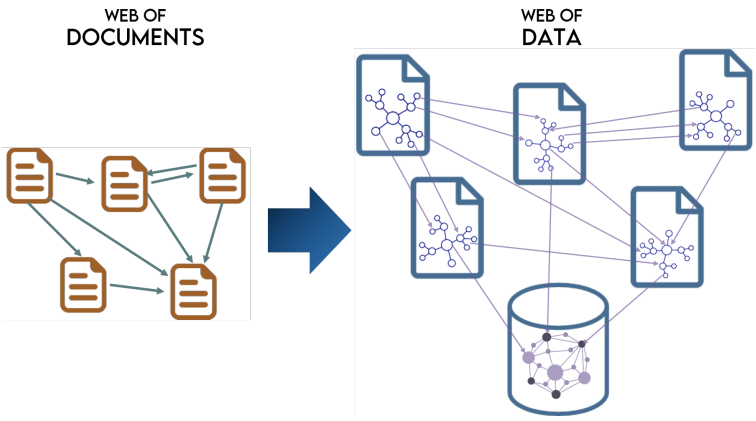



FIGURE 2.1: Transition from a *document-centered* view to a *data-centered* view.

[1] The arrows between documents represent URLs, while arrows linking the graph nodes are IRIs.

### 2.1.1 Linked Data

The current architecture of the Web relies on a set of technologies and protocols, allowing the presentation and access to the resources (documents). These technologies were needed to identify, describe, and access resources on the Web:

- Hypertext Transfer Protocol *HTTP*:[2] to access and retrieve information from existing resources on the Web.
- Hypertext Markup Language *HTML*:[3] to describe the structure of a Web page.
- Uniform Resource Locator *URL*:[4] to identify and refer to resources on the Web.

Changing the view from *document* to *data* required the introduction of new terms and technologies, as shown in figure 2.2:

- ***URI* (Universal Resource Identifiers[5]) replacing URL**. Instead of "*representing what exists on the Web*" (i.e., files, documents, pages, etc.) it allows "*representing, on the Web, what exists*" (i.e., entities, concepts, data) [12]. While URLs can only refer to documents on the Web, URIs (and IRIs after, for Internationalized) allow identifying physical and conceptual entities (like a person, a car, or the concept of knowledge), hence offering the possibility to bridge the gap between the Web and the reality.
- **RDF replacing HTML**. Instead of describing the documents using HTML, the Resource Description Framework *RDF* is used as a standard[6] to describe both entities and the relationships between them. More details about RDF are provided in section 2.1.2.

[2] https://tools.ietf.org/html/rfc2616
[3] https://tools.ietf.org/html/rfc2616
[4] https://tools.ietf.org/html/rfc1738
[5] https://tools.ietf.org/html/rfc3986
[6] https://www.w3.org/TR/rdf11-concepts

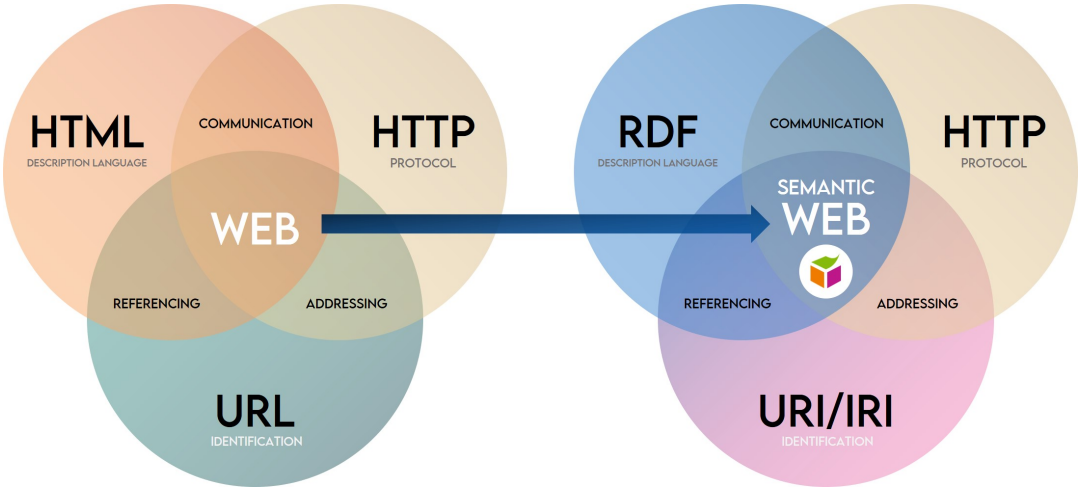



FIGURE 2.2: Transition from the Web to the Semantic Web [12]

### 2.1.2 Semantic Web Stack

The Semantic Web "layer cake" is a stack of technologies summarizing the process of data description and interpretation. The layer-based architecture ensures that information in each layer is conforming with and less general than the lower ones. The version we provide in figure 2.3 is the one of the *W3C*. The technological stack is based on dereferenceable entities, identified each by an IRI. The stack provides multiple layers:

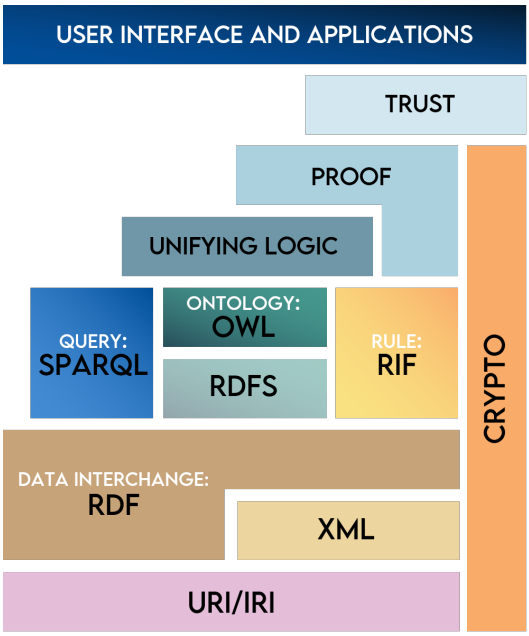



FIGURE 2.3: The Semantic Web "Layer cake" [13]

- **Data**: In the Semantic Web, data is represented following the Resource Description Framework RDF. The standard allows representing data in a machine-readable format that can be understood and exchanged between the different agents on the Semantic Web. Atoms at this level (smallest piece of data) are RDF triples in the form *<subject, predicate, object>*. For instance, expressing that "entity *XYZ* is of the color *Red*", translates to a triple linking the IRI `ex:XYZ`[7] representing the entity *XYZ*, with the IRI `ex:Red` representing the color "`Red`" as a resource, using a predicate (i.e., property) called `ex:hasColor`. In this case, all three elements of the triple `<ex:Apple ex:hasColor ex:Red>` are IRIs. The RDF triples form a directed labeled graph (*i.e.* RDF graph) in which the *predicate* of each triple links its *subject* to its *object*.

  The triple-based architecture and abstraction in the framework are the same, yet RDF requires a serialization syntax for data publication. RDF data can be expressed in many syntaxes, like *XML* or *JSON-LD* (JSON for Linked Data), or more intuitive syntaxes[8] such as *Turtle* and *N-Triples*. Moreover, notations such as *RDFa* permit embedding RDF data in HTML pages.

- **Schema**: The essence of the Schema layer is structuring data existing in the lower level (RDF triples). This layer allows the enrichment of the stack with extra functionalities to describe the "*data about data*" or *metadata*. RDFS (for RDF Schema) is a W3C standard to represent schemas (ontologies), providing a semantic extension to RDF [14]. It allows building relationships between classes (subsumption relationships) and for properties to have additional specifications about their semantics (like domain and range).

  The Web Ontology Language *OWL* extends the capacities of RDFS with more expressive, rigid, and descriptive features [15]. For instance, it enables expressing restrictions or declaring equivalences by introducing predicates such as `owl:sameAs`, but at the same time adds some constraints, such as the prohibition to use classes as instances. OWL enables annotating the ontology itself,

[7] The example prefix "`ex:`" refers to the URI of the namespace in which the entities are defined.

[8] https://www.w3.org/wiki/RdfSyntax

enabling stating metadata such as the provenance of the ontology, authors, version numbers, and backward compatibility information. OWL also comes in different variants (*profiles*, or *flavors*) in expressiveness and complexity.

- **Query**: The role of this layer is to allow access to data in lower levels. Similar to *SQL*, the Semantic Web presents its RDF-flavored technologies and protocols, offering the possibility to read and question the triples using formalized queries and triple patterns. The *SPARQL* query language allows harnessing information in an advanced way, offering the needed tools for complex web-based applications to thrive [16]. SPARQL relies on data and does not provide inferences on its own. Its only role is to transform the description of the application's query, then return its results as an RDF graph or as bindings.

- **Rules**: This layer is introduced to cope with the lack of expressivity in the schema layer. Rules focus on representing a general approach for discovering and creating new links based on the existing data. Rules are not only about controlling the inference steps which the inference engine makes, but also to limit the process with a proper set of constraints. Reasoning and inference are based on the two layers (i.e., schema and rules) to generate new data, transform the existing ones, or verify data integrity and explain what it should look like in shape. The W3C maintained language to explain rules is the *Rule Interchange Format RIF*, mainly destined to exchanging rules among rule systems. Rules can be invoked in both directions: associating given information with an inferred one (bottom-up) or associating a query with sub-queries satisfying its answer (top-down).

  Some rule languages like *SPIN*[9] or *SHACL*[10] can be used for data validation purposes. *SPIN* uses SPARQL queries to mimic the Object-oriented model on the Semantic Web by linking resources (objects) to rules (methods) describing the behaviors or constraints of objects. *SHACL* defines a set of constraints,

[9] https://spinrdf.org
[10] https://www.w3.org/TR/shacl

called shapes, around graph nodes and their property paths. It produces conformance reports to check whether data in a graph is conform with predefined shapes.

- **Unifying Logic**: The idea of standardizing and bridging granularized data together targets mainly reasoning about facts to create new links, assemble the pieces to visualize a bigger picture, and untangle the different stories that the data may tell. The use of ontology languages and rule languages enables, to some extent, inferencing and verifying the integrity of newly generated data. Despite that, the stack does not specify a standard logic to work with data on a global level: in particular, interpretations to use when data is missing, incomplete, invalid (We discuss later an uncertain World Assumption in Section 2.2.1). The unifying logic should enable digesting data from all lower levels into unique ones that make sense to the agents. The different families of logic used all along with the stack (*i.e.* Description logic in OWL, First-Order Logic, and Horn Logic in rules) seem to offer different flavors (but not one) to reason about data. OWL itself comes with many variants, each with a different level of expressiveness.

- **Digital Signature or Crypto**: This layer goes all the way from the bottom of the stack to its top, backing up the other layers and making part of the foundation of the *trust* layer. The idea behind cryptography is to ensure that the presented data comes from a trustworthy source and via a reliable channel. Statements can be either part of signed documents, verified with a key, or have been logically derived from verified statements. The other aspect of this layer is *encryption*, for data needs to be transferred securely.

- **Trust and Proof**: The foundation of *trust* comes from the previous layer (*Crypto*) and the fact that data needs to be traced to its origins, hence *proven*. The role of the *proof* layer is to ensure finding breadcrumbs leading to the processes or the sources that created the data and being able to explain how data was generated and for what purpose. Proof engines need to elaborate a clear path for the *trust* layer to analyze and choose whether the signed and proven

data is to be trusted for applications to use.

### 2.1.3 Linked Open Data

The adoption of the aforementioned technological stack by sources on the Web influenced further the culture of *openness* of information on the Semantic Web: Since Web documents were accessible, data issued from them or directly made public through the Semantic Web is made accessible for everyone too. However, some restrictions/recommendations must be respected to achieve that and the first of them is linked to the previous influence. For the Linked Data to be open, it must comply with the 5-stars recommendations for data publishing. The recommendations require data to be openly available on the Web, in a machine-readable and non-proprietary format, published with respect to the Semantic Web standards and is linked to other data. All sources publishing their data publicly and in open-access on Semantic Web are contributing with *Linked Open Data*. One illustration of this interconnected Web is formed in the *Linked Open Data* cloud or *LOD* cloud,[11] illustrated in figure 2.4. The cloud includes available datasets of more than 1000 triples from the Semantic Web, with some datasets reaching the order of billions of triples. As of May 2020, the cloud assembles 1255 datasets linked with 16174 links and covering several domains (*i.e.* Geography, Government, Life Sciences, Linguistics, Media) [17].

The potential of *Linked Open Data* is enormous: search engines can thrive, definitions enriched, and exploratory search is easier than ever. The interlinks between datasets allow to refer to similar entities and reuse their descriptions, assembling the scattered facts about them and seeing a bigger picture or accessing extra information that one dataset could not afford. The advancements made in the field of Artificial Intelligence permit nowadays to leverage the use of this kind of data as inputs, hence helping produce, curate, share and maintain corpora and datasets [2]. In the following paragraphs, we present some of the significant cross-domain datasets in the *LOD*

[11] https://lod-cloud.net

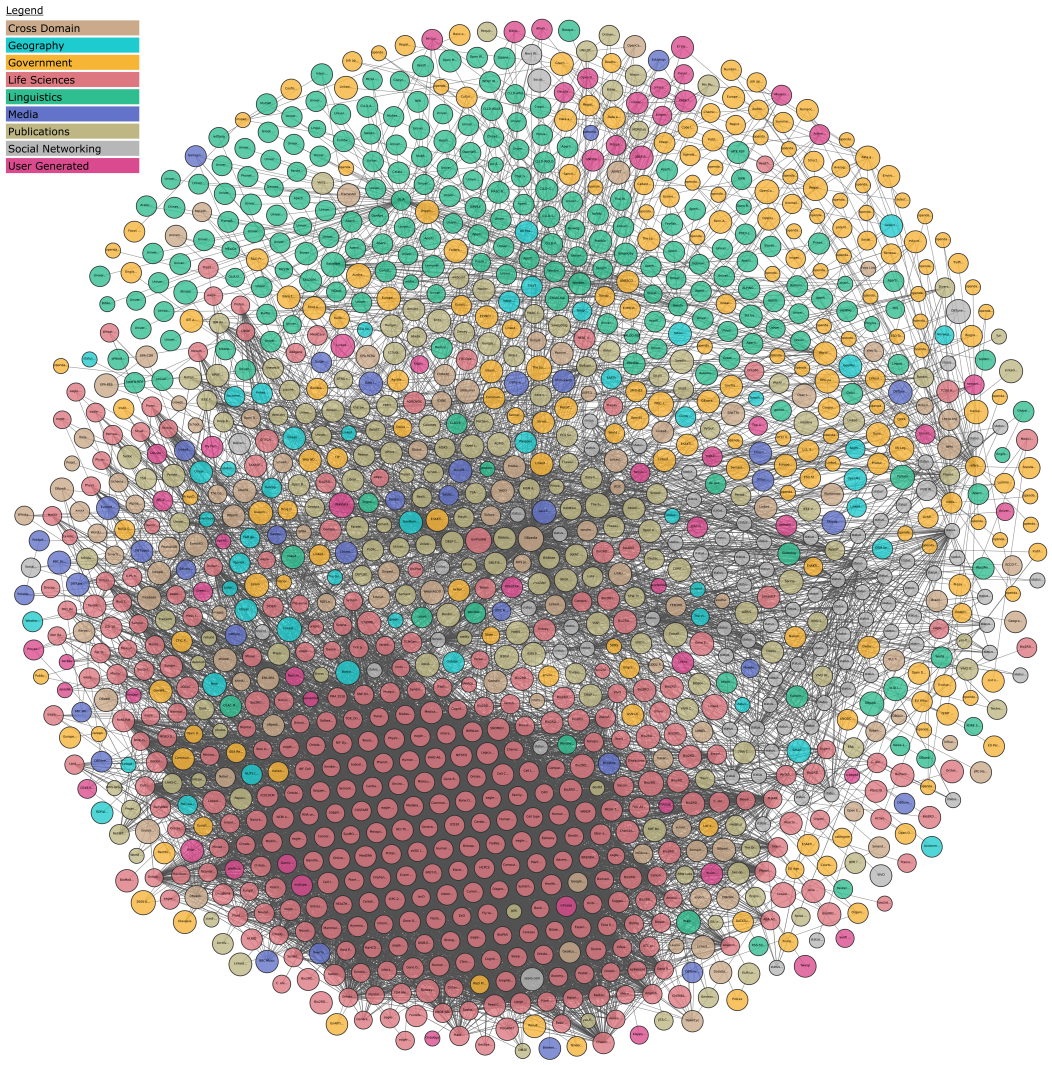



FIGURE 2.4: the Linked Open Data Cloud as of May 2020[17]

cloud. The ones we mention here integrated the cloud in their early stages.

**DBpedia** [12] It is a knowledge base aiming to make the existing knowledge on Wikipedia (unstructured or semi-structured) available on the Semantic Web. The user community of DBpedia contributes by providing the mappings linking the information representations in Wikipedia with the DBpedia ontology [18]. The mapping process focuses mainly on Infoboxes (tables that appear on the side of an article to indicate its relevant facts). As a consequence, users cannot directly alter data on DBpedia. If one does want to, they must edit the corresponding Wikipedia pages and wait for the extraction process (which can be done live in DBpedia-live [19]). DBpedia is one

[12] https://www.dbpedia.org

of the pillars of the LOD cloud, representing one of its central nodes that has been maintained since the first iterations. It covers information in 125 languages, with the English chapter alone containing more than 28 million triples in multiple domains [18].

**Wikidata** [13] It is one of the projects maintained by the *Wikimedia Foundation*. Wikidata started in 2012 and aimed at offering a machine-readable representation of existing knowledge in the Wikimedia projects. The project assembles all languages from the projects in a unique, easily accessed interface. Wikidata is a community-maintained knowledge base where everyone with an account can add, update, or delete triples. It counts more than 93 million interlinked entities.

**Freebase** [14] It was one of the first initiatives to link data on the Web. Freebase shared the same spirit with DBpedia regarding knowledge extraction from Wikipedia but offered a broader view as it harvested other sources. The project was started in 2007 by *Metaweb* and was discontinued in 2015 after being absorbed by Google, with the last dumps of Freebase residing in Google's Knowledge Graph. One common point between Freebase and Wikidata is their openness to changes directly by their users, unlike DBpedia that requires altering Wikipedia articles or the mappings that allow it to extract information. However, unlike Wikipedia and DBpedia, Freebase was maintained by domain experts (and not community members).

**YAGO** [15] It defines itself as a simplified, cleaned, and reasonable version of Wikidata. YAGO aims to upgrade the usability and reliability of Wikidata by imposing a strict type hierarchy with semantic constraints. The project started with a combination of Wikipedia and Wordnet [20]. It is moving now in the fourth version of YAGO towards the combination of Wikidata and schema.org [21].

[13] https://www.wikidata.org
[14] https://developers.google.com/freebase
[15] https://yago-knowledge.org/

Alongside data, the *Linked Open Vocabulary*[16] initiative allows access to open schemas that were published to map the previous datasets or to be used to annotate data as a standard way to represent the concepts and their relations.

For Linked Data to be 5-stars, it has to be (i) available on the Web, (ii) *structured*, (iii) presented in a *non-proprietary* format, (iv) follow W3C standards (*i.e.* such as RDF) to be referenced, and (v) *linked* together to provide a context [22]. The openness of *LOD* cloud requires all sources to comply with the previous 5-stars rule to publish their data. Still, it gives no constraints about the uniqueness and the compatibility of such data. Two different sources may make contradictory statements about a single entity. Moreover, the *Unique Name Assumption* is not respected: one resource may have different IRIs referencing it. For instance, one user may have an account at Facebook, Twitter, and Pinterest. In each of the former websites, the same user has a different set of information about them and a different permalink. For that, standards like OWL and SKOS[17] provide tools to link similar and identical resources together (`owl:sameAs`, `skos:exactMatch`, `skos:closeMatch`).

### 2.1.4 RDF Serialization Syntaxes

In the universe of Linked data, triples are the atoms. They consist of components that provide meaningful information only when put together. The existence of concepts means nothing if the concepts are not interlinked and put in action. Hence, a statement consists of a *subject* linked to an *object* using a *property*. Some views would extend this to include the *context* in which the statement has been asserted (yet no standard definition of a context has been made).

Table 2.1 shows some of the existing syntaxes of RDF serialization and examples of the representation for the triple `(ex:Apples, ex:hasColor, ex:Red)`. The two factors we focus on are human readability and storage (Compression). The first one depends on the ease of distinguishing resources and linking them to their properties, objects,

[16] `https://lov.linkeddata.es`
[17] https://www.w3.org/TR/skos-reference/

and types. The second one is about the size of the serialization with and without the presence of redundancies. Some of these syntaxes were adopted by W3C as standards or recommendations, such as *RDF/XML* [23], *Turtle* [24] (followed by TriG [25]), *N-triples* [26] (followed by N-Quads [26]), and *JSON-LD* [27].

| Serialization format | Human readability | Storage | Example with the triple (ex:Apples, ex:hasColor, ex:Red) |
|---|---|---|---|
| RDF/XML | + | + | <?**xml version**="1.0"?><br><rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22−rdf−syntax−ns#"<br>xmlns:ex="http://example.com/"><br><rdf:Description rdf:about="http://example.com/Apples"><br><ex:hasColor rdf:resource="http://example.com/Red" /><br></rdf:Description><br></rdf:RDF> |
| N-Triples | + | ++ | <http://example.com/Apples> <http://example.com/hasColor> <http://example.com/Red> . |
| Turtle | +++ | +++ | @prefix ex: <http://example.com/> .<br>ex:Apples ex:hasColor ex:Red . |
| Trig | +++ | +++ | @prefix ex: <http://example.com/> .<br>ex:MyWebsite {ex:Apples ex:hasColor ex:Red .} |

TABLE 2.1: Examples of existing serialization syntaxes for RDF

The first and the native syntax was an XML representation of RDF data. It consists of a nested representation of entities and attributes. It is considered quite verbose, which resulted in the adoption of more intuitive syntaxes.

Turtle (for Terse RDF Triple Language), a flavor of N3 stripped of some syntactic sugar, brought more readability to the representation of the code by allowing predicate lists and object lists. Overall, the N-Triples format is the most serializable, with direct use of IRIs, explicit triple structure, and no predicate or object lists. Other formats such as TriG or N-Quads (successors of Turtle and N-Triples, respectively) may be included in the aforementioned listing. Both included the possibility to add a fourth element to statements consisting of the graph in which the statements are asserted.

There exist other embedded formats to serialize RDF. For instance, *RDFa* [18] enriches the HTML representation of Web pages with semantic attributes, making it possible to annotate parts of the Web page with RDF data. On the other hand, JSON-LD became a common alternative with a familiar web-ready syntax. Another example is *Microdata* [19], using the `itemscope` attribute to enforce a semantic reading of parts of a Web page and produce machine-readable labels.

[18] https://www.w3.org/TR/rdfa-primer/
[19] Part of the *HTML* living standard: https://html.spec.whatwg.org/

### 2.1.5 Syntactical Representation of RDF Annotations

To be more thorough, we discuss the different approaches in the literature for annotating RDF statements with metadata in the next part. We focus here on the annotation approaches at the statement level.

**Reification** It is part of the RDF standards presented in 1999 [28], [29]. Reifying an RDF statement consists of adding a resource representing the statement (in addition to the statement itself) and link the new resource with any metadata we want to use as annotation for the data. This representation is verbose. It also touches the mere existence of the statement in its natural form (subject, predicate, object), hence canceling its natural semantics and the assertion of its information. In [30], the authors confirm that the reification is the only standard way to represent annotation for RDF Triples and still the only one compatible with all the RDF Data Repositories for the fact that it does not require any extensions. There exist other abbreviated variants of standard reification, consisting of representing each triple `<s,p,o>` with two triples as `<:i,rdf:subject,:s>`, `<:i,:p,:o>` with the subject of the statement in one triple and the property-object pair in the other [31].

**N-ary properties** It consists of using a middle resource instead of the object of the statement [32]. This middle resource is linked to the object above using the property *rdf:value* and can be annotated with metadata. This approach is less verbose than standard reification but inherits that the statement is no longer asserted.

**Single named graph** It consists of encapsulating a single statement into a named graph and use the named graph as reference to establish the link between the statement and its metadata. Unlike reification, this approach keeps the structure of the statement, preserving its semantics.

**Singleton property** It consists of creating classes of unique properties from the existing property instances linking subjects to objects. The singleton property classes are then used to annotate the statements that use it [33]. For example, one may create `ex:label_1` a singleton property from `rdf:label` that should inherit its semantics and use it instead of the main property to transfer its semantics and annotate the edge. This approach does not preserve the semantics of statements as it is, as it alters the purpose of properties in the ontological definition. Another downside is ending up having many unique properties, singletons of a single one.

**RDF-star** This approach proposed by Hartig et al. [34] consists of adding the notion of *embedded* triples. The latter are triples encapsulated in the annotation `<< >>` and can take the place of subjects or objects in regular triples. Another addition is that embedded triples are not automatically asserted. Hence we may describe metadata about a triple without asserting it with the previous annotation, or use the delimiters `{|` and `|}` to both assert and annotate it. That might be useful in case of redacted triples. Meanwhile, embedding triples may be problematic in statements with long literal objects (i.e., a descriptive text of 1000 words, with 30 attributes). In some direct RDF syntaxes, we must repeat this abstract triple as many times as the meta dimensions. The authors proposed an extension of Turtle to cope with the problem and offer tools for the aforementioned notions.

Table 2.2 illustrates the previous annotation methods when annotating the triple `<ex:Apples, ex:hasColor, ex:Red>` with information about its creation date and provenance. For example, we consider that the triple is issued from DBpedia and created on 01-01-2020.

The previous approaches rely on the native idea of RDF. As mentioned before, some of the serialization syntaxes, such as N-Quads, use n-tuples instead of triples. The additional elements can be used to associate the triple to one graph, give an ID to the statement, or be given specific semantics by the authors of the approaches. One example in [35] used *quintuples* to allow the annotation. The extra elements are

| Annotation method | Example (turtle-star) |
|---|---|
| Reification | @prefix ex: <http://example.com/> .<br>@prefix dc: <http://purl.org/dc/terms/> .<br>@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .<br>ex:S1 rdf:type rdf:Statement;<br>rdf:subject ex:Apples;<br>rdf:predicate ex:hasColor;<br>rdf:object ex:Red;<br>dc:created "01−01−2020"^^xsd:date;<br>dc:source <http://dbpedia.org/> . |
| Singleton Property | @prefix ex: <http://example.com/> .<br>@prefix dc: <http://purl.org/dc/terms/> .<br>@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .<br>ex:Apples ex:hasColor_1 ex:Red .<br>ex:hasColor_1 rdf:singletonPropertyOf ex:hasColor;<br>dc:created "01−01−2020"^^xsd:date;<br>dc:source <http://dbpedia.org/> . |
| N-ary properties | @prefix ex: <http://example.com/> .<br>@prefix dc: <http://purl.org/dc/terms/> .<br>@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .<br>ex:Apples ex:hasColor _:**a** .<br>_:**a** rdf:value ex:Red;<br>dc:created "01−01−2020"^^xsd:date;<br>dc:source <http://dbpedia.org/> . |
| Single Named Graph | @prefix ex: <http://example.com/> .<br>@prefix dc: <http://purl.org/dc/terms/> .<br>@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .<br>ex:Graph_1 {ex:Apples ex:hasColor ex:Red .}<br><br>ex:Graph_1 dc:created "01−01−2020"^^xsd:date;<br>dc:source <http://dbpedia.org/> . |
| RDF-star | @prefix ex: <http://example.com/> .<br>@prefix dc: <http://purl.org/dc/terms/> .<br>@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .<br><<ex:Apples ex:hasColor ex:Red>> dc:created "01−01−2020"^^xsd:date;<br>dc:source <http://dbpedia.org/> . |

TABLE 2.2: Examples of annotation methods for RDF triples

identifiers for the graph in which an occurrence of the statement is asserted and a key to metadata annotation for the statement in this context.

The authors of [31] remodeled subsets of *Wikidata* using the different annotation approaches previously discussed. Using their work, the team working on GraphDB [20]

[20] A graph database engine and RDF store. https://www.ontotext.com/

performed a comparative analysis for the number of triples, loading time, and the dump size in the different annotation approaches. The results in table 2.3 show that RDF-star outperforms the other approaches in the three previous aspects.

### 2.1.6 Lightweight Ontology Development Methodology

The passage between data and schemas requires understanding a set of techniques to harness the similarities and relations between the different instances. Such findings can be used to create and tweak models representing a small part of the world. These models can be compared, linked, and composed to create bigger or better ones. Engineering ontologies aims to build structured hierarchies that rule over data and inspire it.

The iterative process of conceptualizing entities and relationships between them needs to be well defined to ensure a working methodology. Online communities from several countries and throughout several years may gather around to build a hierarchy, and for that, the working method is amongst the first things to agree on. Sometimes the collaboration is done between strangers deciding to collaborate on one project, and collaboration tools offer the possibility to do so (i.e., Git for code). An example of iterative processes used for ontology development is *SAMOD* [37]–for *Simplified Agile Methodology for Ontology Development*–. It is organized in three steps within an iterative process, focusing on the creation, development, and documentation of models by using a set of applicable use cases. The iterations begin with a collection of information about the studied domain by the ontology engineer. This information serves as the basis for the development of a *modelet* (an iteration of a model) that will be checked for compatibility multiple times before publishing in the three different steps of the lifetime of an ontology: initially, after merging a new modelet with the model, and after refactoring the model. The tests consist of verification of the model, data, and both of them together facing query testing. This process iterates every time there should be an extension of the model in the future.

| Approach | Total statements | Loading Time (min) | Repository image size (MB) |
|---|---|---|---|
| **Reification** | 391,652,270 | 52.4 | 36,768 |
| **Single Named Graph** | 334,571,877 | 50.6 | 34,519 |
| **N-ary relations** | 277,478,521 | 56 | 35,146 |
| **RDF-star** | **220,375,702** | **34** | **22,465** |

TABLE 2.3: Comparison of the different annotation approaches [36]

The *SAMOD* methodology provide a test case at each iteration. It consists of the following elements:

- Motivating scenario *MS*;
- Informal competency questions *CQ*;
- Glossary of terms *GoT*;
- a formal model *T-Box*;
- an exemplar dataset *A-Box*;
- a set of queries *SQ*.

A motivating scenario [38] describes a set of intuitive examples of the problem informally. Competency questions [38] relate to this scenario and allow the production of a glossary of terms [39]. The formalization of the motivating scenario and the previous questions in a Semantic Web language such as OWL 2 results in the proposition of a *T-Box* and an *A-Box* of the model. A set of queries reflects the competency questions in a query language such as SPARQL.

## 2.2 Uncertainty in Information

Like any other human innovation, the Web reflects the genius and creativity of its creators as well as their flaws and imperfections. Nowadays, the Web is a place where "anyone can say anything about any topic". A simple user can open a free blog on *Blogger*[21] and start writing what is on their mind. Fake news, conspiracy theories, or misspelled names and places can (and do) exist on the Web. By extension, these imperfections transferred some lack of reliability to the Semantic Web.

The population of the latter with data is the result of two processes: (*i*) automatic extraction and conversion (Web documents to data), and (*ii*) manual (or

[21] https://www.blogger.com

semi-automatic) data entry. Both processes are at risk of error: extraction models can be imperfect or inaccurate, and newly added data can be incomplete, wrong, or invalid. In this case, both may be parts in generating unreliable outputs. This uncertainty can be due to a random event, *i.e.* mistyping a name while knowing how it should be written and entering it the right way. It also can be due to a chosen lack of information or an imposed one. In the first case, information exists but still was not accessed (*i.e.*, a course on JavaScript exists in the curriculum. However, the student still did not take it). In the second case, it cannot be accessed and is lost forever (*i.e.*, small details during historical events that were not documented hence lost).

The following sections discuss the different aspects of uncertainty and how it reflects on the Semantic Web. In an attempt to lay the ground for our research work.

### 2.2.1 Identifying Uncertainty in an Open World

The existing data in knowledge bases can be incorrect, incomplete, vague, etc. Each of the deficiencies mentioned earlier may result in different types of uncertainty [40]. The *Closed World Assumption CWA* presumes that the lack of knowledge refers to its falsehood. On the other hand, the *Open World Assumption OWA* identifies each statement by the existence of its affirmation or negation but treats the lack of information as an "unknown" information. While both assumptions treat the truth of statements as crisp values (0 or 1), they do not deal with uncertainty. In Table 2.4, we present the difference of truth-value interpretation $I(t)$ of an uncertain statement $t$ in a knowledge base $K$ following one of the assumptions as mentioned earlier and add our view over quantifying the presumed lack of knowledge by extending both assumptions. The qualification of `True` and `False` interpretations of statements in the last case is achieved by associating a pair of dual *uncertainty* measures $(\mu_t, \bar{\mu}_t)$. Hereby, any uncertainty measure $\mu_t$ (resp. its dual $\bar{\mu}_t$) linked to a statement $t$ has to be in the interval $[TF, TT]$ with $TT$ (Totally true) and $TF$ (Totally false) respectively.

We believe that when measurable and presented in a machine-readable form, uncertainty can be deduced, compared, and leveraged by applications in their processing.

| **Use-case** | $I_{\text{CWA}}(t)$ | $I_{\text{OWA}}(t)$ | $I_{\text{Uncertainty}}(t)$ |
|---|---|---|---|
| $t \in K, \neg t \notin K$ | $t$ | $t$ | $t$ |
| $t \notin K, \neg t \in K$ | $\neg t$ | $\neg t$ | $\neg t$ |
| $t \notin K, \neg t \notin K$ | $\neg t$ | $t \vee \neg t$ | $t \vee \neg t, (\mu_t, \bar{\mu}_t) \in [TF, TT] \times [TF, TT]$ |
| $t \in K, \neg t \in K$ | $\bot$ | $\bot$ | $t, (\mu_t, \bar{\mu}_t) \in [TF, TT] \times [TF, TT]$ |

TABLE 2.4: Interpretation of statement $t$ in *CWA*, *OWA*, and *OWA* with uncertainty

We note that it is a technical choice to use the affirmative form of a statement when mentioning uncertainty in the `Unknown` case (last row, last column). One may use $t' = \neg t$ instead, and the exact definition applies, or choose uncertainty measures that help to affirm $\neg t$ in contrast with $t$.

### 2.2.2 Origins of Uncertainty

The nature of the term that we are dealing with stresses the need for a clear definition, for what the literature offers has several connected and close meanings. When Shannon introduced *entropy*, he presented uncertainty as *information deficiencies* which can be reduced by obtaining more relevant information [41]. The term pops up in experiments and simulations to imply the range of values (*i.e.* the interval) in which measurements are included. For instance, authors in [42] stress the difference between vagueness and uncertainty. The former deals with the non-exactitude of values in an existing and well-defined interval, while the latter describes the absence of a context while defining the truth value of a statement. Klir *et al.* [40] define three types of uncertainty that already have established measurements: *fuzziness*, *non-specificity*, and *discord*. Marquis *et al.* [43] reduce the definition of uncertainty to two specific notions: (i) the **lack of data** and (ii) the existence of **contradictory** ones. The definition of uncertainty itself is challenging. It can be *epistemic*, *i.e.*, stemming from our ignorance (incomplete knowledge, lack of a model) of an entity or process, or *ontic*,

*i.e.*, representing the inherent randomness of a phenomenon or system (a roulette, for instance). Besides, the border between these two types of uncertainty is somehow blurred and arbitrary. It depends on our point of view and the level of abstraction of knowledge representation. Random uncertainty is irreducible and uncontrollable, but the epistemic uncertainty can be reduced or eliminated by introducing sufficient information.

In logic, uncertainty is often linked to *inconsistency*. One example is the immediate (or delayed) presence of contradictory information. Moreover, information can be true (or partially true) only in certain contexts. Therefore, instead of relying on logic with unretractable conclusions (monotonic), uncertainty appealed to other types of logic. For instance, *non-monotonic* logic allow beliefs to be updated based on newly presented information, and *paraconsistent* logic [44] handles contradictory information and inconsistencies in knowledge bases.

We believe that the term "uncertainty" holds more specification. It includes the different imperfections that data might have (vagueness, fuzziness, ambiguity, invalidity, incompleteness, etc.). In practice, we are leaning towards the definition of *decision-making* uncertainty, *i.e.*, when uncertainty refers to the fact that an agent is uncertain about the decisions to make due to an imperfection in data. The valuation of uncertainty is, in fact, a materialization of the existing imperfection. Here, we provide some examples of what an uncertain statement $t$ might point at:

- **Invalidity**: $t$ is wrong. *i.e.* providing a wrong name for a person.
- **Incompleteness**: $t$ provides incomplete information. *i.e.* the list of songs in one album referred to by the statement is missing some titles.
- **Inconsistency**: $t$ provides information that contradicts another statement. *i.e.* having two different delays for one flight or two birth dates for one person.
- **Risk**: the use of $t$ is risky. It is usually linked to a use case. *i.e.* one road is selected for transit, but there is a risk for it to be closed due to constructions.

- **Vagueness**: $t$ provides a vague, imprecise information. *i.e.* saying a person was born in the '90s instead of providing their date of birth.
- **Ambiguity**: may be confused with *vagueness*, but here $t$ provides multiple possible interpretations, due to the lack of complementary knowledge. *i.e.* Two students in the same classroom have the same family name, but no first name or ID number is provided.
- **Fuzziness**: $t$ states a fuzzy truth. While vagueness means not specifying a value in a defined interval, the problem here is the imprecision of upper and lower cuts of the possible instances of the information (the interval itself is imprecise). i.e., using the concepts *old* and *young* without proper definition, or in a form of a membership function.

In literature, the previous terms are treated differently, some works on uncertainty evaluation in textual documents deal with *vagueness* [45], other works on logic deal with *inconsistency* [46] and other works on ontology alignment deal with *fuzziness* [47]. We should mention that the previous list is not exhaustive.

We perceive Uncertainty as "*a result of imperfections, in the inputs or the model, generating defiant outputs whose mere existence, real value, or real meaning (semantics) we are* ***uncertain*** *of*". Some outcomes are either produced by error, their values are wrong and make no sense (invalid), or do make some sense (incomplete, vague, ambiguous, fuzzy). If we consider machine learning models, insufficient training data or using wrong parameters may provide skewed results. We relate to real-life scenarios of human error, such as mistyping a name during data entry, missing one line during data curation, or not selecting the best regular expression for textual data extraction. The same might happen for models performing Named-Entity Recognition tasks on a text where the validation scores are not perfect on the complete set of tests. The simple logical rule governs the previous relationship: "*the false implies everything*". Imperfect processes and imperfect inputs will generate anything, including errors. For such, we qualify the data as *uncertain*.

### 2.2.3 Uncertainty Theories

The quantification of uncertainty is not a novel intent. Klir [48] specified that uncertainty theories must satisfy four levels of challenges to be able to deal with the uncertainty of a specific type. According to the author, uncertainty theories have to:

- LEVEL 1: provide an appropriate mathematical representation of the conceived level of uncertainty
- LEVEL 2: offer calculus by which the values can be manipulated
- LEVEL 3: enable measuring relevant uncertainty in any situation formalizable in the theory
- LEVEL 4: present developed methodological aspects of the theory, i.e. procedures to make the various uncertainty principles work within the theory.

The first three levels are self explanatory. As for the fourth, it links to the uncertainty principles presented by the same author to be taken as standard rules for the use of theories in different use case and they are similar to . The principles in question are:

- The more informative results are the ones to be selected, to *minimize uncertainty.*
- The results of uncertain information should widen the range of uncertainty to cover all the constraints, to *maximize uncertainty.*
- Uncertainty should be preserved as it is transformed between different theories, for *uncertainty invariance.*

We enumerate some of the theories developed and used by researchers in various domains to cope with uncertainty.

**Probability Theory** The recurrent observation of random (ontic) phenomena allowed understanding more details about them. Observations show that facts remain random, but they show up over time with a certain frequency. As a result, defining a truth value of a fact is tied to defining their likelihood of being in a particular state, either based on observations (*i.e.* the number 3 in one dice shows up once every 20 times), or mere subjective definitions (*i.e.* the dice has six faces, so if we repeat the experience each of them is likely to show up once in 6 times). Thus, probability theory can have *subjective* or *frequentist* semantics. The first represents an assigned *degree of belief* and includes prior background knowledge (like the number of faces in dice and the fact that the dice is not tricked). The second represents the statistical frequency of events. It is based on infinite sampling: rolling the dice infinite (finite enough to tell) times, studying batches of samples from the population, and having the same result. Subjective probability is often described as *Bayesian*, as the first explanations about *prior*, *likelihood*, and *posterior* knowledge used in that calculus were introduced by Bayes in 1763 [49].

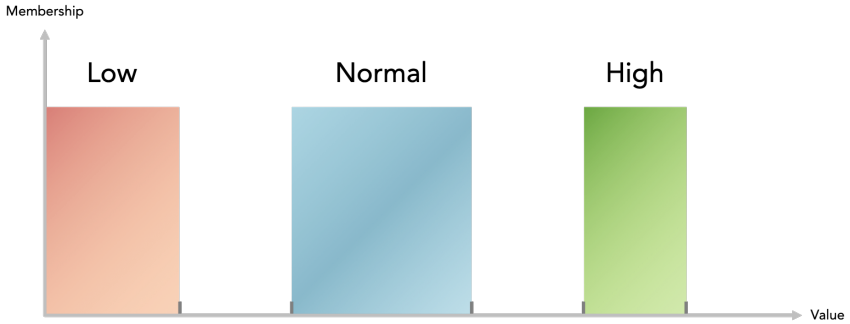
To go further, *Imprecise Probabilities* extend the previous definitions and allow using a probability interval (with upper and lower bounds) to describe how the probability may variate instead of single-valued probabilities.

**Possibility Theory** The reflection behind possibility is based on being close to a certain prototype. A possible outcome is not a certain one, as probability might state, but it refers to the fact that it is normal to have it, not surprising. However, when an outcome has more possibility than another, it is more plausible to be accepted. The degree of possibility $\pi_\Omega(x) \in [0,1]$ of an element $x$ in the universe of discourse $U_\Omega$ represents the plausibility of $x$ to be the correct value of an uncertain variable [50].

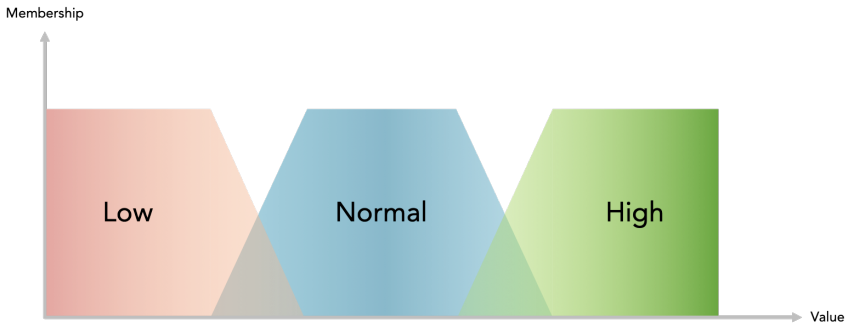
Based on the previous definition, the theory identifies two dual measures: *possibility* and *necessity*. The possibility $\Pi(A)$ of a subset $A \subseteq \Omega$ represents the upper bound of the possibility distribution of the elements in $A$, representing the consistency of $A$ with $\pi_\Omega$. On the other hand, the necessity $N(A)$ of a subset $A \subseteq \Omega$ represents the possibility that elements in the complement of $A$ noted $\bar{A}$ are not satisfied (or

possible), thus representing to what extent the knowledge implies $A$.

**Fuzzy sets** As their name suggests, fuzzy sets are sets with gradual, non-crisp, and *fuzzy* boundaries. Each element in a fuzzy set has a degree of *membership* to the set. The sum of the degrees of membership to all possible sets of each element sums to the unit. One element can be in multiple fuzzy sets at once with different degrees of membership. For instance, normal temperatures cannot be less than 20, and low temperatures cannot be more than 24. If the temperature is 23, it is neither normal nor low, but both. Each one with a degree of inclusion to the sets "normal" and "low". Figure 2.5 illustrates the difference between sets with crisp boundaries (A) and fuzzy sets having gradual ones (B).




(A) Crisp sets




(B) Fuzzy sets

FIGURE 2.5: Crisp sets and Fuzzy sets

There is, however, a difference between *fuzzy sets* and *fuzzy measures*: the latter describes how uncertain we are about the membership of one element to a crisp set (with known boundaries), while the former offers a formal description to the graduality of the edges of the set itself.

**Dempster-Shafer Theory of Evidence** The nature of this theory allows it to provide epistemic-based uncertainty specifications in defined intervals. This theory is built upon the works of Dempster and Shafer [51], offering a framework built upon "evidence" of membership in sets. The theory introduces two non-additive measurement called *belief* $Bel(x)$ and *plausibility* $Pl(x)$.

Considering a universe of discourse $U_\Omega$, the degree of belief that an element $x$ in $\Omega$ is backed up by a subset $A \subseteq \Omega$ or any of its subsets (members of the so-called *power set* of $A$), is the degree of belief in the subset $A$ or $Bel(A)$. Similarly, the degree of plausibility of $A$ noted $Pl(A)$ represents the evidence provided by the power set of $A$ and any set that overlaps with $A$. The *Fuzzified* Dempster-Shafer theory is A generalization of the previous theory by applying it on the fuzzy sets.

The previous list of uncertainty theories is not exhaustive. Nevertheless, it offers a glimpse of what tools are available to value the lack of data and handle the measures agents use in their applications. The projection of these theories on the universe of interest is also linked to the use case. For instance, one may relate the probability information about a set of financial data to investment risks they afford to handle if they choose to act based on one piece of data over the other. We show after in section 3.2.2 that custom uncertainty theories can be represented to use and allow annotating data with custom features.

## 2.3 The Link between Semantic Web and Uncertainty

The Semantic Web is perceived as a formal extension of the Web. In addition to the existing Web documents, it allows accessibility for information from data stores. Data on Semantic Web is either contributed independently (directly in a Semantic Web format, e.g., data dumps) or extracted via specific processes (e.g., mapping semantic

annotations from Web documents to Linked Data, or transforming spreadsheets into Linked Data, etc). Ceolin *et al.* [52] picture one process of extracting significant data from existing Web pages. Their use-case, assessing the quality of Web pages, allowed identifying three different possible sources of uncertainty: (i) automatic extraction and selection of features, (ii) the relevance of those features, and (iii) the sustainability of models. The three sources follow the idea we endorse about inputs and models. Another concrete example we use to illustrate uncertainty on the Semantic Web is this same manuscript, once published in a machine-readable format (i.e., uploaded on *HAL*,[22] or its code shared on *Github*, etc.). We can still be misspelling some words or misusing others, for what will lead the automatic extractors to generate triples with both *implicit* or *explicit* imperfections even after proofreading, examining, and testing.

One of the many concerns here is defining the extent to which uncertainty can be significant. We argue that uncertain data can be trusted or used to extract reliable parts of data. It leads to a problem of defining what threshold is necessary and link that to the use-cases, or determining what parts can be reliable, i.e., linking data with a partial definition of reliable fragments (For instance, day, month, and year information to be extracted from a date).

**Efforts for Uncertainty standardization** The first efforts to present an ontological representation for uncertainty on the Semantic Web were the ones of the Uncertainty Representation on The World Wide Web Working Group or *URW3-XG* [53]. Their proposed vocabulary [53] enables them to annotate data with the type, the model, and the derivation of uncertainty. The group offers a limited list of models (Fuzzy sets, rough sets) with which neither information regarding the quantification of uncertainty nor the specificities of each approach and theory are provided. Some works settled for the use of a more straightforward definition for uncertainty, considering it as a number between 0 and 1 and representing the *confidence* in the annotated statement [54]. Various works proposed their flavors of OWL to cope with

[22] https://hal.archives-ouvertes.fr

specific uncertainty types, such as Poss-OWL [55] for possibility theory handling, Fuzzy-OWL [47] for fuzzy set, and Bayes-OWL [56] for Bayesian probability support. These works do not provide a generic view for uncertainty nor a link between its different components.

**Uncertainty in the Semantic Web stack** The three layers "*crypto*, *proof*, and *trust*" in the Semantic Web Stack (see section 2.1.2) have the potential of backing-up initiatives for uncertainty representation. It will rely mainly on digitally signed and verified proofs describing the course of data to evaluate whether to trust them (or not, or to a certain extent). However, until now, there are no standards to cope with uncertainty for many reasons. One of the main ones was expressed by Tim Berners-lee himself, pointing that scalability is the main issue if one has to reason about data reliability [57]. Moreover, for those layers to be as intended, the lower ones have to integrate uncertainty.

In the current context of the Semantic Web, data can be incomplete, invalid, vague, outdated, ambiguous, imperfect, hence leading to an uncertain understanding or uncertain decision-making [40]. More is due to the distributed nature of the Web architecture, where some processes rely on data aggregation from different sources, opinion polling, crowdsourcing, etc. Following the architecture of the Semantic Web "layer-cake" presented in the section 2.1.2, we can link uncertainty to some if not all of the layers. At the data level, uncertain statements can be annotated with uncertainty metadata. At the schema level, schemas for uncertainty can be proposed to annotate uncertain data as well as uncertain schemas. In the rules layer, uncertain rules can be generated to identify multiple interpretations for inferred data. The query layer can be extended to take into consideration such changes. As for the proof and trust layers, they can be the foundation of standards for uncertainty evaluation. In this thesis, our focus is going to be on the former research questions regarding uncertainty. The study of the contextualization of statements is mentioned but not profoundly studied. As for provenance, it is perceived as another type of metadata that might help evaluate the uncertainty of statements.

One reflection behind elucidating uncertainty is to make an agent feel safer and more confident about getting out of their bubble. An agent may be open to consuming data from known and trusted sources, therefore enclosed in a bubble of *confirmation*. The bubble limits the perception of the agent towards the external world and acts like an *echo chamber* in worst cases. We imagine one agent consuming only information from one language chapter of *DBpedia*.[23] even though the source is being updated regularly, the agent limits itself to that. We believe that the agent here created for itself a "*Local Open World*". One solution to that is to list all of the trusted sources and their confidence level according to the agent and treat the other sources as "unknown". Elucidating uncertainty provides a universal path for the exploration of unknown sources. It allows understanding what to expect from one source or performing an evaluation and casting it in that format. It will not affect agents who still choose to trust their limited list and help those who want to open up for exploration.

**Uncertainty Extraction from Data Sources** Some sources do not provide explicit uncertainty information, whether perceived as errors,incompleteness, or reliability. Detecting uncertainty can be achieved by comparing the data source to itself, reflecting on data patterns or extrapolation to complete missing information and/or detect wrong ones. This also can be achieved by comparing and linking the data source with other external sources for more confirmation. According to Paulheim [58], external error-detection approaches in knowledge graphs are based on interconnections between data sources: they take advantage of the links (identity links or simply IRI reuse) to check for errors in the data source of interest. Paulheim [59] proposes in another work an external approach to detect outlier interlinks between datasets by creating a feature vector representation of each interlink based on types and incoming/outgoing links to all instances of a class. That work is meant to evaluate links, whilst here we check the reliability of data based on presumed correct interlinks. Other works are based on a statistical analysis of feature vectors associated with predicates that are linked to interlinked resources [60], [61]. Another

[23]https://www.dbpedia.org

interesting idea is identity quantification between two linked data sets. It explores the idea of isomorphism quantification between two sets presumably representative of the same real-world entity. Similar works inspiring data-driven ontology alignment were discussed by Shvaiko *et al.* [62].

Christodoulou *et al.* [63] discusses the use of similarity measurements and Bayesian updating to help to align ontologies from different data sources and using precomputed values provided by ontology matchers. The authors depend on the Linked Open Vocabularies[24] to calculate the likelihood of equivalence vs. non-equivalence of two distinct classes and use that measure to update the local probability of similarity between two classes using Bayesian update. Authors of [61] propose a statistical data-driven approach to detect incorrect property mappings among the different language chapters of DBpedia. The work focuses on detecting the wrong mappings and the analysis is run through the whole datasets.

## 2.4 Conclusion: a need for explicit Uncertainty Information

Understanding the nature of uncertainty became crucial to maintaining a view on the reliability of the different information presented on the Semantic Web. One way of doing this is by achieving a consensus between all parties in the information transaction: the provider is transparent about their data. The users are aware of the quality of data they are building their judgments and decisions upon. It fell into the vision of Tim Berners-Lee when he was explaining the relation between AI and the Semantic Web: "*The concept of machine-understandable documents does not imply some magical artificial intelligence that allows machines to comprehend human mumbling. It only indicates a machine's ability to solve a well-defined problem by performing well-defined operations on existing well-defined data. Instead of asking*

[24] https://lov.linkeddata.es/

*machines to understand people's language, it involves asking people to make the extra effort*" [11].

The idea of explicit uncertainty representation raises many challenges: what if the uncertainty information is uncertain on its own? Such an issue requires proposing a specific limitation for the semantics of uncertainty (Local Closures).

In the next chapter, we begin the journey towards uncertainty representation on Semantic Web and present the different factors that should be considered to perform such extensions.

# Chapter 3

# Representing and Manipulating Uncertain Data on the Semantic Web

The Semantic Web is populated from four primary sources: data extracted from the static Web, databases linked to the dynamic Web, data published directly in a Semantic Web format, and data generated from reasoning over all the previous ones. As discussed in the previous chapter, the different processes of extraction, population, publishing, and reasoning are sources of different imperfections. If we consider the Semantic Web as a digital discretization of our continuous reality, our view should be achievable through approximations. The uncertain nature of information on the Web pushes us to consider the possible boundaries we can draw to fit the abnormalities or know the extent of our errors. We dedicate this chapter to the proposition of a model for uncertain data representation on the Semantic Web. In section 3.2.1, we will discuss the physical and logical representation of uncertainty. Following that, section 3.2.2 offers an overview of the manipulation of uncertainty information with the integration of uncertainty calculi. We continue the discussion in section 3.2.2 with an overview of the different readings and mapping modes for uncertainty. We conclude this chapter by showing the means of negotiating suitable definitions and values for one use case. The results presented in this chapter were published in the

International Conference of Conceptual Structures ICCS2019 [64], and the workshop on Linked Data on the Web and its Relationship to Distributed Ledgers hosted in the Web Conference WWW2019 [65].

## 3.1 Understanding Uncertainty in the Web

The textual and structured documents on the Web can be easily parsed. Documents in a non-proprietary format, such as XHTML ones, can be analyzed to get interlinked pieces of data. Data harnessing does not stop at Web-destined documents only. The Web allowed linking other documents, such as audio files, videos, and code. Those types survived the fragmentation of documents into interlinked pieces of data. Nevertheless, they represent valuable resources that AI mines nowadays. Automatic video captioning, Automatic transcription, and code mining are some of the various applications. All of these works generate valuable data to be linked again on the Semantic Web.

For a starter, we chose two publications [66], [67] with the same title "The Uncertain Web". The existence of both publications on the Web makes a practical example of uncertainty in the following use-case: *Google Scholar* [1] offers an add-on for browsers, enabling them to search directly for references in selected texts or the metadata of one open tab in the browser. For instance, if we select the text "Information Management: a proposal" then click on the add-on, a pop-up shows the closest matches to the text and includes links to the papers' Web pages (and texts when available). On the page describing the book "The Uncertain Web" by Benslimane *et al.* [66], the add-on did not show the same book. Instead, it showed another book with the same title [67], but this time for a different author and perspective. We include both resources in the following discussion.

[1] scholar.google.com

**Is the Web meant to be Uncertain by Design?** The direct answer to this question would be a negative one. Like any other technology, Web architects do not mean for it to host errors. The Web, an incomplete but progressing project, is in a continuous movement and improvement. The designers of Web-based applications rely on standardized, or at least community-supported technologies to build up their functionalities. When Web technologies receive updates, improve efficiency, and offer new features, the Web-based applications are updated accordingly: from simple HTML pages to cross-platform applications that can be run over Web-views on different devices. The first book about the "Uncertain Web" [67] mainly discusses the uncertainty of Web engineering and how browsers and protocols have developed over the last two decades. The "uncertainty" here is about the technologies and mechanisms helping to create Web-based content and how to deliver it to the user. If we limit ourselves to the design of the Web or the user experience, the main uncertainty discussed here is the one about the behavior. The Web has to reduce and tighten the constraints leading to common errors.

**Is the Semantic Web Uncertain by Design?** The same answer applies here: the Semantic Web is not meant to be uncertain. The four processes we mention in the introduction of this chapter aim to create richer experiences and enable more applications to thrive on top of the existing piles of data offered by the generous amount of documents, databases, and Linked Data on the Web. The stack of technologies reigning over the Semantic Web inspires a Utopian vision of a universe where every datum can be verified, traced, and be accurate. However, the processes leading to the population of such data are uncertain on their own, and the standards might not be tight enough to prevent issues like inconsistencies from happening. Those problems can still be addressed with off-the-shelf solutions. The second book about the "Uncertain Web" [66] suggests the different factors that should be taken into consideration to cope with uncertainty. It identifies the critical points for uncertain data as the creation, representation, and consumption.

**Imperfections in Web Data** Although one piece of information passes all kinds of tests and verification, we may find that the mere existence of that piece is wrong. One real person may spell their name wrong on an online form, although their name is correct in their minds, and it is the case in another part of the Web (e.g., their *LinkedIn* profile). Sensors connected to the Web may produce data with a random uncertainty due to measurement errors. Some bugs in a code that extracts data from online spreadsheets can lead to the generation of wrong sums or averages shared on the Web. These cases are out of the scope of Web technologies and may not be covered entirely by the Semantic Web. There is no overseeing entity that supervises the whole Web and prevents that one wrong piece of information is entered, based on the Web itself (When we say the Web here, it means all the sources connected to it and not just the hosted content. If external sources are connected, they also make parts of the Web).

To sum up, the uncertainty we discuss here is a general concept that gathers all of the previous definitions. It funnels the uncertainties of designing the Web technologies, Web documents, data in online databases, Linked Data, the transformation generating data from documents, and data augmentation processes. It is an uncertainty that marks the origins and the paths of each statement in the knowledge base.

In the previous chapter, we pointed that uncertainty can be *epistemic* or *ontic* and that it follows different theories and may indicate several meanings according to the use-case. This chapter will dive into the technical details of uncertainty representation and manipulation on the Semantic Web. We present and discuss a model to represent and exchange information about uncertainty on the Semantic Web.

## 3.2 Representing Uncertainty on the Semantic Web

Uncertainty is presented as a piece of information given about another information to tell how *unsure* we are about the semantics of the latter. This doubt may refer to a lack of confidence or be a subjective representation of the backing evidence to this statement. Either one, uncertainty is a special kind of "data about data", i.e., metadata, defined on Oxford dictionary as "*information that describes other information in order to help you understand or use it*".

**Definition 3.1.** *Uncertainty metadata refers to the data describing the uncertainty of one or multiple statements. They include qualitative/quantitative data that evaluate the reliability of the concerning statements and descriptive ones indicating the approach and the meaning of the provided valuations.*

Uncertainty metadata is another type of data about data, similarly to temporal annotations [68] and provenance data [69]. It represents the qualification/quantification of the uncertainty of a statement and describes the theories and semantics of such uncertainty. For example, it measures that 70% of the available evidence points that the person `X` is the author of the paper `Y`. The source describes the uncertainty metadata as one of a statistical nature. They could also be Bayesian, evidential, or following a custom theory of uncertainty (see section 2.2.3).

To integrate uncertainty in the Semantic Web, it must comply with the standards and, like any other data on the Web, must be reusable and publishable. In the next sections, we propose to discuss three aspects related to the representation of uncertain linked data:

- **Syntax**: The serialization of uncertain data with respect to the existing standards requires uncertainty to follow the rules of representation, linkage, and storage of linked data. The representation should allow accessing and querying

uncertainty besides of data and independently from it. Results should be in a machine-readable and interchangeable format.

- **Semantics**: The readability of uncertainty metadata is the first step. The next step is to make it understandable. We need to define proper semantics for uncertainty metadata and clarify the meaning of the link between the annotation and the statement. A valid schema for uncertainty is required for reasoners to infer new triples or validate the knowledge base with the schema.

- **Reading**: We need to address the coverage of uncertainty metadata and specify the borders of its semantics. The reading refers to the different stages involved in providing a piece of single uncertainty information. These stages include the sets of statements affected by or affecting uncertainty, either directly (i.e., a direct annotation and affiliation) or indirectly (i.e., through a link or an inclusion). They also include the methods used to combine the different values and use the indications to understand current uncertainty information or provide a new one.

### 3.2.1 Serializing and Storing Uncertain Data

Statements (triples) in the form of `<subject, predicate, object>` are used to describe relationships between resources on the Semantic Web. The metadata linked to these statements and representing uncertainty can also be considered as a resource. They can be identified using *IRI*s and linked to other resources as an annotation or be annotated by them. For example, an uncertainty *IRI* can be annotated by other metadata such as its creation date or its provenance. If we consider the *IRI* `ex:Uncertainty_1`, we may assert that the statement `ex:Statement_1` is annotated by `ex:Uncertainty_1`, and that `ex:Uncertainty_1` was generated on the first of September 2021 by user `ex:User_1`.

The annotation of a statement with uncertainty metadata requires a link to the

statement as an entity. We discussed in section 2.1.5 the different serialization syntaxes for RDF and the several methods to annotate statements with metadata using such syntaxes. Some of the proposed methods affect the semantics of triples themselves. We argue that annotating a statement requires elaborating a link with the statement as a whole and preferably without losing its semantics. Both the syntax and annotation methods are crucial elements to discuss for scalability.

We believe that the use of RDF-star to represent uncertainty and annotate triples fulfills the needs for assertion, readability, and ease of access. RDF-star is yet to be a standard. However, it is currently adopted by different data stores (like Apache Jena, TopQuadrant TopBraid EDG, and Ontotext GraphDB) and offers multiple implementations (RDF-star, SPARQL-star, turtle-star, ...). Upon the approval of its authors, we registered the representation of uncertainty as one of the use cases in the working documents of RDF-star.[2]

In the next part, we walk through an ontological representation of uncertainty and link that with our current choice of syntax.

### 3.2.2 *mUnc*: an Ontology for Uncertainty Metadata

Uncertainty might differ from one use case to another in terms of nature and requirements. It can take the form of inconsistencies, incompleteness, ambiguity, vagueness (See section 2.2.2). The use of such uncertain data requires a specific representation that the current standards of the Semantic Web cannot fulfill without extensions.

Back in the previous chapter, we provided various elements that can be related to an uncertainty annotation. For instance, the value reading requires two elements: the theory that governs the value/qualification and the meaning of the annotation. The theory can be linked with calculus. We reuse some of the definitions offered by *URW3* with slight changes to the content. We recall that we rely on the *SAMOD* methodology of ontology development to recite different use cases for uncertainty use

[2] https://w3c.github.io/rdf-star/UCR/rdf-star-ucr.html#uncertainty-representation

and then validate our ontology by giving examples of its usage. Besides the benefits of agility, having it in ontology design methodology is a choice for several reasons. One of the reasons is upgrading the model iteratively as long as new data and requirements are available. In addition, the model can be published in several versions containing separated focuses. The example would be focusing now on the main questions about uncertainty representation and then figuring out a way to represent the translatability of uncertainty. Not to mention that this allows having milestones and hence facilitate collaborative working on the design, especially since discussions about uncertainty should be of consensus about concepts before application.

#### 3.2.2.1 *mUnc* Motivating Scenarios

We provide examples of the motivating scenarios *MS* describing some use cases where uncertainty representation is of need. This list cites existing/possible situations with unreliable data in Table 3.1. The list is extended with more examples on the website of the uncertainty ontology.[3]

Other motivating examples are mentioned in the report of URW-XG.[5] Cases of information fusion and Ontology-based reasoning are upon the first motivations we mentioned previously.

#### 3.2.2.2 *mUnc* Competency Questions

The identified questions, issued from the reading of the motivating scenario, would make the basis and the limitation for the proposition of the Glossary of terms. We provide here some of the requirements that the ontology is supposed to handle.

1. What statements are annotated using a particular uncertainty theory?
2. What are the components of an uncertainty approach?
3. How to manipulate an uncertainty value?

[3] http://ns.inria.fr/munc

[5] The use-cases can be found here: https://www.w3.org/2005/Incubator/urw3/XGR-urw3-20080331/#usecases

| Name | Uncertainty of data on the Semantic Web |
|---|---|
| Example 1 | The height of the football player is stated in the context of the French language chapter of DBpedia <http://fr.dbpedia.org/page/Stefano_Tacconi> is 193cm, while in the English chapter <http://dbpedia.org/page/Stefano_Tacconi> it is of 188cm. We counted 27516 football players in the English chapter have a valid height and an owl:sameAs link with the French chapter. After querying and linking both sources, we found 695 instances (i.e., 2.5%) having different height values (We tested for more than 2cm, refer to appendix A). |
| Example 2 | To calculate the age of *Plato* [4] from incomplete birth and death dates (missing one information such as the day, or the month), we must have an approximate representation or an interval. A birth date missing an information about the day of birth can be represented using a fuzzy number, with a membership function based on a statistical analysis of the different birth dates in a specific period. There is no datatype to represent fuzzy intervals and fuzzy numbers. According to Straccia [70], a fuzzy interval may have up to four parameters to describe the shoulders. A fuzzy number can be part of multiple fuzzy intervals at once, each with a degree of membership. |
| Example 3 | The choice to use unreliable information (from a website that we did not verify or without references) is linked to a risk the user must comprehend. Such risk may be linked to the whole website, a portion from it, or specific authors. For example, the reference to *Plato* in *Wikidata* has two different birth dates on the Web page. Moreover, the query service returns a third and different birth date (Figure 3.1). |
| Example 4 | The list of music albums of one artist can be missing some titles. The fact that we judge based on incomplete information should be mentioned and quantified. |
| Example 5 | A human agent provided a list of the employees in their company. The same task was given to a robot to crawl the different phone books and websites, find listings of people working in the same company, and enrich the previous list. The human agent made some typos in the list, and the robot found multiple instances for people with the same name but different phone numbers and addresses. |
| Example 6 | When Named Entity Recognition is run automatically, the model may not be fully accurate and precise with the outputs. The missing labels inform of incompleteness in the generated list, and the falsely (or wrongly) annotated refer to invalidity (or ambiguity). |

TABLE 3.1: The motivating scenario for uncertainty representation on the Semantic Web

4. How to answer a query about two statements, knowing that one of them at least is uncertain?

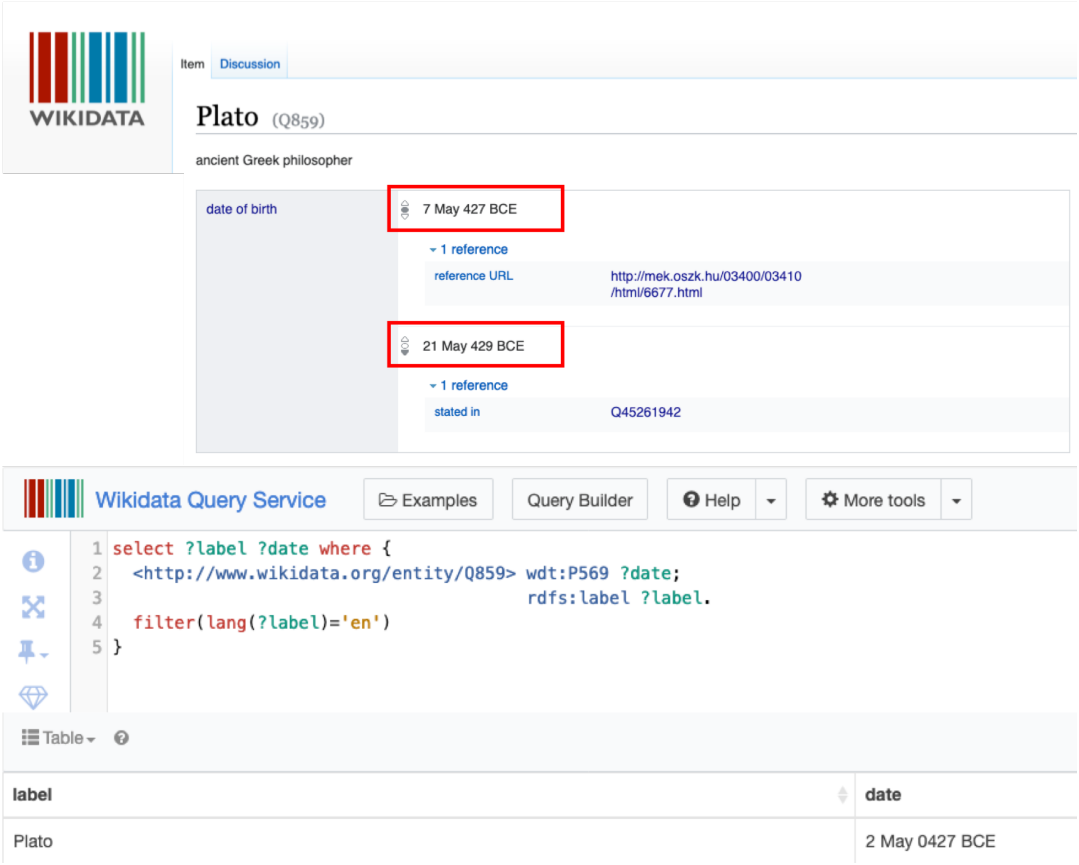



FIGURE 3.1: An example of inconsistencies in Wikidata. The page shows two different birth dates while the query service shows a distinct third one.

5. Given two uncertain statements, how to determine which one is the most accurate one?
6. What is the uncertainty of a statement inside an uncertain context?

The list above is non-exhaustive. Some of these questions are related to the motivating scenario. Others link with some of the research questions we provided in the introduction of this manuscript. These questions can be preliminarily answered as follows.

1. A possible outcome is the list of all the statements with an uncertainty annotation joint with the list of their theories. For instance:
   - statement 1, uncertainty object: (uncertainty value, uncertainty theory)
   - statement 2, uncertainty object: (uncertainty value, uncertainty theory)

2. An uncertainty approach announces a reading and follows a theory that has defined features.
3. An uncertainty value can be manipulated using the calculi linked to the features of uncertainty theories.
4. We need to check first if the two statements are mentioned in the same world or non-contradictory worlds. If the statements are linked to each other to provide an answer, then the answer –if any– must be uncertain, and we need to evaluate its uncertainty based on the two statements.
5. A formal definition of a theory also provides comparability between the uncertainty values. We may choose the best one in terms of the uncertainty reading provided with the two values.
6. One possible outcome is to consider the assertion of the inclusion in the context as an uncertain statement itself and answer according to (4).

#### 3.2.2.3 *mUnc* Glossary

From the previous test cases and the definition of uncertainty we provided in the previous chapter (see section 2.2.2), we introduce some terms that we see as a raw format for what can be formalized later in ontological concepts. Table 3.2 presents the main terms that are related to uncertainty, such as theories and values.

#### 3.2.2.4 *mUnc* Concepts and Properties

Uncertainty information is considered a specialization of the general concept of metadata. That simplifies the future extensions for other types of metadata. For the same reason, we do not include the concept of *Agent*, as it can be included using other vocabularies like W3C *PROV* Ontology.[6] We see the provenance-related metadata as an external component that can annotate uncertainty data to express their origin for more transparency. Figure 3.2 offers an overview of the core concepts and properties of *mUnc* including sentences, contexts (worlds), and uncertainty metadata

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

| Term | Definition |
|---|---|
| Meta/Uncertainty | Data associated with and annotating a statement. This metadata does not have a generic type, for as many types of customized metadata can inherit from it. The Meta relates to a specific world in which the statement is asserted. Uncertainty is one kind of metadata that can be used to annotated a statement. |
| Sentence | The annotated statement with uncertainty metadata. |
| Uncertainty Theory | A theory that allows reading and manipulating uncertainty values. |
| Uncertainty Features | A set of parameters linked to one uncertainty theory and allowing to quantify/qualify the statements with uncertainty values. |
| Uncertainty Indication | An annotation that explains the reading of the uncertainty. That may refer to the risk of having uncertain sentences in the results, the reliability of the information, the ambiguity, or a mistake detected by the agents subjecting the data quality. It may as well refer to the epistemic nature of the uncertainty. |
| Uncertainty Value | The value (or values) qualifying/quantifying the uncertainty of the statement. |
| Uncertainty Calculi | The set of operators and functions that can be applied to uncertainty values to combine them and manipulate them. |
| World | The context in which the statement is asserted and this assertion is evaluated. |
| Select uncertain statements | When two uncertain statements in the same world are selected to output a result, a selection can be performed based on their uncertainty values to decide whether there is a world in which the two statements hold together with respect to their uncertainty theories. |
| Compose uncertain statements | When two uncertain statements in the same world are joined to output a result, a composition can be performed on their uncertainty values to export a new uncertainty value linked to the result. |

TABLE 3.2: A glossary of terms of the uncertainty ontology

(theories, features, calculi). Table 3.3 illustrates the former concepts and their definitions. The extended definitions of the ontology and its documentation are available in its reserved namespace on INRIA servers.[7]

| Class | Definition |
|---|---|
| `munc:Meta` | Data associated with and annotating a statement. This metadata has a generic type, for as many types of customized metadata can inherit from it. The Meta relates to a specific world in which the statement is asserted. Uncertainty is one kind of metadata that can be used to annotated a statement. |
| `munc:Uncertainty` | A specific type of `munc:Meta`. The concept refers to the metadata set that holds information about the uncertainty of the annotated resources. |
| `munc:Sentence` | The annotated statement with uncertainty metadata. |
| `munc:UncertaintyApproach` | An approach for uncertainty evaluation, linking a theory, a set of features, and operators. Understanding the approach allows reading and manipulating uncertainty values. |
| `munc:UncertaintyIndication` | An annotation that explains the reading of the uncertainty. That may refer to the risk of having uncertain sentences in the results, the reliability of the information, the ambiguity, or a mistake detected by the agents subjecting the data quality. It may as well refer to the epistemic nature of the uncertainty. |
| `munc:UncertaintyValue` | The value (or values) qualifying/quantifying the uncertainty of the statement. It can be a literal, a URI, or of a specific datatype. |
| `munc:UncertaintyCalculus` | The function linked to an uncertainty feature to manipulate it under an uncertainty operator. |

TABLE 3.3: List of Classes in the Uncertainty Ontology

A *sentence* is an expression evaluating a truth value, while the *world* represents the context in which a *sentence* is stated. Both sentences and worlds can be annotated with *uncertainty* information. For instance, the sentence `ex:S1` representing the triple ⟨`ex:StefanoTacconi`, `dbo:height`, 188⟩ in the world `ex:DBpedia_FR` referring to the height of the football player is stated in the context of the French language chapter of DBpedia [71], assuming that the latter is consistent [72].

[7] `http://ns.inria.fr/munc`

| Property | Definition |
|---|---|
| `munc:uncertaintyFeature` | A parameter linked to one uncertainty theory and allowing to quantify/qualify the sentences with uncertainty values. |
| `munc:uncertaintyOperator` | The set of operators and functions can be applied to uncertainty values to combine them and manipulate them. |
| `munc:statedIn` | Links the sentences and worlds to their metadata, including uncertainty. |
| `munc:hasMeta` | Links the sentences and worlds to their metadata, including uncertainty. |
| `munc:hasUncertaintyFeature` | Links the uncertainty approach to its features. |
| `munc:hasUncertaintyOperator` | Links the uncertainty approach to its operators. |

TABLE 3.4: List of Properties in the Uncertainty Ontology

```
1    ex:S1 munc:statedIn ex:DBpedia_FR.
```

An *Uncertainty Approach* (i.e., *Uncertainty theory*) links a set of *features*. These are the metrics on which the uncertainty theory is based to indicate the degree of truth, credibility, trust, or likelihood of a sentence (see section 2.2.2). Each *feature* links a *value* to the *uncertainty* entity annotating the sentence. The interpretation of each uncertainty entity is linked to the entity itself, resuming the reading of the different uncertainty features in this case. Indications can be given as plain text, references to documentation, or other datatypes that can be used later as parameters in the uncertainty calculus. To continue with the previous example, the sentence `ex:S1` can be annotated with an uncertainty entity `ex:Uncertainty_1` that follows a probabilistic approach `ex:SubjectiveProbability` having one feature `ex:probability` with the value 0.9. The indication states that the probability here indicates the trust in the author of the sentence.

```
ex:S1 munc:hasMeta [
  a munc:Uncertainty;
  munc:hasUncertaintyApproach ex:SubjectiveProbability;
  ex:probability "0.9"^^xsd:integer;
  munc:hasUncertaintyIndication "ex:probability represents the trust in the
      original author of the sentence. If the provenance information is not given,
      the author is example.com by default."^^xsd:string
].
```

We note that the attribution of a specific uncertainty approach depends on the use-case. For instance, a missing triple cannot be annotated with an approach that represents *incompleteness* but we can manage to annotate its graph with such an approach to describe that there is missing information that was supposed to exist. Such a detail is left for the users under the condition of expressing the indication of each feature.

#### 3.2.2.5 The Logic behind Uncertainty Operators

To understand and manipulate the previous *values*, we divided the force between (*i*) *operators* that can be applied to an uncertainty value, and the (*ii*) functions that are used to apply the effect of operators, i.e., *Uncertainty Calculus*. With respect to the definitions provided in the previous chapter (see section 2.2.3), we offer a generic choice of operators for the different uncertainty approaches. We stress that we do not extend the RDF semantics with uncertainty operators and calculi.

*Uncertainty operators* can be applied to uncertainty values to cover the logical part of uncertainty theories in case of querying uncertain data. To treat the uncertainty metadata linked to sentences beside them in queries, we require operators linked to uncertainty approaches to offer mainly two functionalities:

- establish an order between uncertainty values of the same feature, under the assumption of monotonicity.
- link an uncertainty approach to the calculi necessary for manipulating and binding uncertainty values during query processing.

The use of uncertainty operators in our work is similar in a way to the use of $\otimes$ or so-called *meet* operator in [73]. The authors interpreted the *t-norm* as the existence of one interpretation that satisfies a set of conditions set either by the agent or by the semantics of the uncertainty theory itself. Unlike their work, we do not specify an operator to combine information about the "same" statement, but to check if different statements that can be bound without considering uncertainty will fit in terms of it. For instance, two statements with different uncertainty theories would not fit in some instances because their uncertainty metadata cannot be combined (e.g., where uncertainty features cannot be homogenized). Naturally, two statements following the same theory and respecting the condition of selection issued by the requesting agent (such as a threshold for truth values or intervals for fuzzy values) are selected for the next steps. We mentioned that we do not offer an operator for one statement, but we can consider two instances annotated differently of the same statement as different sentences and work the combination from there.

The example here is about the existence of two sentences in the graph `:G` considered here as their world:

$T_1$ :<`ex:StefanoTacconi, dbo:height, 188`> and

$T_2$ :<`ex:StefanoTacconi, rdfs:label, "Stefano Tacconi"`>.

We define a probabilistic uncertainty approach that associates a subjective probability value $\mu_i$ to each sentence $T_i$ to reflect its reliability. We associate $\mu_1 = 0.4$ to $T_1$ and $\mu_2 = 0.95$ to $T_2$. If the user looks for the different football players' names and heights (using the query in Listing 3.1), the expected result should be: (`ex:StefanoTacconi`, 188, `"Stefano Tacconi"`). This result presents valid bindings satisfying the pattern in the query.

```
1  PREFIX dbo : <http://dbpedia.org/ontology/>
2  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3
4  SELECT ?player ?height ?label FROM :G WHERE {
5     ?player dbo:height ?height.
6     ?player rdfs:label ?label.
7  }
```

LISTING 3.1: Querying for the names and heights of football players

The bindings, in this case, should satisfy the conjunction of the two triple patterns to satisfy the query. When uncertainty is considered, both sentences should satisfy separately (and together) a set of conditions given by the user before executing the query. For instance, the user suggests that they do not accept answers from sentences with a probability value of less than 0.5. In this case, the user has two options:

- to ignore the dependency of probability values, and perform a preliminary selection of sentences with a probability surpassing the threshold;
- to check the dependency between the probability values of the two sentences and adjust each accordingly.

If at least one of the sentences does not satisfy the conditions, both are disregarded. For the sake of example, we consider that no dependency information links the two probability values. For the previous threshold of 0.5, $T_1$ won't be selected and that results of disregarding `ex:StephanoTacconi` from being a binding for `?player`. We expect no multiple occurrences of the same statement with different uncertainty metadata. As we mentioned before, we consider each context a consistent one to work with. If we consider a new sentence:
$T_3$ :<`ex:StefanoTacconi, dbo:height, 192`>
with $\mu_3 = 0.9$, then the expected result should be: (`ex:StefanoTacconi`, 192, `"Stefano Tacconi"`).

For uncertainty operators, we distinguish two important ones for each theory. The first is the *selection* operator, which acts whenever a pair of sentences is selected for a query evaluation. The operator checks if the link between these sentences holds under the conditioning of their uncertainty values. Unlike the regular *meet* definition from the lattice theory, the *selection* operator indicates the existence of a possible world where both sentences can be asserted with indicated conditions on their uncertainty values.

The second is the *composition* operator. It is used to evaluate the new uncertainty value for the selected result components from query answers. For instance, the two

selected sentences in the previous example $T_2, T_3$ are the basis for calculating the resulting information's probability. If we consider each sentence as an independent fact, the resulting probability value would be the product of $\mu_2$ and $\mu_3$. The operations linked to these operators –and any others– are implemented as an *uncertainty calculus*.

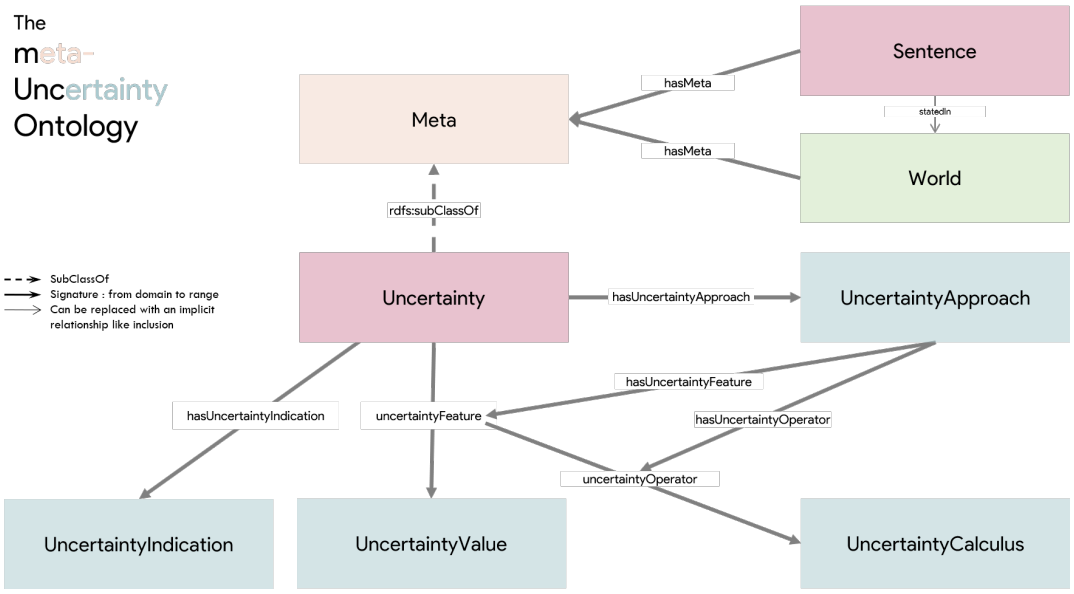



FIGURE 3.2: Overview of the *mUnc* ontology and its core concepts

#### 3.2.2.6 Uncertainty Calculi

Semantic Web ontology languages do not support procedural attachments or functions inside ontologies. We consider linking the features of uncertainty theories to their proper calculi (arithmetic, logical, comparison, selection, composition). Nevertheless, to satisfy the four levels of formalization for uncertainty theories, we rely here on the *LDScript* function definition language [74], a programming language whose objects are RDF entities. It is built on top of SPARQL as an extension of the SPARQL filter expression language. *LDScript* as a language permits variable declaration, assignment, function call, return, etc. Using *LDScript*, we can define functions named with an *IRI* and one or several arguments that are variables in the SPARQL syntax. That enables defining uncertainty operations and linking them to uncertainty features.

One of the previous informal competency questions mentioned the manipulation of uncertainty values. We may provide a function that calculates the probability of a conjunction of two supposed independent events $A$ and $B$. The formula suggests that for the previous events:

$$P(A \wedge B) = P(A) \times P(B). \tag{3.1}$$

Such value can be calculated for the user using the function referenced by `ex:composeProbabilityCalculus` and defined in *LDScript* as shown in the following example:

```
function ex:composeProbabilityCalculus(?pA, ?pB){
  ?pA * ?pB
}
```

Therefore, binding the function `ex:composeProbabilityCalculus(0.7, 0.9)` during a SPARQL query execution will return 0.63. The former definition of the probabilistic approach using *mUnc* can be enriched by linking the *IRI* of the function to the declared feature, simply by adding the triple:

```
ex:probabilityValue ex:composeProbability ex:composeProbabilityCalculus.
```

As stated before, each function is considered a resource due to the *IRI* defining its name. We can store such functions in SPARQL files all over the Web and access their code using their reference. In Appendix B, we provide an overview for a language-independent vision for the use of linked functions. Listing 3.2 illustrates a complete example of an uncertainty approach alongside its calculus, with a pseudo-coded functions to check for the dependency of probability values. The approach is linked first to its features and operators (Lines 1-3). The features are then linked using the operators to the proper calculus, and the type of uncertainty values is attributed (Lines 5-9). The calculus is defined in the last part of the code using *LDScript*, where the selection calculus checks for the conformity of the probability value with the given conditions, the composition calculates the product of the two probabilities assuming they are independent ones.

```
1  # Definition of the approach, features and opeators
```

```
ex:Probability a munc:UncertaintyApproach;
   munc:hasUncertaintyFeature ex:probabilityValue;
   munc:hasUncertaintyOperator ex:selectProbability, ex:composeProbability,
       ex:compareProbability.

ex:probabilityValue a munc:uncertaintyFeature;
   rdfs:range xsd:decimal;
   ex:selectProbability ex:selectProbabilityCalculus;
   ex:composeProbability ex:composeProbabilityCalculus;
   ex:compareProbabilty ex:compareProbabilityCalculus.

# Returns True if both probabilities satisfy the threshold of selection provided by the user.
function ex:selectProbabilityCalculus(?proba1, ?proba2, ?threshold){
   if(min(?proba1, ?proba2) > ?threshold) return True;
   return False;
}

# Returns the product of two probabilities, under the assumption they are independent
function ex:composeProbabilityCalculus(?pA, ?pB){
   return ?pA * ?pB
}

# Returns True if the first parameter is strictly bigger than the second parameter
function ex:compareProbabilityCalculus(?pA, ?pB){
   return ?pA > ?pB
}
```

LISTING 3.2: Representing the Probabilistic approach using *mUnc*

## 3.3 Annotating Uncertain Data with *mUnc*

To illustrate the previous definitions, we offer an annotation for the previous examples in the motivating scenario.

Example 1 uses subjective probability as the feature representing the uncertainty values. We may indicate in the definition of the probabilistic approach that the feature *ex:probabilityValue* reflects the likelihood that one value appears in the different language chapters of DBpedia. The annotations are written in *turtle-star* syntax[34].[8]

[8]The syntax is similar to Turtle, with the addition of two notations for embedded and asserted-annotated triples.

```
ex:StefanoTacconi dbo:height "188"^^xsd:decimal {|
   munc:hasMeta [a munc:Uncertainty;
      munc:hasUncertaintyApproach ex:Probability;
      ex:probabilityValue 0.8
   ]
|}

ex:StefanoTacconi dbo:height "193"^^xsd:decimal {|
   munc:hasMeta [ a munc:Uncertainty;
      munc:hasUncertaintyApproach ex:Probability;
      ex:probabilityValue 0.1
   ]
|}
```

Listing 3.3: Example 1- Two triples from different sources

In the second example, we define an uncertainty approach to deal with fuzzy date values. We define a "fuzzy interval" with two shoulders limited by four uncertainty features with a decimal range. Later on, we can define the selection of two fuzzy intervals as the different sentences with valid intersections between their intervals and the composition as the intersection linked to the query results.

```
PREFIX fuzzy: <http://example.com/fuzzy>

fuzzy:FuzzyDate a munc:UncertaintyApproach;
   munc:hasUncertaintyFeature fuzzy:fuzzyInterval;
   munc:hasUncertaintyOperator ex:selectFuzzy, ex:composeFuzzy.

fuzzy:fuzzyInterval rdfs:subPropertyOf munc:uncertaintyFeature;
         rdf:range fuzzy:Shoulderinterval.

fuzzy:ShoulderInterval a rdf:Seq;
   rdf:1 fuzzy:valA; #left value in left shoulders or crisp, only value in linear modifiers
   rdf:2 fuzzy:valB; #right value in left shoulders or crisp
   rdf:3 fuzzy:valC; #left value in right shoulders
   rdf:4 fuzzy:valD. #right value in right shoulders

fuzzy:valA, fuzzy:valB, fuzzy:valC, fuzzy:valD rdfs:subPropertyOf
    munc:uncertaintyFeature;
         rdf:range xsd:date.
```

Listing 3.4: Example 2- Defining a fuzzy date interval using *mUnc*

In the third example, we may define the combination of the *author* and the *website* as a *world*. These worlds can have a general uncertainty annotation with a risk annotation.

```
ex:World_1 munc:hasMeta [a munc:Uncertainty;
                     munc:hasUncertaintyApproach ex:Probability;
                     munc:hasUncertaintyIndication "This value represents the risk
                         taken for trusting an author on this website; The less the
                         better";
                     ex:probabilityValue 0.1].
```

LISTING 3.5: Example 3- The risk of using information from an author in a website.

The fourth example indicates using a possibilistic feature that the list of songs of the album represented by the entity `ex:Album_1_Songs` is complete by 80%.

```
ex:Album_1_Songs munc:hasMeta [a munc:Uncertainty;
                     munc:hasUncertaintyApproach ex:Possibility;
                     munc:hasUncertaintyIndication "This value represents the
                         completeness of the resource it is attached to";
                     ex:completeness 0.8].
```

LISTING 3.6: Example 4- Incomplete list of tracks

The fifth example represents the fact that one feature can have many indications. The default indication can be attributed by definition, and then the feature can be understood according to each instance. The probability here served as a validity measure in the first and second blocks and in the third block as a similarity.

```
ex:Employee_1 rdfs:label "Fabienn"^^xsd:string {|
   munc:hasMeta [
      a munc:Uncertainty;
      munc:hasUncertaintyApproach ex:Probability;
      munc:hasUncertaintyIndication "This value represents the validity of the
          attached label";
      ex:probabilityValue 0.8;
      prov:wasAttributedTo :HumanAgent.
   ]
|}

ex:Employee_2 rdfs:label "Fabien"^^xsd:string {|
   munc:hasMeta [
      a munc:Uncertainty;
      munc:hasUncertaintyApproach ex:Probability;
```

```
        munc:hasUncertaintyIndication "This value represents the validitiy of the
            attached label";
        ex:probabilityValue 1;
        prov:wasAttributedTo :RobotAgent
    ]
|}

ex:Employee_1 owl:sameAs ex:Employee_2 {|
    munc:hasMeta [
        a munc:Uncertainty;
        munc:hasUncertaintyApproach ex:Probability;
        munc:hasUncertaintyIndication "This value represents the similarity between
            two entities";
        ex:probabilityValue 0.8
    ]
|}
```

Listing 3.7: Example 5- Man vs Machine

In the sixth example, we attributed uncertainty to a graph entity. The features here (accuracy, precision) represent the evaluation of the dataset of recognized entities. This shows that uncertainty can be used as a marker to evaluate both data and processes.

```
ex:Graph_Of_Entities munc:hasMeta [
    a munc:Uncertainty;
    munc:hasUncertaintyApproach ex:Evaluation;
    ex:precision 0.8;
    ex:accuracy 0.9
].
```

Listing 3.8: Example 6- An uncertain graph from an uncertain model

As seen in the previous examples, the definition of *mUnc* allows the introduction of the existing uncertainty theories to the Semantic Web and the proposition of new and custom ones. The Semantic Web stack holds for the previous definitions of uncertainty. We find different syntaxes and methods to annotate data with uncertainty values. Annotating uncertain data with *mUnc* allows to harness uncertainty information from the Semantic Web and communicate with other sources in a unified and formalized language. Uncertainty calculi can also be stored in a referenceable format.

In practice, the Semantic Web engine *Corese* [9] implements *LDScript* [74], allowing functions to be stored in an external SPARQL query files on the Web to be called at the moment of query execution.

## 3.4 Design Choices for Uncertainty Representation

Opting for RDF-star requires providing mappings from the different alternatives to RDF-star, ensuring backward compatibility and easing the transition. For instance, annotations can be reconstructed from reified annotated statements with the query in listing 3.9.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns\#>
PREFIX munc: <http://ns.inria.fr/munc/>

DELETE {
   ?statement a rdf:Statement .
   ?statement rdf:subject ?subject .
   ?statement rdf:predicate ?predicate .
   ?statement rdf:object ?object .
   ?statement munc:hasMeta ?uncertainty .
   ?statement ?p ?o .
} INSERT {
   ?subject ?predicate ?object {| munc:hasMeta ?uncertainty; ?p ?o |} .
} WHERE {
   ?statement a rdf:Statement .
   ?statement rdf:subject ?subject .
   ?statement rdf:predicate ?predicate .
   ?statement rdf:object ?object .
   ?statement munc:hasMeta ?uncertainty .
   ?statement ?p ?o .
   FILTER (?p NOT IN (rdf:subject, rdf:predicate, rdf:object) &&
         (?p != rdf:type && ?object != rdf:Statement))
}
```

LISTING 3.9: Mapping between reified annotated statements and RDF-star

[9] https://github.com/Wimmics/corese

RDF-star appears to outperform other approaches in storage and loading time. To illustrate, we account for four additional statements linked to *mUnc* to clarify: the nature of the metadata as uncertainty, the theory it follows, the reading it offers, and the value annotating it. We consider here an examplar dataset with $n = 50$ million triples and $p = 1500$ properties. The table 3.5 represents the effect of annotating all triples in a dataset with uncertainty values. The different annotation methods consider/add many statements to ensure the encoding of the additional information about uncertainty. For example, standard reification encodes the sentences in four triples, and the four additional triples for uncertainty make the total at $(4+4)n$ triples for each sentence, hence 400 million triples for a dataset with 50 million sentences. We gave examples of the encoding in table 2.1.

For storage purposes, we can rely on customized representation and indexing methods (like S-RDF [75], or HDT [76]) to store uncertainty. This aspect is crucial as the introduction of uncertainty increases the size of data significantly. Later in this chapter, we discuss the reading of uncertainty and the fact that contextualization can be a solution to minimize redundancy on statements with different types of reading that should be implemented.

For the ontological part, we discussed that we could not limit the list of possible indications. Instead, we offer a natural language annotation to let users explain the purpose of this uncertainty annotation. That might seem counter-intuitive, but if the users do not care about annotating their data, they will not reach this step. If one cares about delivering proper definitions about their data, they will take their time to explain what might be problematic with it as well. Moreover, the annotation property we offered can handle both an object property and a datatype property. The indications can be formalized later to give further meaning to their uncertainty values. However, calculus and formalization must be provided to ship such theories.

The representation we offer here is based on data annotation. We do not deal with uncertain ontologies, and their definition is out of the scope of this work. We perceive ontologies as stable entities at a specific moment, but we are aware of works

| **Approach** | **Reification** | **Singleton Property** | **Single Named Graph** | **N-ary relations** | **RDF-star** |
|---|---|---|---|---|---|
| | (4+4)n | (2+4)n | (1+4)n | 2(n+p)+4n | (1+3)n |
| Triples | 400M | 300M | 250M | 300M | 200M |

TABLE 3.5: Number of triples needed to annotate all statements in a dataset with uncertainty values

that dynamize the concepts and roles of the ontology (e.g., concepts that change over time [77]). We provide some alternative to that in the next chapter, where contextualization may replace the uncertainty of an ontology. After all, we stress that designing accurate ontologies may be the only requirement to avoid that situation. The two aspects may be linked together in future works.

One of the goals of *mUnc* is to promote the portability of the uncertainty calculi. Reasoners do not use external calculi to perform entailment. The logic of reasoners is extended in many works [70], [78] to be theory-specific but still dealing with one type of uncertainty. We believe that the best way to deal with the uncertainty from an open world is by publishing the calculi alongside the theories and have small extensions to perform any uncertainty-related operations on the go. That opens the doors for generic reasoners that update their core and logic from the Linked Open Data graph.

## 3.5 Conclusion: Representing Uncertainty Metadata

In this chapter, we provided and discussed a listing of the annotation methods for linked data. We proposed the *mUnc* ontology for uncertainty representation. The ontology allows custom uncertainty approaches and calculi to be linked with the uncertainty values annotating uncertain statements.

Representing uncertainty with respect to the existing Semantic Web stack is a challenging process. For it is a particular type of metadata, different aspects should be considered to publish, access, read, manipulate, and produce uncertain values. In this chapter, we discussed the representation and publication of uncertainty on the Semantic Web. We presented a vocabulary allowing the representation of uncertainty theories and annotating sentences using the Semantic Web standards. We explained the publishing of reusable uncertainty calculus using *LDScript*.

Uncertainty representation is the first step of a long process, including the preliminary calculus of uncertainty values and uncertainty propagation among interconnected Linked Data sources. The next chapter follows with our view to access and read uncertain data. We discuss the notion of contextualization, the mapping of uncertainty values, uncertainty translation and through sources.

# Chapter 4

# Accessing Uncertain Data on the Semantic Web

Understanding uncertainty requires understanding it on different levels. Despite the linking of crucial information to uncertainty entities such as the theories to use and the indication of the value, uncertainty still requires extra attention. Statements are encapsulated in graphs, issued from datasets, and grasped from deeper levels that may affect the reading of its uncertainty, by adding some features or altering others. Moreover, uncertainty requires transformation when meeting different formalism, and *mUnc* as presented in the previous chapter does not account for that.

This chapter discusses the contextualization of uncertainty values, the different scopes for uncertainty reading, the translatability of uncertainty theories and the negotiation of such data. Contextualizing leads to involving extra information about the assertion of sentences to reflect the effect of the encapsulation. The readings discuss ways to reduce the uncertainty on several levels to present a set of uncertainty information to users. Translatability and negotiation of uncertainty are about offering more flexibility and transparency and access to agents selecting to be fed of uncertainty.

## 4.1 Contextualizing Uncertain Linked Data

In our regular communications, we often refer to the context in which our statements are made to clarify their semantics and offer extra elements on which one can rely to understand our message. This contextualization can be both implicit and explicit. The first type makes part of most of our speech and is perceived as the default set of ideas about the speaker. The second type is direct and does not necessarily require prior knowledge of the speaker. For instance, this manuscript commenced with an opening to clarify the context of the research we discussed. However, someone who knows the author of the chapter and their research direction would infer that information and position any discussion accordingly.

As for uncertainty, we may explicitly offer a context with respect to the standards of the Semantic Web using the graph that our statements are asserted in. However, that context is still unclear semantically. The information we have about a graph is its assertion to statements. In an RDF dataset, we may find several graphs encapsulating statements, but having a statement inside a graph is not compulsory unless the semantics of that graph are well defined, as the default graph. Another thing to consider is the effect of the reliability of statements on one of their graphs and vice-versa: we may consider a certain open graph being affected by uncertain statements or for the latter to be considered certain once stated in a graph of the sort.

We included previously in section 3.2.2.4 an explicit relationship between a statement and the context it is mentioned in, saying that a *sentence* is *stated in* a *world*. When using RDF-star as a serialization syntax, we are allowed several ways of realizing that link:

- inserting the embedded sentence (or its identifier in other annotation methods) as a subject of a statement linking it to the world with `munc:statedIn`. This allows elaborating the link between the statement and the world. This link is a fact on its own that can be annotated with uncertainty.
- inserting the sentence in a graph that we consider as a world.

The first method associates uncertainty to the fact of linking embedded triple in a world as a separate thing. This can be time-consuming if compared with the second one. Moreover, it does not assert the embedded triples unless specified with the notation. The second method encapsulates all asserted triples in one place. Both methods require mentioning the triples every time they are linked to a world. In the first, that can be achieved using occurrences of the same embedded triple. In the second, mentioning the triple in the new graph is enough. We use worlds mainly for two purposes:

- Controlling the coherence of statements;
- Optimizing the uncertainty annotations for statements with the same provenance.

We do not think one method is better than the other, but the first one seems counter-intuitive to the idea of the world itself, while the second method seems more natural for both purposes. A query can be directly pointed to a specific graph to control the existence of contradictory statements. The latter can exist as they are within different graphs without specifying the semantics of occurrences. Nevertheless, the first method comes in handy in the absence of an encapsulation method (like named graphs or quads). As for *mUnc*, the annotations are linked to both the world and the sentence, allowing to cover the previous two cases.

We expect agents on the Semantic Web to be querying $n$ uncertain data sources $s_1, s_2, ..., s_n$, each possibly containing several graphs $G_{ij}, i \in \{1, \ldots, n\}$ representing each a set of coherent information. This means that each graph contains a set of triples that do not lead to a contradictory reasoning. For example, if the predicate `dbo:height` is a functional one (accepts a unique value), we cannot have both triples `(ex:StephanoTacconi, dbo:height, 188)` and `(ex:StephanoTacconi, dbo:height, 193)` in the same graph. We still can declare both triples in different graphs. We recall here the definition of RDF Dataset and perform the link with the set of contexts.

**Definition 4.1.** *(RDF dataset) An RDF dataset of a source $s_i$ is a collection of RDF graphs, containing one default graph $G_i$ and a set of named graphs, each consisting of a pair $(u_j, G_{ij})$ where $u_j$ is the IRI of the graph $G_{ij}$. The set of named graphs can be the empty set.*

As cited in [79], named graphs are suitable for context representation as they allow encapsulating a set of triples in a graph and annotate the latter with metadata. Also, each named graph can represent a vision or an opinion over the reality represented in the source. A sentence can be cited in multiple named graphs but with different uncertainty information. For example, two websites can state that tomorrow it will rain. The two websites may not be sure about that information at different levels, so they annotate it with different uncertainty information. Data from the previous websites can be encapsulated in a separate named graph, each representing a *context*.

**Definition 4.2.** *(Context) A context $C_{ij}, j \geq 0$ is a named graph $(u_j, G_{ij}), j \geq 0$ in the RDF Dataset of a source $s_i$.*

Each context can be annotated with a set of uncertainty information triples defined as follows.

**Definition 4.3.** *(Context Uncertainty) A context uncertainty $\mathcal{U}_{C_{ij}}$ is a set of pairs $(UncertaintyFeature, UncertaintyValue)$ representing the uncertainty information about the context $C_{ij}, j \geq 0$ in a data source $s_i$ .*

Triples in the default graph of the source $s_i$ may present a context on their own, and they are moved to a named graph $G_{i0}$ representing a separate context $C_{i0}$. The set of pairs $(\mathcal{U}_{C_{ij}}, C_{ij})$ represents the *contextual dataset* (noted as *CDS($s_i$)*) of data source $s_i$. Figure 4.1 illustrates the following definition.

**Definition 4.4.** *(Contextual Dataset) Given a data source $s_i$ and a set of Contexts $C_{ij}, j \geq 0$, each annotated with a set of metadata triples $\mathcal{U}_{C_{ij}}$, a contextual dataset $CDS(s_i)$ of a data source $s_i$ is a set where $C_i$ is the default context encapsulating metadata about other contexts, $C_{i0}$ is the context encapsulating triples which was*

*stored in the default graph $G_i$ of data source $s_i$.*

$$CDS(s_i) = \{(\mathcal{U}_{C_{ij}}, C_{ij}) \mid j \geq 0\} \tag{4.1}$$

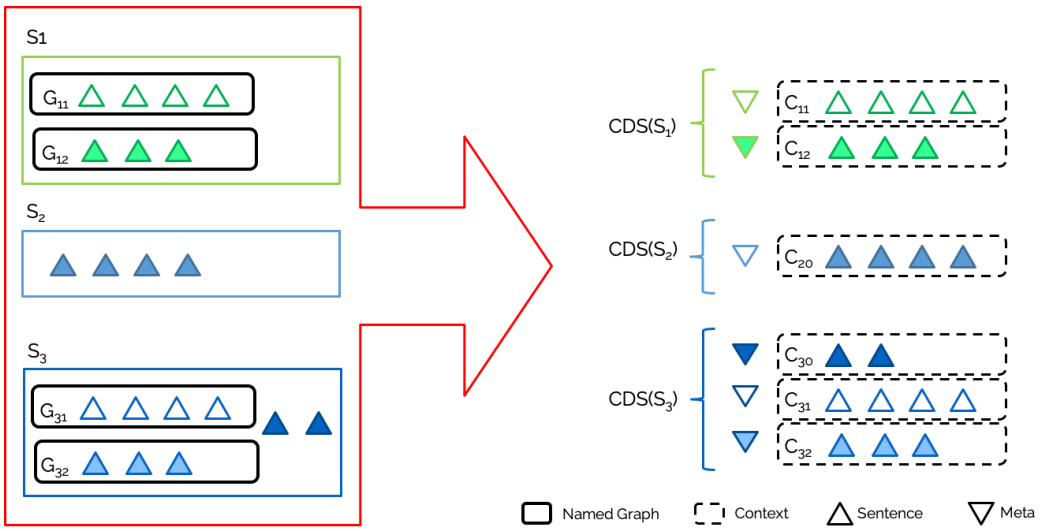



FIGURE 4.1: Example of context encapsulation

Similar to context uncertainty, a sentence can have its uncertainty information defined as follow.

**Definition 4.5.** *(Sentence Uncertainty) A sentence uncertainty $\mathcal{U}_{S_{C_{ij}}}$ is a set of pairs $(UncertaintyFeature, UncertaintyValue)$ representing the uncertainty information about the sentence $S$ in a context $C_{ij}$ .*

## 4.2 Mapping and Querying Uncertainty in Contextualized Uncertain Data

In a Semantic Web supporting the representation of uncertainty metadata, query results should include information about their uncertainty. The latter should be dependent on both sentences and their context.

We illustrate this need with the following example: we associate a subjective probability $p_t$ indicating our trust in one website. We decide that we do not trust this

website in all subjects but the economy, for which we grant a subjective probability $p_e$. We may achieve that with different methods:

- annotating statements of economics with $p_e$ and all other statements with $p_t$, then use a direct reading for each statement.
- creating a context for the economy, and annotate it with $p_e$, and all the other statements outside of it with $p_t$, then use a direct reading for other statements and a hereditary reading for statements in the economic context. This reading allows all statements to get the same uncertainty as to their context.
- considering the whole website as a context annotated with $p_t$, and the context of economics as a nested one annotated with $p_e$. In this case, use the hereditary reading on direct ascendants of the contexts. This allows the statements about the economy to get the uncertainty from their context. All other statements get their uncertainty from the website.

The previous list is not exhaustive, and solutions differ according to the result we want to achieve. In addition, it will be more complex if there is a statement we do not trust as much as the others inside the context of the economy. As a result, mapping each sentence $S$ with its uncertainty information requires defining a *metadata-mapping mode* (see table 4.1) to link between the metadata annotating the sentence and the one annotating the context in which the sentence is stated.

**Definition 4.6.** *(Meta-Mapping Mode) Given two sets of pairs* $A = \{(x,y) \mid x \in F_1\}, B = \{(w,z) \mid w \in F_2\}$. *A meta-mapping mode is the process linking* $A$ *and* $B$ *to a new set* $C$ *where* $C = \{(f,v) \mid (f,v_1) \in A, (f,v_2) \in B, v = v_1 \oplus v_2\}$ *with* $\oplus$ *being the composition uncertainty operator linked to the feature* $f$.

The reading for uncertainty metadata concerns the statement it annotates, the same as any other annotation for a direct resource. However, the scope for uncertainty may override that and allow several readings. For instance, when a graph $G$ containing a set of statements $\{t \mid t \in < s,p,o >\}$ is annotated with uncertainty metadata, the scope of uncertainty becomes less intuitive.

A distributive reading for the uncertainty values distributes or duplicates the uncertainty over all the statements included in $G$. For example, if $G$ is annotated with an uncertainty value $\mu$:

- a *distributive* reading states that $\forall t \in G, \mu_t = \mu$, hence $\mu_G = \mu^{|G|}$. It means uncertainty assigned to the graph via annotation was projected on all statements that the graph asserts. The uncertainty of the graph is then an association of the uncertainties of its statements.
- a *collective* reading states that $\mu_G = \mu$, hence $\forall t \in G, \mu_t = \sqrt[|G|]{\mu}$. It means uncertainty was meant for the graph as a whole, and it is read as the uncertainty of the statements collectively.

The two former readings are mere examples of how uncertainty can be understood. Other readings may establish that the uncertainty value associated with a set of statements touches some but not all the statements. These readings are related to what we define as meta-mapping modes, which select the way of choosing a set of uncertainty features, and what uncertainty value to associate to each feature based on their associated values in each of the initial sets. In the first two modes, only uncertainty information $\mathcal{U}_{C_{ij}}$ linked to the context $C_{ij}$ is considered with a specification of the reading. The third mode considers uncertainty information from the lowest level of granularity only, while the fourth mode enables inheriting context metadata but overrides the values for existing features in uncertainty information linked to the sentence.

The set of uncertainty information linked to a sentence $S$ regarding its context is denoted as *Universal Uncertain Information Set* or $\hat{\mathcal{U}}_{S_{C_{ij}}}$ and defined as follows.

**Definition 4.7.** *(Universal Uncertain Information Set) A universal uncertain information set $\hat{\mathcal{U}}_{S_{C_{ij}}}$ of a sentence $S$ in a context $C_{ij}$ of the data source $s_i$ is a set of $(UncertaintyFeature, UncertaintyValue)$ pairs issued by combining the sets $\mathcal{U}_{C_{ij}}, \mathcal{U}_{S_{C_{ij}}}$ using a meta-mapping mode.*

TABLE 4.1: Metadata mapping modes

| **Considered level of granularity** | **Mode** | **Explanation** |
|---|---|---|
| Context | Imposed | $\hat{\mathcal{U}}_{S_{C_{ij}}} = \mathcal{U}_{C_{ij}}$ |
| Lowest granularity only | Overriding | $\hat{\mathcal{U}}_{S_{C_{ij}}} = \begin{cases} \mathcal{U}_{S_{C_{ij}}} & \text{if } \mathcal{U}_{S_{C_{ij}}} \neq \varnothing \\ \mathcal{U}_{C_{ij}} & \text{otherwise} \end{cases}$ |
| Lowest granularity first | Override if Exists | $\hat{\mathcal{U}}_{S_{C_{ij}}} = \mathcal{U}_{S_{C_{ij}}} \cup \{(F,V) \mid (F,V) \in \mathcal{U}_{C_{ij}}, \nexists V' \ (F,V') \in \mathcal{U}_{S_{C_{ij}}}\}$ |
| Multi-level | Uncertainty Calculus | $\hat{\mathcal{U}}_{S_{C_{ij}}} = \{(F,V) \mid (F,V) \in \mathcal{U}_{C_{ij}}, \nexists V' \ (F,V') \in \mathcal{U}_{S_{C_{ij}}}\}$ $\cup\{(F,V) \mid (F,V) \in \mathcal{U}_{S_{C_{ij}}}, \nexists V'' \ (F,V'') \in \mathcal{U}_{C_{ij}}\}$ $\cup\{(F,V) \mid \exists V_{C_{ij}}, V_{S_{C_{ij}}} \ (F,V_{C_{ij}}) \in \mathcal{U}_{C_{ij}}, (F,V_{S_{C_{ij}}}) \in \mathcal{U}_{S_{C_{ij}}},$ $V = eval(Calculus(F), V_{C_{ij}}, V_{S_{C_{ij}}})\}$ |

### 4.2.1 Mapping a Sentence to its Uncertainty Set

The mapping between a sentence $S$ and its universal uncertainty information set $\hat{\mathcal{U}}_{S_{C_{ij}}}$ is a two steps process:

- Mapping both sentences and contexts to their uncertainty information. We denote $\mathcal{U}_{S_{C_{ij}}}$ the uncertainty information about the sentence $S$ cited in the context $C_{ij}$ and $\mathcal{U}_{C_{ij}}$ the uncertainty information about the context $C_{ij}$.
- $\mathcal{U}_{S_{C_{ij}}}$ is combined with $\mathcal{U}_{C_{ij}}$ using uncertainty operations linked to each feature, in order to evaluate its corresponding value in $\hat{\mathcal{U}}_{S_{C_{ij}}}$. In this step we apply the *metaList* algorithm (see Algorithm 1) translated form the formula in the fourth meta mapping-mode (see table 4.1): $\hat{\mathcal{U}}_{S_{C_{ij}}} = metaList(\mathcal{U}_{S_{C_{ij}}}, \mathcal{U}_{C_{ij}})$.

In our case, *Uncertainty Operations* are stored as Linked Functions. This feature in *Corese* [80] enables storing *LDScript* [74] functions in external SPARQL query files on the web to be called at the moment of query execution. The former feature permits publishing and executing the calculi of uncertainty approaches. Additionally, this approach may be extended to capitalize existing software libraries from other programming languages like C++ or Java.

Also, some sentences might be redundant in different contexts. A possible alternative would be the enrichment of RDF semantics to use occurrences of statements to store them in different named graphs. This gives more flexibility to the process and allows defining new methods and terms, allowing context-overlapping and context-selective querying.

*mUnc* also enables representing uncertainty about uncertainty information by considering the latter as sentences with uncertainty. Nevertheless, combining uncertainty information about uncertainty sentences with the information provided by the sentences themselves is challenging. The framework does not allow combining uncertainty information from multiple data sources using different uncertainty approaches

1: **procedure** METALIST($\mathcal{U}_{S_{C_{ij}}}, \mathcal{U}_{C_{ij}}$)
2: $uncFeatures(x)$: list of all uncertainty features contained in the set $x$
3: $uncValue(f, x)$: the value linked to the feature $f$ in the set $x$
4: $Operation(f)$: the uncertainty operation linked to the feature $f$
5: $eval(o, v1, v2)$: execute the operation $o$ passing the parameters $v1, v2$
6: $\hat{\mathcal{U}}_{S_{C_{ij}}} \leftarrow \varnothing$
7: **if** $\mathcal{U}_{S_{C_{ij}}} \neq \varnothing$ **then**
8: **for all** $f \in uncFeatures(\mathcal{U}_{S_{C_{ij}}}) \cap uncFeatures(\mathcal{U}_{C_{ij}})$ **do**
9: $v \leftarrow eval(Operation(f), uncValue(f, \mathcal{U}_{S_{C_{ij}}}), uncValue(f, \mathcal{U}_{C_{ij}}))$
10: $\hat{\mathcal{U}}_{S_{C_{ij}}} \leftarrow \hat{\mathcal{U}}_{S_{C_{ij}}} \cup \{(f, v)\}$
11: **end for**
12: **for all** $f \in uncFeatures(\mathcal{U}_{C_{ij}}) \setminus (uncFeatures(\mathcal{U}_{S_{C_{ij}}}) \cap uncFeatures(\mathcal{U}_{C_{ij}}))$ **do**
13: $v \leftarrow uncValue(f, \mathcal{U}_{C_{ij}})$
14: $\hat{\mathcal{U}}_{S_{C_{ij}}} \leftarrow \hat{\mathcal{U}}_{S_{C_{ij}}} \cup \{(f, v)\}$
15: **end for**
16: **else**
17: $\hat{\mathcal{U}}_{S_{C_{ij}}} \leftarrow \mathcal{U}_{C_{ij}}$
18: **end if**
19: **return** $\hat{\mathcal{U}}_{S_{C_{ij}}}$ ▷ the set of universal uncertainty information of the sentence S
20: **end procedure**

ALGORITHM 1: *metaList*: Universal Uncertainty Information Set of a sentence S in a context $C_{ij}$ of a data source $s_i$

for the same reasons. Sentences should be annotated using the same uncertainty approaches for the calculus to be executed. Otherwise, this presents no problem with other metadata since they will be appended to the presented information using the *metaList* algorithm. Providing a solution for the latter problem, we can add a third step to the previous two-step process in subsection 4.2 corresponding to the combination of uncertainty information of identical sentences issued from different contexts with different uncertainty information.

### 4.2.2 Querying for contextualized uncertainty

*mUnc* does not provide an extension of RDF Semantics. Instead, we rely on the SPARQL query language to provide a mapping between sentences and the uncertainty information presented to the user. Moreover, we consider *mUnc* as an approach to providing definitions of known and custom uncertainty theories, for which we do not provide any specific semantics. The possibility of defining a calculus alongside the ontology is an alternative to generalize and reuse the shared rules between uncertainty theories, such as maximizing or minimizing a feature.

We note $\mathcal{U}_{S_{C_{ij}}}$ the uncertainty information about the sentence $S$ cited in the context $C_{ij}$ and $\mathcal{U}_{C_{ij}}$ the uncertainty information about the context $C_{ij}$. Each sentence $S$ stated in a context $C_{ij}$ of a source $s_i$, will be mapped to a combined set of pairs (*Uncertainty Feature*, *Uncertainty Value*) issued from both sentence and context metadata (noted $\hat{\mathcal{U}}_{S_{C_{ij}}}$). This requires defining a metadata-mapping mode (see table 4.1).

The modes depend on the purpose of the application, the data itself, and the semantics of uncertainty theories. In the first mode, only uncertainty information linked to context $C_{ij}$ is considered. The second mode considers only pairs from the lowest level of granularity, while the third mode enables inheriting context metadata but overrides the values for existing features in uncertainty information linked to the sentence.

In our approach, we use a specific meta mapping mode which relies on uncertainty

```
@public
function munc:metaList(?xT, ?xG){
   let(SELECT ?xT ?xG (group_concat(?FV;separator="-") as ?metaD) WHERE
      {{
        SELECT ?xT ?xG (CONCAT(?xF,'=',?xV) AS ?FV) WHERE
        {
           ?xG ?xF ?xV1
           OPTIONAL {?xT ?xF ?xV2}
           ?xF rdfs:subPropertyOf munc:uncertaintyFeature
           ?xF ex:and ?xFFunction
           BIND(IF(BOUND(?xV2),funcall(?xFFunction,?xV1,?xV2),?xV1) AS ?xV)
        }
      } GROUP BY ?xT ?xG}
      UNION
      {{
        SELECT ?xT ?xG (CONCAT(?xF,'=',?xV) AS ?FV) WHERE
        {
           ?xT ?xF ?xV
           ?xF rdfs:subPropertyOf munc:uncertaintyFeature
           FILTER NOT EXIST {?xG ?xF ?xV2}
        }
      } GROUP BY ?xT ?xG}
      )
      {?metaD}
}
```

LISTING 4.1: *metaList* algorithm in *LDScript*

calculus to evaluate a new set of pairs based on both information from sentences and contexts. In the *metaList* algorithm we implement and use the last mode in the previous table (see Listing 4.1),

The `munc:metaList` function is declared in the example as "`@public`". This keyword is implemented in *Corese* as many others (`@define`, `@visitor`, `@trace`, ...) defining specific routines in the former Semantic Web engine. The keyword allows the previous code to be accessed globally in the engine through its reference without rewriting the function with each query. The listing 4.1 translates the `metaList` algorithm into *LDScript*. The result of binding this function in a SPARQL query is a string that groups all uncertainty features and their corresponding values from the Universal Uncertainty Information set of the corresponding sentence.

*Corese* also implements *Linked Functions* enabling storing *LDScript* [74] functions in external SPARQL query files on the Web. Such functions, referenced by *IRI*s, may be called at the moment of query execution. The former feature permits publishing and executing the calculi of uncertainty approaches.

The Semantic Web engine also allows defining specific routines preceding the query execution. One can integrate query transformation or precalculations of some variables. We implemented the previous meta-mapping mode in extension to the visitor `"@metadata"` and enabled rewriting SPARQL queries to simplify querying for uncertainty information. Using `"@metadata"` and with `munc:metaList` publicly defined, querying for the height of the football player *Stefano Tacconi* in a data source is as follows.

```
@metadata
prefix ex: <http://example.org/> .
prefix munc: <http://ns.inria.fr/munc/> .

SELECT * WHERE {
  ?player dbo:height ?height .
  ?player rdfs:label ?label .
}
```

Listing 4.2: Query rewriting using visitors implementing the access to uncertainty information

## 4.3 Negotiating Uncertainty on the Semantic Web

In addition to the previous two-step process leading to the generation of Universal Uncertainty Information Sets alongside sentences, users may prefer one theory or another.This section will discuss the translatability between uncertainty theories and how, using HTTP content negotiation (conneg), users may negotiate the theory they want for their results.

### 4.3.1 Translating uncertainty between theories

Many examples reject the claim that uncertainty can be represented only using probability theory. However, the belief about the uncertainty being the lack of information or the deficiencies due to a shortage of knowledge urges researchers to work on their unification, or at least that the different views may be linked. Dubois *et al.* [81] stated that transformation is helpful in any problem considering heterogeneous, uncertain, and imprecise data (e.g., subjective, linguistic-like evaluations and statistical data). Zadeh [82] cites the example of DempsterShafer theory which is a theory of random sets. The latter is a probability distribution of possibility distributions. An interesting analysis of the possibility-probability transformation and its links to graphical models can be found in [83].

With the use of our framework, every context will issue an answer to the user. If the answers are annotated with the same theory and the same set of features, this enables ranking the results or offers more options to control the results. In the example of search engines, this could support uniform criteria to order the results shown to the user. However, on an open Web where several open sources are queried, the results might use different features from different theories.

A translation must offer to transform a Universal Uncertainty Information Set $\hat{\mathcal{U}}_{S_{C_{ij}}}$ of a sentence $S$ annotated following an uncertainty approach $T_1$, to another set annotated with a different uncertainty approach $T_2$. The translatability of theories should consider several issues such as symmetry, reversibility, and the possible loss of information.

To fit in with the previous requirements, we define a translatability relationship between two uncertainty theories as follows:

**Definition 4.8.** *A theory $T_1$ has a translatability relationship with a theory $T_2$, if there exists a mapping $M : F_{T_1} \rightarrow F_{T_2}$ from the set of features $F_{T_1}$ represented in theory $T_1$ to the set of features $F_{T_2}$ represented in theory $T_2$ such that every possible feature of $F_{T_1}$ is mapped to a set of feature of $F_{T_2}$ semantically coherent with the*

*uncertainty initially expressed in* $T_1$. *We note* $T_1 >| T_2$.

The former definition is valid for all theories that have a relationship allowing the conversion of features from one theory to another, regardless of the loss of information. In case the conversion does not generate a loss of information allowing the reversibility of the operation, we define the relationship as follows:

**Definition 4.9.** *A theory* $T_1$ *has an ideal translatability relationship with a theory* $T_2$, *if* $T_1$ *is translatable to* $T_2$ *(*$T_1 >| T_2$*) and there is no loss of information in the translation. We note* $T_1 \gg T_2$

We should mention that an ideal translatability might not be reversible, regardless of the semantics of the translatability. The loss of information disables the backward operation. If the other case is considered, where we have no loss of information, then we can define a full translation as follows:

**Definition 4.10.** *A theory* $T_1$ *has a full translatability relationship with a theory* $T_2$, *iff* $T_1$ *is ideally translatable to* $T_2$ *(*$T_1 \gg T_2$*) and, inversely,* $T_2$ *ideally translatable to* $T_1$ *(*$T_2 \gg T_1$*). We note* $T_1 \otimes T_2$.

Using our *mUnc* vocabulary and the framework previously proposed, we are able to formalize the translation (if it exists) between the different theories. For this, we extended *mUnc* with the set of the following properties:

- `munc:hasTranslation` (definition 4.8)
- `munc:hasIdealTranslation` (definition 4.9)
- `munc:hasFullTranslation` (definition 4.10)

These QNnames respectively identify the previous definitions. Figure 4.2 shows the extension, where each property of the previous set has for domain an uncertainty approach, and for range a blank node pointing to both the destined theory and the *IRI* of the translation function, written in *LDScript*.

By definition, the translativity properties have several algebraic properties:

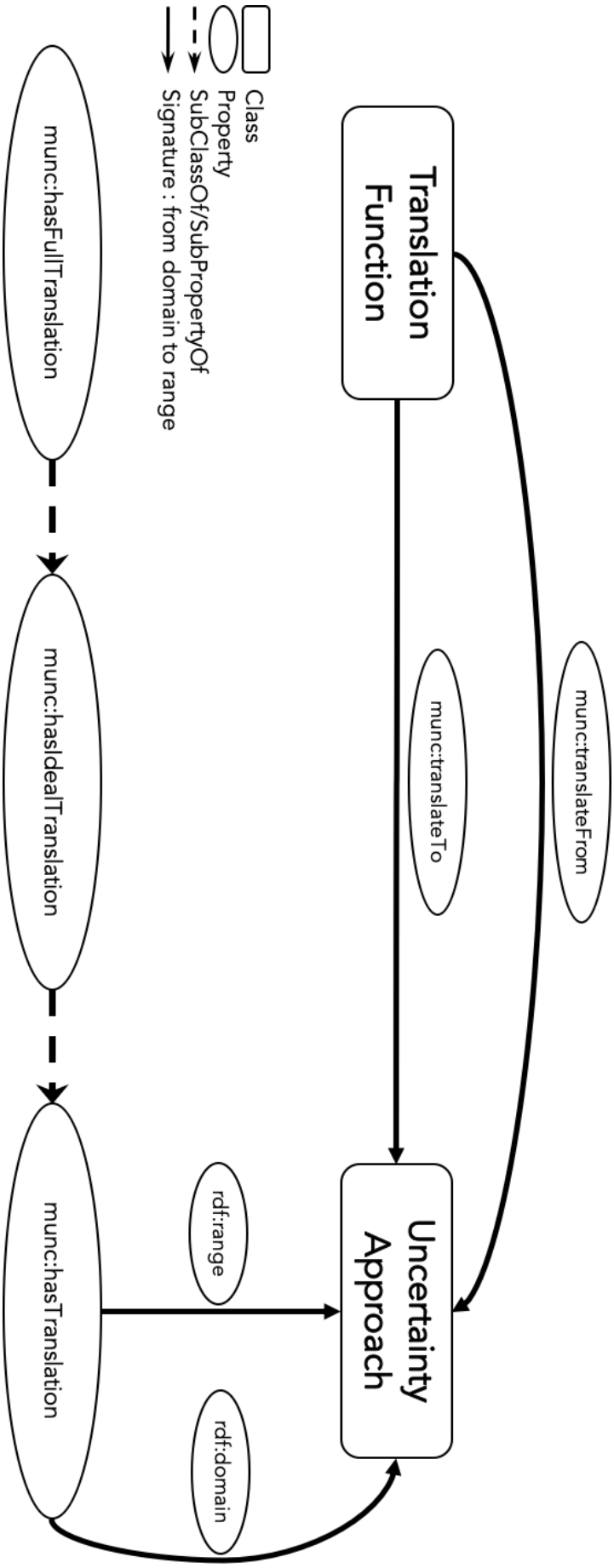



FIGURE 4.2: Extending *mUnc* ontology with translation properties

**Definition 4.1.** *(Transitivity of translatability) Let* $T_i, i = 1, 2, 3$ *be three uncertainty theories. If* $T_1 >| T_2$ *and* $T_2 >| T_3$ *then* $T_1 >| T_3$.

**Definition 4.2.** *(Reflexivity of translatability) Each uncertainty theory has a full translatability with itself. We note* $T \otimes T$.

**Definition 4.3.** *(Symmetry of full translatability) Let* $T_1, T_2$ *be two uncertainty theories. If* $T_1 \otimes T_2$ *then* $T_2 \otimes T_1$.

**Definition 4.4.** *(Full translatability is an equivalence relation) because it is transitive, symmetric, and reflexive.*

We can note that full translatability being an equivalence relation allows us to form equivalence classes by transitive closure $i$, which we have translatability with no loss of information from a theory $T_i$ to any other theory $T_j$ of its class. We note this set $TC_{\mathcal{U}}(T_i)$.

To illustrate the previous extension, we propose to represent the example proposed in [84] about the Optimal Transformation (OT) from probability to possibility. We declare a translatability relationship between probability theory `ex:Probability` and possibility theory `ex:Possibility` representing the two different uncertainty theories. We enrich the data source with the triples below, where `ex:translateProbaToPoss` is an *LDScript* function.

```
ex:Probability munc:hasTranslation ex:Possibility.


ex:translateProbaToPoss munc:translateFrom ex:Probability.
ex:translateProbaToPoss munc:translateTo ex:Possibility.
```

### 4.3.2 Negotiation with Uncertainty Headers

Based on the previous model, we can now support the possibility of negotiating answers annotated with different uncertainty theories. Content negotiation can be based on HTTP headers or non-HTTP methods such as query arguments in *IRI*s.

Following the W3C working draft proposed by Svensson *et al.* [85] we propose that clients may negotiate a representation annotated with a specific uncertainty theory, using $q$-values to express their preference regarding the uncertainty theories they are to receive. Since uncertainty theories are already defined using $mUnc$ and named with $IRI$s, both server and client can exchange and verify the conformity of their options. We propose to handle three use cases:

1. Uncertainty information exist in the queried source under one or many requested uncertainty theories. We answer with the first theory selected by the user. In the example, uncertainty information is issued from a context annotated with `evidence`.

```
GET /some/resource HTTP/1.1
Accept: text/x-turtlestar;q=0.9;uncertain="http://example.org/
    probability",
      text/x-turtlestar;q=0.7;uncertain="http://example.org/evidence";


HTTP/1.1 200 OK
Content-Type: text/x-turtlestar;uncertain=<http://example.org/evidence>
```

2. The data source does not offer direct information about all the requested theories, but a translation from existing uncertainty information to one or more requested theories is available. In this example, uncertainty is available in the probability theory. The returned information is evaluated using the function `ex:translateProbaToPoss` and presented to the client with an indication about the type of translation the data underwent.

```
GET /some/resource HTTP/1.1
Accept: text/x-turtlestar;q=0.7;uncertain="http://example.org/
    possibility";


HTTP/1.1 200 OK
```

```
Content-Type: text/x-turtlestar;uncertain=<http://example.org/
    possibility>;translation=full
```

We note that the selection of a suitable translation starts from the transitive closure of full translations $TC_{\mathcal{U}}(\texttt{possibility})$ offering more information and graduates to the normal translatability relationship.

3. The data source has no information about the theory and no available translations. We answer the user with the existing information. The default uncertainty information proposed by the server is returned in such a case.

```
GET /some/resource HTTP/1.1
Accept: text/x-turtlestar;q=0.9;uncertain="http://example.org/
    probability",
      text/x-turtlestar;q=0.7;uncertain="http://example.org/evidence";


HTTP/1.1 200 OK
Content-Type: text/x-turtlestar;uncertain=<http://example.org/
    possibility>;default=true
```

## 4.4 Conclusion: Leveraging Uncertainty Contexts

The focus on uncertainty translatability was mainly in AI-based applications. We point that the Semantic Web requirements in term of interoperability and information usage are fundamental and the suitability of uncertainty theories to different types of data and applications need to be further explored. We also offered the possibility of translating between uncertainty theories and negotiating uncertainty information following a specific theory. The translation process is also the first step that enables merging uncertain data annotated using different uncertainty approaches.

The next chapter follows with our proposition of an approach to extract uncertainty from sources with respect to specific use cases.

# Chapter 5

# Extracting Uncertainty from Implicit Uncertain Data Source for Task-Oriented Evaluations using Graph Interlinks

We are witnessing an era fulfilling the vision to create a Web of linked intelligent systems [5], thriving through sharing data they own or have processed. In this context, many challenges present themselves to developers of such platforms to retain reliable data that allows enriching their existing knowledge bases using robust reasoning or with the help of more external relevant content. The latter is using links with extra pieces of information revealing new dimensions for users to explore with their requests. On the latter, data can be filtered through different funnels allowing to confirm their consistency with respect to the layers of its stack. In other words, we consider the Semantic Web as a "task-ready" environment for reliability validation and uncertainty assessment, as it offers a granularized and enriched representation of data. Moreover, the interlinking between different resources enables browsing, analyzing, and reasoning over the graphs [4]. Uncertainty is a major issue when related to content brought out on the Web, or Semantic Web by extension. Nevertheless, most

data providers do not present explicit information about the uncertainty of their data. On the other hand, completely mistrusting a data source is unfair: while some data providers may not be reliable on one subject or provide false information about it, they are experts on other subjects and the pieces of information they provide should not be ignored. In some cases, references about data provenance and/or related data are given, from which a data consumer may hope to get further validation from other data sources.

We address in this chapter the need to evaluate uncertainty in linked data sources. In our approach, a data source may auto-evaluate the level of uncertainty of its data according to what is being presented by other data sources and for a specific use-case. We leverage the fact that different knowledge graphs may provide complementary and/or extra information enabling the assessment of the conformity of a target source. We also think that a user's preferences should be taken into consideration while evaluating uncertainty. Our work is built on top of the *mUnc* model [64] introduced in chapter 3 to represent and publish uncertainty on the Semantic Web. The main question we aim to answer is: *How to evaluate uncertainty in a data source, based on its data, other linked data sources, and with respect to a specific use-case?*

To answer this question, we propose an approach to evaluate the uncertainty of a target data source, based on graph interlinks with other reference data sources. We propose to annotate statements with uncertainty values in a publishable format and provide a method to manipulate and update such values if existed. In the first, we propose to evaluate uncertainty by using graph interlinks. To do so, we extract a set of links supporting each interlink, using different metrics for syntactic and statistical semantic similarity. We then represent such measures as publishable, reusable uncertainty information that can be updated when new information presents itself to the data source. The intuition behind this work is that often users who need to confirm a piece of information will look for different sources that confirm or contradict it. For instance, the traditional verification techniques in journalism include the "two-sources rule" asking to verify that at least two independent trustworthy sources confirm a piece of information.

The rest of the chapter is organized as follows. Section 5.1 surveys related work and positions our contribution accordingly. In section 5.2 we discuss similarity assessment between two focus graphs of one resource and our choices of indicators. In section 5.3 we present our main contribution, with a method to evaluate uncertainty based on existing links and transform it into reusable information that annotates statements in the data source of interest. In section 5.4 we discuss the experimental workflow and present our tool for uncertainty evaluation and annotation. The work in this chapter was published in the International Conference on Knowledge Engineering and Knowledge Management EKAW2020 [86].

## 5.1 The Need for Uncertainty Extraction

Users check the consistency of information before assessing its truth. This fact is always context-related: a user looking for information about the last concert of singer *Whitney Houston* might give more importance to the reliability of the resource regarding *Whitney Houston* than other singers. Moreover, asked about their knowledge regarding a specific subject, a person would answer according to their proper expertise and the support they get from their trusted backers, confirming the reliability of their information. One may not be a music expert, but be a fan of *Whitney Houston* with detailed knowledge about her.

We take this hunch to the case where two resources from different data sources are linked using a similarity link (e.g. an `owl:sameAs` link). We want for instance to see if *MusicBrainz*[1] is reliable regarding information about *Whitney Houston*,[2] with respect to her page on *Wikidata*[3] for which the link was provided. Both references are about the same artist, yet in some cases, one reference may provide a wrong birth date or a misspelled name, or does not mention the complete list of singles of the artist.

[1] https://musicbrainz.org/
[2] http://dbtune.org/musicbrainz/resource/artist/0307edfc-437c-4b48-8700-80680e66a228
[3] https://www.wikidata.org/wiki/Q34389

Hence, we translate the link between the two resources into a set of links between two graphs, each representing the description of a resource on one data source. While the similarity link remains the same, the ones selected to perform the analysis alongside the definition of the graphs are a user's preference. The sets of selected links can be analyzed then generalized over other resources. Similarly to when one is tested on how reliable they can provide information about a specific subject.
Hereby is an example to discover what can be wrong with data and to what extent, from a user's perspective and with the help of selected references to trust. Referring to such anomalies might be a trigger for an automatic process of correction, or crowd-sourcing the correct answer. This can be built upon the work presented in the previous chapters [64], [65] to offer referenceable exchangeable uncertainty metadata.

The works mentioned in section 2.3 mostly treated the reliability of the similarity links between data sources or detecting wrong schema-mappings. This differs from our problem that requires analyzing data based on a use-case. The previous works present a promising set of measures to analyze data uncertainty based on links. Nevertheless, we notice the absence of specific sets of interest encapsulating the linked resources. Moreover, the said works are more in the spirit of ontology-matching techniques relying on linking all instances of two classes.

The problem relates in general to ontology alignment approaches and is also inspired by quasi-key detection problems. Most of the literature is assessing the link quality and not depending on the links themselves to assess data quality. We believe that it is original to discuss uncertainty evaluation with a task-centered perspective based on graph interlinks .

To sum up, our problem is about evaluating uncertainty based on graph interlinks. The previous works that treated error-detection were either about evaluating the quality of interlinks themselves, detecting wrong literal values according to the whole graph, or assessing the quality of the mapping between the schemas of different data sources.

## 5.2 Uncertainty Assessment in Linked Data

### 5.2.1 Terminology and Definitions for Uncertainty

We introduce the terminology and the formalism used in this chapter to propose an evaluation of uncertainty based on existing links between graphs.

**Definition 5.1.** *RDF-dataset — a set of statements (triples) in the form* $\langle subject, predicate, object \rangle \in (I \cup B) \times I \times (I \cup B \cup L)$ *where* $I$ *is a set of IRIs,* $B$ *a set of blank nodes,* $L$ *a set of literals,* $I, B$ *and* $L$ *are pairwise disjoint and for every two RDF-datasets* $D_1, D_2$ *the sets of blank nodes are disjoint. we also denote* $I_D$ *the set of IRIs used in statements of the RDF-dataset* $D$*.*

**Definition 5.2.** *Target dataset — an RDF-dataset noted as* $D_t$ *that is the target of the uncertainty evaluation.*

**Definition 5.3.** *Reference dataset — an RDF-dataset noted as* $D_r$ *that represents a reference for the evaluation of the uncertainty of a target dataset.*

**Definition 5.4.** *RDF-graph — a graph* $G = (V, E)$*, where* $V \subset (I \cup B \cup L)$ *is a set of vertices, and* $E \subset I$ *is a set of directed edges.*

**Definition 5.5.** *Focus graph — an RDF-graph noted as* $G_D(e) \subset D$*, where* $D$ *is the dataset including the graph (target or reference) and* $e \in I$ *is a focused resource for which* $G_D(e)$ *is considered representative according to the use-case.*

**Definition 5.6.** *Set of Linking predicates — a non-empty set of predicates explicitly chosen to link between the target dataset and the reference dataset. We note it as* $P_l \subset I$*. Example:* $P_l = \{$`owl:sameAs`, `skos:exactMatch`$\}$*.*

**Definition 5.7.** *Contextual Linkset — as defined in the VOID vocabulary,*[4] *a linkset is a set of RDF triples where all subjects are in one dataset and all objects are in another dataset. We call a contextual linkset the one containing links between focused resources of* $D_t$ *and those of* $D_r$*. A contextual linkset defines the set of focused resources of each dataset as well as the links between them. A link between a target*

[4] https://www.w3.org/TR/void/#linkset

*focused resource $e_t$ and a reference focused resource $e_r$ is also a link between the focus graphs $G_{D_t}(e_t)$ and $G_{D_r}(e_r)$: $LS(D_t, D_r) = \{\langle e_t, p, e_r\rangle \mid p \in P_l, e_t \in I_{D_t}, e_r \in I_{D_r}\}$;*

**Definition 5.8.** *Evidence link — a relationship between two statements $t_t \in G_{D_t}(e_t)$, $t_r \in G_{D_t}(e_r)$ discovered using similarity analysis, that supports the link between two linked focus graphs $G_{D_t}(e_t)$ and $G_{D_t}(e_r)$. The evidence link refers to a considered relationship between the predicates and/or the objects of two statements $t_t$ and $t_r$. We note $E(G_{D_t}(e_t), G_{D_t}(e_r))$ the set of evidence links discovered between the two focus graphs $G_{D_t}(e_t)$ and $G_{D_t}(e_r))$.*

Our purpose is to find a method to assess the reliability of the information in each target focus graph $G_{D_t}(e_t)$ centered around a target focused resource $e_t$. To this end, we translated the existing link between the resource $e_t$ of a target dataset and the resource $e_r$ of a reference dataset ( $\langle e_t, p, e_r\rangle \in LS(G_{D_t}(e_t), G_{D_t}(e_r)), p \in P_l$) to a set of evidence links between the target focus graph $G_{D_t}(e_t)$ and the reference focus graph $G_{D_t}(e_r)$. We statistically analyze the extracted evidence links to obtain a set of indicators enabling the evaluation of the overall semantic similarity between the predicates of linked focus graphs. Finally, we use the extracted evidence links to calculate the uncertainty of each focus graph based on its local ones.

### 5.2.2 Choosing Target Focused Resources

The problem of matching, whether it is data-driven or schema-driven, is context-related and may not be evident to users or useful for their request if done without involving them in the process [87]. We consider the concept of uncertainty to be also context-specific and that it is possible to choose a different evaluation method for each use case.

A focus graph $G_D(e)$ is meant to be the image that represents $e$ in the context of the application. Hence, the choice of the set of focused resources is necessary to ensure that uncertainty assessment is built on a user-centered view. The set of targeted focused resources $e \in I_{D_t}$ ($I_{D_t}$ being the set of IRIs in the dataset $D_t$) depends

on the type of validation a user intends to have within the data-source and depending on the use-case.

In the case of *MusicBrainz*, we believe that uncertainty of the statements representing the song data will not be the same if treated as a part of a focus graph of an artist, or as a focused resource itself. The former definition holds for any use-case that may have custom requirements. Another example is the validation of geographical entities such as points of interests that can be based on specific resources from different classes (city, country, building, etc) and following some constraints of coordinates (ex.inside borders of a country, or linked to a book resource for fact-checking, etc).

In works treating ontology matching from a statistical point of view [58], [62], focused resources are all instances of one class in the target dataset that is a candidate to be mapped to another reference class in the schema of the reference dataset. Also, the whole dataset is seen as the focus graph for each resource.

### 5.2.3 Concise Bounded Description

To bridge user choices with uncertainty evaluation, proposals such as the concept of *RDF Molecules* [88] inspire clustering statements in a way to leverage the inherent structural and data redundancy in RDF streams. Other works propose the concept of *local networks* [89] that uses a fixed level selection of nodes and edges, starting from one centered node on the graph.

We also need to present a sufficient focus graph —in the context of the use-case— reflective of information about the resource. As an example with music artists, a focus graph may contain simple information like their names and birthplaces and deeper-level information like songs from their albums. Our work is based on existing links. In the context of the Semantic Web, there is no such formalization in the current standards to reflect such a definition. One may use named graphs [79] or simply attribute the set of triples to a resource. Yet no specific semantics are stipulated for

the former proposals and their use still depends on the application. We previously proposed to encapsulate triples and their sets of interests on-the-go in contexts and query those for related uncertainty information [64].

To limit the issue in our current use-case, we rely on the proposal[5] made by Strikler, aiming to create a focused subgraph centered around and describing a resource, called a *Concise-Bounded Description* and noted as *CBD*. The formal definiton of a *CBD* is a union between three different subgraphs, noted in Formula 5.1: a set of statements directly related to the resource $C_{direct}$, a set of recursively-linked statements to the resource via blank nodes $C_{blank}$, and the set of reified statements which identifiers are found in the previous two sets $C_{reification}$.

$$\begin{aligned}
C_{direct}(e,D) &= \{\langle e,p,o\rangle | \langle e,p,o\rangle \in D\} \\
C_{blank}(e,D) &= \{\langle s,y,z\rangle | \langle e,p,o\rangle \in C_{direct}(e,D), \\
&\qquad o \in B, \langle s,y,z\rangle \in C_{blank}(o,D), \\
&\qquad \langle s,y,z\rangle \notin C_{direct}(e,D)\} \\
C_{reification}(e,D) &= \{\langle x,y,z\rangle | \langle s,p,o\rangle \in C_{blank}(e,D), \\
&\qquad \langle o, \texttt{rdf:type}, \texttt{rdf:Statement}\rangle \in D, \\
&\qquad \langle x,y,z\rangle \in C_{blank}(o,D), \langle x,y,z\rangle \notin C_{blank}(e,D)\}
\end{aligned} \tag{5.1}$$

In a *CBD* linked to a resource $e$, we find a focused body of knowledge aiming to describe the focused resource, *i.e.* a set of triples linked to the resource $e$ and known to be representative of the resource. Some Linked Data stores like *Virtuoso*[6] propose their proper definition of *CBD* and use it as the mapping of `DESCRIBE` SPARQL queries.

$$CBD(e,D) = C_{direct}(e,D) \cup C_{blank}(e,D) \cup C_{reification}(e,D). \tag{5.2}$$

For our current use-case, we find the definition of *CBD* an intuitive, simple yet interesting one to define our $G_D(e)$. We choose to have:

$$G_D(e) = CBD(e,D). \tag{5.3}$$

[5] https://www.w3.org/Submission/CBD/
[6] http://docs.openlinksw.com/virtuoso/rdfsqlfromsparqldescribe/

More parameters can be linked with the choice of *CBD* definition. The former proposition states also other particular definitions, such as the *symmetric* one where the resource is placed as an object in the first layer of linked triples. Other works [90] use an alternative definition with a fixed depth.

### 5.2.4 Linking Predicates and Contextual Linkset

Unlike the approaches to ontology matching or alignment, we take existing links in the contextual linkset as ground truth. The first links one may find between two data sources can be established by reusing IRIs of resources from one in the other. Moreover, the *RDFS* and *OWL* standards provide predicates such as `owl:sameAs`, `rdfs:seeAlso` with debatable semantics to link between resources [14], [91]. Other commonly used ontologies propose more predicates to indicate the matching between two resources (example: `skos:exactMatch` [92]). The choice of predicates in $P_l$ depends mostly on the data: a user may decide to use a custom set of linking predicates to link two focus graphs from two data sources.

## 5.3 Uncertainty Assessment Approach

We propose a level-based architecture where each level depends on the previous one, from isolating candidate evidence links to exporting update-ready uncertainty values. A link between a target focused resource $e_t$ and a reference focused resource $e_r$ can be seen as a link between the focus graph of each. The evidence links supporting that link are discovered and selected based on defined similarity indicators. The architecture in question is illustrated in figure B.1.

In the next parts, we consider two statements $t_1 : \langle s_1, p_1, o_1 \rangle, t_2 : \langle s_2, p_2, o_2 \rangle$ where $t_1 \in G_{D_t}(e_t)$ and $t_2 \in G_{D_r}(e_r)$ and a prior knowledge indicating the existence of a link between the two resources $e_t$ and $e_r$: $\langle e_t, l, e_r \rangle \in LS(D_t, D_r)$.

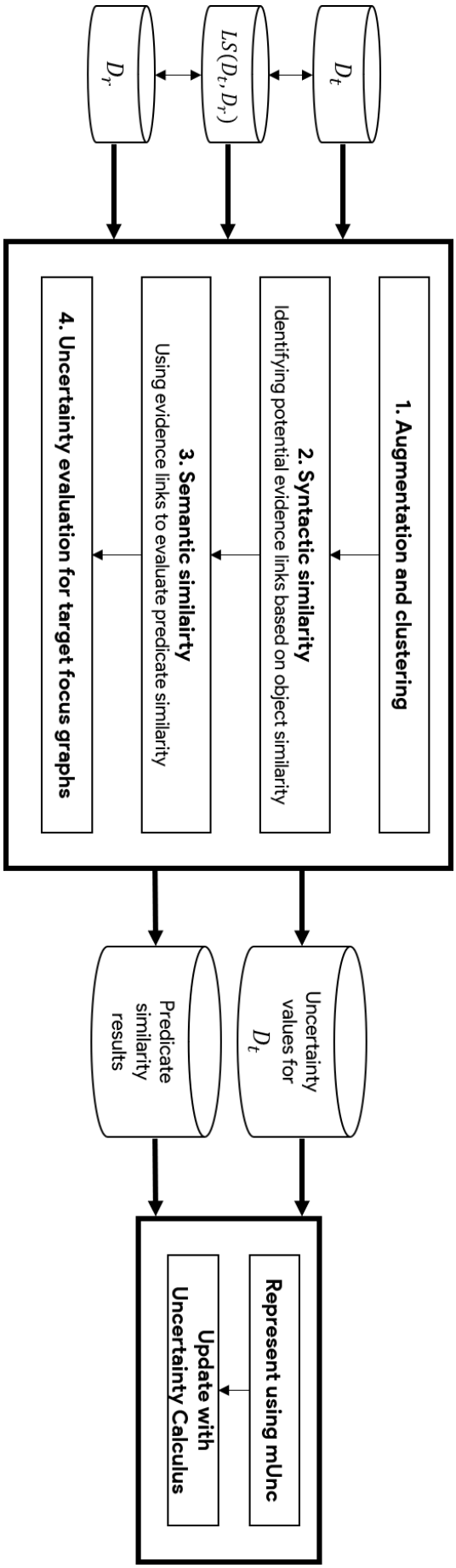



FIGURE 5.1: Pipeline for Uncertainty evaluation based on graph interlinks

### 5.3.1 Precomputing: Augmentation and Clustering

During this step, we apply the chosen definition of focus graphs on $D_t$ based on $LS(D_t, D_r)$. Beforehand, we use *OWL* [91] semantics for properties to augment the data source by evaluating the deductive closure of our target dataset $D_t$. This helps to unveil more potential evidence links between the linked focus graphs. In the example of Whitney Houston, this step leads to the creation of a graph around the resource `ex:WhitneyHouston`. The graph includes the set of statements that are representative to the resource in question and is then enriched via deductive closure.

### 5.3.2 Level 1: Identifying Possible Evidence Links based on Syntactic Similarity between Objects of Statements in Linked Focus Graphs

In the first level, we produce a set of evidence links for each pair of linked focus graphs using an *object similarity* measure defined as follows.

**Definition 5.9.** *Object similarity — We denote by $sym_o(t_1, t_2)$ (eq. 5.4) as the weighted similarity between objects of statements $t_1$ and $t_2$ (between $o_1$ and $o_2$). This measure refers to what extent the two objects share the same nature (literal, URI), the same datatype(`xsd:short`, `xsd:integer`, etc.[7]) and/or the same value:*

$$sym_o(t_1, t_2) = (1 - \omega_{val}) \times typeMatch(o_1, o_2) + \omega_{val} \times valMatch(o_1, o_2). \quad (5.4)$$

The binary function *typeMatch* returns 1 if both nature (IRI, Literal) and datatypes are similar and 0 otherwise. The *valMatch* function can be any syntactic similarity measure (Jaccard, Levenshtein, Jaro-Winkler distance, $n$-grams, etc.). Once the first level measures are established, a positive threshold $\tau_{obj} \leq 1$ restricts the discovered evidence links to ones of higher object similarity. The weight $0 \leq \omega_{val} \leq 1$ gives

[7] https://www.w3.org/2011/rdf-wg/wiki/XSD_Datatypes

preference to one of the matching functions if that information is not likely to be provided by the data sources, or according to the choice of the user.

Object similarity (said *syntactic*) can be considered as a probability measure, reflecting the resemblance of two literals. For example, one measure of the degree of resemblance between two literals containing the complete name of Whitney Houston is the probability that letters in the same position are the same or the probability that some words exist. For instance, we may compare the birth name *"Whitney Elizabeth Houston"* to the name *"Whitney Houston"* and get a measure of 0.67 using Jaccard's distance between words. The quantity here depends on the method chosen for the measurement.

### 5.3.3 Level 2: Identifying Evidence Link Patterns using Semantic Similarity of Predicates in the Overall Linked Focus Graphs

The second level introduces semantic similarity between evidence links while taking into account: the fact that the same predicates are used in schemas of the different data sources, and specific semantics related to the current use case by the mean of predicate similarity indicators. This view is inspired by the example in [63] but adapted to fit predicates due to the generalized, class-independent definition of $LS(D_t, D_r)$.

**Definition 5.10.** *Predicate similarity — We denote by* $sym_p(t_1, t_2)$ *(eq. 5.6) the statistical similarity between predicates of statements* $t_1$ *and* $t_2$ *(between* $p_1$ *and* $p_2$*). This measure is built on all the linked focus graphs and represents the use-case related semantic similarity of the two predicates* $p_1$ *and* $p_2$*.*

To evaluate semantic similarity, we first define five indicators $I_1, .., I_5$ (table 5.1) to be statistically extracted for each pair of linked focus graphs $G_{D_t}(e_t)$ and $G_{D_r}(e_r)$. We then use these indicators for each pair of linked focus graphs $G_{D_t}(e_t)$ and $G_{D_r}(e_r)$ to calculate three local ratios $R_1, R_2, R_3$ (table 5.2).

TABLE 5.1: Semantic similarity indicators for each pair of linked focus graphs.

| Indicator | Definition |
|---|---|
| $I_1(G_{D_t}(e_t), G_{D_r}(e_r))$ | the number of evidence links between the two focus graphs $G_{D_t}(e_t)$ and $G_{D_r}(e_r)$. i.e. the number of links supporting the similarity hypothesis between the two resources $e_t$ and $e_r$. |
| $I_2(G_{D_t}(e_t), G_{D_r}(e_r))$ | the set of predicate pairs in evidence links between statements of the two focus graphs $G_{D_t}(e_t)$ and $G_{D_r}(e_r)$. i.e: the set of pairs $(p_1, p_2)$ where an evidence link exists between $t_1$ and $t_2$. |
| $I_3(G_{D_t}(e_t), G_{D_r}(e_r), p_1, p_2)$ | the count of evidence links relying on two predicates $p_1, p_2$ between the two focus graphs $G_{D_t}(e_t)$ and $G_{D_r}(e_r)$. |
| $I_4(G_{D_t}(e_t), G_{D_r}(e_r), p_1, p_2)$ | the total number of possible combinations between statements using each $p_1$ or $p_2$ in the two linked focus graphs $G_{D_t}(e_t), G_{D_r}(e_r)$ (For instance, if three statements in $G_{D_t}(e_t)$ use $p_1$ and two statements in $G_{D_r}(e_r)$ use $p_2$ then the total number of links would be six. So this represents the maximum possible number of evidence links that can be found linking $p_1$ and $p_2$). |
| $I_5(G_{D_t}(e_t), G_{D_r}(e_r), p_1, p_2)$ | the sum of the quality of evidence links relying on two predicates $p_1, p_2$ between the two focus graphs $G_{D_t}(e_t)$ and $G_{D_r}(e_r)$. i.e. the sum of object similarities of discovered evidence links between $G_{D_t}(e_t)$ and $G_{D_r}(e_r)$ linking statements using respectively $p_1$ and $p_2$. |

To evaluate the semantic similarity between $p_1$ and $p_2$ on the overall contextual linkset, we evaluate three averaged ratios $\hat{R}_1, \hat{R}_2, \hat{R}_3$ for each pair of predicates $p_1$ and $p_2$ with an evidence link between $t_1$ and $t_2$ in all linked focus graphs, and add another indicator $\hat{R}_0$ for the equality $p_1 = p_2$ (as it will stay the same if averaged). We get a vector of averaged ratios $\hat{R}(p_1, p_2) = [\hat{R}_0(p_1, p_2), \hat{R}_1(p_1, p_2), \hat{R}_2(p_1, p_2), \hat{R}_3(p_1, p_2)]$, with

$$\hat{R}_i(p_1, p_2) = \frac{1}{|LS(D_t, D_r)|} \sum_{\langle e_t, p_l, e_r \rangle \in LS(D_t, D_r)} R_i(G_{D_t}(e_t), G_{D_r}(e_r), p_1, p_2) \tag{5.5}$$

and for which we define a vector of semantic weights $\omega_{sem} = [\omega_0, \omega_1, \omega_2, \omega_3]$ with $\sum \omega_i = 1, \omega_i \geq 0$. We select only the predicate pairs having an average of link quality equal or greater than a positive defined threshold $\tau_{sem}$ where $\tau_{sem} \leq \hat{R}_3(p_1, p_2) \leq 1$. Hence, we can define $sym_p(t_1, t_2)$ of statements $t_1$ and $t_2$ as the dot product of the

TABLE 5.2: Normalised local ratios for each pair of linked focus graphs.

| Ratio | Definition |
|---|---|
| $R_1(G_{D_t}(e_t), G_{D_r}(e_r), p_1, p_2)$ | $I_3$ is normalised using $I_1$ to reflect the participation of evidence links between two statements having $p_1$ and $p_2$ as predicates, in the overall evidence links between the two linked focus graphs. |
| $R_2(G_{D_t}(e_t), G_{D_r}(e_r), p_1, p_2)$ | $I_3$ is normalised using $I_4$ to reflect the portion of existing statement that actually participate with a link. If all existing statements between two focus graphs, with $p_1$ and $p_2$ as predicates are linked with evidence links, it indicates that the predicates may be functional, or that this information is a common knowledge that usually have a lower cardinal (like homepages for artists). |
| $R_3(G_{D_t}(e_t), G_{D_r}(e_r), p_1, p_2)$ | $I_5$ is normalised using $I_3$ to get the average quality of each evidence link between statements having $p_1$ and $p_2$ as predicates. |

two vectors $\hat{R}(p_1, p_2)$ and $\omega_{sem}$:

$$sym_p(t_1, t_2) = \omega_{sem} \cdot \hat{R}(p_1, p_2) \tag{5.6}$$

Similarly to the previous level, the overall quality of considered evidence links should also respect the average quality threshold $\tau_{sem}$. To summarize, each statement in the target focus graph is candidate to have two measures: *object* and *predicate* similarity. The first is related to a particular evidence link between two statements while the second requires a general study to the set of all evidence links and is based on the previously defined indicators. We see this measure as an application of the law of total probabilities: the probability that two predicates are similar is a sum of products, with the selected weights that sum up to 1 representing the probability that an indicator is the one responsible for defining the semantic similarity, and the average indicators as the probability that the indicator in question is of quality.

### 5.3.4 Level 3: Evaluating Contextual Uncertainty of Target Focus Graphs

At this level, the previous similarity measures are combined into one value reflecting the degree of uncertainty of a target focus graph $G_{D_t}(e_t)$ regarding its linked reference focus graph $G_{D_r}(e_r)$. For this, we define the notion of contextual uncertainty to be the measure of one of a target focus graph based on its evidence links.

**Definition 5.11.** *Contextual Uncertainty — We define contextual uncertainty of a target focus graph $G_{D_t}(e_t)$ compared to a reference focus graph $G_{D_r}(e_r)$, with a link existing between $e_t$ and $e_r$ in the contextual linkset $LS(D_t, D_r)$, as the sum of products of object(syntactic) and predicate(semantic) similarity scores of the statements linked by each $l \in E(G_{D_t}(e_t), G_{D_r}(e_r))$, on the number of evidence links in $E(G_{D_t}(e_t), G_{D_r}(e_r))$.*

$$U(G_{D_t}(e_t) \mid \langle e_t, p_l, e_r \rangle) = \frac{\sum_{<t_1,l,t_2> \in E(G_{D_t}(e_t), G_{D_r}(e_r))} sym_o(t_1, t_2) \times sym_p(t_1, t_2)}{|E(G_{D_t}(e_t), G_{D_r}(e_r))|} \quad (5.7)$$

This value is not meant to reflect the resemblance of one focus graph to another, but to give an idea of the reliability of data in the focus graph comparing to the reference dataset, and to the overall links between the two graphs.

If one wants to evaluate the uncertainty value for each statement inside a focus graph, we recall the concept of *Meta Mapping Modes* [64] where selective inheritance and the involving of uncertainty calculi issue an aggregated value from the local object and predicate similarities of the statement, and the contextual similarity of the focus graph including it. One may choose to keep the object similarity as a particular uncertainty value for the statement, and when asked to provide its uncertainty, the latter is provided based on the mode: the value of the statement itself, the value of its focus graph, the most specific value, or a combination of the previous values using uncertainty calculi.

### 5.3.5 Offline Uncertainty Extrapolation

When new elements are presented to the contextual linkset $LS(D_t, D_r)$ or as new data are inserted and its uncertainty need to be assessed, one may want to reuse the previously evaluated uncertainty values measured on the semantic level without the need to rerun the whole evaluation process on linked focus graphs. The elements of the vector $\hat{R}(p_1, p_2)$ can be easily updated to match with the new configuration of linked focus graphs. When adding a new link $l$ between two resources $e_t \in I_{D_t}, e_r \in I_{D_r}$ to $LS(D_t, D_r)$, the values of the components $\hat{R}_i(p_1, p_2), i \in [0, 3]$ of $\hat{R}(p_1, p_2)$ are updated as follows:

$$\hat{R}_i(p_1, p_2) = \frac{\hat{R}_i(p_1, p_2) * (|LS(D_t, D_r)| - 1) + R_i(G_{D_t}(e_t), G_{D_r}(e_r), p_1, p_2)}{|LS(D_t, D_r)|} \tag{5.8}$$

Another way to update such values is to treat the predicate similarity of a predicate pair $(p_1, p_2)$ in the current configuration of $D_t$ and $D_r$ as a prior probability, to be used in a Bayesian updating to calculate a posterior representing the probability that the predicates $p_1, p_2$ are similar after the update. The likelihood that the two predicates are similar is tricky to calculate. We can estimate it by using the number of links from the contextual linkset that supported this hypothesis. For example, if in 500 analyzed links we found that only 5 links supported the similarity of $p_1, p_2$, we assume that the likelihood is 1%. Another proposition is to check for the likelihood of the hypothesis of similarity between $p_1, p_2$, by checking a global schema providing information about the equivalence/non-equivalence of predicates, and calculating a contingency table to find the probability that two random predicates are linked with explicit equivalence statements.

## 5.4 Experiment and Evaluation

We evaluate a dataset with 714 artists from *MusicBrainz* against their linked information from the English chapter of *DBpedia*. The used dataset including focus graphs

and contextual linkset is available online [8].

### 5.4.1 Archer: a tool for Uncertainty Analysis and Extraction based on Extrenal Graph Interlinks

To validate our approach, we developed *Archer*,[9] a tool for analyzing and annotating link data with uncertainty values. *Archer* uses the proposed approach to extract the object and predicate similarity with respect to the links between focus graphs. The tool allows the user to query for identity links, extract focus graphs from both the target and the reference datasets, and evaluate the uncertainty of each focus graph in the target dataset. It further allows analyzing and visualizing pairs of linked focus graphs individually as well as the different indicators for the overall analysis.

In the next parts, we give a quick guide from querying for target and reference focus graph, to annotating target data with uncertainty information. We offer an extended guide for each section as an interactive experience within the tool.

#### 5.4.1.1 Querying for Graph Interlinks

The first step is to provide graph interlinks between a target dataset and the reference one. Links are in the form:

```
<IRI_target_resource> <linking_predicate> <IRI_reference_resource>
```

For instance, to assess the uncertainty of information about the resource `dbp:Paris` in *DBpedia* while taking the resource `wiki:Q90` in *Wikidata* as a reference, we should provide the identity link:

```
<http://dbpedia.org/resource/Paris> owl:sameAs <http://www.wikidata.org/entity/Q90>
```

The query interface of Archer, as shown in figure 5.2 allow to execute predefined queries just by precising a set of parameters, or provide a list of identity links to use

[8] see https://github.com/djebR/archer/tree/master/dataset
[9] http://github.com/djebr/archer

as a contextual linkset. The interface allows exploring previous queries, by indicating the target and reference sources queried for what class individuals. It indicates the number of found entries as well.

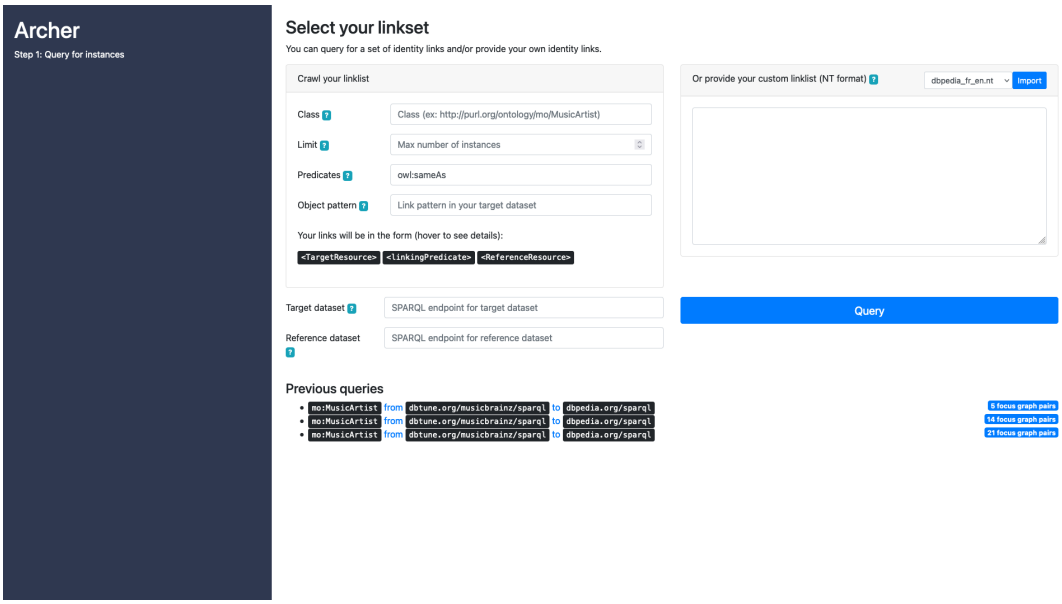



FIGURE 5.2: Querying for Contextual Linksets in Archer

As explained above, identity links can be provided in two ways:

- by selecting a target and a reference SPARQL endpoints, the desired linking predicates (such as `owl:sameAs`, `skos:exactMatch`, etc). Archer will fetch a maximum number of links defined by the user.
- by entering the links manually and directly into the field reserved for custom links or by importing links from a file. Links have to be in N3 format to be parsed correctly.

#### 5.4.1.2 Querying for Focus Graphs

Once the resources from the identity links are parsed, Archer queries for their focus graphs in both the target and reference sources. Figure 5.3 illustrates the view of individuals, and the number of triples on both focus graphs linked to them.

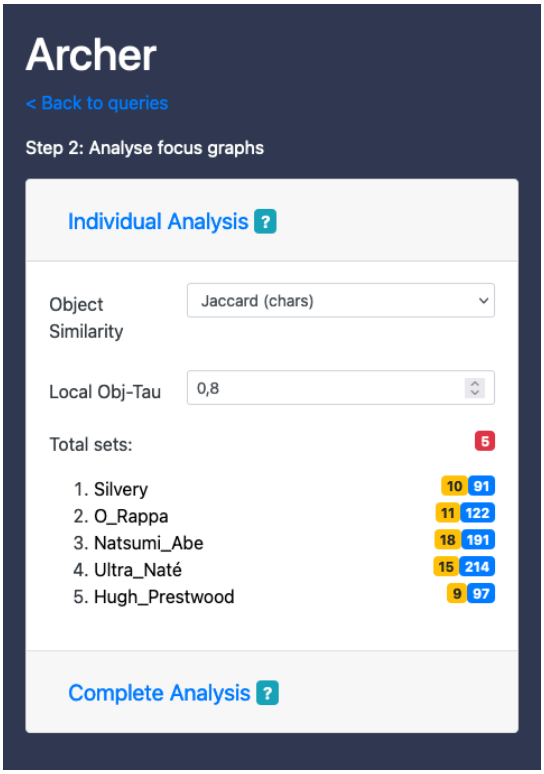



FIGURE 5.3: Entities parsed from the Contextual Linksets

The tools offers two options to analyse linked focus graphs, the first (figure 5.3) is an individual analysis of focus graph pairs by clicking on the label of the desired focused resource. The second (figure 5.4) is a complete analysis to give the overview of the previous pipeline.

The individual analysis provide an overview of the previous indicators $I_1, .., I_5$ with a JSON representation of the same information for debug purposes. The individual (local) analysis is done by choosing a method for the object similarity. The previous indicators can be read as a heat map (figure 5.5) showing statistics about the object similarity and linking it to the predicate-pairs for the next step.

The complete analysis (Figure 5.6) is performed by selecting the method a user judges to be suitable after some trials with individual focus graphs. Once done, the user may select specific threshold to refine the results. The complete analysis is done once and the results are saved to be processed and visualized offline. The figure shows two different types of charts: a three-dimensional heatmap to see the effect of the number of the thresholds on the results, and a box map to offer an idea about

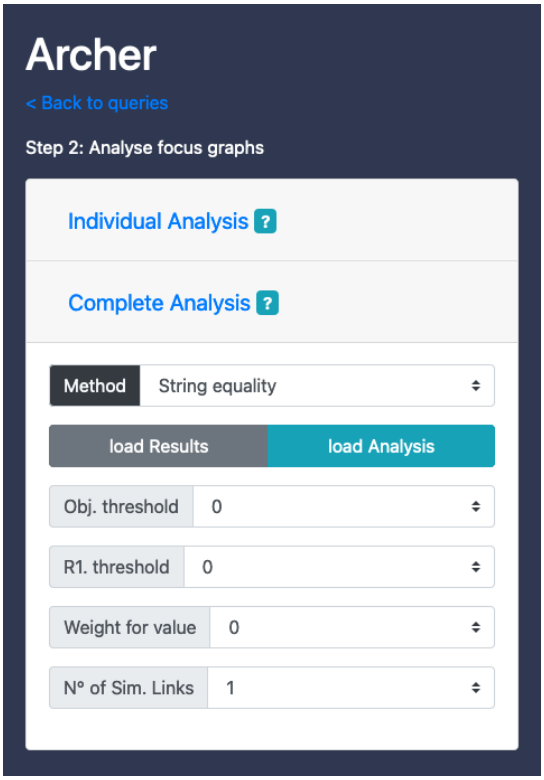



FIGURE 5.4: The parameters for the Complete analysis

the distribution of the local ratios.

#### 5.4.1.3 Annotating Data with Uncertainty

The tool exports the imported data with uncertainty annotations. Exporting the data requires the user to choose a configuration from the previous ones, with what they found convenient and more representative of the reality of their use case. The two tables (Figure 5.7) provide an idea of the uncertainties that triples will be annotated with, and the ones calculated for the future uncertainty extrapolation.

### 5.4.2 Experiment and Results

For the experiment, we chose both a Jaccard distance and a string equality measures as a *valMatch()* function. Plots in figure 5.8 show the effect of the size of the contextual linkset $|LS(D_t, D_r)|$ on the overall count of evidence links $\sum I_1(G_{D_t}(e_t), G_{D_r}(e_r))$ and the number of distinct predicate-pair $|\bigcup I_2(G_{D_t}(e_t), G_{D_r}(e_r))|$. We fixed $\tau_{sem} = 0$ to

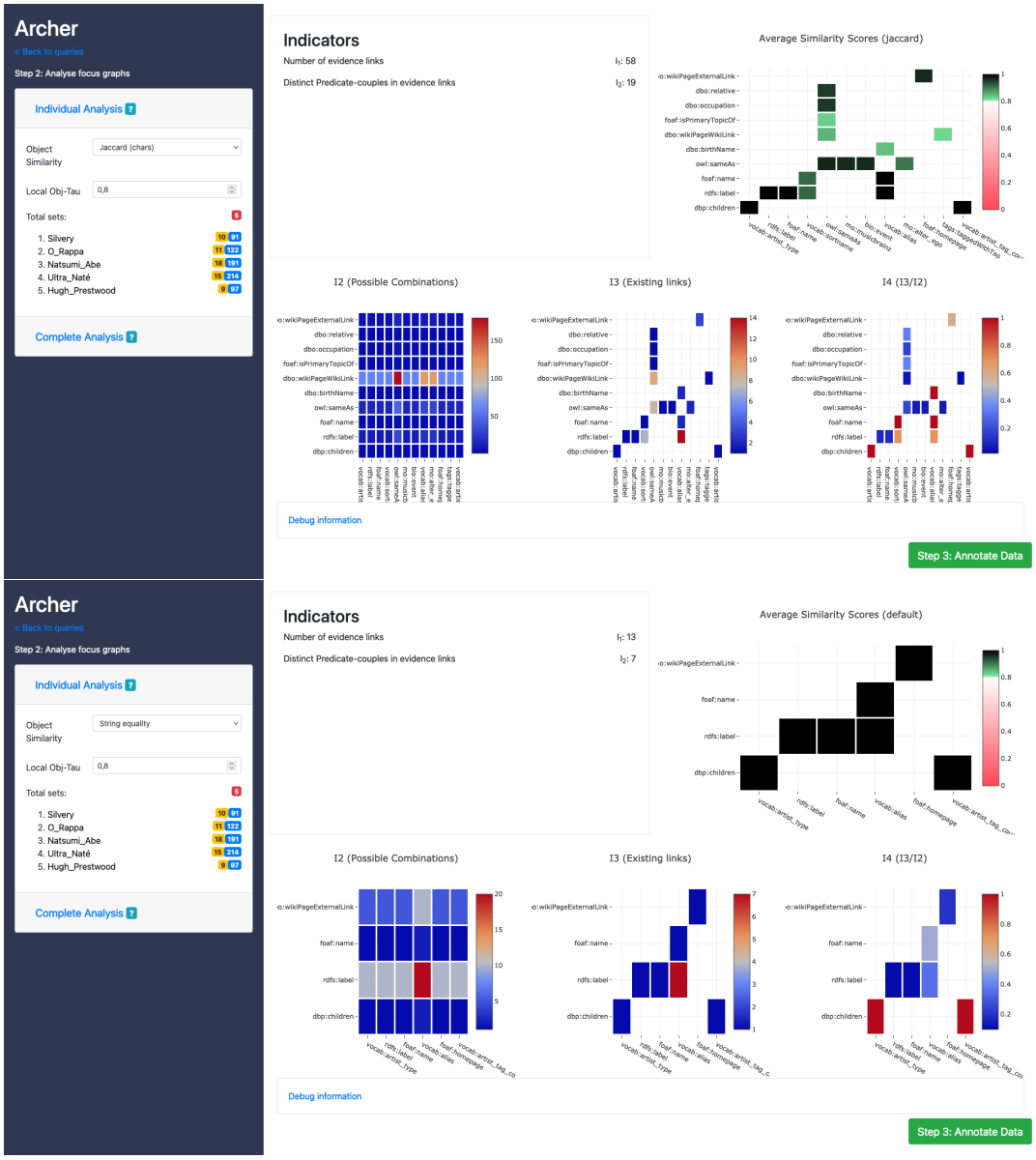



FIGURE 5.5: Individual Analysis of one pair of focus graphs with two different methods (String matching and Jaccard distance for characters)

see the effect of $\tau_{obj}$ on the evidence link count specifically. We then changed the value to $\tau_{sem} = 0.5$ to visualise the effect specifically on the distinct count of predicate-pairs that are considered as similar in the context of the application. For both experiments, we chose $\omega_{val} = 1$ to see the effect of each object similarity function as well.

We notice that:

- in all of $(a_1, a_2, a_3, a_4)$, the number of evidence links is proportional to the number

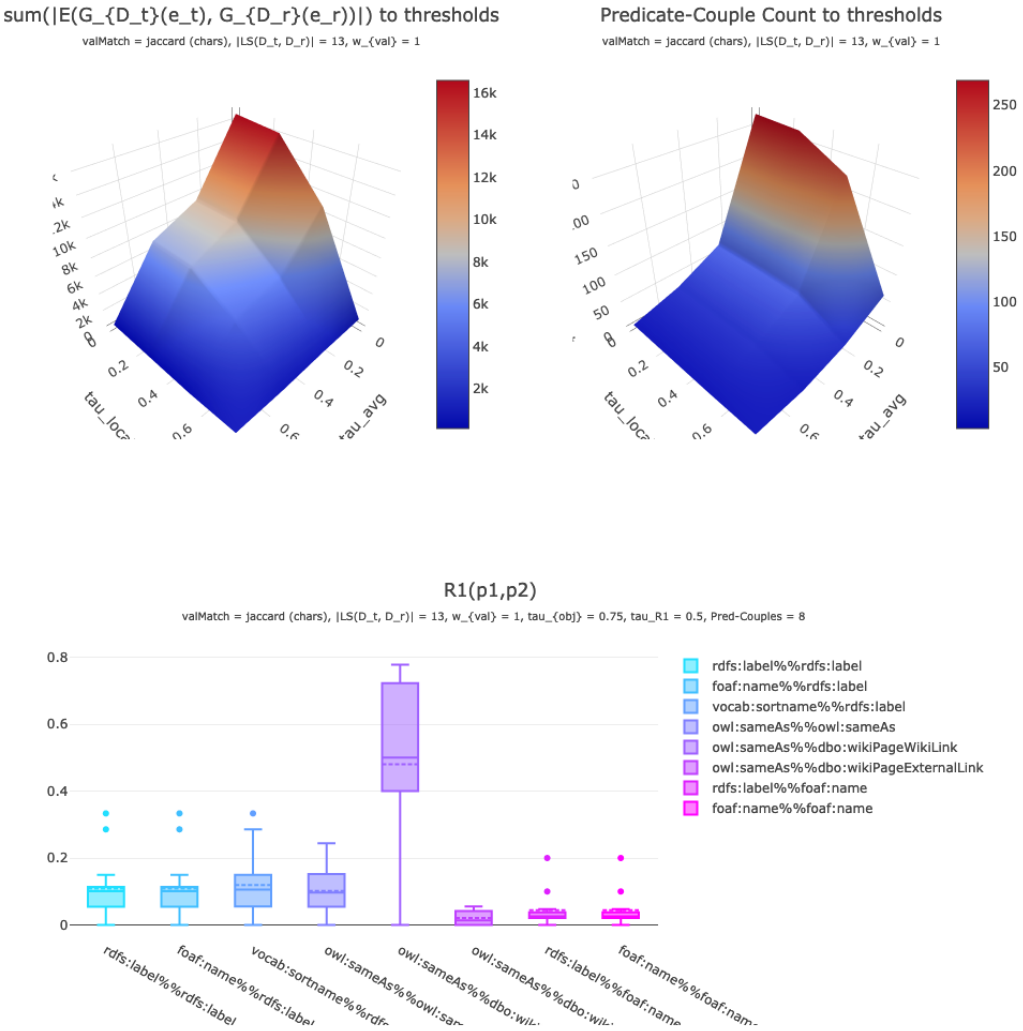



FIGURE 5.6: Visualization of the results of complete analysis.

of analyzed focus graphs. This points to the fact that focus graphs in both sides share a certain structure allowing to maintain a relatively fixed ratio of evidence links per pair of focus graphs. Moreover, in ($a_1$) compared to the absence of a threshold, more than half the evidence links were ignored in ($\tau_{obj} = 0.25$) indicating that those evidence links were of bad quality. As for the string equality, the local threshold is not needed as the indicators $R_1$ and $R_3$ will be the same (for each discovered evidence link, the quality is 1 at $\omega_{val} = 1$), so no evidence links will be dropped.

- as seen in ($b_1$,$b_3$), the effect of $\tau_{obj}$ on the number of evidence links is also predictable. The threshold will only allow links with better quality to be part of the overall evaluation.

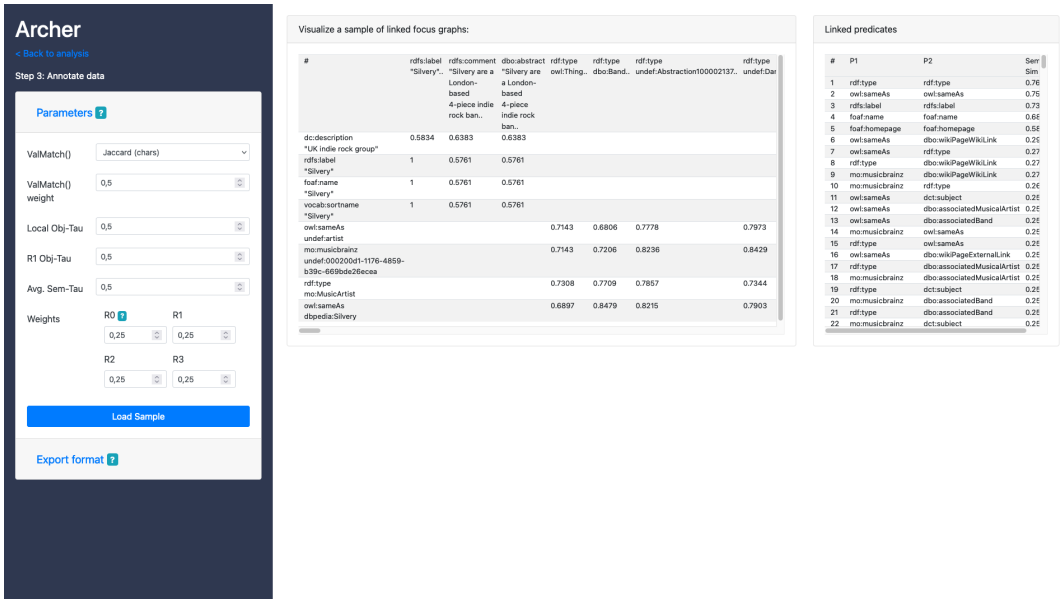



FIGURE 5.7: A sample with the chosen configuration to annotate and export data with Archer

- the number of predicate-pairs increases with the number of linked focus graphs. This is due to discovering predicate-pairs that did not have any evidence links in the first analyzed focus graphs. It is notable that for both object similarity methods, the number converges after analyzing more than 400 pairs of focus graphs. Furthermore, the effect of $\tau_{obj}$ can be observed confirming that some predicate-pairs were dropped as they presented only bad quality links.

- when increasing $\tau_{sem}$, the plots in $(a_2)$ move closer to each other and converging towards $(a_4)$ as it represents strict equality, resulting as well in similar shapes for $(b_2)$ and $(b_4)$. The fluctuation is due to the fact that the overall quality of some predicate-pairs evidence links might drops when considering new pairs of focus graphs that do not support the hypothesis. However, the plot remains constant proving that on the overall analysis, five predicate-pairs can be considered as best candidates to support the graph interlink.

- the difference between the number of predicate-pairs in $(b_3)$ and $(b_4)$ is remarkable. Comparing to 28 predicate-pairs in the first with 6600 evidence links, the second has only 5 predicate-pairs with almost 6000 evidence links. This further provides proof that most of the discovered links were not of general use (not

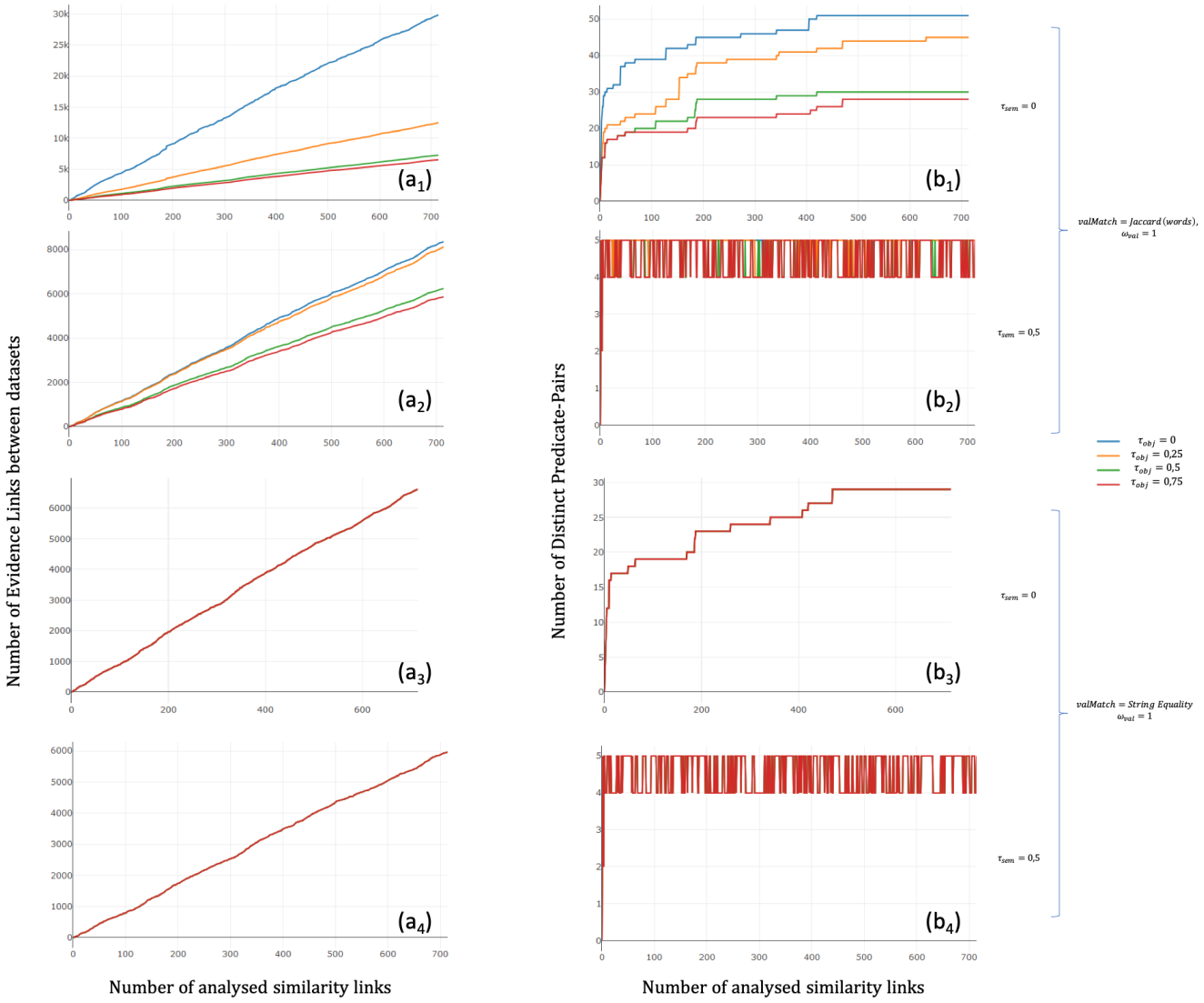



FIGURE 5.8: Results for analysing 714 pairs of linked focus graphs: $(a_i)$ total number of discovered evidence links $(b_i)$ Number of distinct discovered predicate-pairs.

common information between focus graph pairs).

### 5.4.3 Annotating Uncertain Data Using *mUnc*

After the analysis and selecting what evidence links are considered (by selecting the thresholds $\tau_{obj}, \tau_{sem}$ and what predicate pairs are similar to each other, *Archer* enables annotating the statements related to evidence links, the predicate pairs, and the focus graphs with uncertainty information, encoded in RDF using the Meta-Uncertainty ontology *mUnc* [64]. The listing 5.1 is an example of annotating one focus graph with uncertainty values, where the uncertainty approach and the uncertainty value annotating the focus graph are defined using *mUnc*, while *PROV*[69] describes *Archer* as an activity that was used to generate that value.

```
:StatisticalApproach a munc:UncertaintyApproach;
   munc:hasUncertaintyFeatures :contextualUncertainty.

:contextualUncertainty a munc:uncertaintyFeature;
   rdfs:range xsd:decimal.

:focusGraph1 munc:hasMeta [a munc:Uncertainty;
   munc:hasUncertaintyApproach :StatisticalApproach;
   :contextualUncertainty 0.8^^xsd:decimal;
   prov:wasGeneratedBy :Archer101
   ].

:Archer101 a prov:Activity;
   prov:used :contextualLinkset101.
```

LISTING 5.1: Example of annotating a focus graph with uncertainty approach and value- with a reference to the process and the contextual linkset used to generate the value

## 5.5 Discussing Uncertainty Extraction

An argument about statistical extraction of semantics would be the fact that a target dataset can be completely wrong, or somehow unrelated to the reference dataset like having the same information but in a different language. In both cases, this does not affect the semantic analysis of evidence links. For the first case, no links will be discovered and this will raise a flag about the current configuration itself (one is wrong about everything related to a certain subject, or the references were chosen incorrectly). For the second case, the similarity links will not be translated as well, and triggering the intuition of completeness between the two graphs (and not that of negation).

Analyzing the similarity patterns based on graph interlinks may be a good first base to evaluate trustworthiness and inclusion between data sources. If one source is likely to quote the other or has always supported it (in a non-contradicted, complementary way) then it can be thought of as backed and trustworthy. This approach works best if one has already a clustered dataset by structure, and the system is used to see the reliability of its information according to known sources.

Further investigations are scheduled to explore the use of other clustering methods, or customized focus graphs and see the possibility to transform the existing information about focus graphs using graph embedding. Finally, user queries should be one of the main triggers of uncertainty measurement, interlink creation, and evaluation. Presenting a publishable measurement with enabled updates is one of the main motives to ease up the process. It is also important to explore other information about predicates by studying the polarity of evidence links, built upon the previous semantic indicators in the same spirit of ontology-matching but involving a contextual view.

## 5.6 Conclusion: Managing Extracted Uncertainty

For data sources to ensure providing reliable linked data, they need to indicate information about the uncertainty of their data based on the views of their consumers. In this chapter, we introduce a novel approach to evaluate the uncertainty of data in an RDF dataset based on its links with other datasets. We have proposed to evaluate uncertainty in linked data sources by providing graph interlinks. Our approach is based on both object and predicate similarity and operates on different levels to evaluate task-specific uncertainty measurements for the data source of interest. We translate each interlink into a set of links referring to the position of a target dataset from a reference dataset, based on both object and predicate similarities. The results of our experiments show that graph interlinks can be supported with a set of evidence links, depending on the use-case and the user's choice of quality parameters. Using our tool enables us to assess the quality of a dataset regarding a certain task, and annotate its data accordingly while producing reusable and publishable uncertainty measurements.

This manuscript's next and last part gives our final thoughts about the realized work and our research perspectives. Following, we discuss the potential of sharing and propagating the results of experiments that each source performed. We talk about the idea of having a collaborative uncertain linked data universe.

# Part III

# Conclusion

# Chapter 6

# Conclusion

The challenge of understanding all aspects of uncertainty seems somehow limiting its purpose. The world as we know it is meant to be incomplete and some parts of it to be ignored. Data can be erroneous in the Web Utopia we aspire for, but the error should be detected. People can express different subjective measurements about one fact, and their beliefs can be compared and composed into knowledge. This vision of a transparent and reliable Web can be realized with the aid of extra resources, allowing the curating of the existing data and the secure funneling and linking of new ones to a consensual Web.

Uncertainty is not a bad thing. However, exaggeration or ignorance can be bad practices. We ask data publishers to be transparent with their data and processes to serve both the openness of the Web and offer the possibility for future robust decentralized curation processes to be achieved.

This manuscript cannot cover all the different questions one may ask about uncertainty. The purpose was to shed some light on the area we explored and contributed to, and most importantly, to ask further questions. As wide as it is, uncertainty always leaves room for improvement.

Our ideas were built on the assumption of local consistency in contexts. This means that a unique occurrence of the same sentence with a specific set of uncertainty metadata. Although this was not discussed in detail, one way this can be achieved

on the existing and upcoming data is through validation. The use of SHACL, for instance, helps validate the shape of each context to ensure it contains no redundant or contradictory information. The rules can be further set to translate any uncertainty to compare the different occurrences of the same sentence.

To the best of our knowledge, no previous work offered a framework to handle uncertainty using a standard generic vocabulary to represent custom uncertainty approaches. Since the URW3-XG was shut down in 2008, we perceived a decrease in the works treating uncertainty as a generic dimension. The group insisted on the investigation and the proposition of generic support of uncertainty. We offered a way for uncertainty to be representable, publishable, detectable, and reusable using the Semantic Web standards. In addition, the use of contexts, readings, indications, and calculi allows more selectivity towards the metadata presented to the user and allows inferring new uncertainty information. The uncertainty ontology can extend the work by Cabrio et al. [93] treating the composition of information from the different language chapters of DBpedia. This can be done by enriching the proposed fuzzy labeling algorithms with definitions of uncertainty theories suitable to the data. A set of other applications such as fake news detection, argumentation-based systems, and even community-based data population for sources such as DBpedia can use the uncertainty ontology to enrich future content with uncertainty information.

The idea we see as following to the representation and extraction of uncertainty is the propagation of such uncertainty values. In this section, We tackle the questions of sharing, receiving, updating uncertain information. We discuss the best practices on approaching information on the Web and provide some ideas about positioning a piece of information we found on the Web, accordingly to others in the same or different contexts.

## 6.1 Generalizing Uncertainty Extraction and Propagation through the LOD Cloud

The natural course of action to propagate uncertainty is to move it up the different contexts that shape the view of reality and recursively select and compose its values according to the conditions set by the user. Propagating uncertainty between sources requires analyzing their dependency relationship, leading to the study of the macrostructure of the Web of Linked Data (as any other directed graph [94]) and discussing what sources affect/get affected more by others. The in/out links from data sources are represented interactively on the LOD-cloud website.[1] The linking information provided is the number of triples linking the pair of sources with an in-/out link, and it is a good start to build an initial dependency graph. Therefore, we can start implementing uncertainty for the effective central sources (sources mostly used as references by other sources). This should be done by checking the existence of a link between two sources and analyzing the relationship between the data offered by both sources. The LOD cloud provides the number of triples linking one dataset to another but does not specify the relationship of such triples, and –unlike Archer– it does not provide an idea about non-connected triples. Uncertainty cannot be propagated without understanding where it is coming from and having a reference to compare the two contexts in which uncertainty was generated into. For that, all sources must evaluate themselves and position themselves according to all other sources. This evaluation may concern all or part of the data the source is presenting. This must be eventually mentioned to data consumers to promote more transparency about the process.

Sharing and propagating uncertainty requires packaging each level with its uncertainty and unpacking the levels in the same way. For instance, a sentence-level uncertainty must be compared and composed with a similar one from another source during propagation. This process is different from the composition of uncertainty

[1] The links are visible on hover. From one source, red links represent incoming links, and green links are outgoing ones. https://lod-cloud.net/clouds/lod-cloud.svg

from different levels inside the same source (like meta-mapping modes). We point here to the differences between (*i*) serving an answer to a user and (*ii*) comparing two sources. The first one requires a unified view of uncertainty from the different encapsulation levels inside every single source. For instance, we can use the meta-mapping mode to compose the uncertainty of a sentence stating that *Stefano Tacconi is Italian*, with the uncertainty of the section of the website it is mentioned in (e.g., the football section) and then with the uncertainty of the website as a whole. In the end, the user gets a composed uncertainty from the different levels, reflecting the sentence's contextualization. The second one requires understanding the different components composing uncertainty to be transparent on the propagation as well. Comparing two sources can be done while considering the contextualization of sentences and without it. Moreover, uncertainty from the different levels can be linked to more specific provenance information.

## 6.2 A Consensual Uncertain Semantic Web Universe

The question is about how to get everyone to comply with the presented approach. Complying here is not about fully embracing the approach, but at least coping with the fact it exists and some sources adopt it. This question can be answered with a quick "no" or an ambitious "maybe." The quick negative answer implies that the Web cannot handle this additional layer for many reasons:

- Data providers perceive uncertainty as an additional burden to handle.
- Humans do not care about publishing correct or accurate metadata [6].
- Users get confused about the type of uncertainty to use in each use case.
- Some sources serve but do not take feedback.

The level-based architecture we proposed and the use of meta-mapping modes was meant for dealing with scalability issues. Sharing metadata about data is not necessarily increasing the size exponentially. Using uncertainty reading and contextualization, we can save much space and reduce the time of treatment. Dealing with scattered, unorganized metadata can be less challenging relative to scalability. We can overcome such an issue by imposing strict guidelines for data publishing or reuse on the Web. Moreover, it is easier to represent uncertainty when we abide by a published unique vocabulary, or a set of interlinked vocabulary serving the same purpose, with a straightforward mapping from one to another and a more precise explanation for the processes, indications, readings, and calculi of uncertainty. The open nature of the Semantic Web encourages reusability: one can reuse previously defined uncertainty approaches to describe the uncertainty of their datasets and build upon the existing ones. They can decide to position their data according to the ones proposed by their peers, therefore decide for its reliability accordingly. On the Semantic Web, everyone complies with the standards to publish their data. The community can propose "reliability standards," a set of best practices to publish reliable, portable linked data on the Semantic Web. These standards can contribute to the 5-star rating of Linked data publishing on the Semantic Web. We can have something like:

- Published while respecting the Semantic Web and Linked Data publishing standards.
- Data provenance made available.
- Uncertainty data and schema made available.
- Uncertainty Calculi and reading made available.
- Data reproducibility processes made available (full provenance, access to data generation/extraction code)

The last issue is where some sources opt-in for transparency, but others do not. One source can serve data but does not accept receiving suggestions or updates from the external world. Supporting uncertainty starts by filtering, contextualizing, and

profiling every piece of information that hops from the Web into the Semantic Web or is integrated directly into the latter.

The ambitious "maybe" is based on two steps: Homogenization and curation. With *mUnc*, we can enforce transparency on individual sources, especially those linked to official or governmental institutions. The more sources are adopting uncertainty, the better it is for the community. However, homogenization may take a long time, and data providers may not comply with other regulations easily. During that time, curation may be a second resort, where a mediator agent gathers the necessary data for a specific use case from different sources. The mediator performs an uncertainty extraction following the same principles as *Archer* and compares/homogenizes the acquired data for those without explicit uncertainty. An example for homogenization using truth values is offered in [93] where the authors proposed to curate data from the different language chapters of DBpedia using a majority consensus. Afterward, the curated data can be cached or saved on a new copy, or the changes can be publicly communicated to the different data sources used in this operation. The idea of data curation is not meant to create a unique point of view but a set of backed ones. Anyone can say anything on the Web, as long as they can prove it rationally and others approve of their method. Hence, sources accept that there can be multiple opinions on the Web, and users can see them transparently, understand how they relate to each other and judge them according to their search context. This work is not meant to enforce a *unique* view on the Web, but to add a protective shield to the ambivalence it thrives on, making it safer to burst the bubbles of local contexts.

## 6.3 Data transparency: say more and be honest about data

Setting up filters on the doors of the Semantic Web is not an easy task. Any process requires assessing the quality of data before using it with or as a reference. This transparency can be expressed manually, where data providers check every piece of

their data and state its provenance. However, in the case of the Web, lots of data are already out there. AI comes in handy for such a problem, where systems can check and predict missing provenance or simulate data generation, which can be for the service of crypto, proof, and trust layers on the Semantic Web Stack. To deal with unifying logic, we must consider sharing data and the processes that lead to its creation and those manipulating it.

Another transparency issue is the uncertainty of uncertainty-related processes. We elaborate the debate around the fact that uncertainty data can be uncertain on its own. This is another reason for us to be transparent about the processes of uncertainty selection and composition.

In the future, we may face other issues as the uncertain semantics of *silent* triples (i.e., triples without explicit annotation or indication of uncertainty). The idea with these triples is to consider both their default state as an asserted triple in a data source and the assumption of their existence in an open world. By default, silent triples are considered certain for their mere existence, and the absence of uncertainty tells nothing more about their state. Any user can change this setting according to what they think is convenient for their use case. One can decide not to consider silent triples or to treat them differently. This issue opens the door to a debate about the democracy of information on the Semantic Web and whether to consider all triples equal or different.

## 6.4 From Beliefs to Knowledge

According to the *Theory of Knowledge* proposed by *Plato*, a *belief* is the subjective requirement for knowledge, whilst *Knowledge* is defined as a *justified true belief*. In our case, the criteria set by the user to select resulting uncertain data are the justifications we are looking for to consider the beliefs as knowledge. We see uncertain data as a support for a subjective judgment. Therefore, we can say that uncertain data assert the truth of beliefs depending on their uncertainty qualification/quantification. Once

the users select to believe uncertain data because they fit their reality and use case, beliefs then are considered knowledge in that use case. For example, we consider the sentence stating that Stefano Tacconi's height is 192 cm, annotated with an uncertainty value of 0.3. This sentence asserts the truth of this information to the extent of its uncertainty value but cannot be considered knowledge by the users unless they decide to accept all sentences with an uncertainty annotation equal to or greater than 0.3. Nevertheless, the sentence is considered knowledge only in that context.

The previous example reflects on any sentence, assuming each data source possesses a coherent context. Hence, two sentences can still be selected if their uncertainty values are above the threshold while knowing they will not be of a contradiction. However, if we want to widen our view, this should not be the only criterion to use. The Web, in reality, allows the existence of different sentences asserting different beliefs on the same atomic and functional information, such as birthdays or height values. Having a consensus requires the previous selection criteria to apply globally to all sources. For example, consensus by the majority is one form of the conditions we apply to get beliefs to be knowledge. It is an implied selection criterion that allows the information believed by the majority to be the one selected as knowledge. Our work on uncertainty representation and extraction provides another criterion. Uncertainty values can then lead and achieve the consensus while easing the communication of data deficiencies.

# Chapter 7

# Perspectives and Future Works

This work on uncertainty is a precious opportunity to reflect on some basic ideas about epistemology, knowledge, and philosophy. The journey through the different layers of the Semantic Web and the different related domains was of joy. This manuscript covers the works we contributed to the previous, but both time and words limit the expression of some ideas we want to pursue further. We revisit the principles required by Klir to see that the representation and calculi of uncertainty enabled by *mUnc*, uncertainty extraction enabled by *Archer*, the contextualization and the translation mechanisms of uncertainty cover the different points and allow for the implementation of the different principles. It is up to the users to choose the best theories to fit with their use-cases. But this does not conclude the work in this area and will open many doors and possibilities for future improvements.

We aim to dive deeper with uncertainty operators. The plan is to issue a set of recommendations about linking every component in the query answer with the appropriate uncertainty value, with a detailed description of the processes, the considered contexts and levels, and the various user-selected conditions that contributed to the generation of such data. The visualization of uncertainty explanations can be done using procedural/flow diagrams (e.g., *Sankey diagrams*[1]). This work should finally be presented as a generic uncertainty querying and reasoning engine that would support

[1] A diagram illustrating the flow from one set of values to another. It can help to show what elements intervened in the generation of an uncertainty resource. https://developers.google.com/chart/interactive/docs/gallery/sankey

any formalized uncertainty approach. We worked on a tweaked version of Corese to integrate some uncertainty support, which should be our starting reference.

For uncertainty extraction, it would be interesting to focus on two main axes. The first axis concerns learning the most suitable structure for a focus graph. That should generalize our approach to consider a set of reference resources to include more parameters such as scores from ontology matchers. The second is exploring the possibility of using embeddings with the focus resources, especially for the internal analysis of the similarity between the focused resources of the same set.

We aim to make our vocabularies, code, and contribution more accessible and open for sources to use and adapt. This step includes offering solutions to ease the adoption of uncertainty and integrating this dimension in the marketed extensible data stores. Linked Data sources can adopt this approach to enrich federated queries with uncertainty information, allow negotiation, and progressively build a consensus-based Linked Data source. Having enough uncertain data would allow us to study the relationship between use cases and uncertainty theories that are used. We think of specifying the levels in which different readings for uncertainty may operate in order to constraint the approaches to their specific level: for instance, if there is an approach that treats incompleteness, it should be limited to annotate collections of triples and not single ones.

We intend on exploring the question of weighted contexts. Some contexts appear to be more relevant than others in single or multiple sources or contain more relevant information to a specific query. Using relevant and reliable contexts will help the answer be backed by higher quality data, reducing uncertainty. At the same time, the idea of nested contexts can be explored. We require to implement a generalized version of the meta-mapping modes to enable a finer-level architecture for uncertainty. We also like to explore context overlapping, allowing the selectivity inside the source between contexts and optimizing the storage.

The generalization of our approach requires testing Archer on a larger scale. This step includes analyzing the macrostructure of the Linked Open Data cloud and testing

for a feasible way to propagate a simple type of uncertainty, such as a statistical probability of the consensus between the different sources. We aim to upgrade Archer to fit in with more data sources.

Some of the questions we would like to tackle concern annotation to extract valuable data for uncertainty evaluation. W3C Credible Web group discussions are stirring the wheel in that direction, but on an upper application level. We aim to draw attention to the use of low-level data and the chance to consider uncertainty as one dimension in judging a source's credibility. Another question we mentioned above was about the democracy of data on the Web and the reading of silent triples. We want to discover the effects of the different assumptions we can make about such data and the optimal use cases for each. We want to see if uncertainty metadata can be networked on an independent architecture or mediated by external parties of the transactions between data sources. A broad ambition is to pursue the idea of a Linked Open Code [95], where code on the Web can be referenced and reused as Linked data (see appendix B). We find it interesting to see the possibilities it can offer for sharing, understanding, and proposing functional code to serve as a platform for distributing uncertainty calculi.

To conclude, embracing uncertainty should be part of the full experience of the *Web of Intelligence*. Various aspects are to consider, and many technicalities must be solved to establish a universal, collaborative, trustworthy universe.

"*He who controls Metadata controls the Web*"[12] recalls the power of metadata. Being transparent about data prevents this power from falling into the wrong hands and guarantees a fair Web experience for all users. Being familiar with imperfections and using them to our advantage is required to overcome our limits.

# Appendix A

# Querying the English and French chapters of DBpedia for Inconsistencies in Football Players

To give the example of inconsistencies between two language chapters of DBpedia, we performed a query on both the French and English chapters to check for the heights of football players. The query could not be executed due to the limitations offered by DBpedia servers.

The options we had in hand were:

- Downloading the dumps of the both chapters
- Overcoming the limits imposed by the servers

Both language chapters ran endpoints based on *Virtuoso*. The default settings allow to fetch 10000 rows with a single query. Nevertheless, SPARQL offer the option to fetch rows starting a specific index, using the keyword `OFFSET`. In the end, we proceeded like follows:

- we queried for the total number of football players satisfying our condition
- we used a `UNION` over the result of a set of limited queries (using `LIMIT` and `OFFSET`).

In our experiment, we counted the number of entities in the class `dbo:SoccerPlayer` having a valid label and a height value in both chapters. We then use the English chapter as reference because it covered more information, and within looked for the players having an identity link with the French chapter. We counted a total of 27516 entities.

Knowing that Virtuoso does not allow for results more than 10000 rows, we opted for the following solution: we divided the service query into three parts using `LIMIT` and `OFFSET`, allowing to recover 10000 rows at once (3 sub-queries in this case). We then used `UNION` to reassemble the results before passing them to the second service query for the French chapter. The condition we chose for an inconsistency in this case is a 2cm of difference between the two height values. The following query was tested on CORESE 4.2, following with a result of 695 players with inconsistent height values.

```
prefix dbo: <http://dbpedia.org/ontology/>.

SELECT ?English ?French ?enHeight ?frHeight WHERE {
{
   SERVICE <http://dbpedia.org/sparql> {
      SELECT distinct ?English ?enHeight ?French WHERE {
         ?English a dbo:SoccerPlayer.
         ?English dbo:height ?enHeight.
         ?English owl:sameAs ?French.
         filter(contains(str(?French), "fr.dbpedia") = true)
      } LIMIT 10000
   }
} UNION {
   SERVICE <http://dbpedia.org/sparql> {
      SELECT distinct ?English ?enHeight ?French WHERE {
```

```
16          ?English a dbo:SoccerPlayer.
17          ?English dbo:height ?enHeight.
18          ?English owl:sameAs ?French.
19          filter(contains(str(?French), "fr.dbpedia") = true)
20       } LIMIT 10000 OFFSET 10000
21    }
22 } UNION {
23    SERVICE <http://dbpedia.org/sparql> {
24       SELECT distinct ?English ?enHeight ?French where {
25          ?English a dbo:SoccerPlayer.
26          ?English dbo:height ?enHeight.
27          ?English owl:sameAs ?French.
28          filter(contains(str(?French), "fr.dbpedia") = true)
29       } LIMIT 10000 OFFSET 20000
30    }
31 }
32    SERVICE <http://fr.dbpedia.org/sparql> {
33       ?French dbo:height ?frHeight.
34    }
35    FILTER(IF(abs(?enHeight - ?frHeight) > 0.02, 1,0) = 1)
36 }
```

Listing A.1: Query for inconsistencies between the English and the French Chapters of DBpedia

Another solution to overcome the limitations of the server is the use of *SPARQL functions* proposed in *LDScript*, where the sub-queries can be executed in a for loop before passing them to the next query. The function may improve this query by checking for the unit of measurement, and normalize any odd values (e.g., if some values are in meters or inches).

The previous query is just an example, and reflects just a small part of the inconsistencies here. It suggests the existence of a height value, whether in reality, some players do not have an attributed height value as well as other data points.

# Appendix B

# A vision Towards a Linked Open Code: shipping methods with data

The vision in this appendix is part of a work published in the Extended Semantic Web Conference ESWC 2020 [95]. The work aimed to offer a vision towards a Semantic Web with a possibility of publishing, linking, and reusing parts of code as we know it, aiming to apply that in the publishing of Uncertainty Calculi.

The guidelines provided by the Semantic Web community allow to (*i*) homogeneously represent, (*ii*) uniquely identify, and (*iii*) uniformly reference any piece of information. However, the same standards do not allow defining and referencing the methods to exploit it: functions, procedures, algorithms, and code are generally left out of this interconnected world.

For the sake of transparency, the processes to generate and manipulate data need to be open and AI-ready. Understanding the provenance of uncertainty metadata is required to manipulate it. For that, we are required to understand the processes that lead to the generation of such data: whether it is an entry, an automatic generation, or a generation based on other uncertainty metadata. In all cases, access to the code that leads to these operations is crucial.

The existing code repositories on the Web are not ready for Semantic Web. Many challenges arise from this new definition, starting with the fact that the existing code repositories do not provide a "function-based" view. As a consequence, we should figure out how to turn those into referenceable, reusable resources. The following challenges presented in figure B.1 are to be addressed.

## B.1 Referencing Functions

Function structure and signature in code make it easily recognizable. The signatures usually contain information such as the function's name and its typed arguments (*cf.* figure B.2). Such information can be represented as linked data while attributing a unique identifier for function definitions.

The idea is to allow Linked Data providers to publish, following the Semantic Web principles, the code of functions, and their metadata. Furthermore, one may include an additional level of granularity to existing IRIs referencing code entities (repositories, folders, files, fragments), helping to reference functions and keep track of their provenance. For example, a code file archived on *Software Heritage* with the IRI `swh:codeFile` helps addressing the function `fn` using the IRI `swh:codeFile_fn_1` (instead of referencing fragments of code with no defined semantics).

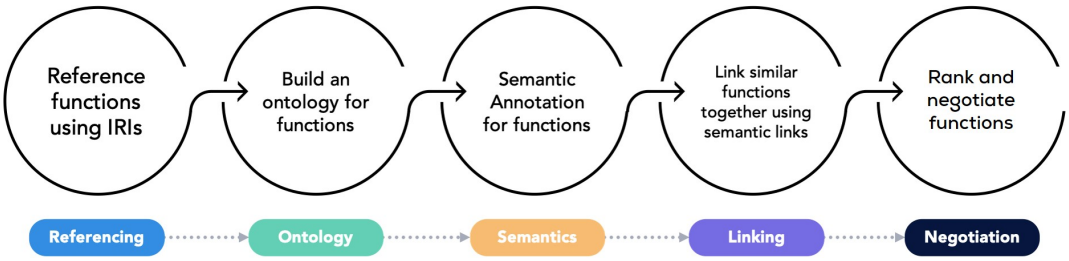



FIGURE B.1: Challenges to achieve a first working prototype of Linked Open Code

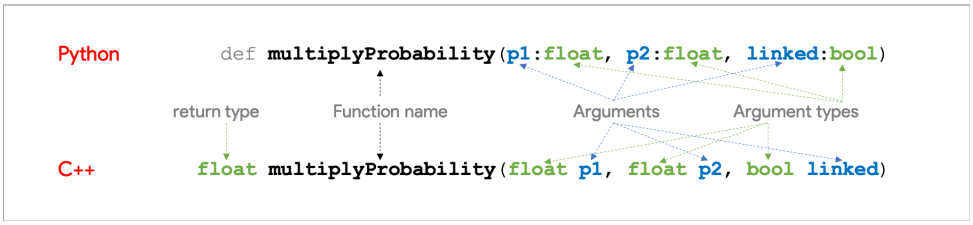



FIGURE B.2: Comparison of metadata provided function signature in Python and C++

## B.2 An Ontology for Functions

A crucial step to bring functions to the Semantic Web is defining an ontology to represent them. Such ontology must describe four aspects:

1. Versioning: the version of the function, programming language, provenance.
2. Relational: relations between functions (inclusion, dependencies).
3. Technical: code, arguments, typing.
4. Licensing: although all open source licenses imply free-use and sharing of code [1], some may impose restrictions on the reuse (*e.g.* crediting the original author), hence this information needs to be provided to the user.

## B.3 Annotating Functions Semantically

During this step, the defined functions are mapped, each with their signature and feature metadata. An Abstract Syntax Tree (AST) analysis is applied to identify the components constituting the function's signature (name, parameters, ...) that will then be used as values for the properties defined in the ontology. As a result, the user will query the knowledge base to retrieve the function matching the given constraints. In parallel, a feature identification process is executed to identify the functionalities implemented by each function and annotate them accordingly. The whole process is

[1] https://opensource.org/licenses

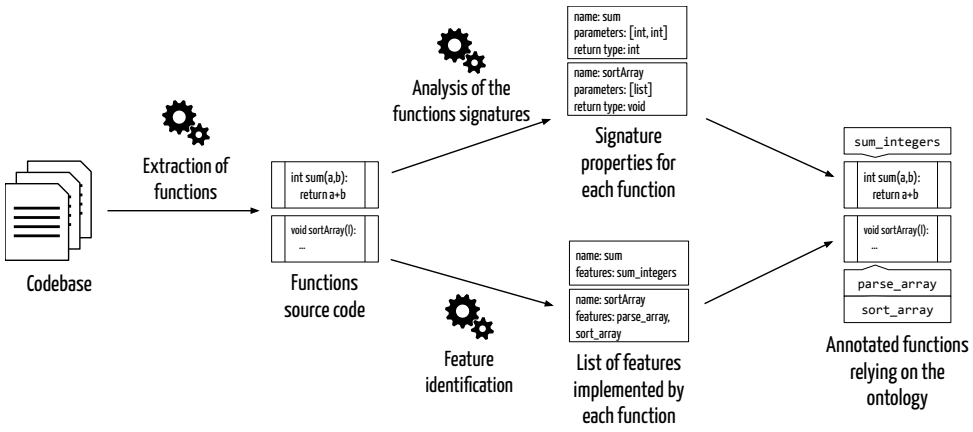



FIGURE B.3: Overview of the process for semantic annotation of functions

depicted as in figure B.3. Multiple techniques for the identification of features have already been proposed [96] and need to be adapted to our context.

## B.4 Linking Functions

After identifying the functions' features, we can use this information to semantically link functions fulfilling similar goals. Indeed, two functions annotated with the same feature can be considered different implementations for the same functionality perceived by the user. Therefore, we can link them with standard predicates such as `owl:sameAs`, `skos:exactMatch`, `skos:closeMatch` or custom predicates offered by other existing ontologies. Alongside semantics, the dependency must be taken into account to link related functions together. Based on this criterion, functions relying on the results provided by other functions (including the function itself in the case of recursive calls) will be semantically connected.

## B.5 Ranking Functions

The same functionality can be implemented in different ways and using different programming languages. To provide the most efficient implementation, we need to rank functions according to several parameters. One example can be the community's

feedback, whereas a repository where usage statistics for functions are being kept for ranking purposes, alongside other information such as the number of times a function was starred, forked, or upvoted by users. It is also possible to signal issues related to security flaws. Performance evaluation can also be used as a ranking criterion. A Semantic Web Engine like *Corese* [2], coded in *Java*, would use functionality implemented in *Java*. However, the same functionality, implemented in *Python*, can deliver better performance for the same tool if used with a *Python* wrapper. This aspect is meant to link code with experience. We can imagine users sharing their execution log, containing hardware specification, operating system, and the language version, amongst other metadata.

## B.6 Negotiating Functions

Users may take advantage of the implemented content negotiation to get suitable function definitions for their use-cases. This is done by using HTTP headers or non-HTTP methods like Query String Arguments (*QSA*). Users negotiate functions that suit their current environment to access and manipulate Linked Data. For instance, a user working with *Corese* may send a request to the function catalog, asking for the *Java* implementation of functions alongside their query for data. Negotiation can rely on the previous step by proposing the best function to the users according to their specifications.

The realization of this vision would be a framework through which the user would use SPARQL to query a catalog of functions for the implementations of needed functionalities meeting architectural and user-defined requirements. The fetched code artifacts can then be composed to build a tailored software system. Concretizing the vision raises other challenges such as the code curators' centrality, scalability, code quality, and distribution. That will need to be addressed when designing the actual solution.

[2] https://github.com/Wimmics/corese

# Bibliography

[1] F. Gandon, "For everything: Tim Berners-Lee, winner of the 2016 Turing award for having invented... the Web," *1024 : Bulletin de la Société Informatique de France*, no. 11, p. 21, Sep. 2017. [Online]. Available: https://hal.inria.fr/hal-01843967.

[2] B. Berendt, F. Gandon, S. Halford, *et al.*, "Web Futures: Inclusive, Intelligent, Sustainable The 2020 Manifesto for Web Science," *Dagstuhl Manifestos*, 2021.

[3] *Digital 2021 Global Overview Report (January 2021) v03*, [Online; accessed 2. Apr. 2021], Jun. 2021. [Online]. Available: https://www.slideshare.net/DataReportal/digital-2021-global-overview-report-january-2021-v03.

[4] F. Gandon, "A Survey of the First 20 Years of Research on Semantic Web and Linked Data," *Revue des Sciences et Technologies de l'Information - Série ISI : Ingénierie des Systèmes d'Information*, Dec. 2018. DOI: 10.3166/ISI.23.3-4.11-56. [Online]. Available: https://hal.inria.fr/hal-01935898.

[5] ——, "Web Science, Artificial Intelligence and Intelligence Augmentation (in Dagstuhl Perspectives Workshop 18262 - 10 Years of Web Science: Closing The Loop)," Dagstuhl, Other, 2019. [Online]. Available: https://hal.inria.fr/hal-01976768.

[6] C. Doctorow, *Metacrap: Putting the torch to seven straw-men of the meta-utopia*, [Online; accessed 3. Jun. 2021], Aug. 2001. [Online]. Available: https://people.well.com/user/doctorow/metacrap.htm.

[7] J. Debenham, "Knowledge decay in a normalised knowledge base," in *International Conference on Database and Expert Systems Applications*, Springer, 2000, pp. 417–426.

[8] T. Blumer and N. Döring, "Are we the same online? The expression of the five factor personality traits on the computer and the Internet," *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, vol. 6, no. 3, 2012.

[9] C. S. Dweck, *Mindset: The new psychology of success*. Random House Digital, Inc., 2008.

[10] T. J. Berners-Lee, "Information management: A proposal," Tech. Rep., 1989.

[11] T. Berners-Lee, "What the Semantic Web can represent," 1998. [Online]. Available: https://www.w3.org/DesignIssues/RDFnot.html.

[12] F. Gandon, *An introduction to semantic web and linked data*, WWW 2014. [Online; accessed 18. May 2021], 2014. [Online]. Available: https://www.slideshare.net/fabien_gandon/semantic-web-and-linked-data.

[13] *W3C Semantic Web Activity Homepage*, [Online; accessed 18. May 2021], Aug. 2017. [Online]. Available: https://www.w3.org/2001/sw.

[14] D. Brickley, R. V. Guha, and B. McBride, "RDF Schema 1.1," *W3C recommendation*, vol. 25, pp. 2004–2014, 2014.

[15] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, S. Rudolph, *et al.*, "OWL 2 web ontology language primer," *W3C recommendation*, vol. 27, no. 1, p. 123, 2009.

[16] T. W. S. W. Group, "SPARQL 1.1 overview," 2013.

[17] J. P. McCrae, *The Linked Open Data Cloud*, [Online; accessed 18. May 2021], May 2021. [Online]. Available: https://lod-cloud.net.

[18] J. Lehmann, R. Isele, M. Jakob, *et al.*, "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic web*, vol. 6, no. 2, pp. 167–195, 2015.

[19] M. Morsey, J. Lehmann, S. Auer, C. Stadler, and S. Hellmann, "Dbpedia and the live extraction of structured data from wikipedia," *Program*, 2012.

[20] C. Fellbaum, "WordNet," in *Theory and applications of ontology: computer applications*, Springer, 2010, pp. 231–243.

[21] T. P. Tanon, G. Weikum, and F. Suchanek, "Yago 4: A reason-able knowledge base," in *European Semantic Web Conference*, Springer, 2020, pp. 583–596.

[22] T. Berners-Lee, *Linked-data design issues*, W3C design issue document, Jun. 2009. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html.

[23] F. Gandon and G. Schreiber, "RDF 1.1 XML - W3C Recommendation," 2014.

[24] D. Beckett, T. Berners-Lee, E. Prudhommeaux, and G. Carothers, "RDF 1.1 Turtle - W3C Recommendation," *World Wide Web Consortium*, 2014.

[25] C. Bizer and R. Cyganiak, "Rdf 1.1 trig-rdf dataset language-w3c recommendation 25 february 2014," 2014.

[26] G. Carothers and A. Seaborne, "RDF 1.1 N-Triples - W3C Recommendation," *World Wide Web Consortium*, 2014.

[27] G. Kellogg, P.-A. Champin, and D. Longley, "JSON-LD 1.1 - W3C Recommendation," *World Wide Web Consortium*, 2020.

[28] F. Manola, E. Miller, B. McBride, *et al.*, "RDF primer," *W3C recommendation*, vol. 10, no. 1-107, p. 6, 2004.

[29] W. W. W. Consortium *et al.*, *RDF Semantics: W3C Recommendation 10 February 2004*.

[30] N. Lopes, A. Zimmermann, A. Hogan, *et al.*, "Rdf needs annotations," in *W3C Workshop on RDF Next Steps, Stanford, Palo Alto, CA, USA*, Citeseer, 2010.

[31] D. Hernández, A. Hogan, and M. Krötzsch, "Reifying RDF: What works well with wikidata?" *SSWS@ ISWC*, vol. 1457, pp. 32–47, 2015.

[32] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandecic, "Introducing wikidata to the linked data web," in *Semantic Web Conference (1)*, P. Mika, T. Tudorache, A. Bernstein, *et al.*, Eds., ser. Lecture Notes in Computer Science, vol. 8796, Springer, 2014, pp. 50–65, ISBN: 978-3-319-11963-2. DOI: 10.1007/978-3-319-11964-9_4. [Online]. Available: http://dblp.uni-trier.de/db/conf/semweb/iswc2014-1.html#ErxlebenGKMV14.

[33] V. Nguyen, O. Bodenreider, and A. Sheth, "Don't like RDF reification? Making statements about statements using singleton property," in *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 759–770.

[34] O. Hartig, "RDF* and SPARQL*: An Alternative Approach to Annotate Statements in RDF," in *International Semantic Web Conference*, 2017.

[35] R. Dividino, S. Sizov, S. Staab, and B. Schueler, "Querying for provenance, trust, uncertainty and other meta knowledge in RDF," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 204–219, 2009.

[36] *What is RDF-Star?* [Online; accessed 29. Jun. 2021], Feb. 2021. [Online]. Available: https://www.ontotext.com/knowledgehub/fundamentals/what-is-rdf-star.

[37] S. Peroni, "A simplified agile methodology for ontology development," in *OWL: Experiences and Directions–Reasoner Evaluation*, Springer, 2016, pp. 55–69.

[38] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review*, vol. 11, no. 2, pp. 93–136, 1996.

[39] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, "Methontology: From ontological art towards ontological engineering," 1997.

[40] G. J. Klir, "Principles of uncertainty: What are they? Why do we need them?" *Fuzzy sets and systems*, vol. 74, no. 1, pp. 15–31, 1995.

[41] G. J. Klir and B. Yuan, "Fuzzy sets and fuzzy logic: theory and applications," *Possibility Theory versus Probab. Theory*, vol. 32, no. 2, pp. 207–208, 1996.

[42] T. Lukasiewicz and U. Straccia, "Managing uncertainty and vagueness in description logics for the semantic web," *Journal of Web Semantics*, vol. 6, no. 4, pp. 291–308, 2008.

[43] P. Marquis, O. Papini, and H. Prade, *A Guided Tour of Artificial Intelligence Research: Volume I: Knowledge Representation, Reasoning and Learning*. Springer Nature, 2020.

[44] C. A. Middelburg, "A survey of paraconsistent logics," *CoRR*, vol. abs/1103.4324, 2011. arXiv: 1103.4324. [Online]. Available: http://arxiv.org/abs/1103.4324.

[45] F. Kerdjoudj and O. Curé, "Evaluating Uncertainty in Textual Document," in *URSW at ISWC*, 2015.

[46] D. Dubois, J. Lang, and H. Prade, "Inconsistency in possibilistic knowledge bases: To live with it or not live with it," in *Fuzzy logic for the Management of Uncertainty*, John Wiley & Sons, Inc., 1992, pp. 335–351.

[47] G. Stoilos, G. B. Stamou, V. Tzouvaras, J. Z. Pan, and I. Horrocks, "Fuzzy OWL: Uncertainty and the Semantic Web.," in *OWLED*, 2005.

[48] G. J. Klir, "The role of uncertainty in systems modeling," in *Discrete Event Modeling and Simulation Technologies*, Springer, 2001, pp. 53–74.

[49] T. Bayes, "LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S," *Philosophical transactions of the Royal Society of London*, no. 53, pp. 370–418, 1763.

[50] D. Dubois, "Possibility theory and statistical reasoning," *Computational statistics & data analysis*, vol. 51, no. 1, pp. 47–69, 2006.

[51] G. Shafer, *A mathematical theory of evidence*. Princeton university press, 1976, vol. 42.

[52] D. Ceolin, L. Aroyo, and J. Noordegraaf, "Identifying and Classifying Uncertainty Layers in Web Document Quality Assessment.," in *URSW@ ISWC*, 2016, pp. 61–64.

[53] K. J. Laskey and K. B. Laskey, "Uncertainty Reasoning for the World Wide Web: Report on the URW3-XG Incubator Group.," in *URSW*, Citeseer, 2008.

[54] C. d'Amato, V. Bryl, and L. Serafini, "Semantic Knowledge Discovery and Data-Driven Logical Reasoning from Heterogeneous Data Sources," in *Uncertainty Reasoning for the Semantic Web III*, Springer, 2014, pp. 163–183.

[55] B.-B. Safia and M. Aicha, "Poss-OWL 2: Possibilistic Extension of OWL 2 for an uncertain geographic ontology," *Procedia Computer Science*, vol. 35, pp. 407–416, 2014.

[56] Z. Ding, Y. Peng, and R. Pan, "BayesOWL: Uncertainty modeling in semantic web ontologies," in *Soft computing in ontologies and semantic web*, Springer, 2006, pp. 3–29.

[57] C. Thomas and A. Sheth, "On the expressiveness of the languages for the semantic webmaking a case for A little more," in *Capturing Intelligence*, vol. 1, Elsevier, 2006, pp. 3–20.

[58] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic web*, vol. 8, no. 3, pp. 489–508, 2017.

[59] ——, "Identifying wrong links between datasets by multi-dimensional outlier detection.," in *WoDOOM*, 2014, pp. 27–38.

[60] A. Hogan, A. Polleres, J. Umbrich, and A. Zimmermann, "Some entities are more equal than others: statistical methods to consolidate linked data," in *4th International Workshop on New Forms of Reasoning for the Semantic Web: Scalable and Dynamic (NeFoRS2010)*, 2010.

[61] M. Rico, N. Mihindukulasooriya, D. Kontokostas, H. Paulheim, S. Hellmann, and A. Gómez-Pérez, "Predicting incorrect mappings: a data-driven approach applied to DBpedia," in *Proceedings of the 33rd annual ACM symposium on applied computing*, 2018, pp. 323–330.

[62] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches," in *Journal on data semantics IV*, Springer, 2005, pp. 146–171.

[63] A. A. Fernandes and N. W. Paton, "Quantifying and Propagating Uncertainty in Automated Linked Data Integration," *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXVII*, vol. 10940, p. 81, 2018.

[64] A. E. A. Djebri, A. G. B. Tettamanzi, and F. Gandon, "Publishing Uncertainty on the Semantic Web: Blurring the LOD Bubbles," in *Graph-Based*

*Representation and Reasoning*, D. Endres, M. Alam, and D. otropa, Eds., Cham: Springer International Publishing, 2019, pp. 42–56, ISBN: 978-3-030-23182-8.

[65] ——, "Linking and negotiating uncertainty theories over linked data," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 859–865.

[66] D. Benslimane, Q. Z. Sheng, M. Barhamgi, and H. Prade, "The uncertain Web: Concepts, challenges, and current solutions," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 1, p. 1, 2016.

[67] R. Larsen, *The Uncertain Web: Web Development in a Changing Landscape*. " O'Reilly Media, Inc.", 2014.

[68] S. Batsakis, E. G. Petrakis, I. Tachmazidis, and G. Antoniou, "Temporal representation and reasoning in OWL 2," *Semantic Web*, vol. 8, no. 6, pp. 981–1000, 2017.

[69] T. Lebo, S. Sahoo, D. McGuinness, *et al.*, "Prov-o: The prov ontology," *W3C recommendation*, vol. 30, 2013.

[70] U. Straccia, "Reasoning within fuzzy description logics," *Journal of artificial intelligence research*, vol. 14, pp. 137–166, 2001.

[71] C. Bizer, J. Lehmann, G. Kobilarov, *et al.*, "DBpedia-A crystallization point for the Web of Data," *Web Semantics: science, services and agents on the world wide web*, vol. 7, no. 3, pp. 154–165, 2009.

[72] E. Cabrio, S. Villata, and F. Gandon, "Classifying Inconsistencies in DBpedia Language Specific Chapters.," in *LREC*, 2014, pp. 1443–1450.

[73] A. Zimmermann, N. Lopes, A. Polleres, and U. Straccia, "A general framework for representing, reasoning and querying with annotated semantic web data," *Journal of Web Semantics*, vol. 11, pp. 72–95, 2012.

[74] O. Corby, C. Faron-Zucker, and F. Gandon, "LDScript: a Linked Data Script Language," in *International Semantic Web Conference*, Springer, 2017, pp. 208–224.

[75] I. Dongo and R. Chbeir, "S-RDF: A New RDF Serialization Format for Better Storage Without Losing Human Readability," in *OTM Confederated International Conferences - On the Move to Meaningful Internet Systems*, Springer, 2019, pp. 246–264.

[76] J. D. Fernández, M. A. Martnez-Prieto, and C. Gutierrez, "Compact representation of large RDF data sets for publishing and exchange," in *International Semantic Web Conference*, Springer, 2010, pp. 193–208.

[77] A. Artale, R. Kontchakov, F. Wolter, and M. Zakharyaschev, "Temporal description logic for ontology-based data access," 2013.

[78] G. Qi, Q. Ji, J. Z. Pan, and J. Du, "PossDLa possibilistic DL reasoner for uncertainty reasoning and inconsistency handling," in *Extended Semantic Web Conference*, Springer, 2010, pp. 416–420.

[79] R. Cyganiak, D. Wood, M. Lanthaler, G. Klyne, J. J. Carroll, and B. McBride, "RDF 1.1 concepts and abstract syntax," *W3C recommendation*, vol. 25, no. 02, 2014.

[80] O. Corby and C. F. Zucker, "Corese: A corporate semantic web engine," in *International Workshop on Real World RDF and Semantic Web Applications, International World Wide Web Conference*, 2002.

[81] D. Dubois, H. Prade, and S. Sandri, "On possibility/probability transformations," in *Fuzzy logic*, Springer, 1993, pp. 103–112.

[82] L. A. Zadeh, "Generalized Theory of Uncertainty: Principal Concepts and Ideas," in *Fundamental Uncertainty*, Springer, 2011, pp. 104–150.

[83] S. Benferhat, A. Levray, and K. Tabia, "On the analysis of probability-possibility transformations: Changing operations and graphical models," in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, Springer, 2015, pp. 279–289.

[84] Y. B. Slimen, R. Ayachi, and N. B. Amor, "Probability-possibility transformation," in *International Workshop on Fuzzy Logic and Applications*, Springer, 2013, pp. 122–130.

[85] Svensson, Lars G. and Atkinson, Rob and Car, Nicholas J., "HTML 5.3," W3C, W3C Working Draft, Dec. 2018, https://www.w3.org/TR/dx-prof-conneg/.

[86] A. E. A. Djebri, A. G. Tettamanzi, and F. Gandon, "Task-oriented uncertainty evaluation for linked data based on graph interlinks," in *International Conference on Knowledge Engineering and Knowledge Management*, Springer, 2020, pp. 204–215.

[87] M. Cheatham, I. F. Cruz, J. Euzenat, and C. Pesquita, "Special issue on ontology and linked data matching," *Semantic Web*, vol. 8, no. 2, pp. 183–184, 2017. DOI: 10.3233/SW-160251. [Online]. Available: https://doi.org/10.3233/SW-160251.

[88] J. D. Fernández, A. Llaves, and O. Corcho, "Efficient RDF interchange (ERI) format for RDF data streams," in *International Semantic Web Conference*, Springer, 2014, pp. 244–259.

[89] C. Guéret, P. Groth, C. Stadler, and J. Lehmann, "Assessing linked data mappings using network measures," in *Extended semantic web conference*, Springer, 2012, pp. 87–102.

[90] H. Farah, D. Symeonidou, and K. Todorov, "KeyRanker: Automatic RDF key ranking for data linking," in *Proceedings of the Knowledge Capture Conference*, 2017, pp. 1–8.

[91] W. W. W. Consortium *et al.*, "OWL 2 web ontology language document overview," 2012.

[92] A. Miles and S. Bechhofer, "SKOS simple knowledge organization system reference," *W3C recommendation*, vol. 18, W3C, 2009.

[93] E. Cabrio, S. Villata, and A. Palmero Aprosio, "A RADAR for information reconciliation in Question Answering systems over Linked Data 1," *Semantic Web*, vol. 8, no. 4, pp. 601–617, 2017.

[94] M. Gabielkov, A. Rao, and A. Legout, "Studying social networks at scale: Macroscopic anatomy of the twitter social graph," in *The 2014 ACM international conference on Measurement and modeling of computer systems*, 2014, pp. 277–288.

[95] A. E. A. Djebri, A. Ettorre, and J. Mortara, "Towards a linked open code," in *European Semantic Web Conference*, Springer, 2021, pp. 497–505.

[96] B. Dit, M. Revelle, M. Gethers, and D. Poshyvanyk, "Feature location in source code: A taxonomy and survey," *Journal of software: Evolution and Process*, vol. 25, no. 1, pp. 53–95, 2013.