Semantic Enrichment and Data Exploitation in an Open Data Context

Rabeb Abida

Jury

Prof. Denis Zampunieris
Luxembourg University

Prof. Sadok Ben Yahia
University of Southern Denmark

Prof. Vincent Englebert
University of Namur, Belgium

Prof. Benoit Vanderose
University of Namur, Belgium

Prof. Wim Vanhoof (president)
University of Namur, Belgium

Prof. Anthony Cleve (Supervisor)
University of Namur, Belgium

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in the subject of Computer Science

Supervised by Prof. Anthony Cleve

University of Namur
Namur Digital Institute
PReCISE Research Center

Cover design: © Presses universitaires de Namur

© Presses universitaires de Namur & Rabeb Abida, 2023
Rue Grandgagnage, 19
B - 5000 Namur (Belgium)
pun@unamur.be - www.pun.be
Registration of copyright: D/2023/1881/22
ISBN : 978-2-39029-186-2
Printed in Belgium

Reproduction of this book or any parts thereof, is strictly forbidden for all countries, outside the restrictive limits of the law, whatever the process, and notably photocopies or scanning.

Toute reproduction d'un extrait quelconque de ce livre, hors des limites restrictives prévues par la loi, par quelque procédé que ce soit, et notamment par photocopie ou scanner, est strictement interdite pour tous pays.

"When you give joy to other people, you get more joy in return. You should give a good thought to happiness that you can give out."

— ***Eleanor Roosevelt, First Lady of the United States***

ABSTRACT

The emerging trend of **Open Data** (OD) holds significant importance as it encompasses diverse datasets ranging from budget and expenditure to health, population, census, geography, and parliamentary minutes. Additionally, it incorporates data that is both directly and indirectly associated with the mentioned categories. Notably, **Open Data** extends to information indirectly "owned" by public administrations, including data from subsidiaries or agencies, such as climate and pollution data, public transport details, congestion and traffic statistics, as well as childcare and education-related information. The expansive nature of **Open Data** positions it as a powerful tool with immense potential to empower data publishers, analysts, citizens, and various stakeholders in numerous ways.

By making these data more discoverable, available, exploitable, accessible, understandable, and usable by all, we can foster data analysis, exploitation, and publication, promote transparency and better decision-making by giving users and citizens access to information about government activities and services, and provide opportunities for public participation and collaboration in government policies and decision-making processes. **Linked Open Data** (LOD) is the process of following a set of best practices for publishing and connecting structured data on the Web. LOD data not only improves the transparency of a government, but it can also lead to innovative solutions for community advancement as well as support public administration functions. While OD can bring many benefits, it also poses many challenges, such as the barriers faced by most data consumers and publishers, which should be addressed in this thesis. **Heterogeneity** is the most significant issue with open data, as the data may come in many various formats and structures, making it challenging to integrate, publish, and analyze the data effectively. This can limit the usefulness and re-use of the data and make it more difficult to extract meaningful information and determine the quality of the data to be published. Yet, limited work exists to support the process of LOD production and publishing in an e-Government context. There is a clear lack of automatically supported, integrated solutions that enable (i) combining appropriate techniques and tools to assist users, especially non-expert users, to efficiently manipulate datasets from extraction to publication LOGD; (ii) having a follow-up on the progress during this process as well as a semantic visualization of its data; and (iii) involving stakeholders with governments to reuse the data to provide public services and applications. In addition, **Data quality**, **Ambiguity**, and **Completeness** issues are another important challenge. For example, Open Data initiative releases raw data via various data portals, which are disparate and difficult

to use, especially by the general public. The same entity can be identified in different data sources. It can make it difficult to add correct information, merge, and analyze data from different sources. Additionally, incomplete data, inconsistencies, or errors can lead to inaccurate or misleading results, data collection errors, or incomplete reporting, which can have significant implications for decision-making and policy development. To address this challenge, it is essential to establish clear data quality standards and guidelines for data collection, annotation, and validation. These standards should include effective guidelines for data extraction, data cleaning, data interpretation, data analysis, and quality control, as well as procedures for handling missing data or incomplete metadata and ensuring data accuracy. **Discoverability** and **Understanding** are important because they help ensure that data can be found, accessed, and interpreted accurately and effectively. When the data files that have been linked on the portal do not provide semantic links between the data sources, it is therefore difficult to understand and establish the links between the data entities. Open Data publishers should promote data fluency among stakeholders so that they can analyze and interpret data effectively. Most existing tools are not capable of automating the process or being interoperable with Open Data portals. Hence, there is still no all-in-one approach that can handle the problems of open Open Data quality, and a lack of semantic annotation tools that integrate well with Open Data portal.

This Phd thesis **focuses on overcoming these barriers for Open Data** and, at the same time, **provides solutions** for improving practical interventions in this area. We propose **an useful and integrated semi-automatically framework** to inter-actively assist Open Data publishers in integrating and publishing LOD. Moreover, we focus **on addressing OD quality issues** such as noise, misspellings or incomplete metadata, missing cell values, and missing significant information. Therefore,we intend **to improve the usability, discoverability, and understanding of OD**, where we provide **all-in-one automatic framework** that can handle the problems of data quality and support the annotation systems that can be easily integrated into any OD portal. Thus, this framework aims **to help data analysts** to reduce the time spent quickly spotting signs of OD problems. In parallel, we deliver **a simpler application to facilitate users' ability to merge related annotated datasets**. Our ultimate objectives are **to improve data transparency** and potentially **lead to innovative solutions for the advancement of the community**. Finally, we evaluate the capabilities, usability, and usefulness of the solutions we have developed.

Keywords: open data, linked open data, semantic enrichment, data exploitation

RÉSUMÉ

La tendance émergente des **données ouvertes** (DO) est d'une importance significative car elle englobe divers ensembles de données allant du budget et des dépenses à la santé, à la population, au recensement, à la géographie et aux procès-verbaux parlementaires. En outre, elles intègrent des données qui sont à la fois directement et indirectement associées aux catégories mentionnées. En particulier, **Open Data** s'étend aux informations indirectement "détenues" par les administrations publiques, y compris les données provenant de filiales ou d'agences, telles que les données sur le climat et la pollution, les détails des transports publics, les statistiques sur les embouteillages et le trafic, ainsi que les informations sur la garde d'enfants et l'éducation. La nature étendue de **Open Data** en fait un outil puissant, avec un potentiel immense pour permettre aux producteurs de données, aux analystes, aux citoyens et aux diverses parties prenantes d'agir d'une multitude de manières.

En rendant ces données plus faciles à découvrir, disponibles, exploitables, accessibles, compréhensibles et utilisables pour tous, nous pouvons favoriser l'analyse, l'exploitation et la publication des données, promouvoir la transparence, et une meilleure prise de décision en donnant aux citoyens l'accès aux informations sur les activités et les services gouvernementaux, et offrir des possibilités de participation et de collaboration du public aux politiques gouvernementales et aux processus de prise de décision. Les **données ouvertes liées** sont le processus qui consiste à suivre un ensemble de bonnes pratiques pour publier et connecter des données structurées sur le Web. Les **données ouvertes liées** n'améliorent pas seulement la transparence d'un gouvernement, elles peuvent également conduire à des solutions innovantes pour le progrès de la communauté, ainsi que soutenir les fonctions de l'administration publique. Bien que les données ouvertes puissent apporter de nombreux avantages, elles posent également de nombreux défis, tels que les obstacles rencontrés par la plupart des utilisateurs et des éditeurs de données, qui devraient être abordés dans cette thèse. **L'hétérogénéité** est le problème le plus important des données ouvertes, car les données peuvent se présenter sous de nombreux formats et structures différents, ce qui rend difficile l'intégration, la publication et l'analyse des données de manière efficace. Cela peut limiter l'utilité et la réutilisation des données, et rendre plus difficile l'extraction d'informations significatives et la détermination de la qualité des données à publier. Pourtant, peu de travaux existent pour soutenir le processus de production et de publication de données ouvertes liées dans un contexte d'administration en ligne. Il y a un manque évident de solutions intégrées, soutenues de manière automatique, qui permettent (i) de combiner

des techniques et des outils appropriés pour aider les utilisateurs, en particulier les utilisateurs non experts, à manipuler efficacement les ensembles de données, de l'extraction à la publication, (ii) d'avoir un suivi des avancées au cours de ce processus ainsi qu'une visualisation sémantique de ses données, et (iii) d'impliquer les parties prenantes avec les gouvernements pour réutiliser les données afin de fournir des services et des applications publics. En outre, les problèmes de **qualité**, **d'ambiguïté** et **d'exhaustivité** des données constituent un autre défi important. Par exemple, l'initiative des données ouvertes publie des données brutes via différents portails de données, qui sont disparates et difficiles à utiliser, en particulier par le public en général. La même entité peut être identifiée dans différentes sources de données. Cela peut rendre difficile l'ajout d'informations correctes, la fusion et l'analyse de données. En outre, les données peuvent être incomplètes, incohérentes ou contenir des erreurs, ce qui peut conduire à des résultats inexacts ou erronés, à des erreurs de collecte de données ou à des rapports incomplets, qui peuvent avoir des conséquences importantes sur la prise de décision et l'élaboration des politiques. Pour remédier à ce problème, il est important d'établir des normes et des lignes directrices claires en matière de qualité des données pour la collecte, l'annotation et la validation des données. Ces normes devraient inclure des orientations efficaces pour l'extraction des données, le nettoyage des données, l'interprétation des données, l'analyse des données et le contrôle de la qualité, ainsi que des procédures pour traiter les données manquantes ou les métadonnées incomplètes et garantir la précision des données. La **découvrabilité** et la **compréhensibilité** sont importantes car elles permettent de s'assurer que les données peuvent être trouvées, consultées et interprétées de manière précise et efficace. Il est difficile de comprendre et d'établir les liens entre les entités de données. Les fournisseurs de données sont tenus de sensibiliser les parties prenantes à la connaissance des données, afin qu'elles puissent les analyser et les interpréter de manière efficace. La plupart des outils existants ne sont pas capables d'automatiser le processus et d'être interopérables avec les portails de données ouvertes. Il n'existe pas encore de solution complète pour résoudre les problèmes de qualité des données, et il n'y a pas d'outils d'annotation sémantique qui s'intègrent bien dans les portails de données publiques ouvertes.

Cette thèse vise à surmonter ces barrières pour les données ouvertes et fournit en même temps des solutions pour améliorer les interventions pratiques dans ce domaine. Nous proposons **un Framework semi-automatique utile et intégré pour aider interactivement les utilisateurs dans l'intégration et la publication des données ouvertes**. En outre, nous nous concentrons sur les problèmes de qualité des données ouvertes tels que le "bruit", les fautes d'orthographe ou les métadonnées incomplètes, les valeurs de cellule manquantes et les informations significatives manquantes. Nous entendons ainsi améliorer **la facilité d'utilisation, de découverte et de comprendre ces données, en fournissant une approche automatique qui peut traiter les problèmes de qualité des données et soutenir les systèmes d'annotation qui peuvent être facilement intégrés dans n'importe quel portail de données ouvertes**. Ce cadre vise à **aider les analystes de données** à réduire le temps passé à repérer rapidement les signes de problèmes de données ouvertes.

Parallèlement, nous mettons en œuvre **une application plus simple pour fa-**

ciliter la capacité des utilisateurs à fusionner des ensembles de données annotées connexes. Nos objectifs ultimes sont **d'améliorer la transparence des données et d'aboutir potentiellement à des solutions innovantes pour le progrès de la communauté**. Enfin, nous évaluons les capacités, la convivialité et l'utilité des solutions que nous avons développées.

Mots clés : les données ouvertes, les données ouvertes liées, l'enrichissement sémantique, l'exploration de données

ACKNOWLEDGEMENTS

The contributions presented in this thesis would not have been possible without God's help and the support of many people.

First of all, I would like to thank God for all the blessings in my life.

I would like to thank my supervisor, the very amicable Professor Anthony Cleve, for his guidance and support over the last four years. These ideas began when I started working with Anthony on the (ERDF) project supported by the European Union and the Walloon Region in 2018. I started thinking about a thesis, as I was very enthusiastic about this project. Anthony had given me a lot of guidance and support, and we were accepted for a CERUNA scholarship at the University of Namur to start this thesis. Thanks also for all his communications, which helped me finalize this thesis. I am fortunate to work with a very sympathetic and patient promoter who trusts me and believes in my capabilities. Thanks to him and his wife Sabine for inviting me to their home for a good "*Barbacue*".

I would especially like to thank the wonderful Professor Vincent Englebert, who invited me to work with him on a three-month internship in the field of the *Internet of Things*, in a wonderful atmosphere. Thanks for your welcome and support.

I would especially like to thank my best professor, Sadok Ben Yahia, who always supported and encouraged me right to the end.

I would like to express my greatest thankfulness and love to my parents, my brother, my sisters, my fiance, aunt Monia who have always encouraged me to pursue my dreams and supported me.

I would like to thank all my colleagues, Jerome, Maxime.G, Pol, Loup, Boris, Maxime.A and friends, Emna, Edilton, Aboubacar, Ahmed, Paterne, Andrea, Yalikha, Maroua for the support and the good times spent in a great convivial environment, at afterwork events, and at the various university events that punctuate the year other special occasions.

All my thanks to the members of the jury for the time spent reading this manuscript and for the feedback: Prof. Denis Zampunieris, Prof. Sadok Ben Yahia, Prof. Vincent Englebert, Prof. Benoit Vanderose, and Prof. Wim Vanhoof.

My sincerest gratitude goes to the University of Namur for providing an inviting environment conducive to learning for international students, and for the financial support that underpinned my research.

CONTENTS

Contents xiii

List of Figures xvii

List of Tables xxi

I Preface 1

1 General Introduction 3
1.1 Motivation 3
1.2 Problem statement 4
1.3 Objectives 5
1.4 Research Questions 6
1.5 Publications 6
1.6 Thesis Outline 7

II Background and State Of The Art 9

2 Conceptual Background 11
2.1 Introduction 11
2.2 Terminology 12
2.3 Adoption Open Data Barriers 15
2.4 Advantages and Challenges of Linked Open Data 24
2.5 Web of Data 26
2.6 Semantic Table Interpretation/Annotation (STI/A) 31
2.7 Conclusion 36

3 State Of The Art 37
3.1 Introduction 37
3.2 A review of the best methodologies for using and publishing Linked Open Data 40
3.3 Synthesis of approaches to using and publishing Linked Open Data 51
3.4 The principals of Semantic Tabular Annotation in Open Data 53
3.5 Comparison of SemTab challenge results 68

3.6 Synthesis of approaches and tools based on the Semantic Table Annotation workflow (STA) 70

III Contributions 77

4 Supporting Semantic-aware Analysis in LOD-enabled OGD Systems 79
4.1 Introduction 79
4.2 Methodology and Comparison 82
4.3 Pub_LOGD: Framework for Integrating and Publishing LOGD 84
4.4 Conclusion 93

5 Integrating, Discovering and Understanding LOGD Based on LOD Semantic Enrichment 95
5.1 Introduction 95
5.2 Requirements identified to support the contributions 97
5.3 Overview of contributions (SINATRA and ODSAG) 99
5.4 Implementation and Demonstration 115
5.5 Evaluation Results of SINATRA approach 120
5.6 Discussion and Limitations of SINATRA approach 123
5.7 Conclusion 125

6 SINATRA Web Application 127
6.1 Introduction 127
6.2 Requirements Analysis 128
6.3 Open Data Analysis 130
6.4 Merging Related Open Tabular Data 134
6.5 SinatraAPP Implementation (SinatraAPI and SinatraUI) 140
6.6 Conclusion 148

IV Evaluation and Discussion 151

7 Evaluation and Discussion 153
7.1 Introduction 153
7.2 Assessment of Pub_LOGD and Pub_LOGD Extension 154
7.3 Limitations of Pub_LOGD framework And Discussion 160
7.4 Data analysis Results 161
7.5 Demonstration of Merging Annotated datasets 171
7.6 Efficiency and Usability of the SinatraAPP 172
7.7 KPIs to assess SinatraApp's performance 179
7.8 Limitation of SinatraApp 181
7.9 Conclusion 182

V Postface **185**

8 Conclusion and Future Directions **187**
8.1 Summary of the Contributions 187
8.2 Future Directions 192

VI Appendix **195**

A Acronyms **197**

B Pub_LOGD Framework Assessments **201**
B.1 Pre-questionnaire for the development of an open data framework . 201
B.2 questionnaire for Evaluation of Pub_LOGD Framework 203
B.3 Survey of Framework Pub_LOGD Extension Evaluation 205
B.4 RDF's representation of the population dataset of Belgium. 207
B.5 RDF representation for the population of Belgium by NOE4J. 210

C SinatraAPP Assessments **213**
C.1 Estimating data analysis performance 213
C.2 Survey of Framework SinatraAPP 215

Bibliography **221**

LIST OF FIGURES

2.1 Datasets from Data Portals. 14
2.2 Multidimensional data (Statistics) modeled as a cube [104]. 15
2.3 Five Star Linked Open Data (figure redrawn from (https://5stardata.info/en/) (CC BY 1.0 Universal). 27
2.4 RDF triples [109]. 27
2.5 Relationship Between OG and LOD [13]. 29
2.6 RDF data cube vocabulary. 30
2.7 Steps of the Linked Data life-cycle supported by the LOD2 Stack [15]. . . 31
2.8 Open data life-cycle [13]. 32
2.9 Excerpts of (a) a Web table about countries and capitals, (b) a real CSV file about broadband data, and (c) a custom table with start-ups from Oxford and their foundation year [90]. 32
2.10 Cloud Diagram of DBpedia Linking Open Data [38]. 34
2.11 Wikidata is KG. 34

3.1 Final Study Selection Process. 43
3.2 Tasks of the Linked Data generation process by [157]. 45
3.3 Methodological guidelines for publishing library data by [77]. 46
3.4 Architecture of the CSV-ANNOTATION by [77]. 47
3.5 Linked Data publishing pipeline by [118]. 48
3.6 Development Workflow by [22]. 49
3.7 Distribution of the topics of the papers selected. 56
3.8 Functioning of MTab-DBpedia 2019. 58
3.9 Functioning of MTab-WikiData 2020. 60
3.10 Functioning of MantisTable 2019. 61
3.11 Functioning of MantisTable 2020. 63
3.12 Functioning of Dagobah 2019. 65
3.13 Functioning of Dagobah 2020. 67

4.1 Pub_LOGD Overview for publishing OGD as Linked Open Government Data (LOGD). 86
4.2 Sparqlify concepts and the query rewrite process [141]. 90
4.3 Screenshot from the platform of Pub_LOGD framework process. 92

5.1 An overview of the SINATRA approach. 99
5.2 The table presents a list of the presidents from T2Dv2 Gold Standard. . 101

5.3 Data annotation. Tabular Data (black) is annotated with the properties (magenta), entities (blue), and types (green) from DBpedia as asked in the CPA (a), CEA (b), and CTA (c) tasks respectively. 103
5.4 The design of MTab method. 104
5.5 Example of data repair by adding cell value "http://dbpedia.org/resource/Honolulu" (light green). 108
5.6 Example of *data augmentation* by adding new column "http://dbpedia.org/ontology/birthDate" (light blue). 109
5.7 An overview of the ODSAG. 111
5.8 Data Annotation. Tabular Data (black) is annotated with the properties (blue), entities (magenta), and types (green) from Wikidata as asked in the CPA, CEA, and CTA tasks respectively. 111
5.9 Screenshots of the features of SINATRA approach using President's list. 116
5.10 Screenshots of ODSAG using a COVID Dataset from the Namur portal. . 119

6.1 A data merging example: merge two tables to complete missing information for a "Polygones des localités" table. 136
6.2 Merging Datasets. 138
6.3 Steps in the table-merging algorithm. 139
6.4 General architecture of SinatraApp. 140
6.5 Screenshot of the SintraAPI Interface. 142
6.6 Screenshot of SinatraUI. 143
6.7 Screenshot of loading dataset phase. 143
6.8 Screenshot of data Analysis phase. 144
6.9 Screenshot of before pre-processing. 145
6.10 Screenshot of after pre-processing. 145
6.11 Screenshot of before pre-processing. 146
6.12 Screenshot of After pre-processing. 146
6.13 Screenshot of annotation data phase. 147
6.14 Screenshot of merging data phase. 147

7.1 User Interaction with the publishing Framework. 155
7.2 Special dataset contains many faults. 165
7.3 API interface for data analysis model. 166
7.4 User Interface for data analysis model. 166
7.5 Percentiles of the distribution of bootstrap estimates [37]. 169
7.6 Screenshot of the annotated data fusion model (MergingAPI). 172
7.7 Screenshot of the annotated data fusion model (MergingUI). 172
7.8 Assumed scale structure of the UEQ [166]. 175
7.9 Comparison of SinatraUI UEQ results in relation to the UEQ benchmark. 178
7.10 Pragmatic and Hedonic Quality. 178
7.11 Importance Ratings. 182

B.1 Population ontology information. 210
B.2 Data visualization by NEO4J. 211

C.1 Distribution of Answers per Item. 219
C.2 Distribution of Answers per Item. 220
C.3 Comparison of SinatraUI UEQ results in relation to the UEQ benchmark. 220

LIST OF TABLES

2.1 Summary of the potential barriers of the Open Data 21
2.1 Summary of the potential barriers of the Open Data 22
2.1 Summary of the potential barriers of the Open Data 23
2.2 Characteristics of the challenge datasets. 35

3.1 Tasks and studies for the identified methodologies. 73
3.2 Evaluation methods adopted in the selected studies 74
3.3 Summary table of the results of the SemTab challenge by round. R= Rank, P= Precision, R= Recall, and Ro= Round. 75
3.4 List of reviewed approaches and tools for semantic annotation. Sup = Supervised, Unsup = Unsupervised, SA = Semi-Automatic, M = Manual, A = Automatic, N = No, Y =Yes and N/A = Not Applicable. 76

4.1 Qualitative Metrics Description [126] 84
4.2 Evaluation of related approaches. 85

5.1 List of requirements identified to implement a SINATRA approach and an ODSAG tool. 98
5.2 List of examined tools for semantic annotation. - = Not to do , ✓= To do and (-/✓) = Depends on. 98
5.3 Features for subject column detection. 102
5.4 Values of the features of the sub_col detection for the president table. . 102
5.5 The characteristics of the datasets were evaluated by SINATRA approach 121
5.6 Evaluation results of the data annotation feature by MTab approach. . . 122
5.7 Evaluation results of the data repair feature. 122
5.8 Evaluation results of the data augmentation feature. 123

6.1 Inclusion criterion for datasets 131
6.2 Datasets selected for the data analysis module. 149

7.1 Dataset Description. 157
7.2 The relevant professional profiles of the participants. 158
7.3 The corresponding questions of survey. 159
7.4 The feedback given by the users. 159
7.5 Median, mean and standard deviation (SD) of survey scores. 160
7.6 Quantity of data faults 164

7.7 The estimates obtained using the bootstrap estimates 168
7.8 Types of participants for assessing the SINATRA process 173
7.9 Results for standard deviation of scale value. 175
7.10 Results of assessment tests. 176
7.11 Mean for each item on the survey. Green indicates a positive rating, and yellow a neutral rating. 177
7.12 Mean for each scale in the survey. Green means positive evaluation, and yellow neutral. 177
7.13 Results of comparison with benchmark. 178
7.14 Average of the three aspects: Pragmatic and Hedonic quality. 179
7.15 Distribution of answers to the various items. 179
7.16 Distribution of answers to the various items. 180
7.17 Distribution of answers to the various items. 180
7.18 Distribution of answers to the various items. 180
7.19 Distribution of answers to the various items. 181
7.20 Distribution of answers to the various items. 181
7.21 Importance Ratings: Mean and Standard Deviation. 181

Part I

Preface

CHAPTER 1

GENERAL INTRODUCTION

1.1 Motivation

In last years, **Open Data** (OD) is becoming an important trend, especially in government data context (i.e., OGD (Open Government Data)). It is a powerful tool that has enormous potential to empower data publishers, data analysts, citizens, businesses, and other stakeholders in various ways. Within this context, multiple government projects (e.g., in the United Kingdom, France, and Belgium) have been executed to supply free access to different data catalogues. Open Government Data (OGD) may contain multiple datasets, including budget and spending, health, population, census, geographical data, parliament minutes, etc. It also includes data that are indirectly ''owned" by public administrations (e.g. through subsidiaries or agencies), such as data related to climate/pollution, public transportation, congestion/traffic, child care/education.

By making these data more **available**, **accessible**, **exploitable** and **usable** to everyone, we can foster the publication of data, promote transparency, and improve social indicators. Hence, the appearance of **Linked Open Data** (LOD) that refers to data, which are **published** on the Web for use by public administrations, business and citizens. Such data a part from being machine-readable, is also **linked** to other external datasets. **LOD** is the process of following a set of best practices for publishing and connecting structured data on the Web. The blending of **OGD** and **LOD** forms **Linked Open Government Data LOGD**. This data not only improves the transparency of a government, it can also lead to innovative solutions for community advancement, as well as support the public administration functions. Applying several techniques to this data, such as **semantic web technologies** and **data mining**, is of great importance to ensure data interoperability and intelligibility, improve data quality and develop business strategies.

For example, researchers can **analyze** and **interpret** data in innovative ways that were previously impossible. They can use and exploit Open Data to develop new knowledge and solutions to complex problems in a variety of fields, while businesses can use it to identify new opportunities, reduce costs and improve their operations. It also promotes **transparency**, and better decision-making by allowing citizens to access information about government activities and services and provides opportunities for public participation and collaboration in government policies and decision-making processes. In addition to these benefits, **LOGD** can also help to drive economic and social indicators and improve quality of life in many ways. For example, by making health and education data publicly **discoverable**, **understand**, and **available**, communities can identify areas of need and target resources where they are most effective. Similarly, Open Data on environmental issues can help to inform policy decisions and promote sustainability. Overall, the potential benefits of Open Data are broad and wide-reaching.

1.2 Problem statement

Open Data (OD) initiatives have the potential to provide significant benefits, such as increased usability, transparency, customized access, and innovation. While it can bring many benefits, it also poses many challenges, such as the barriers faced by most data consumers and publishers, which should be addressed in our thesis.

Heterogeneity challenge in our context is related to open data portals and technical barriers. It is the most significant issue with open data, as the data may come in **many various formats and structures** (i.e., spreadsheets, CSV, EXCEL, HTML Table, relational database dumps, RDF), making it difficult to integrate, publish and analyze data effectively. Different organizations and public institutions may collect and store data in different formats, using different data models and taxonomies, making it challenging to combine and compare data from different sources. This can limit the usefulness and re-use of the data and make it more difficult to extract meaningful information and determine the quality of the data to be published. To address the challenge of **data publishing**, including data heterogeneity, OD initiatives must establish clear guidelines and standards for data publishing, including standards for data formats, metadata, and data documentation. In this regard, data integration and cleansing techniques, such as data normalization, mapping, and enrichment, can help overcome this challenge by transforming data into a consistent format that can be more easily combined and visualized. Overall, it requires a combination of standardization efforts, data integration techniques, and powerful data management and analysis tools. Yet, limited work exists, **for supporting the process of LOD production and publishing in a e-Government context**. There is a clear lack of automatically-supported, integrated solutions that enable **(i)** combining appropriate techniques and tools to assist Open Data publishers, especially non-expert users, to efficiently manipulate datasets from extraction to publication, **(ii)** having a follow-up on the progress during this process as well as a semantic visualization of its data, and **(iii)** involving stakeholders with governments to reuse the data to provide public services and applications.

Data quality, **ambiguity** and **completeness** issues are another important challenge, for example, Open Data initiative release raw data via various data portals which are disparate and difficult to use, especially by the public. What's more, the same entity can be identified in different data sources. It can make difficult to add correct information, merge and analyze data from different sources. Additionally, data may be **incomplete**, **inconsistent** or **contain errors** (misspelled or missing values, missing or incomplete metadata, and missing meaningful columns) can lead to inaccurate or misleading results, data collection errors or incomplete reporting, which can have significant implications for decision-making and policy development. To address this challenge, it is important to establish clear data quality standards and guidelines for data collection, and annotation. These standards should include effective guidelines for data extraction, data cleaning, data interpretation, and data analysis, as well as procedures for handling **missing data** or **incomplete metadata/data** and ensuring data accuracy.

Discoverability and **Understanding** are important qualities, because they **help ensure that data can be found, accessed, and interpreted accurately and effectively**. When, the data files that have been linked on the data portal do not provide the semantic links between the data sources, it is difficult to understand and establish the links between the data entities. **Discoverability** is especially important because open data is often characterized by its high level of heterogeneity, which can make it difficult for users to locate the specific data they need. **Metadata** plays a critical role in improving discoverability by providing information about the data, such as its source, format, content, and usage restrictions. **Understanding** is also essential because **Open Data can be complex and difficult to interpret**. To mitigate this risk, data providers need to ensure that data is presented in a clear and understandable format, and that it is accompanied by appropriate documentation, such as data dictionaries and user guides. Additionally, data providers should work to promote data literacy among stakeholders, so that **they can effectively analyze and interpret the data**. This highlights the need for effective understanding and discovery tools to help users understand metadata and navigate the data landscape more effectively. Most existing tools are not capable of automating the process and being interoperable with OD portals. Hence, there is still, no all-in-one approach that can handle the problems of the data quality and ambiguity, and a lack of semantic annotation tools that integrate well with OGD portal.

1.3 Objectives

This dissertation focuses on overcoming these barriers for OD, and at the same time provides solutions for improving practical interventions in this area. More particularly, the main goals are to

(i) support semantics-aware analysis in LOD-enabled Government systems, where, we propose an useful and integrated automatically framework to interactively assist Open Data publishers in integrating and publishing LOGD.

(ii) address OD quality issues such as noise, misspellings or incomplete metadata, missing cell values and missing significant information.

(iii) improve the usability, discoverability and understanding of OD.
(iv) build an approach to help users analyze OD to reduce time spent quickly pointing out signs of open data issues
(v) and implement a more simple application to assist stakeholders in merging related data sets to supplement existing datasets.

Given these aims, we can formulate the thesis as follows:

An all-in-one approach to implementing open data services that can be integrated into any data portal to help data publishers and analysts analyze and publish data and reduce the effort needed to overcome Open Data barriers, which can result in creative solutions for community progress.

1.4 Research Questions

From this statement we derived several research questions:

- **RQ1.** . How can we supply a standardized solution to ease the use and publication of LOGD?
- **RQ2.** How can we design an approach and a tool that support semantic interpretation of OD?
- **RQ3.** How to build a tool to help data analysts analyze public data to reduce time spent quickly pointing out signs of OD issues?
- **RQ4.** How can we implement a simpler technique and application to help users discover and combine related annotated datasets to get an overview of a particular topic?
- **RQ5.** What is the performance, usability and usefulness of the developed solutions?

1.5 Publications

We provide a list of selected, peer-reviewed publications, knowing that this thesis is based on them.

Book chapters

(i) **Abida, R**. and Cleve, A. (2023, February). Improving the Usability of Tabular Data Through Data Annotation, Repair and Augmentation. In Nordic Artificial Intelligence Research and Development: 4th Symposium of the Norwegian AI Society, NAIS 2022, Oslo, Norway, May 31–June 1, 2022, Revised Selected Papers (pp. 65-77). Cham: Springer International Publishing. [7]
(ii) **Abida, R**, Belghith, E. H., and Cleve, A. (2023). Towards Transparent Governance by Publishing Open Statistical Data. In Digital Technologies and Applications: Proceedings of ICDTA'23, Fez, Morocco, Volume 1 (pp. 355-365). Cham: Springer Nature Switzerland [6]

Conferences

(iii) **Abida, R**, Cleve, A. and Belghith, E.H., 2019. Supporting Semantics-aware Analysis in Linked Open Data-enabled Information Systems. [8]

(iv) Abida, R., Belghith, E.H. and Cleve, A.(2020, September). An end-to-end framework for integrating and publishing linked open government data. In 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (pp. 257-262). IEEE. [5]

(v) Chokki, A.P., **Abida, R**, Frénay, B., Vanderose, B. and Cleve, A. (2022, December). ODSAG: Enhancing Open Data Discoverability and Understanding through Semantic Annotation. EGOV-CeDEM-ePart 2022, p.120. [42]

1.6 Thesis Outline

The remaining of this thesis of eight chapters.

Chapter 1 presents the motivation for this thesis, including the statement of the problem to be addressed, the main objectives to be implemented, the research questions, and a list of selected peer-reviewed publications.

Part I presents the relevance of the research presented in this thesis. **Chapter 2** gives the required basic definitions and techniques that are the foundations of this thesis. It traces Open Data (OD) initiatives and their characteristics, provides an overview of the major barriers faced by consumers and providers, the benefits and challenges of LOD, describes the LOD (Linked Open Data) life-cycle, explains its phases, and introduces the concept of annotation systems. **Chapter 3** explores the states of the art of the research areas addressed. It selected research that has been applied to address Open Data outcomes. These works then summarized and the gaps in the relation to our research questions are highlighted.

Part II presents the contributions developed for each research question in dedicated chapters. **Chapter 4** aims to address the problems of heterogeneity and publication of OD and to resolve the research gaps. First, it presents a pipeline-based framework to publish OGD. It then describes the extension of this process based on the set of engagement's requirements. **Chapter 5** intends to overcome barriers related to data quality, completeness, discoverability and understanding. It describes an all-in-one approach and tool that handle them and that can be easily interoperable with any OD portal.

Chapter 6 specifies the noise categories of public datasets, so it presents the analysis functions used beforehand to help users analyze their data. It also develops discovery and business solutions for the Open Data platform to enable data publishers and analysts to find potential features, find accessible or linkable datasets to supplement existing data and plan the future for decision-making.

Part III reflects on the results presented in the previous part. **Chapter 7** evaluates the efficiency and the performance of the contributions, and discusses their implications for research and presents their limitations.

Part IV draws the general conclusion of this thesis and anticipates future work. **Chapter 8** concludes this thesis by reviewing the research questions, the contribu-

tions made by identifying the directions for future work that were not derived from the limitations.

Part II

Background and State Of The Art

CHAPTER 2

CONCEPTUAL BACKGROUND

2.1 Introduction

The aim of this section is to introduce the background of the thesis, in which we define in depth each concept that we have used in our contributions.

First of all, we define OD (Open Data) and its characteristics. Next, we present the definition of OGD (Open Government Data), the projects and portals that have been launched in this context. We outline OSD (Open Statistical Data), their portals, and a simple example to explain the representation of multidimensional data. In addition, we outline an overview of the most relevant barriers to OD adoption by selecting a 16 papers. We classify these barriers into three categories in the section 2.3 : (1) Open Data client barriers, (2) Open Data publishers barriers, and (3) Open Data barriers for both user and publisher (Knowledge and Experience). We summarize the potential barriers to Open Data and present these in the table 2.1 below.

The concept of LOD (Linked Open Data) refers to a set of practices for usisng, publishing, sharing and connecting data on Web in a machine-readable way. In Section 2.4, we describe the pros and cons of LOD and some of the standardization efforts underway to remove the barriers of the Open Data. The benefits of (Linked Open Data) LOD are, transparency, reliability, re-usability and sustainability, integration and availability, decision making, public participation and social value, and economic development. The challenges of LOD are, complexity, absence of knowledge, quality of the information, and technical challenge.

Secondly, we identify LOD and its characteristics, the RDF (Resource Description Framework) concept and its definitions and the SPARQL concept. Then, we present the Linked Open Government Data and demonstrate the relationship between Open Data and LD (Linked Data). Additionally, we define LOSD (Linked Open Statistical Data) and its vocabularies that were used to model the data in RDF format, such as

QB (the RDF Data Cube) and the SKOS (Simple Knowledge Organization System). Following this, we describe the LOD life cycle and explain its phases.

Thirdly, we show the concept of the STA (Semantic Table Annotation) that we used in our contributions (chapter 5 and chapter 6), the form of TD (Tabular Data) and their type. Finally, we describe some of the KG (knowledge Graphs), datasets, and benchmarks used to assess our work.

2.2 Terminology

To give some context to our work, we define here the most important concepts used in this report to represent the relationships between OD, GD, OSD, LD, LOD and LOGD. In addition, we present the most relevant barriers to OD adoption.

2.2.1 OD (Open Data)

Open data are data that "can be freely used, modified, and shared by anyone for any purpose" (Web ref. [3]). The knowledge commons[1] should be made freely available to the public so that anyone can explore, use, copy, and re-distribute them, in unmodified or modified form, without any restrictions, copyright, or fees. It is a buzzword and this area of research is attracting the attention of many communities. Therefore, the Open Knowledge Foundation [2] enumerates many topics of interest for which Open Data might have applications: scientific data, government accounts, financial markets, data from statistical offices, weather, geographical, and tourist information, pollution (e.g., air and river quality), transport (e.g., timetables, real-time traffic), culture, public sector budgeting, and performance levels of all kinds of data about policies and inspections (food, safety, education quality, etc.). Although Open Data Commons [3] and Creative Commons [4] are generally aim to provide data producers with appropriate tools to distribute their work as Open Data. In additionally, The opening of data by public and private institutions has considerable potential to provide many benefits to citizens, researchers, businesses, and other stakeholders. Additionally, it offers citizens and public institutionstransparent, free, and customized access to such information. It also has the potential to provide stakeholders with a growing economy by boosting innovation and generating new ideas in the public and private sectors by creating new ways of both understanding problems and interpreting data [43, 88, 89].

The deployment of the Internet and web-based applications, which are perfectly positioned to share data with a large audience, has fueled the OD movement's rise in popularity in recent years. In such a context, OD should achieve an OG (Open Government) in which the government acts as an open system and interacts with its environment. Not only should data be published, but feedback should be actively sought to improve government [88].

[1] The Knowledge Commons commonly include data from/about public institutions (e-government), scientific research, financial markets, transportation, pollution, arts, culture, etc.

[2] Open https://okfn.org/opendata/

[3] Open Data Commons: http://opendatacommons.org/

[4] Creative Commons: https://creativecommons.org/

2.2.2 OGD (Open Government Data)

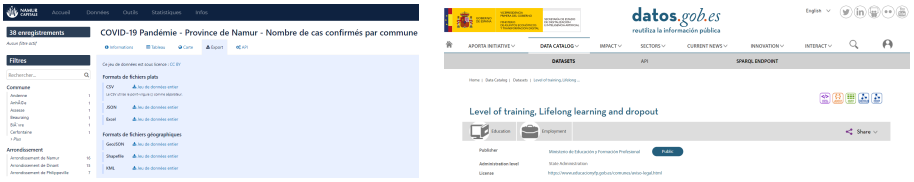
OGD is a global Open Government initiative. This latter was born in 2009, when the President of the United States of America, Barack Obama, published a Transparency and Open Government Memorandum with the goal of “creating an unprecedented level of openness in government." as a primary objective (White House, 2009a [81]). Shortly thereafter, Obama issued another memo on the Freedom of Information Act as “the most prominent expression of a profound national commitment to ensuring an open government” (White House, 2009b [82]). These memoranda served as the basis for the publication of the Open Government Directive, which aimed to establish the concept of open government on the basis of the ideas of transparency, participation, and collaboration within the administration. Whereas OGD promotes transparency and the publication of government data in open formats for public reuse, It also enables the development of innovative services, both in the public and private sectors. In brief, these data are considered a lever to achieve: (i) more transparency in decision-making; (ii) increased collaboration among citizens and organizations in government policies; and (iii) innovation leading to public and private value [35, 182].

In such context, numerous OGD projects have been launched, and many more are emerging every month, in which cities, regions, or governments engage in the open publication of their data (e.g., in France, the United States, Belgium) in order to provide free access to different data catalogues such as Data.europa.eu [44] Europpen Commission [5], data.gov.uk [6] data.gov[7],data.gouv.fr [8], data.gov.be [9], STATBel [10], and EuroStat[11] in diverse domains (i.e., geospatial, health, statistics, finance, transport, education, population, etc.). The European Data Portal is a single point of access to the countless open datasets and data portals of cities and regional and national public bodies in European countries. It contains interesting accounts of datasets by country, label, subject, or data format, as well as examples of applications leveraging these data (e.g., cartographic).

The European Data Portal and data.gov.be [12] contain the data portals, that we used in our work, such as, Portal Data of Namur [13](see, Figure 2.1,(a)), Open Data liege [14], Open Data Wallonie-Bruxelles [15], STIB [16]. These portals are designed to publish all government data in an open manner by default, as well as principles to increase the quality, quantity and reuse of data. they offer a count of datasets by topic, Keyword, id-dataset, description, data format, license, publishers, acess

[5] European Data portal: http://www.europeandataportal.eu/ : data from member states
[6] https://www.data.gov.uk/search
[7] https://www.data.gov/open-gov/
[8] https://www.gouvernement.fr/en/openness-of-public-data
[9] https://data.gov.be/fr
[10] https://statbel.fgov.be/en
[11] https://ec.europa.eu/eurostat/web/main/home
[12] https://data.gov.be/fr
[13] https://data.namur.be/pages/accueil/
[14] https://opendata.liege.be/pages/accueil/
[15] https://www.odwb.be/pages/home/
[16] https://data.stib-mivb.brussels/pages/home/

URL, size, modified, View icon, and etc. It includes APIs that allow access to all data available on the portal. These APIs are based on the APIs version of Opendatasoft [152]. As a result, these portals' data only support CSV, EXCEL, JSON and KML formats for each dataset and do not support RDF format, unlike the Spanish portal data [17]. They provide RDF and Turtle format for each dataset. This portal data distributes the availability of data through the deployment of open data catalogs that expose the metadata schema of these datasets by using DCAT [150][18]. It is an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web. In addition, this portal provides a SPARQL Endpoint interface, which allows users to configure their own queries in SPARQL language on the RDF graphs of the portal (see, Figure 2.1,(b)).



(a) Dataset from Portal Data of Namur, Belgium

(b) Dataset from Portal Data of Spanish

Figure 2.1: Datasets from Data Portals.

Also, the United Kingdom (data.gov.uk), Belgium (data.gov.be), and France (data.gov.fr) provide the same structure of portal as the previous data portals (counts of datasets, topic, data format, license, publishers). An interesting openness score, referring to the star rating system of linked open data proposed by T. Berners Lee [27], has been calculated on the UK portal. StatBel is the statistical portal for Belgium. It offers a list of datasets by subject, data format, license, and publisher. It also contains news related to statistics (e.g., population), engages the participant, and the most relevant datasets by subject. EuroStat is the statistical portal for Europe. It offers the same structure as STATBEL, also offering publications, applications, and tools to explore, as well as a guide to its website and contact information. Also, this portal site requires user authentication to explore the tools and other features.

2.2.3 OSD (Open Statistical Data)

A large part of OD concerns statistics [13, 36, 198], such as demographic (e.g., census data), population figures, and economic and social indicators (e.g., number of new businesses, unemployment, and workers). The majority of datasets published on the European Commission's open data portal are of statistical nature. The main suppliers of statistics at the international level are Eurostat, STATBEL, the World Bank, the OECD, and the CIA World Factbook. Statistical data are organized in a multidimensional manner, where a measured variable (e.g., unemployment or workers) is represented in terms of a number of dimensions (e.g., geography and time). In certain cases, statistical data is viewed as a cube, where each cell contains

[17] https://datos.gob.es/en/accessible-sparql
[18] https://www.w3.org/TR/vocab-dcat-2/

a measure or set of measures, and so here the multidimensional statistical data is referred to as data cubes (QB) or simply cubes [104].

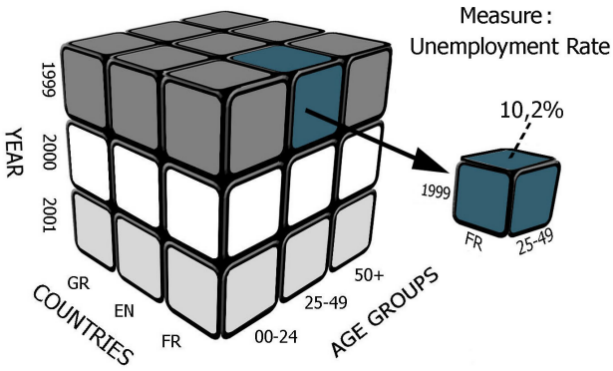


Figure 2.2: Multidimensional data (Statistics) modeled as a cube [104].

A set of dimensions and a set of measures specify a data cube. The dimensions produce a structure that includes a number of cells. Each cell contains a numerical value for each measure of the cube. *For example, a Eurostat cube with three dimensions, i.e., time in years, geography in countries, and age group, measures the employment rate. A cell in this cube would indicate that the percentage of unemployed people aged 25 to 49 in France in 1999 was 10.2 %* (see Figure 2.2) [104]. These statistical data can be important primary material for added-value services and products, increasing government transparency, contributing to economic growth, and providing social value to citizens [89, 92].

2.3 Adoption Open Data Barriers

Open data's definition alone pinpoints some of the many roadblocks towards its seamless adoption. This section outlines an overview of the most relevant barriers to Open Data adoption. We gathered a base of papers on Open Data barriers from various online databases, such as Google Scholar, Science Direct, ACM Digital Library, Scopus, and Springer Link, a citation-indexing service. The term "**Open Data**" has been established as a central keyword in the search, which we combined with the following keywords in our search strings: *"barrier", "challenge", "impediment", "problem", "issue", "consuming", and "publishing"*. We manually reviewed the referenced publications in the literature found in the digital libraries. The search and a manual scan for relevance performed by looking at titles and abstracts yielded 31 publications. We selected 16 articles from which we extracted a list of barriers. Most of the research on open data was published from 2012 onwards and is still ongoing.

As mentioned in the second chapter of this thesis, OD movement emerged after the launch of the U.S. Open Data portal in 2009, which may explain the lack of literature on Open Data before then. However, we have also included an article from 2007 [10], which is a good example. It discusses information retrieval in the public

sector using SW (Semantic Web) technologies, a subject that we believe is highly relevant to open data.

We intended to focus on barriers that are relevant to most **consumers** (users) and **publishers** of Open Data. For this reason, we also excluded barriers only applicable to government organizations. The selected barriers have often been mentioned or described as necessary in the literature.

2.3.1 Open Data Consumer Barriers

In this section, we provide an overview of the barriers that make consuming and reusing published open datasets challenging. The process of using Open Data usually starts with retrieving the data, which is most often done by downloading it from an Open Data portal. Once recovered, users are often stuck with corrupted, incomplete, or unusable data. In addition, the complexity of licensing can restrict users' rights to use this data in arbitrary applications freely. In addition, the lack of guidelines, support, or help desk services makes it difficult for new users or non-experts to use open data.

Open Data Portals

It's miles essential that Open Data portals meet certain standards and quality requirements in order to meet the demands and expectations of potential customers and providers. Nevertheless, there are, at present, no unified standards and quality enforcement guidelines that would ensure a certain level of quality. As a result, different data portals offer different experiences to their users. Indeed, in their analysis of the quality of open government data portals. [162, 201] concluded that there is **no single model** for an open data portal. Instead, it is a wide range of structures, with different processes and features for searching for information, different formats for presenting information, a variety of ways to classify information, and different infrastructures that do not always provide clear interfaces that show the search possibilities and enable clear navigation. Indeed, we can affirm a high degree of **heterogeneity** not only between portals but also within them with respect to formats, as well as completeness and accuracy of metadata descriptions [101, 179]. The heterogeneity of portals is mainly a problem when the developed features limit the user in some way; as a matter of fact, it challenges the discrepancies of how Open Data is made available in terms of access conditions, licenses, use permissions, file formats, and data structure [124, 140, 158]. A thorough review of the limitations of Open Data adoption by [88, 123] reviews many issues, from limitations that only worsen the user experience, in the absence of data on the datasets provided by the portal, to limitations that severely restrict the consumer. Some portals do not allow access to data without the user subscribing to their website or, even worse, place the data behind a paywall. A paywall can stop the users, especially if there is no way to express feedback on or experiment with the data.

However, immediately downloading the data from the portal website is not the best way for users to retrieve the data sets. Many customers choose to use a API (application programming interface) because the availability of such an interface

allows them to incorporate the data records into their application right away without first having to download the data manually [24, 79, 197].

In another seminal study on the limitations of OD platforms, [33, 140] outline the availability of APIs as a characteristic of Open Data platforms, which is evidently required for the automated consumption and reuse of data in other apps. Yet, in their analysis of more than fifty open data portals in 2012, they discovered that 43% of them had no APIs. The situation may have improved in recent years, as quasi-standardized platforms such as CKAN, which supply an API by default unless disabled voluntarily, are becoming more prevalent. According to the Open Data Portal Watch site [101, 179], CKAN is currently the most widely used Open Data platform, followed by Socrata, OpenDataSoft, and other proprietary solutions. APIs must also satisfy certain standards to be considered usable. One such interview, conducted as part of an in-depth study on socio-technical barriers to OD by [190, 201], points to the lack of a good API or an API only downloadable data as a barrier. It is important to keep in mind that the aforementioned study was conducted in 2012 and that Open Data platforms such as CKAN or Socrata were not as prevalent at that time as they are today. The reported lack of a good API could therefore also refer to the general lack of APIs available on Open Data portals in 2012 [33]. Nevertheless, quality of APIs may also have some limitations [24, 201, 202]. Table 2.1 provides a summary of all the barriers to user-side data portals that we examined in more detail in this section.

Data Quality

After searching, users may find that the quality of the data obtained does not meet and address their needs [34, 42, 195].The deficiency of open datasets'information quality on multiple levels: **inaccurate**, **incomplete** or **inaccessible** data, **obsolete** or **non-valid data**, **lacking metadata descriptions**, and simply unclear purpose for use [88, 101, 195]. We consider **transparency** barriers, which refer to the difficulties of identifying "the right data" [88, 120], **understanding** its content, and the consistency of conclusions that can be drawn from its analyses. Perhaps the biggest obstacle for users who desire to reuse the data in their own applications is that the data is not in a machine-readable format. The number of available datasets referred to the total number of datasets in the above formats and in **less advanced** formats: "pdf" (typical non-editable format), "xls"/"xlsx" (typical licensed tabular formats), "csv" (typical licensed formats). Other less typical formats were ignored in the analysis because of their lesser importance. The popular HTML standard was also ignored because it often refers to another standard and makes it difficult to assign the maturity level [190]. While it can be argued that the data is technically open, the use and reuse of this data is extremely difficult. Formats inadequately documented or proprietary formats can hinder the integration of data and its long-term sustainable use [39, 105]. The format of the original storage system can also be a problem, as shown by Kassen [96] *"...some data are really challenging to **publish** in a computer-readable format due to the fact that they only exists as a paper document..."* (p. 317). In the same vein, [10, 39] also note that the lack of structure and semantic

representation (**metadata**) of the data makes it almost impossible for machines to understand the data. [33, 60] argue that while some prefer to operate with data in a human-readable form, this form does not allow for reuse of the underlying raw data. Furthermore, the process of transforming the data into a machine-readable form can be very difficult [13, 17, 90, 142]. Table 2.1 presents a summary of the major data quality barriers from the existing literature.

User Legal barriers

Users have to deal with various legal constraints on datasets, imposed by terms of use and again heterogeneous - licenses. One might think that, by definition, it is easy to understand the legal position of the user, since Open Data is often defined by the fact that users can **freely access**, **reuse**, and **redistribute** the data. A license can be used as a condition for the publication and use of the data [133, 197]. The license can become an impediment when it is not explained, incomprehensible or complicated. Another part is if the license restricts the use too much or is incompatible with other licenses [134]. An example is *"... the UK Open Government Licence for instance requires that the re-user does not mislead others or misrepresent the Information or its source ..."* [[197], p. 7]. Moreover, in terms of licensing, most registers operated under an open license, a Creative Commons one or a national equivalent, whereas six registers provided access to the data without a license. Only 12 of the of the business registries provided a publication date, which was after 2013. The data's update frequency varied from daily or weekly to monthly or even yearly [95]. [13, 124, 172, 200] have found multiple sources that confirm that the interpretation of these licenses is rather complicated. Restrictive and complex licenses are only one side of the problem. In many cases, published datasets do not clearly indicate the license under which they are published. Just as publishing code without choosing a specific open source license, publishing data without a license does not indicate whether the data can be freely reused and re-distributed. Users may have concerns about legal consequences resulting from lawsuits. Although the probability of winning such a legal case is high, many open data users are independent developers or start-ups, who lack the means and resources to fight a judicial battle. To render open data attractive to newcomers, legal complexity must be reduced and efforts made to publish the data with an included license, which clearly defines the user's rights. Table 2.1 gives a summary of the main legal restraints as they have emerged from the literature.

2.3.2 Open Data Publishers Barriers

While, we have previously discussed the barriers to OD consumption on the user-side. In this section, we present the orthogonal barriers from the provider-side.

Strategical and Business

Strategic and business decisions are common reasons why many potential suppliers do not publish their data. [45, 99, 190] demonstrated that **publishing** Open Data was **not a priority** for local governments and was not part of the regular work of

many data professionals. The obvious benefits of publishing data are **not always explicit**, making it an extra task with no clear **encouragement** or **value**. They lack enabling **mechanisms** for an enterprisewide open data strategy implementation [45], implying a need for a strategic shift with elaborate processes for open data use. In this area, some business models depend on maintaining data confidentiality, as opening up the data would disturb the current business strategy of generating leads from the data owned [99]. In the same vein, many software developers **do not open** their code for risk of losing sales. For example, [136] describe giving away source code as a barrier to value. Releasing the data would allow third parties to profit from the data, which could potentially disrupt sales of the providers' product. Besides, the general **lack of time**, **resources**, or **simply additional expense** is a major barrier to publishing open data. The costs do not finish with the publication of the data. Data must be maintained and updated to meet user demand. For example *"... current challenge facing data journalism in Sweden is the lack of time to collect, analyze and present data. This may be a result of streamlined organizations as well as not having enough competence and skills to efficiently work with data."*[[12], p.404]. In summary, principal strategy and business obstacles found in the literature are again presented in Table 2.1.

Provider Legal Barriers

Legal studies, how law and norms interfere with publishing. Publishing studies deal with barriers related to the publishing process. The collection and processing of data must be consistent with the legal system, and only some things available in terms of information and technology can be used. The first important legal aspect is the right to use third-party data for business purposes, as well as copyright and licensing issues. Providers also face the threat of lawsuits, not only for license violations but also when they disclose private or otherwise confidential data. [45] state that as a result of vertical data management, data sharing between departments has not always been the norm. A summary of the legal restrictions on providers is shown in Table 2.1 .

Technical Barriers

Several technical barriers might make publishing data difficult for providers. They are often expected to publish data in different formats. The data might be the wrong format, and unreliable. In this context, we must consider the opening barriers: **Access** [5, 13, 24, 79, 88, 174, 200], **Metadata** [7, 13, 24, 42, 83, 88, 174, 193, 197], and **format** [5, 13, 24, 88, 96, 174, 193, 202] may cause problems for users. [123, 124, 174] affirm that there is currently no systematic policy of opening datasets in multiple formats and **no information** about the quality of the data. [88, 197, 201] also point to the lack of standard software for data processing, insufficient tools for using Open Data, and lack of clear interfaces (searching and navigation), consider as a technical barrier. Finally, the **lack of metadata standards** has been mentioned as a barrier in the literature [13, 24, 83, 154, 202].

Our research thereby draws attention to the necessity of domain-specific semantic models that make Open Data more open and usable by organisations. Therefore, some standardization efforts are already underway, the Resource Description Framework specification, recommended by the World Wide Web Consortium, that allows describing Web resources and used to create semantic webs [28, 29] with the Linked Open Data idea. Table 2.1 outlines the technical barriers encountered by data publishers.

2.3.3 User and Publisher: Knowledge and Know-how

Some Open Data barriers, which we summarize in this third category, can impact users and publishers, but from different directions. Effective use and publication of data is often difficult, especially for newcomers to the field, in terms of available support, leading to costs to providers on the one hand and lack of use on the other hand. **Users** [12, 34, 78, 83, 174, 197] have different requirements and goals. For example, visualizing data or integrating data into an application. If they are not satisfied, users may not be able to use the data. On the other hand, the **publisher** may find it to recognize and meet the needs of the users. Furthermore, [197, 202] affirm that data opening is a complex and not well understood activity, as there are many barriers to these processes. In this process, guidelines and instructions published on data portals may help potential providers. These instructions and steps would also improve the quality of published data. Moreover, the **tool/software** [13, 24, 42, 79, 88, 173, 197, 201] can help users, but the question then becomes who should apply them and to what extent. We also have the other side where data may require special tools to be used, which can be a barrier if they are unusable or expensive. In additional, the lack of **support** or **helpdesk** was identified as a barrier for data users by [88]. However, providers generally do not offer any support on how to use their data. while suppliers can't be expected to offer help-desk services or provide users with professional **guidance** in addition, especially if the data is free, simple instructions and guidelines would be a first step to overcome these barriers. **Evaluation quality** [79, 88, 96] can be a challenge, as **feedback** may be many and varied or users may not have **documentation** and **support tools**; the user may also not know what to look up. Furthermore, The data provider generally does not discuss it with users or obtain their feedback on the dataset to improve its own processes. It is clear that there is a lack of opportunity to engage open datasets in a talking environment committed to a certain dataset. Table 2.1 provides an overview of the OD barriers, which apply to both data users and publishers.

Table 2.1: Summary of the potential barriers of the Open Data

Sources	**Adoptions Barriers**	**Barriers**	**Open Data**	
			Consummer	Publisher
[162](2016), [201](2012), [179](2015), [101](2020), [124](2013), [158](2017), [140](2016), [88](2012), [123](2014) [24](2017), [197](2014), [140](2016), [101](2020), [190](2019), [201](2012), [202](2012), [201](2012), [24](2017), [8](2019), [42] (2022)	Open Data Portals	- No API is available - API slow or unresponsive - API restrictions - Registration is required to access data - Difficult navigation/searching - No information about the quality of the data - No information is provided on the content of the dataset - Language barriers - Data is not available for upload - Duplicate datasets - Only non-value added data published - No central Data Portal - Access limited to a certain group of users - The data is behind a paywall	X	
[195](2019), [34](2016), [88](2012), [101](2020), [105](2018), [190](2019), [39](2016), [195](2019), [34], [101](2020)(2016), [190](2019), [140](2016), [8] (2019), [5] (2020), [42] (2022)	Data Quality	- Data inaccuracy and incompleteness - Non-valid, obsolete data - Missing content - Missing Value - Not machine readable - Not available, inaccuracy metadata - No Potential to use data	X	

Table 2.1: Summary of the potential barriers of the Open Data

Sources	**Adoptions Barriers**	**Barriers**	**Open Data**	
			Consummer	Publisher
[197](2014), [133](2021), [134](2020), [197](2014), [95](2020), [172](2010), [200](2015), [124](2013), [13](2015), [45](2014), [7] (2023)	Legal Barriers	- Threat of judicial action - Licenses unclear (unclear conditions of reuse) - Restricted licenses - Complex, hard to understand license	X	X
[190](2019), [45](2014), [99](2017), [136](2014), [12](2014)	Strategical and Business	- Absence of business models - Not a priority of Open Data - Resistance to change, culturally risk-averse - No selling of data possible when an Open Data license is involved - Embedding open data in strategy and workflow - Uncertain financial impact - Lack of availability of resources and lack of time - Cost problems		X

Table 2.1: Summary of the potential barriers of the Open Data

Sources	**Adoptions Barriers**	**Barriers**	**Open Data**	
			Consummer	Publisher
[13](2015), [24](2017), [79](2015), [88](2012), [174](2015), [200](2015), [83](2017), [193](2015), [197](2014) [96](2017), [202](2012), [124](2013), [123](2014), [201](2012), [154](2011), [83](2017), [5](2020), [7] (2023)	Technical Barriers	- Lack and few of metadata standards - Lack of clear interfaces (searching and navigation) - No systematic policy of openning datasets - Lack of understanding and discoverability of datasets - Insufficient tools for Open Data -Semantic interpretation is complicated - No generic tool/software for data processing (enhancing/integrating/linking/combining) (Software fragmentation) - Many different standards and forms of machine-readable data		X
[12](2014), [34](2016), [78](2014), [83](2017), [174](2015), [197](2014), [202](2012), [13](2015), [79](2015) [88](2012), [173](2015), [201](2012), [79](2015), [96](2019), [201](2019), [42] (2022), [5] (2020)	Knowledge and Know-how	- No helpdesk - No support - Absence of documentation and tutorials -Absence of Forum and contact form - Request/Suggest form (that users request new type or format type of data) - No guiding principles/instructions (steps) - No engagement users (giving feedback)	X	X

Although we have previously discussed the barriers to Open Data. In the next section, we outline the benefits and drawbacks of Linked Open Data and some of the standardization efforts underway to address certain barriers to Open Data.

2.4 Advantages and Challenges of Linked Open Data

In this section, we describe some advantages and disadvantages of Linked Open Data, which we will tackle in our research.

2.4.1 Advantages of Linked Open Data

Transparency

Linked open data provides transparency between different sectors of government. In addition, the ordinary citizens of the country would be able to get a look at the achievements of the government departments. Linked Open Data will establish confidence among the general public towards the government [13, 120]. Everyone will have equal access to information, and this can help the government to enhance services and processes for citizens [161]. It also means that stakeholders can see, use and distribute the data freely. To achieve a transparent democracy, the country must open up data for stakeholders to be able to see the activities of the government [198].

Reliability

The Open Data Initiative will provide the foundation for a robust and open government focused on working and engaging with people. The data available from the government will assist people to believe in the government. They can make clear decision-making based on the data sources provided by the government [42, 165].

Data Re-usability and Sustainability

This is the most basic criteria for the quality of open data. Re-usability refers to the extent to which published data can be reused [7]. It means that data sets collected once can be used multiple times without having to collect the data again. This will reduce the needless duplication and associated costs. In addition, open data will ensure data sustainability [198].

Integration and Availability of Information

Open data will deliver the concept of information availability. All stakeholders and enterprises can access useful information from data portals. The business and technical gain of open data is the integration, merging and linking of private and public data [107].

Decision Making

Open data will provide a huge benefit to fair decision-making by allowing the comparison of various related data sources. Novel data sources can be created by combining available or existing data sources [8, 106].

Public Participation in Government and Social value

Open data offers the public the opportunity to engage broadly with government. The community can raise questions about certain areas (Education, Health, Environment, Agriculture, Employment and Budgets). This will help the government make equitable policies and improve decisions [42, 88, 123]. By openly publishing this data to the public, the government can encourage people to use it and invent new ideas, helping others create social value [201, 202].

Economic Development

One of the main benefits of linked open data will be economic development and the boosting of competitivity. It will drive innovation and contribute to the enhancement of a variety of processes, services, and products [93].

2.4.2 Challenges of Linked Open Data

Complexity

Open government initiatives are rapidly expanding, but the ability to discover the appropriate data is still lacking. The understanding and interpretation of the data is very complex. The data format and datasets are too complex to read and use. There is no support for retrieving and discovering information from these large initiatives. The emphasis is on a single dataset, therefore, the real benefit will come from linking multiple datasets together [21].

Absence of knowledge

The users can't use the information. In addition, they are not attracted to the data for several reasons, such as licensing fees, significant data initiatives, redundant pre-upload records, lack of statistical knowledge, an anticipated increase in costs, unsupported government organization behavior, and threats of lawsuits [7, 88].

Quality of the Information

Most of the data is invalid and obsolete [7]. The incomplete information is a challenge, where part of the information is visible and the rest of the information is missing or not loading [199]. However, the information may seem unrelated when considered in isolation, but when add information, linked and analyzed, it can provide new information. In addition, similar datasets are being stored in multiple locations which confuses data consumers.

Technical

It is hard to access data if it is not machine-readable and if its format is not well defined and established. Currently, there is a lack of architecture and standards for handling open data [194]. Likewise, there is no complete process for dealing with Linked Open data, from extraction to publication and providing informative data visualization [7, 93].

2.5 Web of Data

The Web has implicitly become the natural platform for publishing and sharing Open Data in digital formats. The sheer publication of open data on the web may not be enough to make it truly powerful and useful. The real power of OD lies in the fact that good-quality, understandable, and interlinked data elements can increase their respective value. The Linked Data Principles recommend best practices for publishing data on the web while preserving the existence and meaning of the relationships between data elements.

2.5.1 LOD (Linked Open Data)

The term "Linked Data" is used only for data that complies with the following four rules proposed by Berners-Lee in a famous paper from 2006:[19] [27]. It relates to a set of best practices for "*data published on the Web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external datasets, and can in turn be linked to from external datasets*" [29, 94].

Five principles of LD (Linked Data) / LOD (Linked Open Data) outlined by Tim Berners-Lee are as follows [27]:

(i) making your content available on the Web (in any format) under an open license;
(ii) Using URIs to identify all elements of a dataset;
(iii) Using HTTP URIs allows looking up the item identified through the URI;
(iv) When looking for a URI, provide useful information using standards (RDF*, SPARQL);
(v) Include links to the URIs of other datasets to allow discovery of other data.

Figure 2.3 depicts Berner's 5-star LOD rule.

Linked Data (LD) is based on the philosophy and technologies of the Semantic Web, but unlike the vision of the fully-fledged Semantic Web, it is primarily about publishing data structured in RDF using URIs rather than focusing on the ontological level or inferencing [75]. It offers the opportunity to create the Web of Data because data from decentralized and heterogeneous sources can be linked together by typical links. The Web of Data is about replacing isolated "data islands" with a massive distributed dataset built on top of the Web architecture [94].

[19] Berners-Lee (2006)

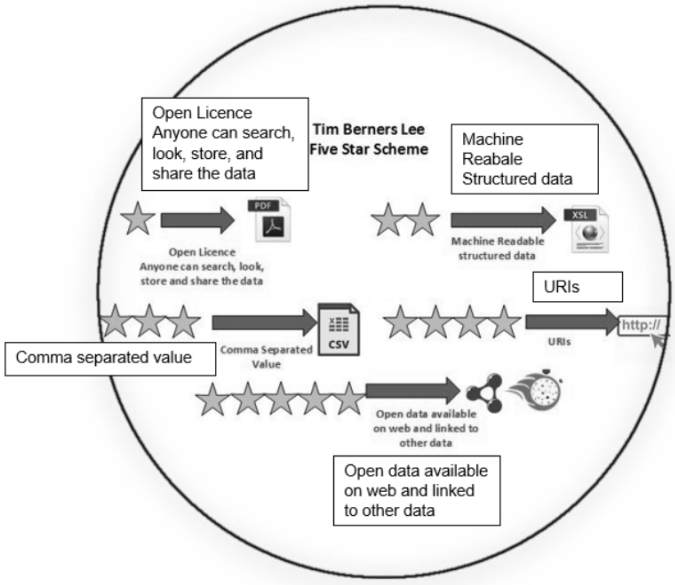


Figure 2.3: Five Star Linked Open Data (figure redrawn from (https://5stardata.info/en/) (CC BY 1.0 Universal).

Major benefits, such as data with uniform resource identifiers (URIs) that are globally unique and modeled according to the RDF specification [20], recommended by the World Wide Web Consortium (W3C), provide a high degree of interoperability. The flexibility of RDF enables data to be modeled according to ontologies that represent the concepts of the data. This renders linked data highly re-usable in a wide variety of contexts [94].

RDF, is a type of graph-based database that stores facts and semantic information. It is a model web publishing model standardized by the (W3C) World Wide Web Consortium that supports semantic queries. The principal purpose of RDF is to allow applications to move data to the web without losing its original significance. In addition, RDF data is stored in triples that are composed of three elements: subject, predicate, and object.

Figure 2.4: RDF triples [109].

[20] Cf. W3C (2014).

The RDF format is capable of taking any subject or concept and linking it to any other object using the predicate that indicates the type of relationship between the subject and the object [20].

Here is the formal definition of RDF triples as given in the W3C RDF standard [141].

Definition 1 (RDF Triple). *Assume there are pairwise disjoint infinite sets I, B, and L represents IRIs, blank nodes, and RDF literals, respectively. A triple* $(v_1, v_2, v_3) \in (I \cup B) \times I \times (I \cup B \cup L)$ *is called an RDF triple. In this tuple,* v_1 *is the subject, and* v_2 *the predicate and* v_3 *the object. We call* $T = I \cup B \cup L$ *the set of RDF terms* [141].

An RDF document is an oriented graph in which the nodes and edges are marked with identifiers, also known as URIs. In general, URIs are used for the subject and predicate, while the object can be either another URI or a literal such as a string or a number. In addition, a literal can be assigned a type that can be a URI [97].

Definition 2 (RDF Graph). *A finite set of RDF triples is called an RDF graph. The RDF graph itself represents a resource that is located at a certain location on the Web and thus has an associated IRI, the graph IRI* [141].

RDF encoding allows for rapid search using SPARQL [97], which is a semantic query language similar to but different from SQL (Structured Query Language).

SPARQL, a recursive acronym for SPARQL Protocol and RDF Query Language, is the standard for RDF query languages developed by the W3C. It can be used to fetch and handle data stored in RDF, which is the basic SW (Semantic Web) representation format.

2.5.2 LOGD (Linked Open Government Data)

The blending of OGD and LOD forms LOGD (Figure 2.5). In fact, LOGD does not only enhance transparency but can also result in economic benefits, innovative solutions for community advancement, and supporting public administrations' functions. Besides, applying several techniques to these data, such as SW technologies or data mining, is of high importance in order to ensure interoperability, create public value, and enhance data quality. This may involve the management of various steps, such as identification, the collection and processing of data to be used for operational decision-making or developing business strategies.

An existing life cycle for LOD can be a hybrid life cycle for LOD and metadata, covering both the offer and the demand.

2.5.3 LOSD (Linked Open Statistical Data)

National statistical institutes and public authorities have adopted the LD paradigm to publish their Statistical Data on the web. Several vocabularies have been created to enable modeling statistical data as RDF graphs, such as the RDF data cube (QB) vocabulary [54] and the SKOS (Simple Knowledge Organization System vocabulary) [130]. The RDF Data Cube[21] QB vocabulary [54] is a W3C standard for

[21] https://www.w3.org/TR/vocab-data-cube/

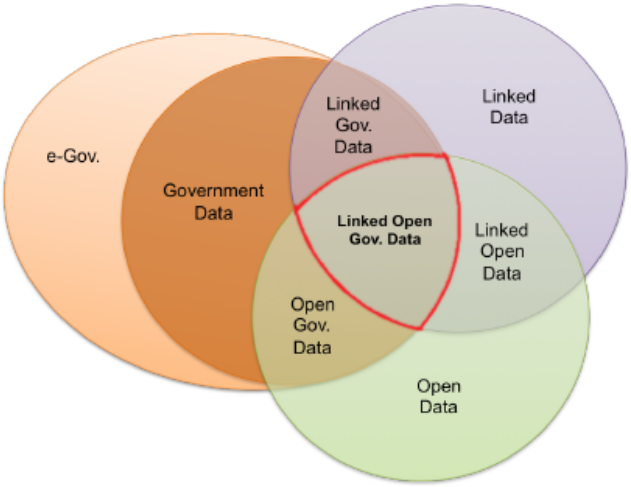


Figure 2.5: Relationship Between OG and LOD [13].

modeling data cubes as graphs and thus to adhere to the RDF model and LD principles (see Figure 2.6). The core class of the vocabulary is *qb:DataSet* which defines a cube. A cube has a *qb:DataStructureDefinition* that defines the structure of the cube and several *qb:Observation* that describe each cell of the cube. The structure is specified by the abstract class *qb:ComponentProperty*, which has three subclasses, namely, *qb:DimensionProperty*, *qb:MeasureProperty* and *qb:AttributeProperty*. The first one determines the dimensions of the cube, the second the measured variables, and the third structural metadata such as the unit of measurement. In general, component values are populated using preset code lists that can formulate hierarchies such as a geographic or administrative split. These code lists can be specified using the *SKOS vocabulary* [130] or the *QB vocabulary* [54]. *SKOS* is a W3C standard used for expressing the basic structure and content of conceptual schemas such as thesauri, taxonomies, and classification systems. The set of values is patterned as a *skos:ConceptScheme* and a value as a *skos:Concept*. In addition, *skos:broader* and *skos:narrower* are used to enforce a direct hierarchical link between two *skos:Concepts*. When reusing RDF data that is not modeled using SKOS, the QB vocabulary entered the*qb:HierarchicalCodeList* class that defines a set of root concepts in the hierarchy (*qb:hierarchyRoot*) and a parent-child relationship (*qb:parentChildProperty*). An unofficial transformation of Eurostat data, created as part of a research project, includes over 5,000 linked data cubes. A few statistical data sets from the European Central Bank, The World Bank, UNESCO, and other international organizations have also been transformed into linked data as a third part of the activities [104].

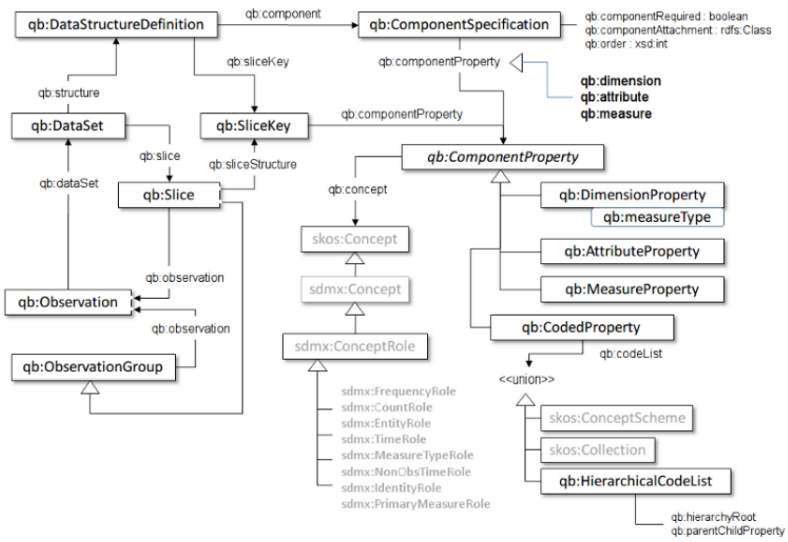


Figure 2.6: RDF data cube vocabulary.

2.5.4 Linked Open Data (LOD) Life-Cycle

The steps to LD or LOD (which involve free licensing and data accessibility) are not standardized. The LOD life-cycle can be supported by the LOD2 [22] stack, which is an integrated distribution of software to support the LOD life-cycle [15]. The various phases of the linked data life cycle are presented in Figure 2.7. They comprise:

Storage. Employing column-store technology, dynamic query optimization, adaptive join caching, optimized graph processing, and cluster/cloud scalability [15].

Creation. LOD2 aims to facilitate the creation of rich semantic knowledge bases, leveraging semantic Wiki technology, the WYSIWYM paradigm, and distributed social and semantic networking and collaboration techniques [15].

Interlinking. Creating and sustaining links in a semi-automated fashion LOD2 had looked for interconnection methods with high precision and recall that are configured automatically or based on feedback from the end-user [15].

Classification/ Enrichment. LD on the web is mostly raw instance data. For data integration, fusion, and search, as well as many other applications, this raw data must therefore be linked, enriched, and integrated with higher-level structures or ontologies [15].

Quality. The quality of data web content varies, as does the quality of web document content. LOD2 has developed techniques to help assess quality based on characteristics such as provenance, context, coverage, or structure [15].

Evolution/Repair. Data on the web is dynamic. The evaluation of data must be facilitated while keeping things stable. Changes and modifications to knowledge bases, vocabularies, and ontologies must be transparent and observable. LOD2 has

[22] https://semantic-web.com/project/lod2-the-linked-data-technology-stack-for-enterprises/

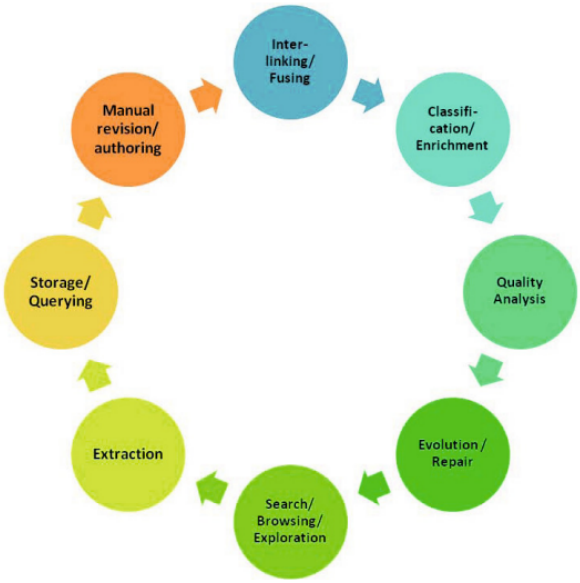


Figure 2.7: Steps of the Linked Data life-cycle supported by the LOD2 Stack [15].

also developed methods to spot problems in knowledge bases and to automatically suggest repair strategies [15].

Search/Navigation/Exploration. LOD2 has developed search, navigation, exploration and visualization techniques for different types of linked data (i.e. spatial, temporal, statistical), which make the Web of Data relevant for real users [15].

On the other side, [13] have proposed a process based on the previous data life-cycle (LOD2), the goal of which are to provide a standard process that government open data stakeholders can follow.

The suggested life cycle, shown in Figure 2.8, consists of three sections, namely a pre-processing section (rectangle), an operation section (oval), and a maintenance (post-processing) section (hexagon). The remaining sections, in order, deal with prepare data for publication, use published data, and maintain published data for sustainability.

2.6 Semantic Table Interpretation/Annotation (STI/A)

Most types of data used and published on the web are tabular. This section presents the semantic processes applied on these data.

2.6.1 TD (Tabular Data)

Tabular data in the form of CSV, TSV, EXCEL, and spreadsheet files is the most common input format in a data analytics pipeline. Then, TD (Tabular Data) can be

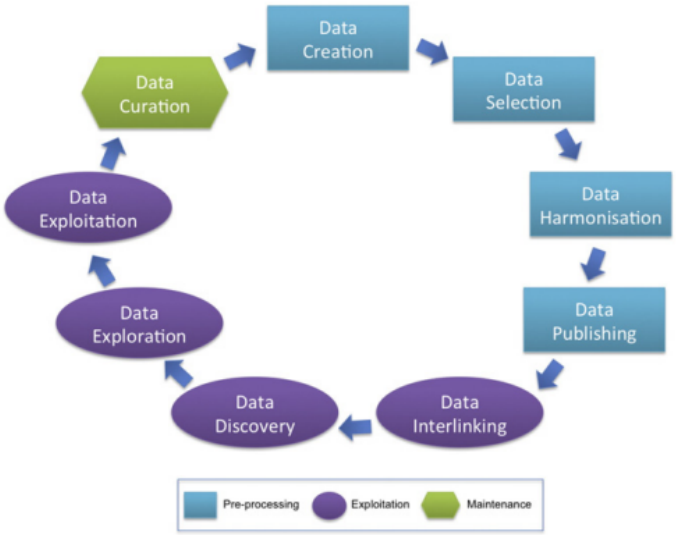


Figure 2.8: Open data life-cycle [13].

considered as a set of columns $C = c_1, \ldots, c_m$, a set of rows $R = r_1, \ldots, r_n$, or a matrix of cells $T = t_{1,1}, \ldots, t_{n,m}$, where a column $ck = t_{1,k}, \ldots, t_{n,k}$ and a row $r_k = t_{k,1}, \ldots, t_{k,m}$ are cell tuples. Searches assume that all columns and rows are the same size, possibly with cells with empty values. In arbitrary TD, unlike relational tables, column names and row identifiers (i.e., primary keys) may be missing. In tables and relational tables, rows typically identify an entity, whereas in arbitrary tabular data (e.g., typical CSV files in data science), there may not be a primary entity in each rows [90](see, Table 2.9).

In Section 5.3.1, we explain in detail that the mapping tasks (CTA, CPA, CEA) can be viewed as sub-tasks that can serve the wider purpose of mapping an integer table to a class or a row of a table to an entity (see Table 5.3).

China	Beijing
Indonesia	Jakarta
Congo	Kinshasa
Brazil	
Congo	Brazzaville

(a) Web table

Virgin	60	London
BT	60	East
BT	40	Scotland
Virgen	40	Wales
Orange	30	West Midlands

(b) CSV file

OST	2017
DeepReason.ai	2018
Oxstem	2011
Oxbotica	2014
DeepMind	2010

(c) Custom table

Figure 2.9: Excerpts of (a) a Web table about countries and capitals, (b) a real CSV file about broadband data, and (c) a custom table with start-ups from Oxford and their foundation year [90].

2.6.2 KG (Knowledge Graph)

KGs are semantic networks that formally describe things or entities in the real world and their relationships [80]. The searches consider RDF-based KGs that are represented as a set of RDF $triples < S, P, O >$where S represents a subject (a class or instance), P represents a predicate (a property), and O represents an object (a class, an instance, or a data value, *e.g.* text, date, or number). RDF entities (i.e., **classes**, **properties**, and **instances**) are represented by uniform resource identifiers (URIs). A KG consists of a TBox (terminology component) and an ABox (assertion component). The TBox [90] is composed of RDF Schema constructs such as class subsumption (e.g., **dbo:President rdfs:subClassOf dbo:Person**) and property domains (e.g., **dbo:PresidentAdvisor rdfs:domain dbo:President**). The ABox [90] contains relationships between entities and semantic type definitions (for example, **dbr:Barack_Obama rdf:type dbo:President**). Then, we describe the KGs, which we used in chapter 5.

An encyclopedic KG is typically large, covers multiple domains, and is often edited in an open and cooperative way. The Linked Open Data cloud [129], which consists of the following larger encyclopedic KGs (DBpedia and Wikidata), represents this openness.

DBpedia [28] is one of the principal hubs of the LOD cloud because of its multiple links to other KGs. It was founded by researchers and scientists from the University of Leipzig. and the University of Mannheim in Germany by retrieving multilingual structured data from Wikipedia (e.g., infoboxes). It is kept up to date with regular extracts from Wikipedia. At the beginning of 2016, DBpedia contained over six million instances and 200 million individual facts. In addition, the DBpedia project provides tools and services such as the DBpedia Spotlight tool [23] [129] that are useful for automatically annotating mentions of DBpedia resources in unstructured data into KG entities. DBpedia search service [24] that resolves keywords into resource identifiers It can be used to enrich simple tables with DBpedia URIs. The linking of labels to resource identifiers provides a gateway to the world of linked data for the task at hand.

Wikidata [25] is a free and open knowledge base. It is a project owned by the Wikimedia Foundation that aims to populate the infoboxes that appear on every page of Wikipedia [113, 188]. It is a free and open knowledge base that is readable and editable by both humans and machines. Like **Wikipedia**, it is produced collaboratively by many thousands of volunteers. By the beginning of 2021, Wikidata had listed approximately 90 million entities with labels expressed in over 350 languages. It delivers a separate page for each entity, has a unique numerical identification mechanism, and has a lineage system that allows for the tracing of facts to their source. Wikidata also supports many other sites and services beyond the Wikimedia projects. Its content is available under a free license, can be exported in standard formats and can be linked to other open datasets on the web of linked data.

[23] https://www.dbpedia.org/resources/spotlight/
[24] https://www.dbpedia.org/resources/lookup-linking/
[25] https://www.wikidata.org/wiki/Wikidata:Main_Page

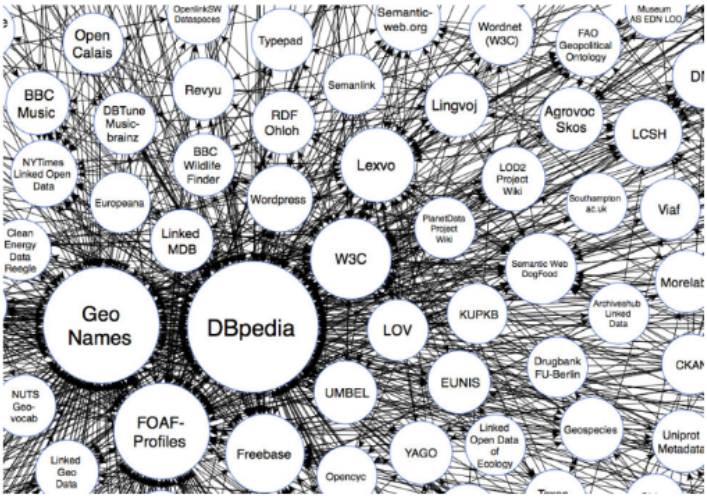


Figure 2.10: Cloud Diagram of DBpedia Linking Open Data [38].

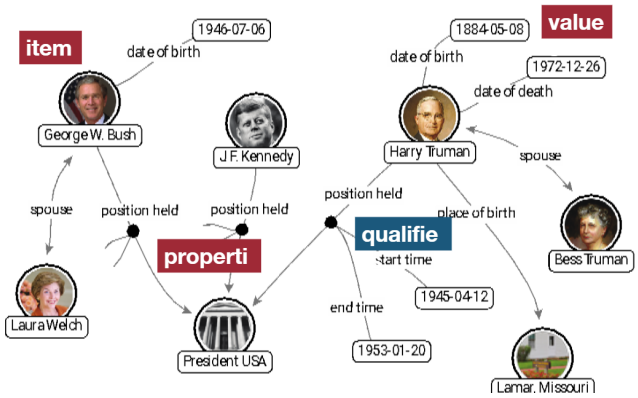


Figure 2.11: Wikidata is KG.

Freebase [30] was deployed by Metaweb from 2007 until its acquisition by Google in 2010. Its content also derives from collaborative editing and structural data automatically loaded from Wikipedia and other websites. At the beginning of 2014, Freebase had 68 million entities and nearly 1 billion facts. It stopped operating in May 2015, and most of its data was migrated to Wikidata.

2.6.3 Datasets and Benchmarks

Many datasets have been suggested to evaluate STA/ STI (Semantic Table Annotation/ Semantic Table Interpretation) approaches. Some datasets seek to establish standards of excellence in which table components (cells, rows, columns, or cell pairs) are associated with KG components (entity, class, or property), while others

collect high-quality tables to provide support for STA training. In this section, we elaborate on the most popular datasets used by the SemTab 2019, 2020 and 2021 competitions.

T2D [159] comes from the Web Data Commons project[26]. DBpedia is used like a target KG and complete metadata such as the context of the table and if the table has a header is not provided. A second version of the gold standard adding negative samples has been published and named T2Dv2 [110]. It contains common tables from the Web. T2Dv2 consists of 779 tables, with about 400 entity columns covering content on person, organization, place, work, species, etc., about 26,000 DBpedia entity matchings, and about 420 DBpedia property matchings. It has become the default reference dataset to be used to evaluate STI approaches.

SemTab competition [27] coincident with the International Semantic Web Conference in 2019, 2020, and 2021 delivers the largest datasets. This competition has drawn nearly 50 participants over the past three years.

The SemTab 2019 datasets [90] use DBpedia as the target KG, while SemTab 2020 [91] uses Wikidata KG. This competition is composed of four rounds in 2019 and 2020. In the 2019 edition, each round contains the datasets and target datasets (CEA-Targets, CTA-Targets, and CPA-Targets) that were represented in the table 2.2. Besides, this challenge extended the Gold Standard T2Dv2 datasets for round 1.

Table 2.2: Characteristics of the challenge datasets.

Round	Tables	CEA	CTA	CPA
1 (T2Dv2)	64	8418	120	116
2	11,924	463,796	14,780	6,762
3	2, 161	406, 827	5, 752	7, 575
4	817	107,352	1,732	2,747

Round 2 dataset was composed of 1,924 tables; there were 463,796 instance annotations (CEA-Targets) for CEA, 14,780 class annotations (CTA-Targets) for CTA, and 6,762 property correspondences (CPA- Targets) for CPA. Round 3 dataset was made of 2, 161 tables; there were 406, 827 instance annotations (CEA-Targets) for CEA, 5, 752 class annotations (CTA-Targets) for CTA, and 7, 575 property correspondences (CPA-Targets) for CPA. Round 4 dataset was composed of 817 tables; there were 107,352 instance annotations (CEA-Targets) for CEA, 1,732 class annotations (CTA- Targets) for CTA, and 2,747 property correspondences (CPA-Targets) for CPA. However, Rounds 3 and 4 were composed of automatically generated annotations with enhanced characteristics and a focus on non-trivial annotations.

Format of ground truths

To execute the matching tasks (CEA, CPA, CTA) with the target table elements we used this format: CEA-Targets were in a CSV/EXCEL file with lines having these grounds: “Table ID”, “Column ID”, “Row ID” and “DBpedia entity (only one)” (e.g., “table1”,“0”,

[26] http://webdatacommons.org/webtables/goldstandardV2.htm
[27] https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

"121","dbr:Belgium"). Similarly, CPA-Targets had the following grounds per line: "Table ID", "Head Column ID", "Tail Column ID" and "DBpedia property (only one)" (e.g., "table1","0", "1","dbo:releaseDate"). For CTA-Targets,there was more than one type annotation, separated by a space, having these grounds: 'Table ID", "Column ID" and "DBpedia classes (1 or more)" (e.g., "table1","0", "1","dbo:Country dbo:Place") [90].

In the 2020 and 2021 editions, the challenges are based on the Wikidata Dump (April 24, 2020) [28]. SemTab 2020 and 2021 are structured in four evaluation cycles with the objective of testing different datasets of increasing difficulty. Each round contains the datasets and target datasets (CEA-Targets, CTA-Targets, CPA-Targets) as SemTab 2019. Rounds 1 through 3 are conducted with the support of AIcrowd [29] , which provided automatic evaluation of submitted solutions and relied on an automatic dataset generator [90, 91]. Round 4 is a completely blind round (i.e., without evaluation of submissions via AIcrowd) using a combination of: an automatically generated dataset (AG) as in the previous cycles, and the Tough Tables (2T) dataset for CEA and CTA [50].

2.7 Conclusion

In this chapter, we have defined in depth the fundamental concepts, which are: OD, OGD, OSD, LOD, RDF, LOD, LOSD, LOD life-cycle, and STA, its forms, KGs and the datasets and Benchmarks that we have applied in our contributions. We have also organized the OD barriers into three categories: (1) Open Data users barriers, (2) Open Data providers barriers, as well as (3) Open Data Consumer and Provider barriers. Besides, we have summarized the potential barriers to Open Data. Afterwards, we described the benefits and the challenges of LOD and showed that the LOD act was effective in solving the mentioned challenges.

In the next chapter, we will present the weaknesses and strengths of the approaches that exist in the context of LOD, LOGD and LOSD. We will also discuss the Semantic Table Annotation (STA) approaches.

[28] https://doi.org/10.5281/zenodo. 4282941
[29] https://www.aicrowd.com/

CHAPTER 3

STATE OF THE ART

"...will take appropriate action, consistent with law and policy, to disclose information rapidly in forms that the public can readily find and use."

— Barack Obama, 2009

3.1 Introduction

Open data (OD) has an enormous potential to empower citizens, data analysts, data publisher, and other stakeholders with many benefits, such as a growing economy and social indicators by driving innovation and increased transparency. But, OD present many barriers to its seamless adoption, including the following, heterogeneity formats, no available and incomplete data, no single model, lack of good API, lack of semantic representation, incomprehensible and complicated license of datasets, no clear value, lack of mechanisms for implementing a companywide strategy, no open source code, lack of time and resources, heavy costs of publishing the data, lack of consistency with legal systems, no systematic policy of opening datasets, no information about the data quality, absence of standard software, insufficient tools, no easy and clear interfaces, more demands and needs of the users, lack of support and helpdesk or documentation, no guidance for data users, feedback and evaluation quality can be challenges, and lack of timeliness to engage open datasets in a talking environment (Section 2.3).

Against the background in Chapter 2, many governments have decided to use LOD to create platforms, portals, services, and applications as an effective form of OD. With the release of Open Data by many countries, different methodologies for publishing LD on the web have been proposed. The publication of data can be used for many purposes, such as for consuming or integrating into their applications or to provide more informed business decisions. In Section 3.2, we present the best practices on methodologies for using, publishing, sharing and linking data on the Web. Before discussing related LOD work in Section 3.2.2. We started using the systematic mapping method to identify work for supporting and publishing LOGD. The research methodology in Section 3.2.1 consists of three steps: 1) defining search questions, 2) searching for primary studies, and 3) selecting articles for inclusion and exclusion. In the research methodology, we proposed to engage users' comments to support the execution of the systematic literature review method and the best articles can be selected. In the end, we selected 23 academic articles, web pages and reports (See Figure 3.4, presents the methodology used to select the final studies). Through this methodology, we aimed to find the best methodologies and approaches for using and publishing open government data that can be used or recommended, as well as any type of quality processes involved in the methodologies to address some of the barriers to open data.

Some principles and processes were proposed, with varying degrees of sophistication, practices, and tools. In Section 3.2.2, we present the related work on three aspects such are: (i) LOD, (ii) LOGD, and (iii) LOSD. We explain in details each steps of methodologies and approaches for using and publishing LOGD, designed to solve its problems. In Section 3.3, we summarize these selected works around the following three topics(see the table 3.1): Common phases (Steps) include, data extraction, data transformation, data interlinking, data storage, data visualizing, data publishing, create Apps, and engaged with community; the associated tasks are select data, clean-up, design URIs, define vocabularies, map vocabularies to data, enrich data, convert to RDF, clean-up RDF data, link to other data sources, merge data sources, store RDF data, enable discovery, publish data, create API, build apps on top of data, and engage with community; and quality assessment validations and evaluation of the approach. We classify the different evaluation methods applied in the selected studies (see, the table 3.2). The evaluation methods adopted which can be qualitative and quantitative measures (functional requirements), a prototype, an illustrative scenario or a case study and can be arguments and logical reasoning.

However, in Chapter 4, we aim at covering all the data phases from data extraction to data publishing, and visualization. In additional, we also **evaluate our contributions** in chapter 6 using some of the evaluation methods presented in the table 3.2 below. The production and publication of LD is an intensive engineering effort to achieve high quality, and existing general guidelines may not be sufficient to make the processes consistently repeatable. In additional, the widespread availability of datasets is driving the use of ML and AI technologies to gather information, study trends, and predict behaviors that are not visible in the data world. Therefore, the gathering and integration of data from different sources is an essentially manual

activity that requires the knowledge of expert users and is costly in terms of time and money.
Against this backdrop, tables on the Web can be the source of highly precious data. They often contain valuable data, but can be difficult to use because of missing, incomplete, or obscure metadata , and missing information. Therefore, it is essential to understand their meaning. However, a lack of understanding of the semantic structure and meaning of the content can hinder the data analysis process. Indeed, before being able to use the data, consumers must be able to find data relevant to their needs. Once they discover the relevant data, they must be able to understand the metadata and content of the data in order to exploit it, such as for data integration, data cleaning, data mining, machine learning and knowledge discovery tasks. Moreover, adding semantic information to Web tables can enhance a wide range of applications, such as Web search, question answering, and KB (knowledge Base) building. It is therefore necessary to make the process of collecting and linking data from many different sources affordable so that the data sets are ready to perform the required analysis. In the section 3.4, we present many approaches that have been proposed to attempt to recover the semantics of tables by enriching them with KGs, which have aimed to address the challenges of data quality, table usability, discoverability, and data understandability. We therefore describe the STA (Semantic Table Annotation) or STI (Semantic Table Interpretation) workflow and its tasks for assigning semantic tags from knowledge graphs (KGs) (e.g., Wikidata, DBpedia) to data elements.
Before conducting, in section 3.4.1, we present a methodology to select and identify the work related to the different tasks and methods that have been proposed so far to STA and can answer the research question that were presented in Chapter 5, Section 5.3.1. We selected 32 from among the articles selected in the SemTab competition(2019, 2020, and 2021) and the most cited articles in this area. In the next section 3.4.2, we describe in details the semantic annotation approaches that have been created to facilitate the deployment of open data on the web or their semantic interpretation by users. Next, we present the comparison of the SemTab contest results in table 3.4 and show the best tool in this competition for each round. Based on the results obtained, we show what we decided to choose as methods and integrated into our contributions in the chapter 5. Besides, we explore other existing tools that offer many possibilities to transform and create data transformation pipelines.
At the end, in section 3.6, we synthesize the overview of the selected approaches and tools that met certain criteria to indicate their characteristics, strengths and weaknesses. We present the approaches and tools reviewed according to the following criteria presented in the table 3.4, such as, learning techniques, annotation method, data pre-processing capability, subject-column (the main column) detection capability, semantic annotation, data repair capability, data augmentation capability, technologies based on, KGs or ontologies used, graph generation capability, export capability, interoperability capability with open data portals, and working and availability of these tools. Despite all their achievements , these proposed approaches are not capable to assist users in overcoming data quality issues and none of the

tools are both able of being interoperable with open data portals, automatically annotating data, and generating graphs from the annotated data. Sadly, in the R&D community, there is a significant lack of work community there is a significant lack of work to support these tools by combining appropriate functionalities and improve the STA workflow.

3.2 A review of the best methodologies for using and publishing Linked Open Data

Data publishers have been increasingly publishing content as **Linked Data** on the Web. They are publishing data that can be used for many purposes, such as for consuming or integrating into their applications, or to producing more informed business decisions. Hence, this process includes **guidelines** (i.e., best practices) for **publishing**, **sharing**, and **linking data** on the Web. Based on the data quality barriers, Open Data formats and the challenges of the Linked Open Data presented in the previous chapter 2.4, such data have been published on the web in various formats, for example, PDF, CSV; spreadsheets, tables embedded in documents and many other forms of raw text. These records are usually linked using HTML page links. Therefore, it is not easy to access, search or reuse this data automatically. **Linked Open Data** LOD consists of a set of practices for publishing data on the Web using international standards of the World Wide Web Consortium (W3C) [29, 116, 191]. It adopts a generic abstract data model to describe resources, called Resource Description Framework (RDF). We presented this data model in more detail in Chapter 2.

In the following section, we use the systematic mapping method to identify work on methodologies for publishing Linked Open Data.

3.2.1 Research Methodology

In this section, we employed the systematic mapping method [153], to identify research related to a specific topic to answer a broad question, essentially exploratory (e.g. How to x ?), preserving the reproducibility of the research. The systematic mapping method was composed of two main steps:

In the first two publications in [2019] and [2020], we conducted a literature review on methodologies to publish structured Linked Data in an e-Government statistical data context. Most of the publications found are dated from [2010-2022]. From these publications, an additional selection was made based on their relevance to our research, leaving 18 academic articles, web pages that were looked at more thoroughly. The selected papers were then used to collect appropriate methodologies, framework and tools to ensure better data quality, to publish and visualize related Open Government (Statistics) Data allowing stakeholders to be aware of the knowledge retrieved to improve their decision-making.

In the third publication in [2022-2023], we led a literature review on user assistance to publish structured Linked Data in a Statistical Data context. Most of the publications found are dated from 2010-2022. We found that 80% of the articles were

identical to the ones I selected first. In this step, we used the StArt tool [58] to remove duplicate papers. This tool was used to support the execution of the Systematic Literature Review (SLR) method. Moreover, we have decided to involve the users in the area of Open Data. We aimed to help stakeholders meet the challenges and resolve known issues, methodologies and best practice guidelines for publishing other documents were developed. Thus, we created an online pre-survey [1] to collect feedbacks from data analysts and data scientists who used Open Data (See Annexes B). The survey was pre-tested with three users and then shared via the following communication channels: UNAMUR mailbox, Facebook, LinkedIn, and at an event that talked about the context of open data. In total, we have engaged 35 participants who completed the pre-survey. In addition, we have watched many online reports and interviews on OGD to enhance our knowledge and can be in line with the field of LOD and meet the needs of users and data publishers. The literature review and expert users' feedback left 29 academic articles, web pages, and policy reports that were examined in more detail.

The combination of literature review, our knowledge, and expert users feedback was used to answer the research questions and improve the features of the first contribution, which aimed to increase the quality of Open Data that becomes explicitly available and accessible to data consumers, to provide guidelines to Open Data publishers and to address other barriers. The systematic mapping involved three steps: (1) defining search questions, (2) searching for primary studies, and (3) selecting articles for inclusion and exclusion.

Research questions

The research questions identified in this work are intended to gather information on how to integrate and analyze Linked Open Government data and how to effectively publish Linked Open Data, both in terms of the steps involved and the tools developed to achieve this. We believe this is an important contribution to the scientific and practitioner community to describe what has been done and what shortcomings need to be addressed to systematically analyze and publish LOD (Linked Open Data). By this methodology, we have sought to find the best methods for publishing Open Data that can be used or recommended and any type of quality processes involved in the methodologies to ensure data quality. Thus, we defined the following research questions:

(i) *RQ1. "How can we supply a standardized solution to ease the use and publication of Linked Open Government Statistical Data?"*

(ii) *RQ2. "How can we design an approach and tool that supports semantic interpretation of Open Government Data OGD?"*

The responses to these questions provide an overview of the relevant literature, with important steps to suggest a good methodological framework for publishing LOGD.

[1] https://tinyurl.com/5n7ejrv4

Search strategy

The following datasets were used for this systematic mapping because they are the most important repositories in the following areas topics involving computing: Springer Link, ACM Digital Library, IEEE Explore, Science Direct, ISI Web of Knowledge and Scopus. The keywords used to cover the research questions were process, methodology, publication, integration, users, stakeholders and related open government data, and their synonyms were considered in developing the search string 3.1.

```
(methodology OR process OR pipeline OR guideline OR best practices OR framework) (
    ↪ publishing OR publication OR integrating OR opening)
(linked government data OR linked open government data OR linked open statistical data OR
    ↪ government open linked data OR (open
government AND (linked data OR linked open data) AND (users OR stakeholders)))
```

Listing 3.1: Search query.

Study selection

The selection of studies should mirror the primary work to recognize the different types of methods used to publish linked open government data. To this end, we developed the following criteria:

- Inclusion criteria
  - The main contribution of the study is a process of integrating and publishing Linked Data in Statistical (Governmental) contexts;
  - The study is based on user assistance to publish structured Linked Data;
  - The study is from a peer-reviewed source
  - The study is in English language;
- Exclusion criteria
  - The study does not present a process for publishing LOD or LOGD;
  - The study focuses on a single step in the process;
  - The study does not include user requirements;
  - The study does not focus on linked data, but on open data in general.

Figure. 3.1 shows the methodology used to select the final studies.

3.2.2 Proposals for Linked Open Data

In the present section, we discuss related works on three aspects: (i) Linked Open Data (LOD) technology, (ii) Linked Open Government Data (LOGD) [2], and (iii) Linked Open Statistical Data (LOSD).

In this way, through Semantic Web technologies, OGD (Open Government Data) has become part of the Web where it can be connected to other data that provide documentation, and additional context. Numerous works have confirmed that the real value of OD is enhanced by the application of Semantic Web technologies to link data and provide unexpected and unexplored information in different domains

[2] https://www.w3.org/2011/gld/wiki/GLD_Life_cycle

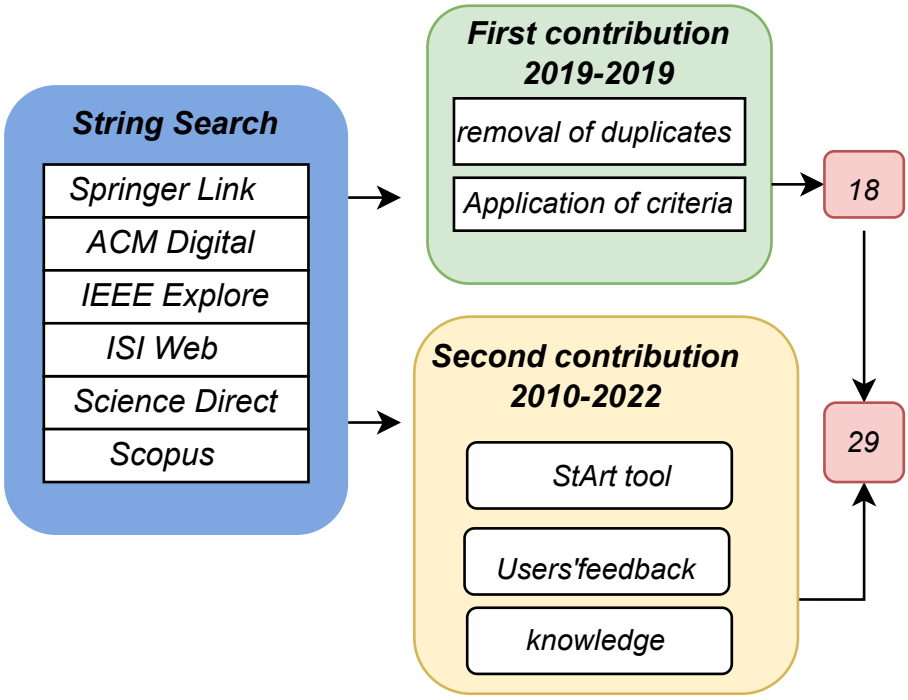


Figure 3.1: Final Study Selection Process.

and problem areas [11, 93, 121, 157]. By adopting Semantic Web methodologies, governments can publish raw data instead of spending time designing data access APIs for specific purposes that assume the needs of citizens, data analysts, data publishers, and other stakeholders.

On the one hand, previous researches on the creation and publication of Linked Data, which is an intensive engineering process requiring considerable effort to achieve a high level of quality. Existing general guidelines are not always sufficient to make the processes reproducible [157, 167]. Most approaches have agreed on a set of basic phases, e.g. identify model, and publish; and there is no standardized LOD life-cycle. The current life-cycles mainly focus on the supply of open data, such as the identification and selection of LOGD, the Modelling of LOGD, and the publishing and linking of data. Several studies presented methodologies and the life-cycles of Open Data initiatives but not linked data in particular, and authors are not specifying what are the criteria for selecting primary studies of methodologies [180]. Different steps in the linked Data life-cycle [16] was depicted in Figure. 2.7 in Chapter. 2. Several challenges exist, in Linked Open Data context, in terms of the different data phases: Extraction, Transformation, Enrichment, Storage, Interlinking, Exploration and Publishing aiming at an efficient exploitation of such data [11, 29, 65]. Information that is represented in an unstructured form, or in other structured or semi-structured representation formalisms, needs to be mapped to the RDF data model (Extraction and Transformation). Mechanisms must be put in place to efficiently store, index

and query this RDF data, once there is a critical mass of RDF data (storage). Links between these different information resources must be established when different data publishers provide information about the same or related entities (Linking). This deficiency can be addressed by data enrichment approaches with higher-level structures to be able to aggregate and query the data more efficiently (Enrichment). Last but not least, users must be able to browse, search and explore the structural information available on the Data Web in a fast and user-friendly way (Search, Browsing, and Exploration). The proposed works have separately deal with these steps.

For instance, F. Benedetti et al. [23] have suggested an approach for automatic extraction of relevant knowledge sources from LOD sources and the creation of a set of indexes that enhance the description of the dataset. These indexes collect statistical information about the size and complexity of the dataset (e.g., the number of instances), but also describe all instantiated classes and the properties between them, providing the user with a synthetic view of the source LOD. Freitas et al. in [69] present a survey of existing approaches tackling the heterogeneity of web-based datasets for searching and querying linked data.

Radulovic et al. in [157] have presented a set of guidelines for generating and publishing Linked Data in the context of energy consumption in buildings (one aspect of Building Information Models). These guidelines offer a comprehensive description of the tasks to perform, including a list of steps, tools that help in achieving the task, various alternatives for performing the task, and best practices and recommendations. Figure 3.2 shows an overview of the generation process; consecutive tasks and its inputs are represented with full lines, while the inputs from non-consecutive tasks are represented with discontinuous lines. Outputs that are final resources are represented with double lines. The Linked Data generation process consists of eight tasks. After a data source that will be transformed into Linked Data is selected and access to that data source is obtained, the license has to be analyzed in order to determine the terms of use. Next, the data source is analyzed in detail, and a resource naming strategy is specified. Afterward, an ontology for describing the data is developed, and the data is transformed into the RDF format. Finally, generated data are linked to data from other Linked Data data sets. These guidelines help both private and public organizations that work with data about energy consumption in buildings to generate Linked Data from already-existing data and in publishing the generated data. This work has a specific context and is not easy for non-expert users (researchers and practitioners of energy consumption in buildings) who are not in the business of Linked Data technologies.

Sebbaq et al. in [167] have proposed a unified data model that serves as a basis for the design and implementation of a Linked Data repository. This approach is based on semantic web technologies: Linked Data and Ontology. Moreover, this model is enriched with metadata related to the pedagogical aspect of MOOCs using Bloom's Taxonomy as a theoretical foundation. The Linked Data repository gives semantically rich descriptive metadata extracted from different MOOC providers which can be accessed via a *SPARQL endpoint*. However, the limitation of this work is that the classification remains manual, which is time consuming given its large-

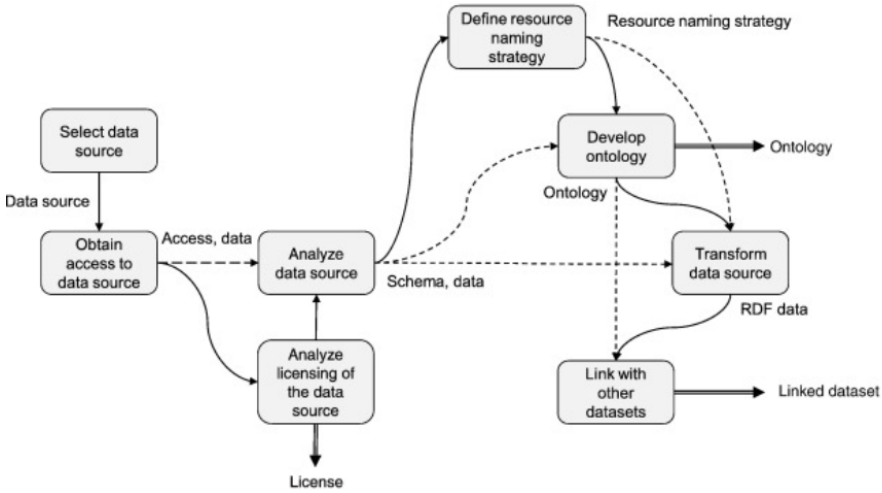


Figure 3.2: Tasks of the Linked Data generation process by [157].

scale aspect. Hidalgo-Delgado et al. [77] have proposed methodological guidelines to formalize the publishing process of LOD taking into account data quality. It is based in [186] for publishing library data as Linked Data. These methodological guidelines consist of five activities: (1) data extraction, (2) data pre-processing, (3) data modelling, (4) data publishing, and (5) data exploitation and follows an iterative and incremental approach. It does not claim to be exhaustive, nor applicable to all data domains. The Figure. 3.3 shows the proposed methodological guidelines.

While, Villazón-Terrazas et al. [186] have Presented a methodological guidelines for publishing Government Linked Data. It is covering all the necessary data steps, such as specification, modelling, generation, publication, and exploitation. This methodological is based on the continuous improvement and extension of the Government Linked Data resulted from performing several iterations. Publishing CSV data as a Linked Open Data (LOD) on the Web [3] is an active field of research. However, there are very few effective applications have been developed with this purpose. For instance, Sarkar et al. [164] outlined a framework for generating RDF data from the existing university database and publishing it as Linked Open Data (LOD) on the web. However, the main components of the proposed framework are legacy information (contains RDBMS), RDF triples generation method (generated RDFS using D2R tool) and provenance control (adding metadata). In light of this, Kumar Sharma et al [168] suggested an ontology-based approach to convert CSV data into RDF datasets. The author has focused on extracting concepts and relationships from CSV files. This ontology-based approach is implemented to contain the CSV files and describe the RDF datasets. The CSV data is extracted into RDF using the suggested method. On the another side, the W3C recommendation document [177] defines the process and rules for converting tabular data into RDF. However, most of the

[3] https://www.w3.org/TR/csvw-ucr/

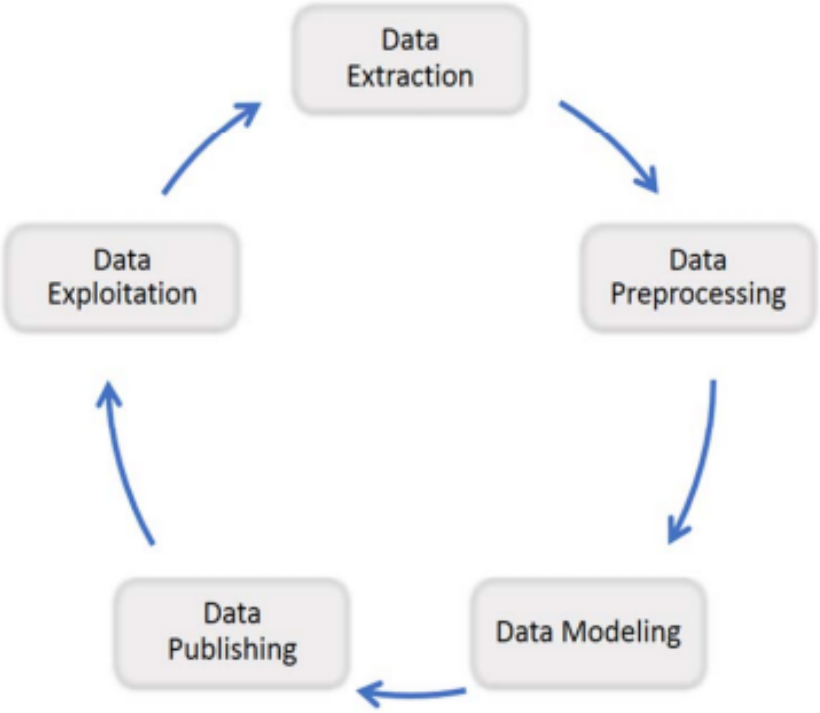


Figure 3.3: Methodological guidelines for publishing library data by [77].

previous work was not expanded by the W3C Recommendation document. On the other hand, Mahmud.SM et al. in [121] have developed a Semantic approach that can effectively convert CSV data into RDF Data with rich semantics and release RDF data on the web using LOD principles. They have used Semantic Web resources and W3C recommendation rules in automatic data publishing method, which enables distributed system for scalability. In such context, [122] have designed a parsing algorithm and development techniques to demonstrate the annotated tabular data model (column, row, and cell). This tool (CSV-ANNOTATION) aimed to generate annotated tables from CSV file. However, annotated table for CSV provides possibilities for data publishers or stakeholders to refer data validating, converting, displaying and inputting by following the Semantic Web standard. It is a desktop application for annotating tabular data (CSV) to help users convert semantic data (tabular-CSV). The architecture and process flow of the prototype are depicted in Figure. 3.4.

An experiment in this work was performed to observe and compare the temporal efficiency of annotating four CSV files. However, this evaluation did not consider the quality of the annotated data and the source implementation of this prototype is not available. Whereas, some works have used the various data steps in different domains such as recommender systems [138] to propose solutions that leverage the data available within LOD datasets in order to recommend specific data to

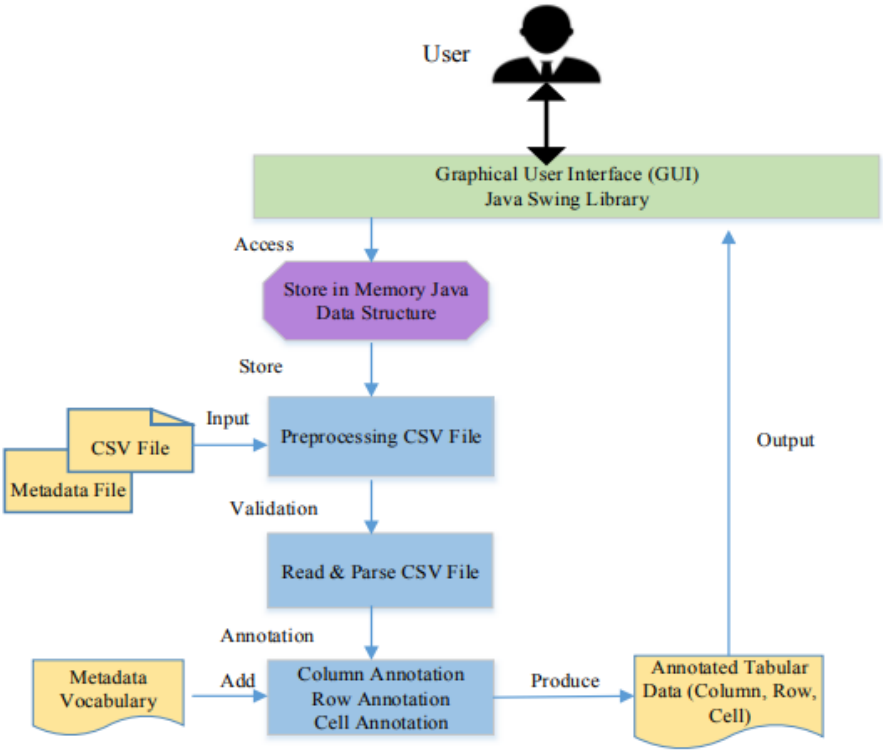


Figure 3.4: Architecture of the CSV-ANNOTATION by [77].

end-users, and recommend systems for smart cities [156], or data indexing [18] to develop lookup indexes over Semantic web resources. Unlike our work, we aimed to propose a framework for government data (See Chapter 4).

On the other hand, several proposals have more focused on Linked Open Government Data (LOGD) and Linked Open Statistical Data (LOSD) in order to make such important data more transparent and fully exploited to public institutions as well as stakeholders. Although there are still gaps to be addressed on how to properly meet the supply and demand of OGD. In Section 2.4 of this chapter, benefits and drawbacks about LOGD are outlined. Methodological guidelines for publishing Government Linked Data are also presented in [185, 186] covering all the necessary data steps. Moreover, The proposed publishing framework is also still at a high level of abstraction [13]. In such context, diverse works have adopted a pipeline methodology to define their approaches. For instance, Maali et al. [118] have suggested an approach to convert raw data into high-quality LOD. We also consider data quality in our work but with all the data phases forming its life-cycle. This pipeline is based on Google Refine (Open Refine), a data workbench that has powerful capabilities for data massaging and tidying up. All the functionalities involved in this pipline are available through a single workbench and the steps are independent from each other. Moreover, They have been applied this pipline to a local government catalog

in Ireland, the Fingal County Catalogue. However this work did not choose a proper RDF model for data modeling, which is an important aspect to be addressed. Figure. 3.5 outline the Linked Data publishing pipeline.

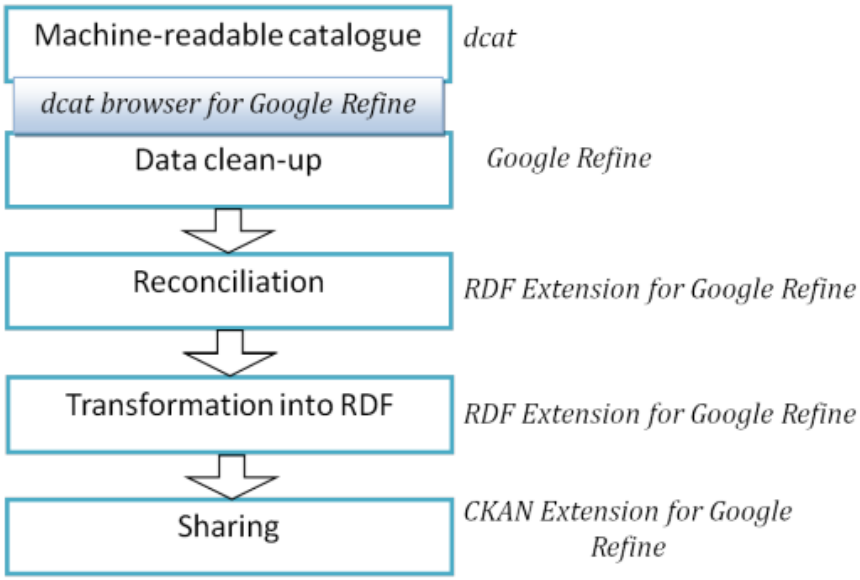


Figure 3.5: Linked Data publishing pipeline by [118].

Moreover, Beluzo et al. [22] have explored and applied “Re-public” generic process that aims to publish OGD in RDF format. It has been proposed to improve data connectivity and availability to consumption both can be achieved by adding a semantic layer to data. This approach is composed of the following workflow for this process: 1) definition of the first dataset context to be integrated in CSV format by using the manual method; 2) creation an ontology to describe the data from this datasetby using "Protégé" tool[4]; 3) CSV to RDF transformation from the ontology mentioned in the previous step by using RDFLib[5] and Panda[6]. After this process, they have added two other steps for the validation of the process, which are: 4) link the dataset(s) described in RDF (LOD) with the common element(s) with the dataset from step 1; and 5) use an endpoint ("SparqlWrapper" library[7]) to perform aggregate queries between the two sets datasets(the original dataset and the LOD). Figure 3.6 illustrates this workflow.

They have applied this process on the government budget field by using Constitutional Transfers to Cities dataset and DBpedia Open Data. The assessment results indicated that this approach made it possible to use semantic web tools to publish

[4]https://protege.stanford.edu/
[5]https://rdflib.readthedocs.io/en/stable/
[6]https://pandas.pydata.org/
[7]https://sparqlwrapper.readthedocs.io/en/latest/

Figure 3.6: Development Workflow by [22].

the original CSV files in a new format. However, the infrastructure for storing, as well as the processing and availability architectures its cost.

AlSukhayri et al. in [11] have proposed Linked Open Saudi Government Open Data Framework (LOSGDF) for producing LOD from different data sources such as CSV files, Excel sheets, and structured documents and linking them to other existing open datasets. This framework consists of four modules for collecting, processing, generating RDF datasets, interlinking these datasets with other open datasets and storing them into a triple store server. They have used DBpedia khnowldge graph, because, it being one of the biggest datasets in the LOD Cloud is considered a hub of this Web of Data. DBpedia datasets are extracted from Wikipedia articles [14] and have a bidirectional link to many datasets in the LOD Cloud. However, this framework is not user-friendly for non-expert users in the Semantic Web area and the source of the implementation code is not available.

On the other side, a major part of the open government data concerns statistics data [13, 36, 197, 200]. These data are arranged in a multidimensional manner, where a measured variable (e.g., unemployment or workers) is represented in terms of dimensions (e.g., geography and time). Statistics data consist of primary material for added-values services and products, increasing government transparency, contributing to economic growth, and providing social value to citizens [89, 92]. In such context, multiple Methodological guidelines for publishing Statistical Linked Data are also presented in [13, 54, 92, 94].Yet statistical data plateforms are not a mature technology yet [126]. While E. kalampokis [92] have focused on benefits and challenges of Open Statistics. They have presented the heterogeneity of Open data as a big issue to the reuse of Open Statistics Data. Moreover, they described multidimensional data, which could be reused, and extended methods from several areas such as open data, Statistics, Data Warehouses, and the Semantic Web. In such context, E. Tambouris [175] used such methodology to present multidimensional

linked open data publication and reuse. They have proposed a process that can be used for publishing, expanding and consuming statistical open data and that can improve the quality of raw data to create value. By going through this process, data producers can transform their raw data into RDF data cubes. In addition, they have to link their concepts (e.g., measures) with the same concepts in other OGD portals. In addition, data consumers can more easily find, merge, and analyze their data. Meanwhile, the use of this process still raises some technical and knowledge problems. [94] have proposed a prototype is based on the Linked Data paradigm, which aims to create a layer of data interoperability on the Web through the use of the RDF data model and the creation of typed links between distributed data sources [28]. In additional, the implementation was run on a hypothetical use case scenario regarding real-world DMOs regarding public agencies and schools in Greece. Besides,it was based on Semantic Web tools such as the D2R server, which aims to map relational data to RDF data, the SILK framework [187], which identifies and links different datasets, the Sig.ma semantic web browser [178], and the RDFa specification for expressing structured data in HTML pages. [176] have proposed an architecture and implementation that allows developing custom visualization and analysis tools without knowledge of LOD technologies. This work was intended to facilitate the publication and reuse of Linked Open Statistical Data to improve decision-making and transparency and thus support public administration. Despite these works, there still some technical and economic development challenges for exploiting the full potential of LOGD and especially reusing related statistical data. Although, some relevant policies must be established and implemented.

Several challenges still remain, in terms of processing the various data phases of the publishing process, which are: extraction data which is important to consider relevant data, in the sense that they have a high request in society, reflect important aspects of public administration and that present quality indicators (completeness, accuracy, and timeliness) [131]. In [121, 163] authors did not consider this phase in their workflow. The data transformation phase involves the cleaning and design of URIs that define the unique and persistent identification of data resources, which many works have not considered [63, 66, 102, 121]. In contrast, research studies have included defined vocabularies to reuse existing vocabularies or build new ones, depending on the context of the data [5,54,92,102,119,125]. All these works considered modelling the "raw" data into a linked data representation or serialization (RDF) and preferred to publish their linked data in a simpler form, unlike [5,8,11,40,56,108,118], which aimed to link and enriching the data with external data sources (knowledge base schema) before publishing it on the web. In a broad outline, the primary purpose of creating linked datasets is to promote the reuse of government information, to provide public applications, and to encourage commercial reuse of the data, since all studies emphasize the importance of creating applications from the linked open data. Furthermore, the publishing phase can be used to promote it and make it visible to search engines or to engage with the community of users and consumers.

Despite these investigations, a big challenges remain in providing the public with the opportunity to engage broadly with government, whereby, after the publication of linked data, the government must receive feedback from data consumers. In

addition, it will help the government develop equitable policies and better decisions. While, [5, 6, 108, 120, 192] considered the importance of engaging with a data community to receive their feedback, which aim to enhance the quality of the published and their applications.

On the other side, the results of the evaluation of a LOGD and LOSD platforms or frameworks are important to show their functionality and usability. Hence, it is therefore important to use strategies to assess the quality of data published on the Web of Data (Quality Analysis). In this context, several works have been proposed to evaluate open government data [25, 120, 126, 131, 170]. In such context, Matheus et al. in [126, 131] purposed a model for evaluation Open Statistical Data and Open Government Data by using using a list of many requirements derived from the ISO 25010:2010 standard for Systems and Software Quality Requirements and Evaluation. Based on this list of requirements, a questionnaire was developed which was used to evaluate six cases which makes use of the same platform for processing LOSD. Moreover, Minotas et al. in [131, 184] have focused on the open data platforms and datasets, which provided a measurement framework to quantitatively assess the quality of open government data. They have tested the adequacy of the framework by applying it to compare two cases of Italian OGD datasets: centralized, i.e., controlled at the national level with data aggregated from several regional sources and in the presence of a data quality process, and decentralized, without any type of orchestration and likely of lower quality, selecting a dataset from municipalities in the same areas.

3.3 Synthesis of approaches to using and publishing Linked Open Data

We have conducted a systematic mapping in the literature to synthesize different methodologies and approaches **for using and publishing LOD** designed to solve problems in a particular field and to address some of the barriers to Open Data (Section 2.3). We have undertaken to synthesize these works around the following themes: common phases (Steps), associated tasks, quality assessment validations, and evaluation of methodology.

According to the synthesis phases and studies used for the identified methodologies in the table 3.1, These studies start **to select datasets to be published as linked data** and to exploit existing or new open datasets and identify their structure. Then some researches consider unifying and cleaning the data, to remove inconsistencies, typos, to fix errors, remove duplicates, and get the data ready to prepare for data transformations. One of the key features of linked data is the unique and persistent identification of data resources. Any type of information represented by the data (digital document, person, company, organization, location, etc.) must be identified by a URI. By requesting this URI, the web server returns information about the thing identified by the URI (called dereferencing). Therefore, one must also consider the careful design of URIs. Since different data sources may present the same information in different forms, it is necessary to achieve a general consensus

on how to represent this data. Defining vocabularies [8] is one step, analyzing when to reuse existing vocabularies or to create new ones, depending on the context of the data. Publishers should outline what it takes to study how others use and represent these same concepts and reuse this conceptualization to facilitate data merging and reuse, using existing authorized and widely used vocabularies to describe common data types. This includes creating vocabularies for specific concepts and making them available on the Web so that other users can reference them to understand the data or reuse those concepts. Next, the vocabularies are carefully mapped to the data. This modeling step is necessary to create the RDF structure to be represented from the original data. Some approaches adopt a field-to-field mapping (such as direct mapping[9]), where a column of the initial tabular data is transformed into an RDF property, and the files or tables are transformed into ontology classes in the resulting graph. Some searches use this link to enrich the dataset, by importing data from external sources and enhancing the knowledge base schema such that data enrichment can be performed before or after conversion to RDF data. Once the original data, metadata and mappings to vocabularies are obtained, the step of transforming the "raw" data into a linked data representation or serialization (RDF) is performed. There are different serialization formats, such as RDF/XML, Notation 3 (N3), Turtle, N-Triples or HTML with semantic tags (RDFa). The serialization format chosen depends on the purpose of the conversion process (file transfer, readability, visualization, upload to a triplestore). Some actions performed during this phase can become provenance metadata that can also be added to the dataset, such as the tool used, the steps of the conversion, the parameters used, etc., so that third-party users can trace how it was generated. A final task in data conversion is RDF cleaning, sometimes considered a separate step. It can discover and activate the correction before deploying the graph to production, transforming it into a higher quality data graph. The great advantage of linked data is the ability to reuse data from external data sources,to discover additional information, to merge multiple data to create complete information, and to bluid a Web of Data. However, it is necessary to identify and select the external sources with which to link the data, such as Dbpedia or Wikidata ; specify the method of accessing the target datasets (RDF dump, SPARQL endpoint); specify how to relate the source and target resources to be linked; create the links; and validate the links created. The generation of additional information or the creation of new information can be stored in a different file or triplestore. The publication phase can be exploited by making it known and discoverable through a simple and clear visual representation and by engaging with the user and consumer community. Some studies emphasize the importance of creating APIs for data and applications with data to help the community be more aware of them. Given that the primary purpose of creating linked datasets is to promote the reuse of government information, to enable transparency, to provide public applications, and to encourage commercial reuse of data, it is essential to create APIs for the data and public applications must be developed and made available as generic or domain-specific

[8] In most of the studies the following terms are used interchangeably: vocabularies, taxonomies and ontologies.

[9] https://www.w3.org/TR/rdb-direct-mapping/

applications. After the publication of linked data, the government must receive feedback from data consumers, engaging with a data community. This community is typically composed of a wide range of individuals and entities with different skill sets (researchers, analysts, citizens, intermediaries, other government agencies, etc.). The data community contributes to the process by [108]: (i) providing feedback on the data to be published; (ii) contributing to the quality of the data; (iii) collaborating with other members to create solutions from the data. To this end, the studies identify the evaluation methods adopted which can be considered qualitative and quantitative measures (functional requirements) [68], a prototype, an illustrative scenario or a case study and can be arguments and logical reasoning, which are presented in Table 3.2. Thus, we have derived the work of Design Science Research to classify the different evaluation methods applied in the selected studies.

According to the evaluation method used in these researches in Table 3.2, more than half of these studies did not provide an empirical evaluation in the paper, simply providing a list of metrics and recommendations, most often justified by the basic principles of linked data and the 5-star schema. Most papers have given illustrative scenarios of the application of the methodology and practice-driven. The actual validations were varied, ranging from visualizing Open Statistics Data to government data lots. According to this evaluation framework, illustrative scenarios differ from case studies. Besides, some works focus on how a tool could sustain the process and detail its capabilities and features. In addition, some studies have used measures to quantitatively assess the quality of open government data. Some of the work builds on the authors' experience in previous open government data projects or perhaps on validations performed at other stages of a research project that could not be gathered because they were not included in the narrative.

However, in the chapter 4, we aim at covering all the phases from data extraction to data publishing and visualization and the engagement the users community with government. We propose a pipeline-based framework to publish Open Government Data (OGD). Our framework aims at shifting the burden of processing, converting, and publishing LOGD to data consumers rather than the government. Secondly, we extend the data transformation step with an automated approach to the modeling step, which was based on the set of requirements that are participation feedback. These contributions are intended to strengthen the quality of open data and improve transparency of governance through the publication OGDs. Furthermore, We also assess and validate in more details our approaches in the chapter 7 by using certain evaluation methods presented in the table above 3.2.

3.4 The principals of Semantic Tabular Annotation in Open Data

In recent years, a vast amount of information is provided on the Web and the Open Data portals in unstructured text, semi-structured data, and in the form of tables (Tabular Data (TD)) [11, 19, 22, 90]. They often contain high-value data, but the meaning of each table is rarely explicit from the table itself. Then, it can be challenging to use (usability) due to data quality issues, such as missing, incomplete, or obscured metadata, ambiguities in table cells, missing cell values, and missing

significant columns. In addition, header rows exist in few cases, and even when they do, attribute names are usually useless. Tables potentially contain rich semantic information that must also be interpreted. Unfortunately, many data producers also are not familiar with LOD technologies and are not willing to invest time to integrate their data with KGs (semantic annotation).

Therefore, several approaches have been proposed to attempt to recover the semantics of tables by enriching them with additional annotations [11, 46, 142, 145, 146], which sought to tackle some of the above issues. Among the challenges related to **data quality** and **usability** of tables are the **discoverability** and **understandability** of data [13, 26, 51, 103]. Indeed, before being able to use the data, consumers must be able to find data relevant to their needs (discoverability) [103, 120, 128] . Once they discover the relevant data, they must be able to understand the metadata and content of the data to exploit it, such as for data integration, data cleaning, data mining, machine learning and knowledge discovery tasks (understanding) [25,51,91]. Beno et al. in [26], Bizer et al. in [28], Křemen and Nečaský in [103], and others [25, 108, 120, 135, 142] have demonstrated the use of Linked Open Data (LOD) and Semantic Annotation are efficient in solving the mentioned challenges.

STA (Semantic Table Annotation), is the process of assigning semantic tags from knowledge graphs (KGs) (e.g., Wikidata [188] and DBpedia [14]) to the data columns elements. It mainly consists of the following tasks (see Figures 5.3 and 5.8 in Chapter 5): (1) **Column Type Annotation** (**CTA**), (2) **Cell Entity Annotation** (**CEA**) which involves mapping each cell to an entity in KGs and (3) **Column Property Annotation** (**CPA**). Although the benefits of semantically annotated data have been widely recognized, there are still a vast number of datasets without any semantic annotation that are published every day on open data portals, probably because semantic enrichment to data or adding meaningful semantic information to data is a laborious, error-prone and difficult task for editors ([25]) and also because the majority of existing tools are not able to automatically automate the process and be both interoperable with open data portals. On the one hand, some approaches have focused on data annotation (semantic annotation), which has been shown to be effective in solving the problem of spelling errors and missing or incomplete metadata [19, 47, 90, 91, 142, 151]. On the other hand, some approaches are not able to repair data, which aims to deal with the problem of missing cell data (values), and some other approaches did both data annotation and data augmentation, which dynamically add meaningful columns and corresponding cell values to the data ([151], OpenRefine [10]).

In the next section, we describe our methodology using the systematic mapping method to identify the work related to the different tasks and methods that have been proposed so far to perform Semantic Table Annotation (STA) or Semantic Table Interpretation (STI).

[10] https://openrefine.org/

3.4.1 Research Methodology

The methodology used to answer the research questions that were presented in Chapter 5. The aim was to identify, through existing tools and tasks, a list of requirements presented in the section that should be included in a fully automated approach that can easily solve the problems and barriers related to open data (Section 2.3), in particular, the quality of information (data), the absence of knowledge, the lack of transparency and the reuse of data, and **that can be easily interoperable with any open data portal**.

We observed that most of the work was published between 2000 and 2021. Consequently, our work covered articles published between 2010 and 2021. To collect relevant articles, we used "Google Scholar", "Semantic Scholar", and "Science Direct".

We first started with a combination of keywords from two lists: a list of terms reflecting the tasks we are interested in "semantic annotation", "annotation", "semantic interpretation", "knowledge graph matching", "semantic matching", "semantic labelling", "type prediction", "entity linking", "knowledge graph mapping","relation extraction" and another list of terms covering the field "web table", "tabular data", "structured data", "table annotation" . This list of terms was manually created when reading relevant articles such as those in the SemTab challenge [11]. For example, an appropriate combination is "web table semantic annotation". For each query, we fetched the top 11 most cited articles among the relevant results. We read the abstract of each retrieved article extracted and judge whether the article is relevant to STA (e.g., presenting a method to match a table item with a given KG, introducing a new dataset) or not. Our research was not limited to the titles or abstracts, we also searched the full text of the articles. In addition, we have also collected 21 articles from the SemTab challenge, which is the most relevant competition for this area. We drew on [113] manually categorized each article into five categories, such as Annotations systems, Ontologies, Datasets, Table processing, and NLP (Figure 3.7). Note that some articles may address more than one research topic. For example, [112] provided both a dataset and a method for performing STI or (STA).

3.4.2 Related Works of STA

It can be difficult to understand and exploit Open Data to its full potential. Some tools have therefore been created to make it easier for users to deploy open data on the web or to semantically interpret it. In this section, semantic annotation tools will be described and compared. If there are several versions of these approaches, we use the last two existing ones. For the selection of these approaches, we have decided to use the tools that arrived in the top 3 of the SemTab 2019 and 2020 challenge that took place in 2021 (MTab [142, 145, 147], Dagobah [84, 114] as well as MantisTable [46, 47], LinkingPark [41], MAGIC [151], and JenTab [4], that come back with good results. In addition, other popular tools such as OpenRefine, SemanticBot [135], Odalic [100], TableMiner+ [196] and DataGraft [160] have been proposed to annotate a structured

[11] https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

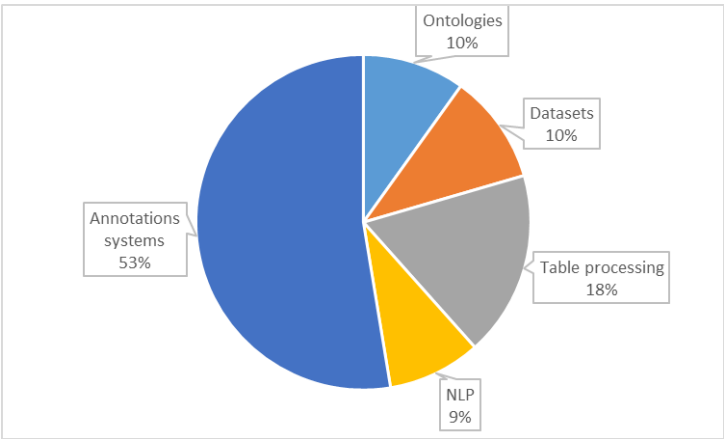


Figure 3.7: Distribution of the topics of the papers selected.

dataset. Besides, an annotation evaluation tool also be presented. The purpose of the latter is to evaluate the annotations of semantic annotation tools.

MTab is an automatic semantic annotation tool. In the SemTab 2019, 2020 and 2021 challenges, it got the best results. In this section, we talk about the two versions of MTab, the 2019 version is based on Knowledge (KG) DBpedia and the 2020 and 2021 versions are based on Knowledge (KG) Wikidata. The proposal of the latter, we explained its algorithm in more detail in the chapter 5.

Like **MTab**, **MantisTable** are an automatic semantic annotation tool. It received good results in the SemTab 2019[12] and 2020[13] challenges. In 2021[14] challenge, MTab [147] and MantisTable [17] did not concentrate on annotated tabular data. MTab [147] in version 2021, focused on entity search, which aimed to address the challenges of finding relevant entities. MantisTable [17] has focused on optimizing the system by providing API endpoints to retrieve data, which can be used with any backend. Unlike the other tools, LinkingPark and JenTab only participated in SemTab 2020. It is therefore not possible to compare with earlier versions. JenTab is a simple semantic annotation tool. In the SemTab 2020 challenge, it was also at the very low end of the rankings. JenTab and LinkingPark did not include a graphical user interface to interact with their proposed novel approaches to semantic table interpretation.

Dagobah is an automatic semantic annotation tool. In the SemTab 2019 challenge, it was at the very low end of the rankings, but for the 2020 challenge, it managed to move up to third place. In this section, we will discuss both versions of Dagobah (2019 and 2020).

Magic only participated in SemTab 2021, it is a system able of both annotating and augmenting a dataset. Unfortunately, it was at the very low of the rankings for the challenge. It is therefore not possible to compare with an earlier version.

[12]https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2019/results.html
[13]https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2020/results.html
[14]https://www.cs.ox.ac.uk/isg/challenges/sem-tab/2021/index.html#results

MTab-DBpedia 2019

We will briefly explain the MTab process because it will be discussed in more detail in Chapter 5. The MTab tool must have fulfilled certain preconditions before implementing the STA principle.

- They have assumed that the KG is complete and accurate.
- They have assumed that the type of a table is a vertical relational table.
- They have assumed that there is no information that is shared between tables.
- They have assumed that the values of the columns have the same entity types.
- They have assumed that the first row of the table is the leading row and that the first cell of the column is the heading.

Once these five preconditions are validated, it is now possible to work out the principles of CEA, CTA and CPA. MTab tool separates all these principles into seven steps.

1.Pre-Treatment, The first step is quite classical and can be found in most STI tools or methods. It is necessary to correct the set of characters that may be missing the correspondence with the entities of the KG. It is therefore necessary to prevent special characters, many spaces, ... To solve this problem, the MTab team used the Ftfy tool [15] which allow to solve all the writing errors and the Fasttext [16] models which allows to predict the language of values in the cells correspond (English, Spanish, French...). Then, the Duckling model [17] is used, which make a prediction on the type of data that is found in each cell (numeric, URL, email...). Finally, SpaCy models [18] are used to predict what type of entity is in each cell. The search services such as DBpedia Lookup, Wikidata Lookup are used to find interesting candidates matching the cell [142].

2.Treatment, The second step allows to estimate the candidate entities that were found by the different search services during the first step. Here, a relevancy score (from 0 to 1) is used to highlight the most relevant entities. The third step deals mainly with columns. It is necessary to be able to distinguish between entity columns and numerical columns. A majority vote is done using the Spacy and Duckling models [19]. If the vote returns a text or an integer tag then the column is type entity otherwise it is numerical. For numeric columns, MTab gathers all the values in a list. This list used in the DBpedia Endpoint [20] to retrieve a set of candidates that can match the type of the numeric column.

For example, in a table containing books, we have dates, so the MTab tool tries to understand what these dates correspond to using the DBpedia Endpoint, it will retrieve an ontology such as "PeriodicalLiterature". For the entity columns, a set of probabilities is aggregated into a score (0 to 1) to give scores to the different entity candidates. The fourth step establishes the relationships between the different candidates (the columns). For this task, there are two types of possible entity-entity or

[15] https://pypi.org/project/ftfy/
[16] https://fasttext.cc/
[17] https://github.com/facebook/duckling
[18] https://spacy.io/usage/models
[19] https://pypi.org/project/duckling/
[20] https://dbpedia.org/sparql

entity-non-entity relationships. For entity-entity relationships, it is sufficient to see if the entities have common relationships using the DBpedia Endpoint (relationships or properties). For entity-non-entity relationships, an estimation is done between the candidate entities and the cell values, which results in entity candidates with paired relationships [142].

3.Post-treatment,

CEA, Step five is a re-estimation of the entity type candidates. A set of probabilities must be considered: (1) The probabilities of the candidate entities given by the search services; (2) The probabilities of the candidate entities given the probabilities of their type; (3) The probabilities between the cell value and the entity using the Levenshtein algorithm [21]; and (4) The probabilities of the candidate entities according to the values of the cells on the same row. With the set of these probabilities, the entities corresponding to each cell can be retrieved.

CPA, Step six corresponds to the selection of the highest probabilities of the candidates to establish their relationship via a majority vote.

CTA, Step seven corresponds to the selection of the highest probabilities of the candidates to establish their type via majority vote.

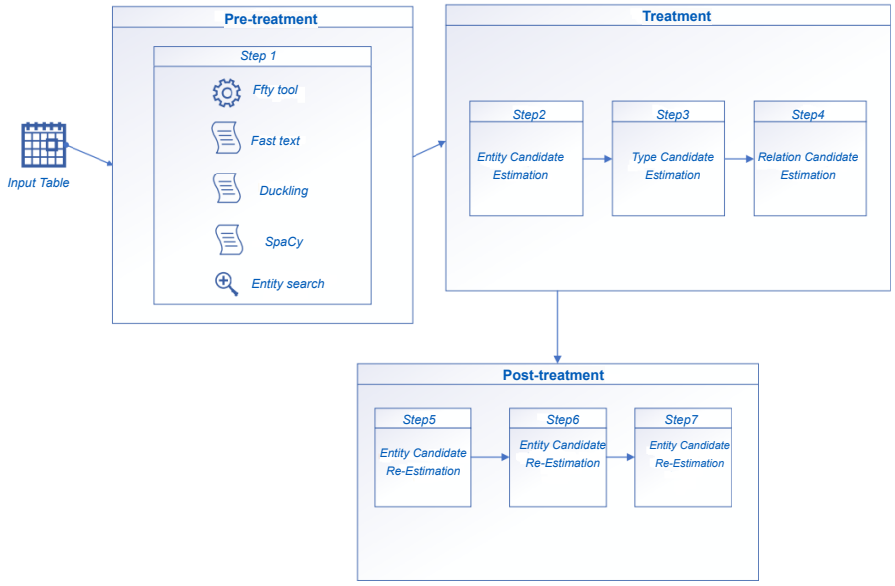


Figure 3.8: Functioning of MTab-DBpedia 2019.

MTab-WikiData (2020)

The MTab tool fulfills the same role as the 2019 tool. Here, the knowledge Base (KB) used is Wikidata [90, 145, 147]. Compared to its 2019 counterpart, the preconditions

[21] https://towardsdatascience.com/text-similarity-w-levenshtein-distance-in-python-2f7478986e75

are almost the same, but there are still some differences. Preconditions 1,2,4 and 5 of the 2019 MTab are found in the 2020 one. Precondition 3 is no more included and is replaced by this one: The tool does not take into account column analysis and it assumes that the first column of the table corresponds to the subject column. At the level of automation of CEA, CTA, CPA tasks, some improvements are also noted: (1) Improved matching performance using historical revisions [22] available on Wikidata; (2) Improved correction of spelling errors/noise in table cells; (3) Reduced number of relevant candidates by assuming a meaningful relationship between table cells. The number of steps also changes, from seven steps to three.

1.Pre-Treatment, In the first step, they retrieve all the information from Wikidata during a given period. Then, they retrieve the revision history in RDF (subject - predicate - object) form; this allows to enrich the Wikidata statements. They use a hash table to retrieve a set of entities containing multilingual labels, and aliases. Finally, they index article statements ("item - property - item").

2.Treatment, The second step consists in retrieving the candidate entities of the cells. To do this, two types of searches are made.

The searching by cell. To search for the entity corresponding to the cell, a use of Levenshtein's algorithm (approximate search) and a score list is used. With that, they can retrieve all the relevant entities related to the cell.

Searching via two cells. To remove any ambiguity in the table cells, a relationship is created between the cells in the same row. The first cell is used in the subject-column (The main column) of the table, the second cell is on the same row as the first cell. All the correspondences between these two cells form lists of responses and only the responses containing valid statements will be kept. The responses provided by the search via two cells are given preference if this search returns nothing, then the results of the cell searches are taken. Unlike MTab 2019, the match between the two cells is done before post-treatment.

3.Post-Treatment, In the third step, the correspondence between the candidate of the subject-column and the other cells located in the same row is done. A computation is produced to check the similarity between the candidates of the subject-column and the candidates of the other columns.

CEA, The candidate with the highest similarity to the cell value is used as the CEA annotation.

CPA, The set of candidate Properties is aggregated and a majority vote is applied to retrieve the CPA annotations.

CTA, The CEA annotations are taken and the set of their type is aggregated and a majority vote is applied to retrieve the CTA annotations.

MantisTable (2019)

The MantisTable tool does not require any preconditions to be met before the STI principles can be implemented [47]. However, there are two points to consider. The first is the fact that the team had to develop and adopt a new way of loading input files, since basically the MantisTable tool only reads JSON files but the SemTab 2019

[22] https://www.wikidata.org/wiki/Wikidata:History_Query_Service

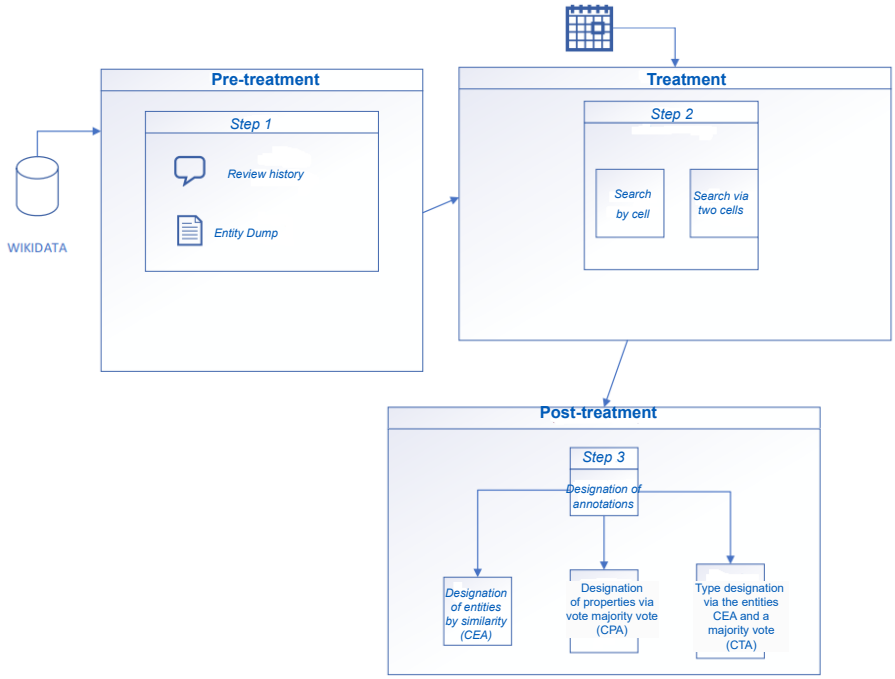


Figure 3.9: Functioning of MTab-WikiData 2020.

challenge only uploaded CSV files. The second point is that the Column Analysis step has been eliminated since the columns for the destinations are already given.

1.Pre-Treatment, This step is quite common in methods/tools using the STI principle. This step consists of data preparation, i.e. transformation of text into lower case, resolution of acronyms/abbreviations, units of measurement, etc. To make this task, the use of regular expressions and Oxford English Dictionary [23] has been applied. Thus, we noticed that this task is very rich compared to the MTab pre-processing, because the MantisTable teams focused on acronyms and abbreviation. For more details on this level, you can see the chapter 5, Section 5.3.1.

2.Treatment, The second step (Column Analysis) is roughly similar to step 3 of the MTtab 2019 tool. It discriminates column types between entity or literal. This is done via 16 regular expressions that allow to define types (address, URL, hexadecimal color code...) [47]. If the column is over a certain limit of occurrence, then the column is defined as a literal otherwise it is an entity. Then, it takes the entity columns and looks for the column that could best represent the subject of the table (the subject column). To achieve this, a calculation is made of the average number of words for each cell, the number of empty cells in the column, the number of cells with a unique value and a distance from the first entity column. The column, which has the highest value after this calculation, is considered as the subject column.

[23] https://www.oed.com/

CEA, The third step (Entity linking) is also used in MTab 2019, but is done differently. Instead of doing cell by cell, the team does linking for all cells in a row to try to link the cells to the KG entities. If there is more than one entity to return, the team use the *Wagner Fischer* distance [47] and take the one with the lowest edit distance. For the literals, they use a simple matching algorithm that is based on different data types (Dates, Numbers and Strings).

CPA, The fourth step (Predicate Annotation) aims at identifying a relationship (predicate) between the subject column and the other columns to understand the general context of the table. Here, it is necessary to take the predicates that were found in the third step and to take the one that appears the most frequently.

CTA, The fifth step (Concept and Datatype Annotation) aims at matching the column headers with semantic elements such as concepts or datatypes. The goal is therefore to retrieve the RDF: type of the candidate entities of the column and then to find out if the links are possible between them. If true, we need to retrieve the RDF: type that appears the most times in the column and that has the most links with the other RDF: type of the column [47].

3.Post-Treatment,

In this step, there is no post-treatment, because there is no re-evaluation or re-estimation at the CEA, CTA and CPA tasks.

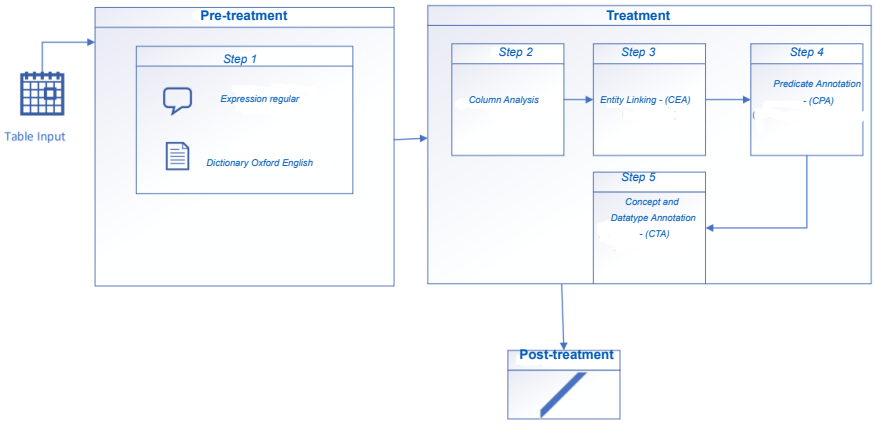


Figure 3.10: Functioning of MantisTable 2019.

MantisTable (2020)

The MantisTable tool serves the same purpose as the 2019 version. Here, the Knowledge Base (KB) used is WikiData [46]. In this version, there are no preconditions used in this version. Unlike MantisTable 2019, it does not use SPARQL queries directly. Instead, it used another open source tool called LamAPI [24], which allows several things: (1) Label matching : It uses elastic search to find the entity that most closely matches the text in the cell; (2) Predicate matching: It allows to return the

[24] https://bitbucket.org/disco_unimib/lamapi

set of predicates between two entities; and (3) Object/concept correspondence: It allows to return the set of objects/concepts linked to these entities (dbo, foaf...). The label correspondence uses JSON files that have been pre-computed. As a search engine, an elastic search supporting "HTTP GZIP" has been implemented for best performance. The complete working of MantisTable 2020 includes 8 steps.

1.Pre-Treatment, This step is similar to the one in 2019: a treatment is applied on each cell of the table to remove special characters, parenthesis, spaces in excess...

2.Treatment, The second step (Column Analysis and Subject Detection) is also very similar to 2019. The goal being to discriminate entities from literals using regular expressions. To find the column-subject, they used scores based on the content of the columns as for the 2019 tool. The third step (Data Retrieval) is not in the 2019 tool. For each cell that has been normalized, they use the label matching from the LamAPI tool. *For example, the data "Jurassic world" is inserted, a list of candidates shows: Q3512046 (" Jurassic World" - Movie); Q18615494 (" Jurassic World" - Comic Strip); ...*

CEA, The fourth step is the CEA step: for each cell (entity), a score will be obtained using the *Levenshtein distance* [25] algorithm. The more closely the normalized text of the cell and the entity is related, the higher the score will be. For literal columns, the treatment is different if the cell is type string, numeric or date. For numeric and date types, a special formula created by the MantisTable team has been implemented. Dates are considered as numeric values in the format "YYYYMMDDHHmmSS". For strings that are not long, the score calculation works as for entities. For strings that are long, the calculation is done via the *Jaccard distance* [26] because it offers a more reliable similarity score. **CPA**, The fifth step is the CPA step, it is a rather fast process where we have to retrieve the set of predicates from the candidate couples. The candidate with the highest occurrence will be ranked first and so on. This allows a reduction in the number of candidates for each cell.

CTA, The sixth step is the CTA step, the LamAPI tool searches for all the concepts related to the candidates from the CPA step. A principle of occurrence (score from 0 to 1) is still carried out and only those with an occurrence higher than 0.95 will be retained. Then, the concept containing the most links in the Wikidata KG is selected.

3.Post-Treatment, The seventh step (Revision) takes the dataset provided by the CTA step and allows to correct the different entities obtained in the CEA step. The predicates are also rearranged. Now, each entity is consistent with the column and the row to which it belongs [46]. The last step (Export) allows to export all the results obtained by the previous steps. This makes it easier to evaluate the results for each step.

Dagobah (2019)

The Dagobah tool does not require any preconditions for the implementation of STI principles [114]. The database used for this version is DBpedia.

[25] https://en.wikipedia.org/wiki/Levenshtein_distance
[26] https://en.wikipedia.org/wiki/Jaccard_index

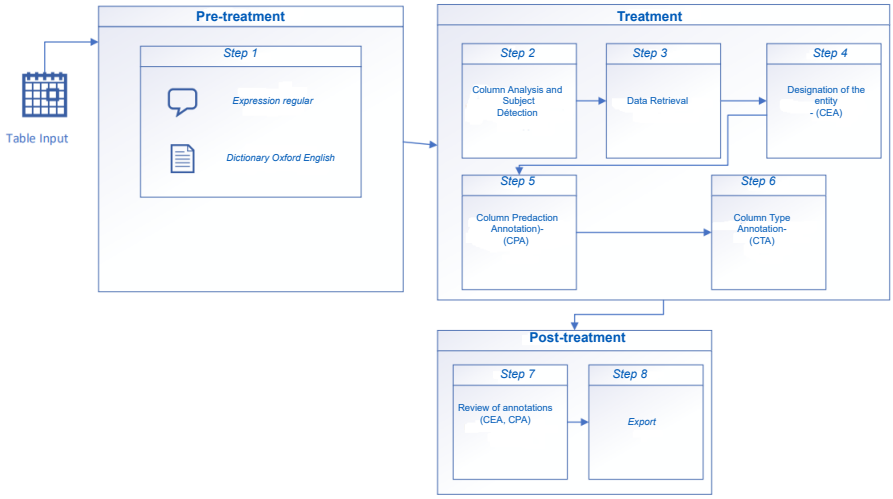


Figure 3.11: Functioning of MantisTable 2020.

1.Pre-Treatment, The first step in the tool is to use the DWTC-Extractor [27] . It returns important information about the chart. The first information returns the orientation of the chart (vertical or horizontal). To do this, it rely on 11 types (String, float, date...). Then, a score is calculated for each column and row. The averages of each column and each row are compared to each other. Depending on the result, a table is defined as horizontal or vertical. The second piece of information concerns the precision of the column headers. The tool relies on the fact that the header contains a string and that the string does not have the same type as the cells in its column. With these two items of information, the tool can determine whether the header is accurate enough to represent the column. The third information element concerns the detection of the key column (subject). To define this column, the team relies on the information that contains many unique values and that is located on the left side of the table. This information is sufficient to start the STI processing . But before that, a text checker is applied to solve the problems of special characters, parenthesis...

2.Treatment, Two approaches are used to fulfill the CEA, CTA, CPA tasks. The first is called Baseline and the second is called Embedding.

i) *Baseline Approach.* The second step of the tool concerns the search for candidate entities. The team uses several search services to retrieve candidate entities of interest (Wikidata API, Wikidata Cirrus Search Engine, DBpedia API). To retrieve the most relevant candidate entities and their type, they retrieve the ones that appear most often in the searches. In order to match the results of the different search services, a correspondence is made between the results obtained by Wikidata and the entities of DBpedia (via SPARQL) [114].

CTA, The third step is to retrieve the type of the column. To do this, a basic

[27] https://github.com/JulianEberius/dwtc-extractor

coverage of cell criteria is not relevant, as good types may be more specific but not frequent enough to be considered target types. A threshold based on relative scores is used. Therefore they retrieve the types that appear at least 70% (configurable threshold by Levenshtein percentage) of the time in the column [114].

CEA, Through the CTA step, it is possible to choose the entity from the list of candidate entities that appeared in the CEA step. This fourth step concerns the resolution of the ambiguity of the entities. To this end, they verify if the first candidate in the list has the type that was retrieved in the CTA step: if yes, then the candidate entity is chosen as the entity of the cell; if no, they select the candidate entity that corresponds to the type retrieved in the CTA step.

ii) *Embedding Approach*. This approach takes into consideration that entities in the same column have common semantic features and form coherent groups. The nested approach is done in the following manner : For the second step, they use the pre-trained Wikidata embedding [74] but this one does not have all the information (alias, label...), so an elastic search must be added. The strategy used to retrieve candidate entities is the use of REGEX as well as the *Levenshtein* algorithm. For a candidate to be retrieved, it must have a *Levenshtein percentage* higher than 75% and/or the candidate entity must contain all the words found in the cell. Next, a grid search strategy is applied. This calculate the precision of the grouping of candidates and bring out the grouping with the highest precision ratio.

CTA, The third step is to retrieve the types with a high occurrence score per candidate grouping. Then, they retrieve the type with the most specificity with respect to DBpedia (SubClasses of *owl:Thing*).

CEA, If the candidate entity has a type corresponding to the result released during the CTA step then it is given a score of (1.5). If it has a related type to the result of the CTA step, then a score of (1.0) is given to it. With this score, a new algorithm is produced. This one bring out the required entity.

CPA, Whether for the basic approach or the embedded approach, the method used for the CPA is always the same. For this fifth step, a search technique is either used on the column headers (this does not give good precision) or a relationship search between the occurrences of two columns with a majority vote is applied (this offers higher precision).

3.Post-Treatment, Here, there is no post-treatment because there is no re-evaluation at the level of the CEA, CTA and CPA tasks.

Dagobah (2020)

The Dagobah tool fulfills the same role as the 2019 tool [84]. Here, the KB used is Wikidata. No preconditions are used here. Compared to Dagobah 2019, the basic approach has been dropped to make room entirely for the nested approach.

1.Pre-Treatment, No pre-processing for this version of the tool.

2.Treatment, The first step is to find the candidate entities related to the different cells. This step matches the first step of the nested approach of Dagobah 2019. At the REGEX level, they remain on the principle that Alias must contain all the words. For Levenshtein's algorithm, the ratio increases to 65% (instead of 75% from last year).

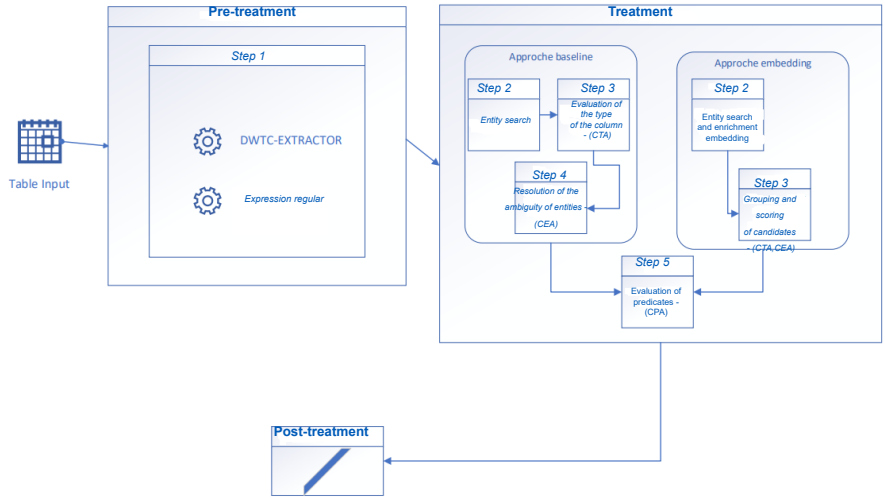


Figure 3.12: Functioning of Dagobah 2019.

The ratio has been reduced to ensure that there is at least one candidate per cell. A maximum number of 50 candidates is selected per cell [84]. They decided to create their own search service since Wikidata API has too many restrictions (concurrent connection, size of result sets, query time...). This new search service includes the Wikidata Toolkit. Only entities that have a QID and documents that have a PID are used. The search was done in python and has a rather poor performance. Their search ran for 11 days for Round 3. An optimization is possible by integrating the Cython Levenshtein [28] library which considerably reduces the search time. The second step aims at giving a confidence score to the different candidates resulting from step 1. The goal is to calculate the highest Levenshtein ratio between the cell and all the labels of the candidates. To calculate this ratio, it is also necessary to recover all the cells on the same row as the cell where the formula is applied. *For example, if a cell contains the text "University College Cork" and the candidate entity is "Q1574185" then its score is 1 since "Q1574185" has exactly the name of this university as a label and all the cells on the same row have information related to "Q1574185".*

CPA, The third step is to find the relationships between the columns. The table is divided into two parts, the "head" part which contains the candidate entities of the "head" column and the "tail" part which contains the candidate entities of the "tail" column. To calculate the score, the types of values had to be considered:

- If entity identifiers are compared, then the score is equal to 1 if the identifiers are the same otherwise it is equal to 0.
- If numerical values are compared, then the score is equal to $1-(num1-num2)/(num1+num2)$.
- If strings are compared, the Levenshtein algorithm is used.

[28] https://pypi.org/project/python-Levenshtein/

- If dates are compared, then the score is equal to 1 if dates match even if they have different formats.

The relationship with the highest occurrence is included and, in case of a equality, the one with the highest score via the previously mentioned criteria is included.

CEA, The fourth step is to define the final entity in the corresponding cell. For this, the scores obtained in step two are retrieved and a score of 1 is added if the CPA step succeeds in finding the relations with the entity in question with respect to the other cells. By increasing the score by 1, the candidate has a much higher chance of being selected as the final entity.

CTA, The fifth step consists in retrieving the type of the entities of a same column. To do this, a majority vote is done on all the entities of a same column. The types are divided into 3 levels:

- Level 0 corresponds to the type related to the "instance of" predicate of the CEA entity.
- Level 1 corresponds to the type linked to the predicate "*subclass of*" of the CEA entity.
- Level 2 corresponds to the type of the ancestor of the CEA entity.

The preferred type is the direct type. A classification system is also used. This one respects the annotation principles of WikiData. There are 3 ranks : PREFERED-2, NORMAL-1, DEPRECATED-0.

Then, all the types of the 3 levels are retrieved and they take the types that appear the most often. If there is a equality, first the occurrence is taken into account, then the level and finally the rank. Finally, a relationship is looked for between the different columns and if there are several, the most specific one is taken. If there are none, the type that is closest to the cell entity is used. If there is still a equality, a random choice is done.

3.Post-Treatment, Here, there is no post-processing because there is no re-evaluation at the CEA, CTA and CPA task level.

Unlike the previous tools, in this section we briefly discuss these tools: LinkingPark, JenTab and Magic, because these tools focused only on the treatment part. Therefore, it is not possible to compare with previous versions or the 2021 version as Magic tool. Moreover, JenTab and LinkingPark did not include a graphical user interface to interact with their proposed novel approaches to semantic table interpretation.

JenTab

JenTab [4, 90] began by analyzing the row context and column context of a table. This method enveloped each unit of computing into independent modules so that they can be easily and repeatably recalled. These modules included handling row information, disambiguation using CTA output, etc. The system exploits different combinations of modules to support CEA, CTA and CPA tasks. The evaluation demonstrated that JenTab has excellent performance on synthetic datasets. The result showed that considering two skips has a low probability of improvement, while considering multiple skips results in a meaningful decrease in precision.

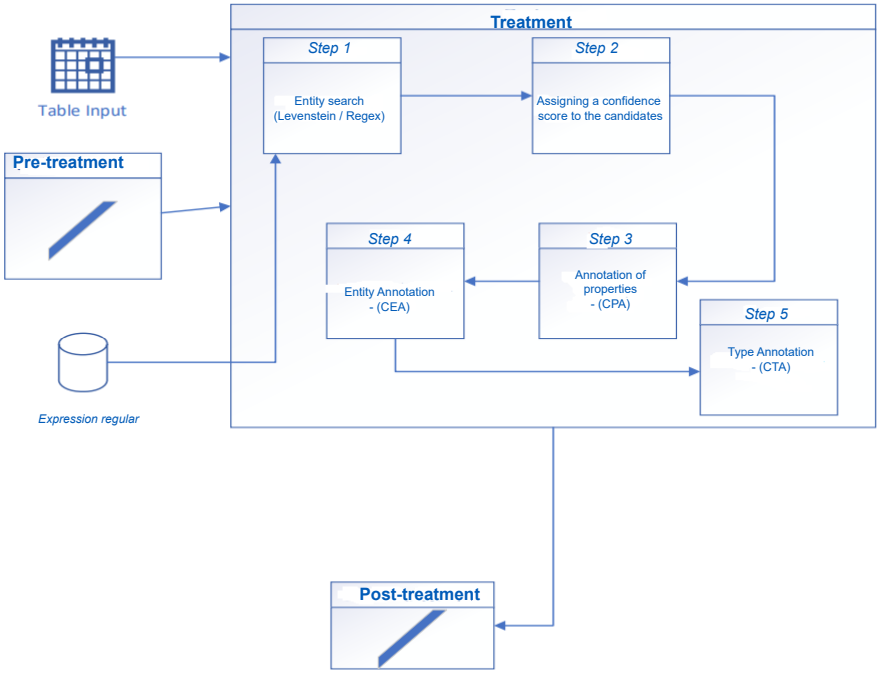


Figure 3.13: Functioning of Dagobah 2020.

LinkingPark

LinkingPark [41] have leveraged the Wikidata MediaWiki API to generate lists of candidate entities for cells. They also adapted a fine-grained Elasticsearch index to rank these candidates. This system first implemented a cascade pipeline to generate candidate entities. Then, the system disambiguated each cell through a coarse-to-fine iterative algorithm by considering the results of the CPA annotation. The system finally produced the CTA annotations from the disambiguated cell annotations. The LinkingPark team also affirmed that the Wikidata's type ontology is noisy, which makes it difficult to assign types during CTA annotation.

MAGIC

Magic [151] adopted the approach of generating comparison matrices (called INK Embeddings) to make the calculation more efficient. INK embeddings were representations of the attributes and values of an entity or the table context of a cell reference. The complete comparison matrix was generated by merging multiple candidates. The system generated CEA annotations by measuring the compatibility between the KG and table INK Embeddings. Moreover, the INK Embeddings of the entities in the same column were collected to perform the CPA and CTA. For the annotation, the system focused on the key column: it conducted the search (via public access points) for each cell in the key column, and then used its neighbor-

hood to find candidates for neighboring cells in the same row (it did not search the entire table due to limitations in using the public API). Additionally, In addition, the information supplied by the INK embeddings of new relationships in the KG can be used to enrich the table with information from the same dataset. However, spelling mistakes can be a problem for Magic, which cannot detect synonyms for attributes. This system also provided a very simple graphical user interface to augment the selected columns, which is accessible to non-expert users. It did not provide any feedback on the annotations retrieved by the process for the tables.

3.5 Comparison of SemTab challenge results

Whether, it's the 2019, 2020 and 2021 SemTab, **MTab approach comes out on top**. The tool is very **often ranked first** during each round in Table 3.3. This annotation approach can be considered the best.

LinkingPark tool comes second in the ranking for 2020. In some rounds, it was able to outperform the MTab tool especially in the CTA step where it won in every round except the last one. For the other steps, LinkingPark comes very often 2nd or 3rd.

Dagobah tool had the greatest progression. During SemTab 2019, it failed to have very convincing results. It was rarely in the top 3 or top 5 during each round. This is probably due to the baseline approach which was not enough precise. At SemTab 2020, they used the embbeding approach and succeeded in coming third in the ranking. It didn't win a single round, but it was still fairly consistent with second or third place finishes.

MantisTable did not make it into the top 3 but it was able to have very good results. Even coming to a level with MTab on some rounds. At SemTab 2019, it managed to take some rounds from the Dagobah tool.

With all these results, we had to choose which tools we were going to use during our work. We decided to use the MantisTable and Mtab tools. LinkingPark was not part of SemTab 2019 and was not included. We took back Mtab 2019 because it wins the SemTab 2019/2020 competition by a large majority which makes it the best candidate to have the best annotations. We also included MantisTable because this team has a subject column detection algorithm that will be very useful for the practical part of our project. Dagobah was not chosen because the algorithm is not as powerful as Mtab and it does not have a column detection algorithm such as MantisTable.

Other many desktop and web applications out SemTab Competition, can also display and manipulate tabular data. The tools that come along offer many possibilities to transform and create data transformation pipelines.

OpenRefine, is a very powerful tool for working with messy data through interactive user interfaces available as a desktop application. It offers capabilities for cleaning, unifying and transforming data from one format to another format. OpenRefine can perform a semi-automatic reconciliation process against any database

that exposes a Web service using Reconciliation Service API [29] specification or a SPARQL endpoint. This tool requires the user to manually correct a cell that has multiple entities (CEA). In addition, it is also able to create new columns through facets, where the user has to formulate the URL to fetch the URIs. A complete list of currently available services is specified in their GitHub [30] documentation, while the community is still working to continuously add new services. Despite the designed framework is nicely documented and supported, the tool is still accessible to domain experts and specialists.

DataGraft, [160] is a cloud-based service which provides data hosting and transformation. DataGraft processes CSV files as input data. Datagraft relies on two different tools: Grafterizer [127] and ASIA [52]. The first provides features for data preparation, transformation and cleaning, while the second offers schema and instance level annotations through reconciliation services and data extension with extension services. Semantic annotation must be performed manually by users; the tool only provides a function to auto-complete concept and data type labels using ABSTAT abstracts [171].

Odalic, [100] is a tool for STI based on the TableMiner+ [196] algorithm. it annotates columns using a KG, and links the cell values to the entities of a KG. It takes as input CSV files and one or more KGs. Odalic allows users to make changes to the annotation task product to improve the final result, which also allows to remove the suggested class/property for the annotation and propose a new one. It is possible to export the annotation performed on the table data, in different types of formats like CSV, JSON-LD, and RDF.

TableMiner+, [128] is a tool for STI based on its homonym approach [196]. It is based on an automatic process. The application requires the URL of a web page that contains one or more web tables as input, and provides a preview for each table. The entire process is run in batch and users can be notified by an email when it completed. The outputs are formatted in JSON and then parsed for display in the two main components that make up the user interface. The first component shows the loaded and annotated tables and allows users to interactively explore the data and annotation metadata. The second component visualizes the annotation data as a graph. Each row has a contextual action that allows to visualize its data as a graph. For example, the header row is represented with as many nodes as there are column cells in the table, plus more nodes for each candidate concept. The links between column nodes represent the relationships between the columns in the table, whereas the links between the node concepts and the column cells represent their individual candidates. Nevertheless, it was not possible to run and test TableMiner+ because, although the tool code was available, it could not be executed due to interdependencies in the code.

SemanticBot, [135] is a chatbot-like a semi-automatic tool that generates a mapping for the column names of a structured dataset available on OpenDataSoft portal. The mapping is generated using DBpedia, YAGO and LOV KGs combined with the answers obtained from the questions asked to the users on their selected

[29] https://github.com/OpenRefine/OpenRefine/wiki/Reconciliation-Service-API
[30] https://reconciliation-api.github.io/specs/latest/

dataset. The mapping is represented as a graph between entity types and properties. (CTA and the CPA) and can also be serialized using the RML[31] turtle syntax or the YARRRML[32] syntax. This mapping language is not easy to learn for non-experienced users.

3.6 Synthesis of approaches and tools based on the Semantic Table Annotation workflow (STA)

In this section, we focus on **the most recent and most cited approaches and tools** in the literature, as we have outlined in the previous section. In Table , we presented **the reviewed approches and tools according to the following criteria**: learning techniques (C1), annotation method (C2), data pre-processing capability (C3), subject-column detection capability (C4), semantic annotation (C5), data repair capability (add missing value) (C6), data augmentation capability (add columns) (C7), technologies based on (C8), KGs or ontologies used (C9), graph generation capability (C10), export capability (C11), interoperability capability with open data portals (C12), and working and availability of these tools (C13).

Table 3.4 summarized the overview of the selected approaches and tools that have satisfied certain criteria to indicate their features, strengths and weaknesses. These works have been proposed, mainly with a particular emphasis and not fulfilling all the criteria.

Tools that participated in the SemTab competition, such as MTab, Mantistable, JenTab, Dagobah, LinkingPark, and Magic. Each of them has proposed novel techniques to improve and provide high-quality annotations to address the issues of misspelling and missing or incomplete metadata. Thus, They used unsupervised learning techniques, which could be applied to general-purpose domains, and utilized Open Source knowledge graphs (KGs) that was freely available on the Web (Freebase, DBpedia, and WikiData). These tools, as automatic semantic annotation systems, could handle the three tasks CTA, CEA and CPA. MTab achieved impressive empirical performance for the three annotation tasks of the STA process and won the first prize at the SemTab challenge (2019 and 2020). However, it did not focus on subject column detection but has excellent annotation results and MantisTable did not offer great results like MTab but provided a perfect way of Sub_Col detection. In Additionally, they have provided a user-friendly graphical interface and annotation of the data was done with minimal time for the user. Although these tools have advantages, they also have shortcomings, such as the lack of ability to handle missing values or add an additional column to augment the annotation with additional KG and also the lack of export annotated data. More than that, Dagobah, Magic and LinkingPark did not include the pre-processing phase in their work, which led to weak annotation results and their source codes are not available. However, Magic has offered systems capable of both annotating and augmenting a dataset. Despite that, spelling errors ans missing values can be a difficulty and the system is not able

[31] https://rml.io/specs/rml/
[32] https://rml.io/yarrrml/

to detect synonyms and abbreviations of attributes. It may not outperform existing state-of-the-art techniques for generating these annotations. In the same vein, they require users to install before using them, unlike MTab, which is online and provides a user-friendly graphical interface.

In such context, other tools have been proposed to deliver annotated tabular data. SemanticBot is a semi-automatic tool, which generates a mapping for column names and relationships between columns in a structured dataset available only on the OpenDataSoft portal. This mapping is represented as a graph between the CTA and the CPA and can also be export in RML Turtle. This tool requires user intervention to create the mapping (semi-automatic tool) and must the users to discover the content of their dataset in advance. In addition, it requires additional user time to answer questions, focuses only on semantic annotation of the dataset header, and does not offer semantic annotation of cell values and therefore no graphical representation of their correspondence. Unlike TableMiner+ has provided a representation graph to visualize the annotated data. It visualized the CTA, CEA, and CPA. Nevertheless, it was not possible to execute TableMiner+ because its source code was not available, like Dogobah and LinkingPark. Unlike DataGraft, is a manual tool, was proposed online. The semantic annotation might to be done by users, which is the main drawback of this tool. It also did not offer a graphical representation of the annotated data and did not has the capability to pre-process the data, add new columns or add missing cell values, like the Odalic tool. The latter is an automatic tool for annotating data in CSV format using any KG, accessible via the SPARQL query language and which is available on docker. Odalic also allows users to correct the generated annotation themselves and export it later in RDF or JSON-LD format. This tool has the same shortcomings as OpenRefine.

Despite all their achievements and results, these proposed approaches are not able to assist users in overcoming data quality issues and none of the tools are both capable of being interoperable with open data portals, automatically annotating data, and generating graphs from the annotated data. Unfortunately, in the R&D community, there is a significant lack of work to support these tool by combining some suitable features defined in the table 3.4. In addition, many data providers are not familiar with Linked Open Data (LOD) technologies and are not willing to invest the time necessary to integrate their data with (KGs) (semantic annotation) to discover and publish data.

3.6.1 Conclusion

In this chapter, we have conducted by a methodology to select the best papers, that aimed to find the best methodologies an approaches for using and publishing OGD, where this section was made by selecting the 22 best academic papers. We have categorized these related works on three main components: (i) Linked Open Data (LOD) technology, (ii) Linked Open Government Data (LOGD), and (iii) Linked Open Statistical Data (LOSD). In additional, in the section 3.2.2, we have explained in details each phases of methodologies and approaches; and in the section 3.3, we have synthesized these selected researches in terms of common phases, associated

tasks (Table 3.1), and evaluation methods applied in these studies (Table 3.2). In Chapter 4, at least, we intend to cover all phases of data, from data extraction to data publication to data visualization. Furthermore, in Chapter 7, we will assess our works and its effectiveness using certain evaluation methods presented in this chapter. Unfortunately, large tabular data on the Web are not usable because of their poor quality. They have a high level of heterogeneity, lack of metadata and interoperability, making them difficult to discover and understand. Additionally, many data creators have no knowledge of LOD technologies and have no interest in spending time integrating their data into KG. In Section 3.4.2, we have presented many approaches and tools that have been proposed to solve the aforementioned issues by corresponding tabular data elements to KG concepts. This section was made by selecting 32 articles from the selected articles in the SemTab competition (2019, 2020, and 2021) and the most cited articles in the STA/I field. We have described in details the semantic annotation tasks of each approach. In Chapter 5, we compare the results of the SemTab competition, which are represented in the table 3.3 and we will show, why we integrate these tools in our contribution. In the end, we synthesized these approaches and tools examined according to the following requirements presented in the table 3.4.

Table 3.1: Tasks and studies for the identified methodologies.

Phases	Tasks	Studies
Data Extraction	select data	[11], [65], [23], [157], [77], [186], [177], [118], [22], [11], [13], [66], [102], [125], [108], [40]
Data transformation	unify and clean up data source	[11], [118], [11], [108], [40]
	design URIs	[157], [167], [77], [186], [168], [22], [175], [94], [176], [66], [40]
	define vocabularies	[11], [23], [157], [77], [186], [11], [102], [125], [108]
	match vocabularies to data	[11] , [65], [167], [77], [186], [164], [177], [121], [138], [122], [156], [118], [11], [102], [63], [125], [40]
	enrich the dataset	[65], [167], [118]
	convert to RDF	[11], [65], [23], [157], [77], [186], [164], [168], [177], [121], [122], [138], [156], [18], [118], [22], [11], [175], [94], [176], [66], [102], [125], [40]
	clean-up RDF data	[77], [186], [94]
Data Interlinking	link to other data sources	[11], [22], [11], [13], [66], [63], [108] [40]
	merge data sources	[175], [176]
Data storage	store RDF data	[65], [164], [18], [175], [94], [176]
Data visualizing	enable discovery	[77], [186], [13], [175], [94], [176], [66]
Data publishing	publish data	[11], [23], [157], [77], [186], [121], [18], [118], [22], [11], [13], [175], [94], [176], [66], [102], [63], [125], [108], [40]
Create Apps	create API	[157], [40]
	build apps on top of data	[11], [23], [157], [167], [77], [186], [168], [121], [156], [118], [22], [11], [13], [94], [66], [40]
Engaged	engage with community	[108], [40], [66]

Table 3.2: Evaluation methods adopted in the selected studies

Evaluation method	Description	Studies
Qualitative metrics of LOGD	Measurement to quantitatively assess the quality of open government data	[77], [121], [175], [94], [131], [54], [126], [170]
Prototype	Implementation of an artifact to demonstrate its utility or usefulness	[167], [168], [177], [121], [125], [92], [157], [164], [156], [22], [102], [118], [92], [175], [186], [94]
Illustrative scenario or case study and practice-driven	Application of the method to real-world data to illustrate the practicality of the artifact.	[23], [69], [157], [176], [122], [18], [11], [163], [66], [63], [108], [192], [77]
Logical arguments and reasoning	An argument with face validity; without any empirical experimentation.	

Table 3.3: Summary table of the results of the SemTab challenge by round. R= Rank, P= Precision, R= Recall, and Ro= Round.

Ro	**Tasks**	**CEA**			**CTA**			**CPA**		
	Tools	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**
Ro1	Mtab 2019	1.0	1.0	2	1.0	1.0	1	0.987	0.975	1
	Dagobah 2019	0.897	0.941	3	0.644	0.580	9	0.415	0.347	4
	MantisTable 2019	1.0	1.0	/	0.929	0.929	/	0.965	0.991	/
	Mtab 2020	0.987	0.988	1	0.885	0.884	2	0.971	0.991	1
	LinkingPark	0.987	0.988	2	0.926	0.926	1	0.967	0.978	2
	Dagobah 2020	0.922	0.944	6	0.834	0.854	6	0.914	0.962	7
	MantisTable 2020	0.982	0.989	3	0.746	0.753	9	0.888	0.942	8
Ro2	Mtab 2019	0.911	0.911	1	1.414	0.276	1	0.881	0.929	1
	Dagobah 2019	0.713	0.816	8	0.641	0.247	7	0.533	0.919	5
	MantisTable 2019	0.616	0.673	/	1.049	0.247	/	0.460	0.544	/
	Mtab 2020	0.995	0.995	1	0.984	0.984	2	0.997	0.997	1
	LinkingPark	0.993	0.993	3	0.984	0.985	1	0.993	0.994	2
	Dagobah 2020	0.993	0.993	2	0.983	0.983	3	0.992	0.994	3
	MantisTable 2020	0.991	0.993	4	0.966	0.973	4	0.961	0.966	6
Ro3	Mtab 2019	0.970	0.970	1	1.956	0.261	1	0.844	0.845	1
	Dagobah 2019	0.725	0.745	7	0.745	0.161	7	0.519	0.826	6
	MantisTable 2019	0.633	0.679	/	1.648	0.269		0.518	0.595	/
	Mtab 2020	0.991	0.992	1	0.976	0.976	2	0.995	0.955	1
	LinkingPark	0.986	0.986	2	0.978	0.979	1	0.985	0.988	3
	Dagobah 2020	0.985	0.985	3	0.974	0.974	3	0.993	0.994	2
	MantisTable 2020	0.974	0.979	4	0.958	0.965	4	0.941	0.957	5
Ro4	Mtab 2019	0.983	0.983	1	2.012	0.300	1	0.832	0.832	1
	Dagobah 2019	0.578	0.599	8	0.684	0.206	7	0.398	0.874	7
	MantisTable 2019	0.973	0.983	/	1.682	0.322	/	0.787	0.841	/
	Mtab 2020	0.993	0.993	1	0.981	0.982	1	0.997	0.997	1
	LinkingPark	0.985	0.985	2	0.953	0.953	4	0.985	0.988	5
	Dagobah 2020	0.984	0.985	3	0.972	0.972	3	0.995	0.995	3
	MantisTable 2020	0.812	0.985	9	0.725	0.989	8	0.803	0.988	7

Table 3.4: List of reviewed approaches and tools for semantic annotation. Sup = Supervised, Unsup = Unsupervised, SA = Semi-Automatic, M = Manual, A = Automatic, N = No, Y =Yes and N/A = Not Applicable.

Approach	**C1**	**Criteria**													**C13**
Tools	**Learning techniques**	**C2**	**C3**	**C4**	**C5**			**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**Working & Available**
					CTA	**CEA**	**CPA**								
Mtab	Unsup	A	Y	Y	Y	Y	Y	N	N	N/A	DBpedia WikiData	N	N	N/A	+(online)
MantisTable	Unsup	A	Y	Y	Y	Y	Y	N	N	ABSTAT	DBpedia WikiData	N	N	N/A	+(installer)
Dagobah	Unsup	A	N	N	Y	Y	Y	N	N	N/A	DBpedia WikiData	N	N	N/A	-
JenTab	Unsup	A	N	Y	Y	Y	Y	N	N	N/A	DBpedia WikiData	N	N	N/A	+(docker)
LinkingPark	Unsup	A	N	N	Y	Y	Y	N	N	N/A	DBpedia WikiData	N	N	N/A	-
Magic	Unsup	A	N	Y	Y	Y	Y	N	Y	N/A	WikiData	N	N	N/A	+(exe)
Open Refine	Sup	SA	Y	N	N	Y	Y	N	Y	Reconciliation API	WikiData FreeBase	N	Y	N/A	+(exe)
DataGraft		M	N	N	Y	Y	N	N	N	Grafterizer ASIA ABSTAT		N	N	N/A	+(online)
Odalic		A	N	N	Y	Y	N	N	N	TableMiner+	DBpedia WikiData	N	Y	N/A	+(docker)
TableMiner+		A	N	Y	Y	Y	Y	N	N	TabelMiner	FreeBase	Y	N	N/A	-
SemanticBot		SA	N	Y	Y	N	Y	N	N	N/A	DBpedia LOV	Y	Y	OpenDataSoft	+(online)

Part III

Contributions

CHAPTER 4

SUPPORTING SEMANTIC-AWARE ANALYSIS IN LOD-ENABLED OGD SYSTEMS

"If I had asked people what they wanted, they would have said faster horses."

— ***Henry Ford, An American industrialist.***

4.1 Introduction

In recent years, there has been a significant rise in the use of Open Data (OD), particularly in relation to government data, sometimes referred to as Open Government Data (OGD). This movement has gained prominence due to its ability to provide individuals and public institutions with transparent, unrestricted, and tailored access to pertinent information. Furthermore, it facilitates the development of novel services, including both the public and private domains [13, 88].

Open Government Data (OGD) encompasses a variety of datasets, such as budget and expenditure information, minutes from parliamentary sessions, and other relevant data, etc. Moreover, a large part of this OGD concerns statistics [36,53], such as demographic (e.g., census data), population figures, and economic and social indicators (e.g., number of new businesses, unemployment, and workers). It also includes data that is indirectly ''owned" by public administrations (e.g., through subsidiaries or agencies), such as data related to climate and pollution, public transportation, congestion/traffic, child care/education. These data can be important primary material for added value services and products, increasing government transparency, contributing to economic growth, and providing social value to citizens [89, 92]. In such a context, several government projects (e.g., in the United Kingdom, United States, France, and Belgium) have been launched in order to pro-

vide free access to different data catalogues such as GSS [1], data.gov [2], data.gouv.fr [3], data.gov.be [4], and STATBel [5] in diverse domains (i.e., geospatial, statistics, transport, education, population, etc).

Despite this increasing attention, OD available on the web are still of a high level of heterogeneity, they are available in various formats (i.e., spreadsheets, CSV, relational database dumps, RDF), which make challenging to determine the quality of data to be published.

LD has been introduced as a promising paradigm for opening up data [117] and the advent of Linked Open Data (LOD) [6] technologies allows for more efficient access and facilitates knowledge extraction and enrichment from heterogeneous data sources [29]. Besides, many vocabularies have been created to enable modeling data as RDF graphs, such as WikiData [62], DBpedia [63], the RDF data cube (QB) vocabular [54] and the Simple Knowledge Organization System (SKOS) vocabulary [130] , which have been proposed as recommendations by the World Wide Web Consortium (W3C) and are thus widely adopted.

4.1.1 Linked Open Government Data (LOGD) Challenges

Several works have been proposed in detail in chapter 3, section 3.2.2. In this section, we highlight some of the challenges we have focused on. Many challenges remain unaddressed, which prevents them from reaching their full potential. LOGD still suffers from classic problems due to **the heterogeneous and dispersed nature of the data**. Indeed, **the heterogeneity of the published datasets and their representation is quite an obvious setback in such context**. This issue is due to the availability of LOGD in various formats in terms of semantics, standards, and, most importantly in this case, schema. This leads **to interoperability issues** and **challenges to aggregate existing metadata in a way that would be useful for data consumers**. Unfortunately, many governmental entities still publish data in a large variety of formats. Thus, a lot of data is close and not accessible. In addition, other factors lead to difficulties in finding useful data quickly [115]. Furthermore, **Semantic ambiguity** would require extra efforts to **link and understand the data** in question. Again, the **discoverability** and the transparency of open data are bound to the quality of the metadata describing the data itself, which is not always complete or accurate [9, 11, 13, 70, 175, 176].

Despite these drawbacks, many approaches have been proposed in LOD-based systems in the chapter 3 (section 3.2.2). For instance, [23] have suggested an approach for automatic extraction of relevant knowledge sources from LOD sources. [69] presents a survey of existing approaches tackling the heterogeneity of web-based datasets for searching and querying linked data. [185] have presented methodological guidelines to formalize the publishing process of LOD, taking into account data quality. Yet, limited work exists in the R&D community, for supporting the

[1] https://gss.civilservice.gov.uk/
[2] https://www.data.gov/open-gov/
[3] https://data.gov.be/fr
[4] https://data.gov.be/fr
[5] https://statbel.fgov.be/en
[6] https://www.w3.org/DesignIssues/GovData.html

end-to-end process of LOD production and publishing in a government context. There is **a clear lack of automatically supported, integrated solutions** that enable **(i) combining appropriate tools to assist users, especially Open Data publishers, to efficiently manipulate datasets from extraction to publication**, **(ii) having a follow-up on the progress during this process as well as a semantic visualization of its data**, and **(iii) involving users with governments to reuse the data to provide public services and applications**.

This chapter aims to fill the above research gap by proposing a useful and integrated semi-automatically framework to interactively assist users in the **P**ublishing of **L**inked **O**pen **G**overnment **D**ata (**Pub_LOGD**). Therefore, our research questions are presented in section 4.1.2. To answer the research questions, we first conducted a literature review in the chapter 3 to identify the tools and the existing peplines and methodologies for publishing LOGD in previous studies (see, chapter 3 (section 3.2.2)).

Our aim is to support semantics-aware analysis in LOD-enabled Government Systems. We propose a complete and unified process that covers all phases of the LOGD life cycle, iterative, reusable, and interactive with domain experts and application developers. It also enables for data distribution, support, and grouping of process steps in a coherent manner.

As a first step of our work, we developed **a pipeline-based framework to publish Open Data** (OD). Our framework aims at shifting the burden of processing, converting, and publishing LOD to data consumers rather than the government. Our framework consists in **an end-to-end solution that covers five main data analysis steps (i) data transformation, (ii) data interlinking, (iii) data storage, (iv) data visualization and (v) data publishing**. As a second step of our work, we extend the data transforming step with **an automated approach to the modeling step**. This extension is based on **the set of requirements that are feedbacks of participation**, which they completed the pre-survey. Our main goals in this work are to provide a mechanism to promote user engagement with governments and the reuse of government information in order to complete the post-surveys and access to a publishing solution.

This framework relies **on a Docker-based implementation using a combination of software tools** in order to provide users with a facilitated and customized access to the data sets available on the web. Our purpose is to transform raw data to a high-quality government dataset published on the web, link them together, by also providing semantic visualizations which make citizens aware of the extracted knowledge to enhance decision-making.

In Chapter 7.2, an evaluation of our proposals through a **case study of the statistical portal of Belgium**, based on a local population dataset used by community users. We compare our approach with other existing approaches using a number of **qualitative parameters**. At the end, we aim to engage users to evaluate and demonstrate the **practicality and usability** of our proposals.

4.1.2 Research questions and Publications

The content of this chapter is based on the following research questions and scientific publications:

Abida, R., Cleve, A. & Hachicha Belghith, E., Supporting Semantics-aware Analysis in Linked Open Data-enabled Information Systems, Grascomp Doctoral Day 2019 [8].

Abida, R., Hachicha Belghith, E. & Cleve, A., 2020, An End-to-End Framework for Integrating and Publishing Linked Open Government Data, Proceedings of the 29th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2020). IEEE. Under review [5].

Abida, R., Hachicha Belghith, E. & Cleve, A., 2023, Towards Transparent Governance by Publishing Open Statistical Data. International Conference on Digital Technologies and Applications (ICDTA 2023). Springer. Under review [6].

These scientific publications cover this research question such as:

RQ1. "How can we supply a standardized solution to ease the use and publication of LOGD?"

4.1.3 Outline

This chapter is organized as follows: Section 4.2 details the review protocol that guides this solution and compares our approaches with other existing approaches using a list of criteria and a set of **qualitative parameters** (see Table 4.2). There are three parts to Section 4.3. They give an overview of the **Pub_LOGD** framework, go into detail about the different steps, talk about the extension of **Pub_LOGD_Extension**, and show how they are used. Section 4.3.3 addresses the threats to validity. Section 4.4 provides a conclusion that summarizes the contribution of this chapter.

4.2 Methodology and Comparison

To address the research question of this study, we conducted two steps: in the first publications, we conducted a literature review on user assistance to publish structured Linked Data in an e-Government data context. it was defined in line with the systematic literature review [76, 98]. The need for a review was discussed in Section 3.2.1. The literature review was conducted using the databases "SpringerLink", "Scopus", "Google Scholar", AND "Science Direct" with the keywords "open data" OR "open government data", + "linked data life cycle", OR "pipline", OR "process", + "modeling", OR "publishing" + "users" OR "citizens" OR "stakeholders" + " decision-making ". We also extended our search to the gray literature and policy reports. Most of the publications found were dated. from 2010-2022. From these publications, an additional selection was made based on their relevance to our research, leaving altogether 31 academic articles, web pages, and policy reports, which were looked at more thoroughly. The selected papers were then used to collect appropriate methodologies and tools to ensure better data quality and to publish

and visualize related open government (statistics) data, allowing stakeholders to be aware of the knowledge retrieved to improve their decision-making. In the second contribution, we tried to involve the users in order to obtain feedback about the **Pub_LOGD framework**. Then, we created an online pre-survey [7] (See Appendix B) to collect feedback from stakeholders, researchers, data geeks, analysts, and data scientists who use open data. The survey was pre-tested with three users and then shared via the following communication channels: UNAMUR mailbox, Facebook, and LinkedIn. We also distributed it at some Open Data events. In total, 35 participants completed the pre-survey [8]. The literature review along with user feedback will be used to improve the functionalities of the **Pub_LOGD** framework, which aims to increase the quality of open data that becomes explicitly available and accessible to data consumers and to provide guidelines to data publishers (governments). You can find more details about the methodology in Section 3.2.1.

Despite the amount of work (See Section 3.3), there **still some technical challenges for exploiting the full potential of LOGD** and especially reusing related **statistical data** and supporting the end-to-end process for producing and publishing OGD in a public context.

As shown in table 4.2, we evaluate some approaches that mainly publish LOGD according to a set of criteria. These criteria represent (i) the different **data processing steps**, which are: Data transforming, data interlinking, data storing, data visualization, data publishing, User Assistance, Functional Completness, Interoperability, Operability, Adaptability, Installability, reusability, and replaceability; and (ii) a set of qualitative metrics, specific to the evaluation of the **Linked Open Statistical Data** (LOSD) platform, which are defined in the literature by Matheus et al. (2017) [126]. We present the description of the qualitative metrics that we have adopted in Table 4.1.

From the related work in Section 3.2.2, We have focused on approaches and frameworks that have been presented as methodological guidelines for the release of **Government Linked Data** or **Open Statistical Data**.

Table 4.2 presents the related approaches that were presented in Chapter 3, Section 3.3. We note that contrary to our work, some works **do not cover data interlinking** and **do not take into account user assistance**. Whereas other works set **do not offer to the user data visualizing step** and **do not highlight user guidelines**. Therefore, Beluzo et al., Musto et al., and AlSukhayri et al. **only convert the CSV format and do not take into account the preparation phase**. In addition, Beluzo et al. and Musto et al. have a specific context (university, building, etc.) and **are not easy to use for non-expert users** who are not familiar with Linked Data technologies.

To this end, our contributions aim at covering all the data phases, **from data extraction to data publishing** and visualization, in the context of Open Statistical Data (OSD) and meeting the expectations of **Open Data Publishers**.

[7] https://tinyurl.com/55m33rwu
[8] https://github.com/123rabida123/WebAppOpenData

Table 4.1: Qualitative Metrics Description [126]

Qualitative Metric	Description
Functional Completness	The purposed platform has tools to open and to browse, expand, analysis and visualisation, and is easy to use for beginners.
Interoperability	The platform is created interoperable. Considering data source on an interoperable format such as tabular CSV, and output data (Data Cubes, RDF and TTL).
Operability	If proper documentation, manual and also examples of usage are provided, the operation of platform is easy and transparent.
Adaptability	Platform components were written in different languages, but all components are communicating via (Docker or APIs).
Installability	Platform components are web services and can be easily installed/uninstalled and run at any environment.
Reusability	Platform components can be reconfigured to work with other data sources/applications. As an example, the JS libraries can be used.
Replaceability	Platform components can be reconfigured to work with other data sources/applications. As an example, visualisation service can be run using any Fuseki instance containing data RDF.

4.3 Pub_LOGD: Framework for Integrating and Publishing LOGD

In this section, we present an overview of our framework, **Pub_LOGD**, which consists of a pipeline of five main data processing steps **for publishing Open Government Data as Linked Data**. Our purpose is to allow public institutions as well as Open Data publishing to access to **an end-to-end framework** that guides them with the appropriate tools during the entire publishing process, thereby offering them relevant data visualizations.

As shown in Figure 4.1, our framework follows **an iterative** and **incremental process** which covers: (1) Transforming data (i.e., extracting, unifying, cleaning, and converting to RDF), (2) Interlinking data with other data sources, (3) Storage, (4) Visualization, and (5) Publishing on the Web. The integration of the tools supporting each step has been achieved using **the Docker Container Technology** [9].

At first, an **extracting phase** is applied to multiple data sources given as input, which involves a set of discovery actions depending on the user requirements. Two additional phases are then realized: **data unification** and **data cleaning**. All these phases are achieved via *Open Refine* (Web Ref. [1])[10] which is an open-source tool that offers powerful capabilities for data formatting and tidying up. Afterwards, an **interlinking phase** is applied to enrich our **LOGD** with useful related data extracted from other data sources. The latter step is realized by the *SILK* (Web Ref. [2])[11] tool. A **data storage phase** is then performed on the generated LOGD by means of the *Fuseki* [12] tool. A **semantic visualization** can then be triggered by the user via the use of a *neo4j* [13] based tool that reveals implicit contexts and hidden relationships

[9] https://www.docker.com/resources/what-container/
[10] https://code.google.com/archive/p/google-refine/
[11] http://silkframework.org/
[12] https://jena.apache.org/documentation/fuseki2/
[13] https://neo4j.com/

Table 4.2: Evaluation of related approaches.

	Data transforming	Data interlinking	Data storing	Data visualizing	Data publishing	User Assistance	Functional Completness	Interoperability	Operability	Adaptability	Installability	Reusability	Replaceability
Hidalgo-Delgado et al. [77](2017)	+++ (M)	-	-	+	+	-	++	+	+/-	+/-	-	-	-
Villazón-Terrazas et al. [186](2011)	+++ (M)	-	-	+	+	-	++	+	+	+	-	+	+
Sarkar et al. [164] (2011)	++ (SA)	+	-	-	+	-	+	+	+/-	+/-	-	-	-
Musto et al. [138] (2019)	++ (M)	-	-	-	+	-	+	+	+/-	+/-	-	-	-
Maali et al. [118] (2012)	+++ (SA)	-	-	-	+	-	+	+	+	+	-	+	-
Beluzo et al. [22] (2021)	+++ (SA)	+	+	-	+	-	+	+	+	+	-	+	-
AlSukhayri et al. in [11] (2019)	++ (SA)	+	+	-	+	-	+	+	+	+/-	-	-	-
E. kalampokis et al. [94] (2013)	+++ (SA)	+	+	+	+	-	++	+	+	+	-	+	+
E. Tambouris et al. [175] (2016)	+++ (SA)	+	+	-	+	-	+	+	+	+	-	+	+
E. Tambouris et al. [176] (2017)	+++ (SA)	-	+	+	+	-	++	+	+	+	-	+	+
Pub_LOGD	+++ (SA)	++	++	++	++	++	++	+	++	+	++	+	+
Pub_LOGD Extension	+++ (SA/A)	++	++	++	++	++	++	+	++	+	++	+	+

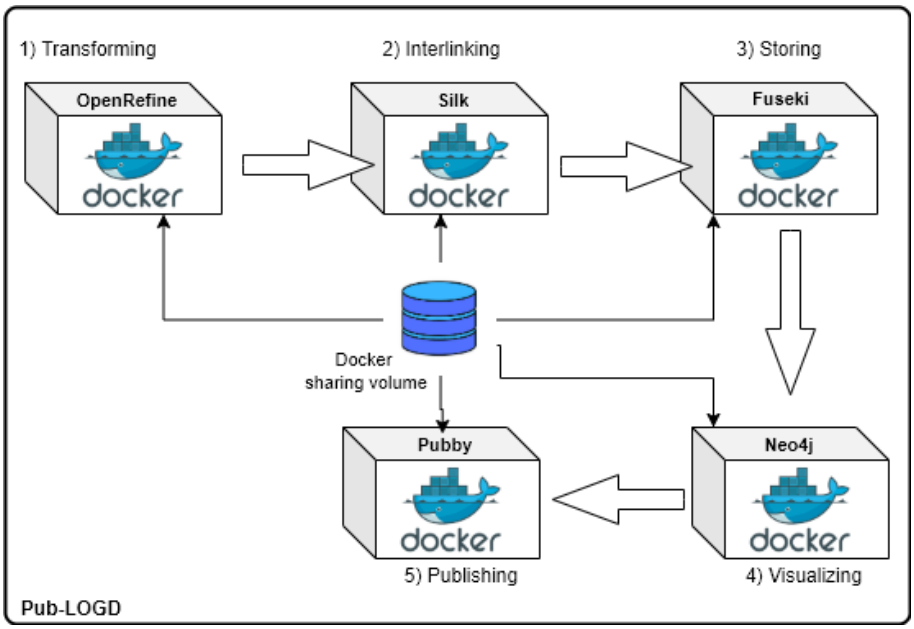


Figure 4.1: Pub_LOGD Overview for publishing OGD as Linked Open Government Data (LOGD).

among the data. Finally, a **publishing phase** is carried out via the use of the *Pubby* tool [14] in order to publish the according RDF triplets on the web.

4.3.1 Assistance for publishing LOGD

In this section, we detail the different steps supported by our Pub_LOGD Framework. More details about our framework, its source code, and implementation aspects are available online [15] .

A. Data Transformation

The data transformation step consists of **extracting** statistical data from heterogeneous data sources. To this end, we use DCAT (Data Catalog Vocabulary Technologies) Browser as an extension to Open Refine. This tool uses *DCAT standard* [16] to describe datasets of the public sector in Europe (e.g., http://data.gov.uk, http://open-data.europa.eu, etc.). *DCAT development* is pursued under the *W3C Government Linked Data Working Group* [17]. Therefore, a growing adoption of it is plausible.

Then a **data unification and cleaning** step is applied. The goal of this step is to unify and clean up the data by fixing errors, removing duplicates, and preparing them

[14] http://wifo5-03.informatik.uni-mannheim.de/pubby/
[15] https://github.com/123rabida123/WebAppOpenData
[16] http://rd-alliance.github.io/metadata-directory/standards/dcat-data-catalog-vocabulary.html
[17] https://www.w3.org/egov/wiki/Data-Catalog-Vocabulary

for transformation. *Open Refine* is the main tool that is used in this step. We found it suitable in our context since it provides a user-friendly user interface, especially for non-technical, non-expert users, which is the case for most OD publishers [73]. Additionally, it enables data clustering, which further **helps understanding** the data.

Once we have cleaned and unified the data, it is necessary to **define an ontological model** to share and annotate our Statistical Governmental Data. To do so, we rely on a domain expert for modeling statistical metadata. Then, we develop the corresponding *RDF Extension* in *Open Refine* to enable modeling and exporting data in RDF format. This step contains two sub-tasks: data reconciliation and data modeling.

Data annotation is a feature that allows to semi-automatically link data to external databases (e.g., Wikidata, DBPedia, and NCBI taxonomy) in order to align the data with these databases. Its process is semi-automatic because the end-user has the possibility to approve the results interactively and/or to select those to modify from a list of results.

Afterwards, it is necessary to define a data structure to model the statistical data of LOGD in an ontological manner. The **data modeling** step aims at building a skeleton to define the Statistical Data structure by reusing available vocabularies/ontologies as much as possible to define namespace prefixes. At this stage, URIs play an important role in the discovery and interoperability of statistical data in the web space [94]. The vocabularies, representations, and even formats of these data sets can be customized. We adopt a standardized RDF vocabulary for representing datasets and their structures.

The **RDF Data Cube Vocabulary** (QB) [54] is an ontology[18] for describing multidimensional datasets on the Semantic Web where the structure of a *dataset* can be prescribed via so-called *Data Structure Definitions* (DSDs) where *observations* are recognizable by their textit dimensions, use textit measures to capture one or more observed values, and observed values can be annotated with *attributes*. The vocabulary defined is available in these formats: [19].

For more explication of the data modeling to build Skeleton to define the open data structure, which you find here [20] a set of Linked Open Statistical Data files in RDF format (See Appendix B). A listing 4.1 shows a part of the ontology of a worker in Belgium.

```
<http://example.org/Region de Bruxelles Capitale> a qb:Observation ;
  qb:dataSet <http://example.org/dataset/file–travailleurs> ;
  rdfs:label "Region de Bruxelles Capitale" ;
  <http://example.org/def/statistical–dimemsion/province–tra> <http://example.org/Region
      ↪ de Bruxelles Capitale> ;
```

[18]While RDF provides us with the data model, data is usually integrated using a common model captured in a so-called ontology. Ontologies are commonly defined as "a [formal] explicit specification of a [shared] conceptualization" [71] – are also developed for a particular purpose, but the ontologies (or vocabularies) we observe on the Linked Data Web are often lightweight and meant for information exchange. Applications that consume such Linked Data are not (necessarily) known beforehand and are often published with very accessible licenses, such as as Creative Commons.

[19]https://raw.githubusercontent.com/UKGovLD/publishing-statistical-data/master/specs/src/main/vocab/cube.ttl

[20]https://github.com/123rabida123/WebAppOpenData/tree/master/RDF%20Cube%20LOSD

```
5   <http://example.org/def/statistical–dimemsion/Pays de citoyennet niveau> "Belges" ;
6   <http://example.org/def/statistical dimemsion/travailleurs> "256,666" ;
7   <http://example.org/def/statistical dimemsion/Pays de citoyennete niveau> "Etrangers" ;
8   <http://example.org/def/statistical dimemsion/travailleurs> "113,147" ;
9   <http://example.org/def/statistical dimemsion/Pays de citoyennete niveau> "Apatrides" ;
10  <http://example.org/def/statistical dimemsion/travailleurs> "12" ;
11  <http://example.org/def/statistical dimemsion/Pays de citoyennete niveau> "Non indique" ;
12  <http://example.org/def/statistical dimemsion/travailleurs> "555" ;
13  <http://example.org/def/statistical dimemsion/Pays de citoyennete niveau> "Total" ;
14  <http://example.org/def/statistical dimemsion/travailleurs> "370,380" .
```

Listing 4.1: Part of the ontology of the worker in Belgium.

RDF extensions [21] allow to import vocabularies available on the web regardless of the format used (e.g., RDFa, RDF/XML, Turtle) as long as their deployment is compatible with the best practices recommended by the W3C [118]. Figure 4.3 represents the user interaction interface of the *Open Refine* tool for cleaning and transforming data to triplet format.

B. Data Interlinking

During the data interlinking step, we **generate links between the RDF graphs generated** in the previous steps. This step relies on the *Silk Link Discovery Engine* [189]. It accesses the data sources that should be interlinked via the SPARQL protocol and can thus be used against local as well as remote SPARQL endpoints. This interlinking process is used to express heuristics for deciding whether there are semantic relationships between two related entities. In our approach, we generate a new set of RDF triples from this process and upload them into the *Fuseki* server instance. Then, we use the similarity measure *Levenshtein* distance to compare the labels (*rdfs: label*) of data in our statistical dataset.

Definition (Link Discovery as Similarity Computation). *Given two sets S (source), and T (target) of instances, a (complex) semantic similarity measure* $\sigma : S * T \rightarrow [0, 1]$ *and a threshold* $\theta \in [0, 1]$*, the goal of the link discovery task is to compute the set M (Mapping) = (s, t),* $\sigma(s, t) \geq \theta$.

In general, the similarity function used to carry out a link discovery task is described by using a *link specification* (sometimes called linkage decision rule [86]). Therefore, we obtain semantic relationships of type *owl:sameAs* to express that two entities correspond to the same entity among our RDF graphs. In figure 4.3, we represent the screenshot of this step.

C. Data Storage

The data storage step aims **to store the generated RDF triples** which are prepopulated with metadata information. The *Apache Jena Fuseki* tool provides a user-friendly Graphical User Interface (GUI) to mount ontologies (RDF representations) on servers [149]. It is used to manage the RDF store, which is a SPARQL server offering RDF storage capabilities and a SPARQL endpoint for querying RDF data.

[21] https://github.com/sparkica/rdf-extension

This step was represented in Figure 4.3 (e)).

D. Data Visualization

In the current development stage of our framework, we enhanced the data publishing process with **visualization functionalities** in order to facilitate decision-making and increasetransparency. To this end, we use the *neo4j* datastore, which is the most popular and highly scalable graph database toprovide the user with graphical representations of the data and the relationships that exist among them (see Figure 4.3(f)).

D. Data Publishing

The *Pubby* server is used to publish RDF triplets on the web. This step allows you to rewrite URI requests into SPARQL to describe queries against the underlying RDF store. It also provides a simple HTML view over the data stores [29]. Pubby provides a Linked Data interface to local or remote SPARQL protocol servers. Thus, a page is created for each resource on the Pubby server. Each page includes links to the corresponding resource and dataset as well as several mappings, which allow users to navigate between datasets resources through the properties used in their mappings. For more details on these tools, we explained them in more depth in Chapter 3.

4.3.2 Extension of the Pub_LOGD Framework

This section presents **an extension of the Pub_LOGD** Framework [9], which is based on **the set of requirements** that are feedbacks of participation, which they completed the pre-survey. And,it is based on *Sparqlify*, which converts each SPARQL query to a single SQL query. For example, the RDF conversion process is very time consuming for large amounts of data that are *crowdsourced*. Furthermore, changing the **data modeling** requires many changes to the extracted RDF or creating an entirely new dump. This allows all optimizations of the underground database to be applied and can help to achieve better scalability.

Previously, we created an online pre-survey [22] to collect feedback from researchers and analysts who used OD. The survey was pre-tested with three users and then shared via the following communication channels: UNAMUR mailbox, Facebook, and LinkedIn. In total, **35 participants** completed the pre-survey (See annexes B). The literature review, along with user feedback, was used to improve the functionalities of **Pub_LOGD** framework, which aims to increase the quality of OD that becomes explicitly available and accessible to data consumers and to provide guidelines to **Open Data publishers**.

Pub_LOGD Extension relies on *Sparqlify* [23] **to automate the data modeling step** in the data transformation phase (Section.4.3.1 (A)) of the framework. In this phase, we extend the data transforming step with an automated approach to the modeling task using *Sparqlify* which is a *SPARQL-SQL* query rewriter that allows the definition

[22] https://tinyurl.com/5n7ejrv4
[23] https://github.com/SmartDataAnalytics/Sparqlify

of RDF views using a *Sparqlification Mapping Language* (SML) or (Sparqlify-ML (mapping language)). This mapping language **is easy to learn for users** who are experienced in *SPARQL* and *SQL* and more compact than other syntactic variants such as *R2RML*[24] [141]. Figure B.2 presents the query rewriting workflow in *Sparqlify*.

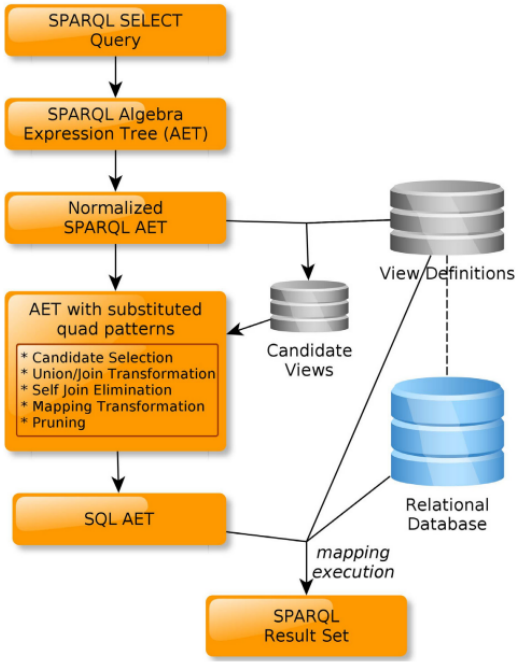


Figure 4.2: Sparqlify concepts and the query rewrite process [141].

The main concept behind *Sparqlify* is to maintain the original schema of the relational database and create RDF views on it. SPARQL queries can then be written against these views, which are expressed in the Sparqlify ML (Mapping Language). The left part of figure B.2 presents all steps that are performed to answer a query. First, the query is converted into an algebraic expression. This expression is subsequently converted to a normal form. Given the query patterns, relevant Sparqlify ML views must be identified. Once this is done, the algebraic expression is rewritten to include the relevant views. The next step is to optimize the algebraic expression to improve efficiency. Finally, this algebraic expression can be converted into an SQL algebraic expression. For accomplishing this, [141] defines a general relational algebra for RDB-to-RDF mappings. The SQL query, which was obtained, is executed against the relational database. Using the defined mappings, the SQL result set returned by the relational database can be transformed to a SPARQL result set.

We present an example of *Sparqlification Mapping Language* or *Sparqlify ML* in Listing. 4.2.

[24] http://www.w3.org/TR/r2rml/

```
1
2   /* This is a comment
3   * /* You can even nest them! */
4   */
5   // Prefixes are valid throughout the file
6   Prefix dbp:<http://dbpedia.org/ontology/>
7   Prefix ex:<http://ex.org/>
8
9   Create View myFirstView As
10      Construct {
11          ?s a dbp:Person .
12          ?s ex:workPage ?w .
13      }
14  With
15      ?s = uri('http://myexample.org/person', ?id) // Define ?s to be an URI generated from the
            ↪ concatenation of a prefix with mytable's id–column.
16      ?w = uri(?work_page) // ?w is assigned the URIs in the column 'work_page' of 'mytable'
17  Constrain
18      ?w prefix "http://my–organization.org/user/" // Constraints can be used for optimization, e.
            ↪ g. to eliminate unsatisfactory joining conditions
19  From
20      mytable; // If you want to use an SQL query, the query (without trailing semicolon) must be
            ↪ enclosed in double square brackets:
21
22      [[SELECT id, work_page FROM mytable]]
```

Listing 4.2: An exmaple of Sparqlification Mapping Language (SML).

Data modeling step that aims at transforming the structure of the Statistical Data (SD) into RDF format by reusing available vocabularies/ontologies to define namespace prefixes. At this step, *URIs* play an important role in the discovery and interoperability of SD in the web space. To represent the LOSD in RDF, we choose the *W3C RDF Data Cube Vocabulary*[25]. Thus, we obtain LOSD and store it in a *PostgreSQL* relational database on the docker volume.

The user is provided with a Workspace file that consists of a Docker Sharing Volume in order to share and facilitate access to his data. Then, it is necessary to define a data structure to model Statistical Data in RDF format in an **automated** fashion. For the data transformation from the relational database to the *RDF Data Cube*, we define a mapping comprised of two views: **(i) Static** and **(ii) StatResults** [61].

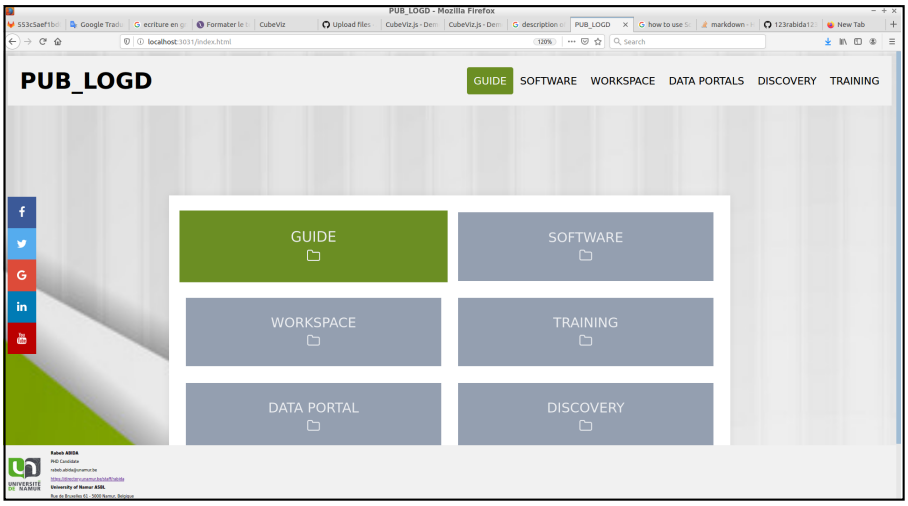
(i) **The *Static view*** describes the static elements of the RDF Data Cube Model: *qb:DataSet*, *qb:DataStructureDefinition* and several *qb:component* elements. There are five *qb:component* elements in *qb:DataStructureDefinition* with one associated *qb:Dataset*. Three *qb:dimension* elements (time of measure, statistical criterion, source dataset) explicitly identify any particular observation. This model includes exactly one *qb:measure* and one *qb:attribute* element.

(ii) **The *StatResults view*** operates on the statistical data from the relational database. It retrieves available statistical criteria from the database and constructs

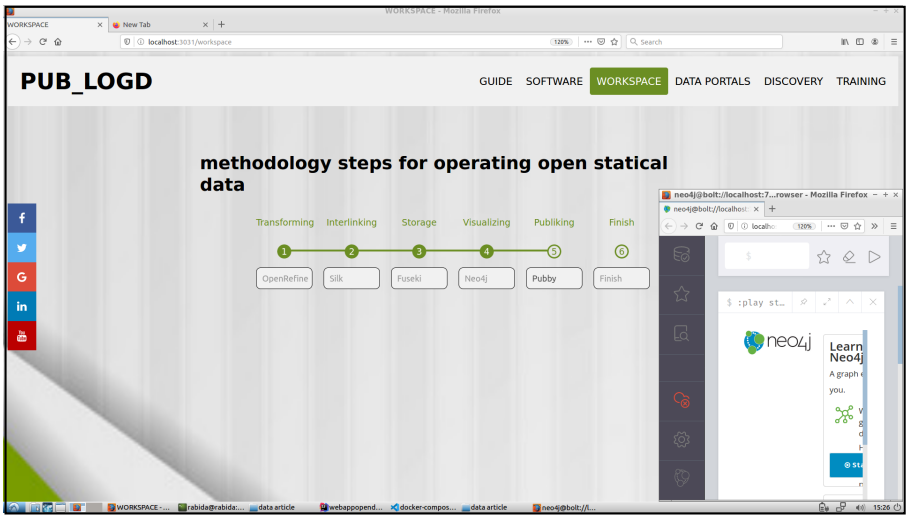
[25] https://www.w3.org/TR/vocab-data-cube/

an observation for each. Observation *URIs* are created by combining a Linked Open Statistical Data (LOSD) namespace URI with a unique hash.

More details about the complete listing of the mapping are available at our GitHub repository [26]. We have enabled the import of vocabularies available on the web, regardless of the format used (e.g. RDFa, RDF/XML, Turtle), as long as their deployment is compatible with the best practices recommended by the *W3C*.



(a) Home page



(b) Pub-LOGD framework process

Figure 4.3: Screenshot from the platform of Pub_LOGD framework process.

The platform **Pub_LOGD** is a Web Application built using the *SpringBoot frame-*

[26] https://github.com/123rabida123/WebAppOpenData

work and powerful Linked Open Data Technology and Docker technology. Figure 4.3 shows a screenshot of the **Pub_LOGD** process containing most of the tools that were requested by users in the pre-survey. Our proposal significantly reduces the required time for the right tool research and combination to model Statistical Data (SD) in RDF format and publish it on the web.

4.3.3 Threads to Validity

Some potential threats to validity exist in our study. Firstly, our framework is based on the use of a set of predefined tools. We certainly need these tools yet the user is not confused, neither in the choice of software ensuring the same functionalities nor in the dependencies between them, in order to achieve his goal. Besides, we have added the visualizing step that is fulfilled through *neo4j* which clarifies data knowledge. Additionally, we aim at extending each of these tools with novel functionalities that further customize the framework to meet user requirements. We extend the framework by improving the functionality of the data modeling. We plan to add an automated framework that ensures the interoperability of LOGD at the beginning of our proposed process.

Secondly in Section 7.2, we shown **the feasibility of Pub_LOGD and its extension** through a real case study from a local government datasets. This work requests experimentation to further evaluate the effectiveness of our proposal. Indeed, the qualitative and quantitative validation came from user panels, including both experts (data analysts and researchers) and non-experts (without skills in the field of the Semantic Web). Indeed, we engage the user community to use the **Pub_LOGD framework** and complete the pre-survey to gather their feedback and contributions to improve certain features. Furthermore, this investigation **aims to raise awareness of the government to receive feedback from data consumers by engaging with a community, which is composed of a wide range of Open Data publishers**. Then, we ask users about the effectiveness and usefulness of the Pub_LOGD extension through a post-survey (See Annexes B).

Finally, our work aims at achieving semi-automated LOGD publishing. Manual intervention is important in this process to allow the users to be aware of their data as well as giving them access to eliminate unnecessary information, visualize semantically-enriched LOGD, and exactly detect the wrong step.

4.4 Conclusion

The aim of this chapter is to support semantics-aware analysis in LOD-enabled Government Systems, which is intended to enable Open Data publishers to have customized access to their LOGD and reuse their data. Semantic visualizations have also been provided to make users aware of the extracted knowledge to enhance decision-making.

As a first step in our work on this thesis, we proposed a complete process that covers all phases of the LOGD-lifecycle, iterative, reusable, and interactive with domain experts and application developers. It also enables the distribution, support,

and grouping of process steps in a coherent manner. We developed a pipeline-based framework to publish Linked Open Government Data (**Pub_LOGD**). Our framework aims at shifting the burden of processing, converting, interling, and publishing LOGD to data consumers rather than the government. It consists of an end-to-end solution that covers five main data analysis steps: (i) data transformation, (ii) data interlinking, (iii) data storage, (iv) data visualization, and (v) data publishing. This framework includes a user-friendly interface that can be executed either from a web browser or through a standardized RESTful web API in order to enable easy integration into more complex workflows. We engaged the user community to use the **Pub_LOGD** framework and complete the pre-survey to gather their feedback to improve the data modeling feature. Then, we presented an extension of the Pub_LOGD framework (**Pub_LOGD extension**) to assist Open Data publishers in automatically modeling data as Linked Open Data for publishing on the web. The research contributes to the practice by implementing the identified research questions into a framework and providing the source code on *GitHub* for use in upcoming work. Besides, we have shown some potential threats to validity in this study. This work requires an automated and easy-to-follow process for non-expert users (See Chapter 7.2).

In Chapter 5, we aim to extend the potential of this study to overcome the obstacles associated with OD. Moreover, Section 7.2 illustrates the application of our Pub_LOGD in a particular **Belgium Government Catalogue** in order to evaluate the usefulness of our solution by engaging community users to complete a survey based on TAM and present the experimental results. Section 7.3 presents the limitations of the **Pub_LOGD** framework and its extension (**Pub_LOGD extension**) in order to address these shortcomings in the near future and to cover all user requirements.

CHAPTER 5

INTEGRATING, DISCOVERING AND UNDERSTANDING LOGD BASED ON LOD SEMANTIC ENRICHMENT

"We don't need more information. We need more meaning."

— ***Paul Salopek***

5.1 Introduction

This chapter aims to address **web data quality issues**, such as noise, misspellings and missing or incomplete metadata, ambiguity in table cells (It's important to understand how the same entity is identified in different data sources, or the same entity in different rows), missing cell values, and missing significant columns (the correct information can be added to the existing dataset), and to improve **usability**, **discoverability**, and **understanding** of the data to make it ready to perform the analysis desired by **Open Data publishers** and **data analysts**. To address these gaps, many works analyzed in Section 2.4. Based on these analyses, a vast number of datasets without any semantic annotation being published on open data portals every day, probably because adding semantic annotations to data is a laborious, error-prone, challenging task for publishers and also because the majority of existing tools are not capable of automating the process and being both interoperable with open data portals. Hence, there is still **(i) no all-in-one approach that can handle them**, **(ii) no other features that can further support Semantic Table Annotation (STA)**, **(iii) and a lack of semantic annotation tools that integrate well with OD portals**.

We present **a methodology to tackle these shortcomings** is structured in three parts. First, we establish **a set of needs** that should be part of a tool-based strategy that can fulfill user expectations and automatically and accurately annotate open datasets using already-available technologies. Secondly, we outline a table 5.2, which provides a comparison with the proposed tools. Hence, our objective is to mitigate certain constraints associated with the **Pub_LOGD framework** 4, as well as **its restrictions outlined** in Section 7.2, via the following additions.

This section is composed of two contributions: (1) the **SINATRA** approach (SemantIc aNnotation AugmentaTion and RepAir) and (2) the **ODSAG** tool (Open Data Semantic Annotation and Graph) is an extension of SINATRA. As a first contribution, we implement the requirements in the first sub-section 5.3.1 using **an all-in-one automated approach named SINATRA**. Furthermore, **the SINATRA approach is enhanced by the use of the ODSAG tool**. The proposed method has the same procedural steps as SINATRA, while also including **a visualization component** to enhance the understandability and accessibility of the annotated dataset and improve some steps of the later approach. In addition, it is **an easy-to-use Chrome plugin** that works with any OD portal, in the second sub-section 5.3.2.

The **SINATRA** approach implements its features through the following four phases:

1) **data pre-processing** and **subject column detection**, are used to prepare the data inside the table and to detect the subject column (The Main Column);
2) **data annotation**, is used to automatically resolve spelling errors and missing or incomplete metadata;
3) **data repair**, is used to handle missing or incomplete cell value in the dataset;
4) **data augmentation**, is used to dynamically add meaningful columns and their corresponding cell values to the dataset.

The **ODSAG** tool is **an extension of the SINATRA approach**, which consists of four phases:

1) **data selection** consists in taking as input a URL to access a dataset on any open data portal or on the user's computer;
2) **data pre-processing**, is used to remove specific numeric columns;
3) **data annotation** is a practical expansion of the SINATRA technique, whereby the use of the Wikidata Knowledge Graph (KG) is employed;
4) **Graph generation**, is used to generate two graphs from annotated dataset to better discover relationships between data elements and to detect hidden relationships between them.

Finally, we evaluate their efficiencies using benchmark datasets, showing promising results of the SINATRA approach in terms of **F1 measurement and accuracy**. We use a real COVID dataset available on the Namur, Belgium portal to show the capability of their features by the ODSAG tool. **The target users of the approach are primarily data publishers who can use the tool to annotate their data and add additional information meet their requirements before publishing them, but can also be extended to Data analysts who can use the tool to self-annotate unannotated datasets found on OGD portals to improve their understanding before using**

them.

5.1.1 Research questions and Publications

The content of this chapter is based on the following research questions and the scientific publications:

Abida, R. and Cleve, A. (2022). Improving the Usability of Tabular Data through Data Annotation, Repair and Augmentation. in Book "Nordic Artificial Intelligence Research and Development", 4th Symposium of the Norwegian AI Society, NAIS 2022, Oslo, Norway, 2022,Springer International Publishing. Under review. [7].

Chokki, A. P., **Abida, R.**, Frénay, B., Vanderose, B., and Cleve, A. (2022). ODSAG: Enhancing Open Data Discoverability and Understanding through Semantic Annotation. in 2022 in Electronic Government, Suede. Under review. [42].

These scientific publications cover these research questions such as:

(i) RQ.2. "How can we design an approach and tool that supports semantic interpretation of Open Data OD?"
   (a) RQ2.1."How can we improve the quality and the interoperability of the open tabular data?"
   (b) RQ2.2. "How can we leverage the existing metadata to understand the original OD?"

5.1.2 Outline

This chapter is organized as follows: Section 5.2 presents the requirements identified on the basis of the weaknesses of the Pub_LOGD framework, users' expectations, the approaches and tools examined, to support these contributions (Sinatra approach and ODSAG tool). Section 5.3 is composed of two subsections to describe in detail the different phases involved and present their implementation. Section 5.5 evaluates these proposals, assesses the effectiveness of their phases and demonstrates a real use case. Section 5.6 discusses the implications of SINATRA and ODSAG, and the answer to the research questions. Finally, Section 5.7 summarizes the contributions of this chapter.

5.2 Requirements identified to support the contributions

In Section 3.6, we have described selected approaches and tools, **we noticed that these works did not have adequate coverage** and that **they satisfied all the requirements**. The identified requirements are a summary of existing features in the reviewed Approaches and tools (Table 5.2). The detailed requirements of **a SINATRA approach** and **an ODSAG tool** are presented in Table 5.1.

Table 5.1: List of requirements identified to implement a SINATRA approach and an ODSAG tool.

No	Requirements
R1	Facilitate the selection of data from any OGD portals
R2	Provide a pre-processing of selected data, this feature will allow user to clean and uniform Data inside the dataset
R3	Provide a subject column detection. This feature will allow to identify the main column and not to use the first column as the main column every time
R4	Provide an automatic annotation of data, which can enrich the entity class (CEA), entity property (CPA), and entity types (CTA) with the KG
R5	Provide an automatic data repair feature, which can add missing or incomplete values to the dataset
R6	Provide extension features to support semantic annotation approaches, which will allow the user to add new columns to the dataset
R7	Facilitate the export of the annotated data
R8	Generate a graph of annotated data, this feature will allow users to visually understand their datasets before applying them
R9	Gather **all functionalities in one fully automatic system**, which does not require the knowledge of expert users and reduces high costs in terms of time and money
R10	Easy to use and install

Table 5.2 provides a comparison with the proposed tools, SINATRA and ODSAG. The previous analysis of related tools and the state-of-the-art, presented in Section 3.4, revealed requirements **that should be met by implementing an all-in-one, and fully automated approach that can be easily interoperable with any OD portal**.

Table 5.2: List of examined tools for semantic annotation. - = Not to do , ✓= To do and (-/✓) = Depends on.

Tools	**R1**	**R2**	**R3**	**R4**			**R5**	**R6**	**R7**	**R8**	**R9**	**R10**
				CEA	**CTA**	**CPA**						
TableMiner+	-	-	✓	✓	✓	✓	-	-	-	✓	-	-
SemanticBot	Only OpenDataSoft	-	✓	-	✓	✓	-	-	✓	✓	-	✓
Open Refine	-	✓	-	✓	-	-	-	✓	✓	-	-	✓
Odalic	-	-	-	✓	✓	-	-	-	✓	-	-	-
DataGraft	-	-	-	✓	✓	-	-	-	-	-	-	-
MantisTable	-	✓	✓	✓	✓	✓	-	-	-	-	-	(-/✓)
MTab	-	✓	✓	x	✓	✓	-	-	-	-	-	✓
JenTab	-	-	-	✓	✓	✓	-	-	-	-	-	-
Magic	-	-	-	✓	✓	✓	-	x	-	-	-	(-/✓)
Dagobah	-	-	-	x	✓	✓	-	-	-	-	-	-
SINATRA [7]	All	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	(-/✓)
ODSAG [42]	All	x	x	✓	✓	✓	-	-	-	✓	✓	✓

Looking at Table 5.2, none of the reviewed tools are able to interoperate with Open Data portals, automatically annotate the data, add missing information, add an additional column, and generate a graph from the annotated data. This justifies the necessity of our SINATRA approach and ODSAG tool which, when compared to other tools, can meet all these features. Therefore, we implement these requirements in all-in-one and automatic approach that we will present in the next section.

5.3 Overview of contributions (SINATRA and ODSAG)

5.3.1 SINATRA approach

This section describes the SINATRA approach, which aims to address the associated difficulties with Web data quality, the shortcomings mentioned in the approaches and tools previously discussed in Table 5.2 , to incorporate the requirements identified in Section 5.2.
Instead of starting from scratch, this proposal adapts existing methods and tools *Mtab* [142] and *MantisTable* [48] into an integrated system by adding certain functionalities to STA approaches. **It implements its features: data pre-processing and subject column (sub_col) detection (the main column), data annotation, data repair, and data augmentation**, which **Figure presents an overview of the proposal**.

1) **Data pre-processing** and **subject column** detection, which aims to prepare the data inside the table and to analyze the type of column to detect the subject column;
2) **Data Annotation**, which focuses to automatically annotate Tabular data elements with DBpedia KG;
3) **Data repair**, is based on the automatic addition of the missing cell entities or data type to the column;
4) **Data augmentation**, which provides the user with the possibility to add meaningful columns to the annotated dataset;

Figure 5.1 presents an overview of the SINATRA approach and the following paragraphs describe each of these phases in more detail.

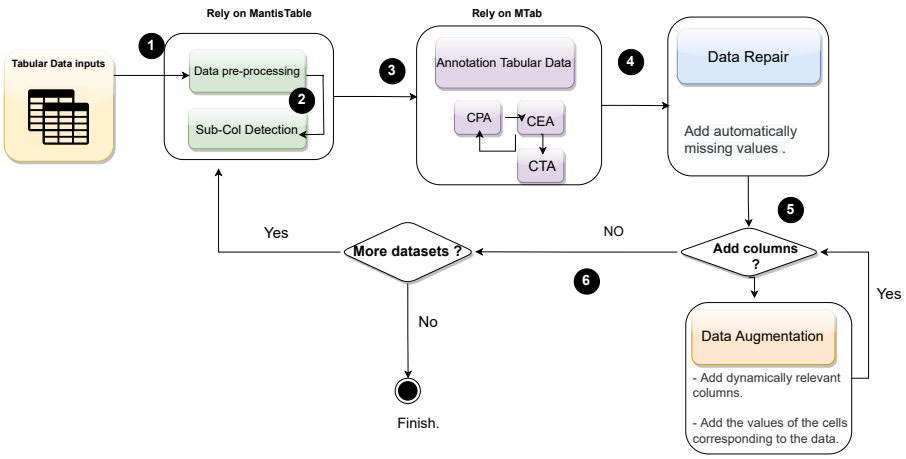


Figure 5.1: An overview of the SINATRA approach.

1. During the **data pre-processing** and **subject column (sub_col) detection** phases, the **SINATRA** approach allows the users to automatically prepare the datasets and detect the **sub_col** before applying the data annotation phase. This phase is based on the *MantisTable approach* [48] and consists of two steps: (i) data preprocessing

and (ii) sub_col detection.

(i) **Pre-processing step**, it aims at providing a good quality annotated result, because the result is strongly influenced by the quality of the input data. A pre-analysis of the data is, therefore, necessary and is often the first step of an effective STA system (Section 6.3). This step is advisable to apply standard transformation rules to the values and the structure of the table so as to preserve only the relevant information (**R2**). Examples of transformation are: deletion of HTML tags, stop words and some characters (i.e. " '), transformation of text into lowercase, deletion of text in brackets, additional white-spaces, explanation of acronyms and abbreviations, and normalisation of units of measurement. The Oxford English Dictionary [1] is used to decrypt many acronyms and abbreviations. The normalisation of units of measurement is performed by applying regular expressions. They normalize units of measurements using around 200 manually generated conversion rules and using about 100 manually defined regular expressions to detect string, numeric value, timestamp and coordinate. The final data type decision for an attribute is based on majority vote, as described in [159]. MantisTable approach [17, 46, 48] extends the original set of regular expressions to cover a complete set of units, which includes area, currency, density, electric current, energy, flow rate, force, frequency, fuel efficiency, information unit, length, linear mass density, mass, numbers, population density, power, pressure, speed, temperature, time, torque, voltage and volume. For each type of unit of measurement, a set of conversion rules has been defined, like the ones presented in Listing 5.1, which shows the equivalent ways to express measures of area.
Table of Figure 5.2 shows the transformation of Table after the completion of the Data pre-processing step.

```
1  area | squareMetre | squaremetre, m2, m^2, µ2, µ^2, Γ.µ.
2  area | squareMillimetre squaremillimetre, mm2, µµ^2, µµ^2
3  area | squareCentimetre squarecentimetre, cm2, cm^2
4  area | hectare | hectare, ha
5  area | squareMile | squaremile, sqmi, mi2, mi^2 | 2589988.110336 (COORD)
```

Listing 5.1: Conversion listing for units of measurement of area [46].

[1] http://public.oed.com/how-to-use-the-oed/abbreviations/.

col1	col2	col3
Barack Obama	Honolulu	1961-08-04
Donald Trump	Queens	1946-06-14
George W.Bush	New Haven	1946-07-06
Bill Clinton	Arkansas	1946-08-19

(a) Before pre-processing

col1	col2	col3
barack_obama	honolulu	1961-08-04
donald_trump	queens	1946-06-14
george_w.bush	new_haven	1946-07-06
bill_clinton	arkansas	1946-08-19

(b) After pre-processing

Figure 5.2: The table presents a list of the presidents from T2Dv2 Gold Standard.

(ii) Once this step is complete, the system switches to **detect the sub_col** (**R3**). Firstly, the process assigns **Literal columns** (L-column) for datatype values (e.g. strings, numbers, dates, such as 4808, 11/05/1993), or **Named-Entity columns** (NE-column) for concept values (e.g. president, country, person, description, such as Barack Obama, Hanolulu). The activity requires the identification of good L-columns candidates. The method exploits Regular Expressions (RegEx) to check if the content of the cells of a column can be classified as empty, date, number and long text as described in [111, 137, 155, 159, 196]. In addition, the set of RegEx is extended with 14 new Regextypes: geo coordinates, iso8601 date, street address, hex colour, url, image file, credit card, email address, ip address, isbn (International Standard Book Number), boolean, id, currency and iata (International Air Transport Association) codes. If the number of occurrences of the most frequent Regextypes detected exceeds a given threshold, the column will be annotated as L-column and the most frequent Regextype will be assigned to the column under analysis; otherwise the column will be annotated as NE-column. For example, in the Presidents Table 5.2, this step assigns the **L-column** to the Birthdate (Col2) column,

and the **NE-column** to the President (Col1) column. Secondary, the process considers the subject column (**sub_col**) detection that takes into account the NE-columns. The **sub_col is the main column in the table: it contains entities for which the other columns provide additional information. To identify the sub_col, the method considers the following list 5.3 of features**: (i) fraction of empty cells (emc), (ii) fraction of cells with unique content (uc), (iii) distance from the first NE-column (df), and (iv) average number of words (aw). The first three are taken and adapted from [196], while the fourth is taken and adapted from [183]. In contrast to [196], *MantisTable* does not consider features that are expensive (in computational, economic, and maintenance terms), such as the Web search (ws) and context match score (cm). The former requires the use of a search engine, while the latter analyses the external contexts (e.g. webpage title, table caption and surrounding paragraphs). The work in [183] inspired them to consider the average number of words (aw) feature, which calculates the average number of words within the cells of each column.

Table 5.3: Features for subject column detection.

Feature	Notation
Fraction of empty cells	emc
Fraction of cells with unique content	uc
Distance from the first NE-column	df
Average number of words in each cell	aw

It has been empirically demonstrated that the best candidate is the column with the highest average number of words per cell. The four features, after the normalisation, are combined to compute the $sub_col(c_j)$ score for each **NE-column** using Formula (5.1), which is an adaptation (due to the different features considered) of the one presented in [196].

$$\frac{2uc(c_j) + aw_{norm}(c_j) - emc_{norm}(c_j)}{\sqrt{df(c_j)+1}}. \tag{5.1}$$

Table 5.4: Values of the features of the sub_col detection for the president table.

Feautres	Col1	Col 2
emc	0	0
uc	1	0.1
df	0	1
aw	2	1.2
sub_col	4	1.6

The column with the **highest score** will be selected as the sub_col for the considered table. Table 5.4 shows the values of the features related to the presidents table 5.2, and the sub_col scores that identify the col1 as the sub_col of the table 5.3 . In order to promote columns with unique values, as a typical

characteristic of the object columns established in [183], more weight is given to the *uc*. We summarize the subject column (sub_col) in Figure 5.3 based on different statistic features, such as the average number of words in each cell, fraction of empty cells in the column, the fraction of cells with unique content, and distance from the first-named entity column [17, 48]. Once the phase has finished, it moves on to the second phase, which consists of annotating the dataset.

2. The Data annotation is used to resolve spelling errors and ambiguous, and missing or incomplete metadata.

Using the Open Data portal, it is important to understand **how the same entity is identified in the various data sources, so that the correct information can be exploited and added to the existing dataset**. *For example, "Commune", "Nom de localité", or "Quartier statistique"*[2] *for the column name; and for the entities "Velaine" and "Jambes Velaine"*[3] *might refer to different entities, such as geographical location or organizational division.*

This phase is based on the **Semantic Table Annotation** (STA) process to automatically annotate Tabular Data (TD) elements with **DBpedia KG** (**R4**), which consists of three main tasks: Column type Annotation (**CTA**), Column property annotation (**CPA**) and Column Entity Annotation (**CEA**) . They assigned the data elements to the concepts in the knowledge graph (DBpedia KG), as shown in Figure 5.3 to describe each task in the STA process [142], we consider a table of real dataset in Figure 5.3, which presents the names of the presidents (col1) and their place of birth (col2).

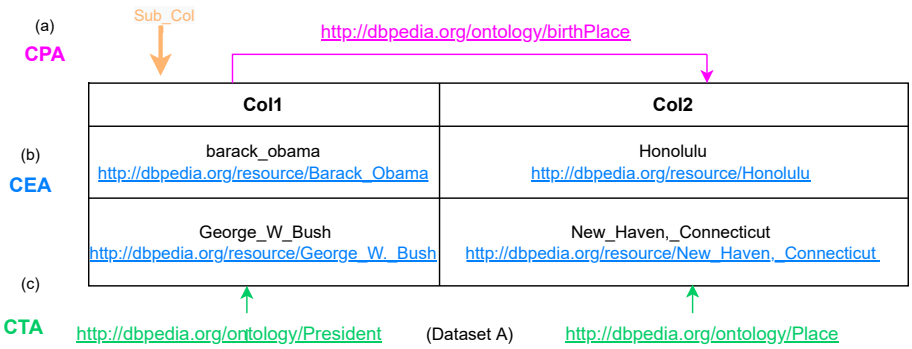


Figure 5.3: Data annotation. Tabular Data (black) is annotated with the properties (magenta), entities (blue), and types (green) from DBpedia as asked in the CPA (a), CEA (b), and CTA (c) tasks respectively.

This phase relies on the MTab approach [142] to generate the three tasks: the Column Entity Annotation (**CEA**), whose task is to map table cells (values) to entities in DBpedia (Figure.5.3b); the Column property annotation (**CPA**) to map column-

[2] https://tinyurl.com/mreeyufe
[3] https://tinyurl.com/mrywz2jp

pairs to an ontology property (Figure.5.3a); and the Column type Annotation (**CTA**) whose task to map table columns to an ontology class (Figure.5.3c). The mapping process in **MTab is based on the joint probability distribution of multiple tables to KG matching**. It improves the matching by using multiple services including, DBpedia Lookup, and DBpedia endpoint, as well as a cross-lingual matching strategy. We consider the MTab mapping process to be done in six steps in the design. 5.4, as we used the pre-processing step of MantisTable [48], and more explained in Section 3.4.2.

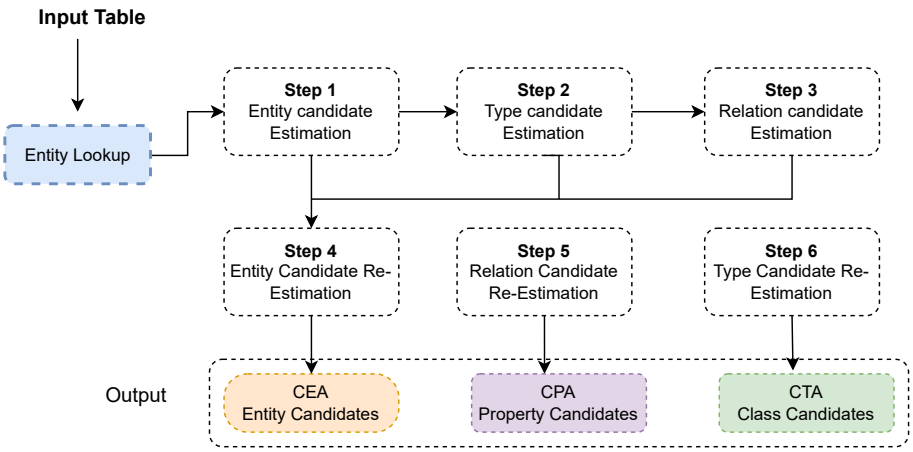


Figure 5.4: The design of MTab method.

(i) The first step is **to estimate the most candidate entities** (**CEA**) that were found by those different search services. Given a cell value $c_{i,j}$, they have a set of ranking result lists from lookup services $Q_{c_{i,j}}$. q is a list of ranking of entities ordered by degree of relevance of a lookup service, where q $\in Q_{c_{i,j}}$. In MTab, they adopted the four services as DBpedia lookup [4], DBpedia Endpoint [5], Wikidata lookup, and Wikipedia lookup. However, they can use any services as long as their output is a ranking list of relevance entities. Denote $\overline{E}Q_{c_{i,j}}$ is a set of relevance entities in $Q_{c_{i,j}}$, s_e^q is a confidence score of an entity e where e $\in \overline{E}Q_{c_{i,j}}$. The confidence score of entity e is calculated as s_e^Q = max(s_e^q). s_e^q is the confidence score of entity e in q, $s_e^q = \alpha - rank_e$, where $rank_e$ is the ranking index of entity in q. They normalize those entity confidence score to [0, 1],
where $Pr(\overline{E}Q_{c_{i,j}}|Q_{c_{i,j}}) = 1, Pr(e|Q_{c_{i,j}}) = \frac{s_e^Q}{\sum_{e\in\overline{E}Q_{c_{i,j}}}} s_e^Q$ and associate those scores as the potential probability of entities given lookup results.

(ii) The second step is **to estimate the Candidate type by inferring the most classes** (**CTA**). They categorize table columns to entity columns and literal

[4]DBpedia Lookup, link: https://wiki.dbpedia.org/Lookup
[5]DBpedia Endpoint, link: https://dbpedia.org/sparql

columns. They aggregate the data type (Duckling Tags and SpaCy Tags) from each cell in a column using majority voting. If the majority tag is text or entity related, the columns is an entity column, else a numerical column. Regarding numerical columns, they perform semantic labeling with EmbNum method [142] to annotate relations (DBpedia properties) for numerical columns [6]. Then, they infer types (DBpedia classes) from these relations. **Numerical Column**, The set of numerical columns in table S is M_{num} . Given a numerical column m_j , they use re-trained EmbNum model on DBpedia [144] to derive embedding vector for the list of all numerical values of the column and then search the corresponding relations from the database of labeled attributes [7]. The result q_{m_j} is a ranking of relevance numerical attributes in terms of distribution similarity. They also use α[8] as the limit for ranking result. The confidence score of a relation r is calculated as $s_r^{m_j} = \alpha - rank_r$, where $rank_r$ is the ranking index of r. These scores are also normalized to a range of [0,1] to associate the probability of potential of relation for numerical columns $Pr(r|m_j)$. Next, we use DBpedia endpoint to infer the classes (types) from the relations. $T_{q_{m_j}}$ is a set of inferred types, t denotes a type $t \in T_{q_{m_j}}$. The types will be used for entity columns. The confidence score of types is estimated as $srt = max(S_r^{m_{num}})$. Then, they normalized the scores to [0,1] so that $Pr(T_{qM_{num}}) = 1$, the confidence scores are associated as the probabilities of type potential $Pr(t|M_{num})$ given M_{num}. **Entity Column**, Given a set of entity columns in table S is M_{ent}, we consider these signals from:

(i) $Pr(t|M_{num})$: the probabilities of type potential from numerical columns.

(ii) $Pr(t|m_j, Q_{m_j})$: the probabilities of type potential aggregated from the types of entity lookup for the all cells in column m_j ($Pr(t|m_j, Q_{m_j})$) = $\sum_{c_{i,j \in m_j}} Pr(t|Q_{c_{i,j}})$). We normalized these aggregated potentials and associates these as potential probabilities.

(iii) $Pr(t|m_j, SpaCy_{m_j})$: the probabilities of type potential aggregated from *SpaCy* entity type prediction for the all cell in column m_j . They used majority voting and normalized these voting value to [0,1]. Then, they associate these normalized voting value type potential probabilities.

(iv) $Pr(t|c_{1,j})$: the probabilities of type potential given header value of the column m_j . They associate the normalized Levenshtein distance as potential probability that a type (DBpedia class) correspond with a header value.

The probabilities of type potential is derived from the four signals as $Pr(t|m_j) = w_1 Pr(t|M_{num}) w_2 Pr(t|m_j, Q_{m_j}) w_3 Pr(t|m_j, SpaCy_{m_j}) w_4 Pr(t|c_{1,j})$,
where $w1, w2, w3, w4$ are learnable weights. Note that, some probabilities of signals might be 0 or too small, and aggregate those might add too much noise to the final aggregation. Therefore, if any signal probabilities less than β[9], they omit those signals. After aggregation, they also perform normalization for

[6] They only use EmbNum for these columns have at least 10 numerical values
[7] They used all numerical attributes of DBpedia as the labeled data
[8] they set $\alpha = 100$
[9] $\beta = 0.5$

$Pr(t|m_j)$ to a range of [0,1] so that $Pr(T_{m_j}|m_j) = 1$

(iii) The third step **establishes the relationships between the different columns** (**CPA**). Given two columns m_{j1} and m_{j2}, they estimate the probabilities of relation potential of $Pr(r|m_{j1}, m_{j2})$. They consider two type of relation between two columns: Entity column to Entity column and Entity column to non-Entity column. To be simple, they associate the first entity column is m_{j1}. If the second column is entity column, we denote it as m_{j2}^{ent}, else $m_{j2}^{non-ent}$.
Entity - Entity columns $Pr(r|m_{j1}, m_{j2}^{ent})$: Given $c_{i,j1}$ is a cell value of the column m_{j1} and the row r_i, $c_{i,j2}$ is a cell value of the column m_{j2}^{ent}. They assume that there is a relation between entity candidates of $c_{i,j1}$ and $c_{i,j2}$, therefore they use DBpedia endpoint to find how many links (relations or properties) between entity candidates of $c_{i,j1}$ and $c_{i,j2}$. The confidence score of relation is calculated as $S_r^{c_{i,j1},c_{i,j2}} = 1$ if there is any relation between entity candidates of two columns. Then, they aggregate those scores of all rows to get the candidate score for two columns as $S_r^{m_{j1},m_{j2}^{ent}} = \sum_{i\in[1,N]} S_r^{c_{i,j1},c_{i,j2}}$. Then, they normalize those score to a range of [0,1] so that $Pr(R_{m_{j1},m_{j2}^{ent}})$ and associate it as the probability of relation potential of Entity and Entity Columns $Pr(r|m_{j1}, m_{j2}^{ent})$.
Entity - Non-Entity columns $Pr(r|m_{j1}, m_{j2}^{non-ent})$: Given $c_{i,j1}$ is a cell value of the column m_{j1} and the row r_i, $c_{i,j2}$ is a cell value of the column $m_{j2}^{non-ent}$. They estimate the relevance ratio between entity candidates and non-entity value $c_{i,j2}$. Given an entity candidate e have pairs of relation (r_e)-values(v_e), they compare the non-entity value $c_{i,j2}$ with all attribute values v_e. They select those pairs have ratio larger than β. They only compare two values of $c_{i,j1}$ and v_e based on there data types (textual type or numerical type).

- For textual values: They use *the normalized Levenshtein distance* to estimate the relevance ratio between v_e and $c_{i,j2}$ as $s(v_e,c_{i,j2})$.
- For numerical values: the relevance ratio is calculated as:

$$s(v_e,c_{i,j2}) = \begin{cases} 0, if & max(|c_{i,j2}|,|v_e|) = 0 \quad and \quad |c_{i,j2} - v_e| \neq 0 \\ 1, if & max(c_{i,j2}|,|v_e|) = 0 \quad and \quad |c_{i,j2} - v_e| = 0 \\ & 1 - \frac{|c_{i,j2}-v_e|}{max(|c_{i,j2}|,|v_e|)} \end{cases}$$

They aggregate all relevance ratio with respect to relations. Then they normalize those aggregated ratio to [0,1], and associate this as probability of relation potential. $Pr(r|m_{j1}, m_{j2})$. If the column of m_{j2} is numerical columns, they also aggregate the re-calculated probability from $Pr(r|m_{j2})$ (step 2) as:

$Pr(r|m_{j1}, m_{j2}, m_{j2}$ is numerical) = $w_5 Pr(r|m_{j1}, m_{j2}) w_6 Pr(r|m_{j2})$,
where w_5, w_6 are learnable parameters.

(iv) Step four is **the re-estimation of the highest probabilities of the candidates** (**CEA**). They present a method to re-estimate the probabilities of entity candidates $Pr(e|S)$. Given a cell $S_{i,j}$ containing a cell value $c_{i,j}$ at row n_i, and column m_j , they consider these signals from:

- $Pr(e|Q_{c_{i,j}})$: The entity candidate probabilities given look up results.
- $Pr(e|m_j)$: The probabilities of entity candidates given their type's probabilities (Step two). This can be estimated by $Pr(e|m_j) = max(Pr(t_e|m_j, Q_{m_j}))$, where t_e is a type of the entity e.
- $Pr(e|c_{i,j})$: The probabilities of entity candidates given the cell value $c_{i,j}$. They get the mean ratio of the normalized *Levenshtein distance*, heuristic abbreviation rules (first character of words, titles, dates, time).
- $Pr(e|n_i, m_{j1})$: The probabilities of entity candidates given cell values in a row $c_{i,j} \in n_i$. We do the same procedure as Step three to compare all entity values with a cell value, and compute the mean probability for all cell value in a row. $Pr(e|n_i, m_{j1}) = mean(Pr(e|m_{j1}, m_{j2}))$, where $j1 \neq j2$.

Overall, the equation is as follows:
$Pr(e|S) = w_7 Pr(e|Q_{c_{i,j}}) w_8 Pr(e|m_j) w_9 Pr(e|c_{i,j}) w_{10} Pr(e|n_i, m_{j1})$.

(v) Step five is **the selection of the highest probabilities of the candidates** (**CEA**) in step four for each cell $S_{i,j}$ to re-estimate their relationship (**CPA**) via a majority vote.

(vi) Step six corresponds **to the selection of the highest probabilities of the candidates** (**CEA**) in step four for each cell $S_{i,j}$ to re-estimate their type (**CTA**) via the majority vote. More details about each step of MTab can be found in [142]

Our contribution in the first two phases is that combined the strengths of MantisTable and MTab to perform both sub-steps. MTab does not offer a sub_col detection phase but has excellent results in annotating data solves misspelling issues; and MantisTable does not offer excellent results like MTab but allows sub_col detection. These two phases address identified requirements such as: **R2**, **R3** and **R4**. Once the **data annotation phase** completes, we get a well annotated dataset. During the processing of this phase, **we discover that some cells in this dataset still have blank values (null)** *"nan"*. We can observe the MTab system's shortcoming, which **cannot add missing cell values** in this dataset, as shown in this example (See Figure 5.10, section(c)). In the next section, we apply extensions to support Semantic Table Annotation system in order **to improve the usability of tabular Data** and meet the requirements identified: **R5** and **R6** .

3. The Data repair aims at automatically adding **the missing entity** into annotated datasets or **undefined entity** (value) "*nan*", as show in Figure 5.5 .

http://dbpedia.org/ontology/President	http://dbpedia.org/ontology/birthPlace
http://dbpedia.org/resource/Barack_Obama	
http://dbpedia.org/resource/George_W._Bush	http://dbpedia.org/resource/New_Haven,_Connecticut

http://dbpedia.org/ontology/President	http://dbpedia.org/ontology/birthPlace
http://dbpedia.org/resource/Barack_Obama	http://dbpedia.org/resource/Honolulu
http://dbpedia.org/resource/George_W._Bush	http://dbpedia.org/resource/New_Haven,_Connecticut

Figure 5.5: Example of data repair by adding cell value "http://dbpedia.org/resource/Honolulu" (light green).

We explain in this section, how to estimate **the best entity candidate** to add in missing cell value c_{ij}. The final objective is to identify the entity with **highest confidence score** $HcfSr_e = Max(s_e)$ **to optimize the ambiguous entities**. We adopte these services in this phase as DBpedia Lookup, DBpedia Endpoint, DirectLink, DBpedia SpootLight, Edit Distance Ratio, Fuzzy Ratio and Service Engine to calculate the similarity and get the score of each entity.

Related to the President Table in Figure 5.5 with the query to extract e_{ij} of cij from DBpedia KG . There is a relation $(R(m_{j1}, m_{j2}))$ (predicate/ property) between two columns, where $r \in R$, the sub_col and second column. We assume par example that there is a relation (property) between the subject-entity and entity-candidate $r(e_{i,j1}, e_{i,j2})$. We use these services to find the missing cell value (the object) $c_{i,j2}$.

The query return E_{ij} a Set of relevance entities in a Set of Score result List (S_E) from services and a set of Score (S_E) that represent the similarity between the name of entity $e_{ij} \in$ DBpedia KG and the cell value c_{ij} by using the Services: Edit distance Ratio, Fuzzy Ratio and service Engine, Where $e_{ij} \in E_{ij}$ and $s_e \in (S_E)$. Then, $HcfSr_e = Max(s_e)$. Before Computing **the highest confidence score** $HcfSr_e$, all scores S_E have to be normalized by diving each score s_e by the maximum value obtained for that score S_E, where, $\frac{s_e}{Max(S_E)}$,[0,1].

$HcfSr_e$ is used to retrieve the best entity candidate. We assume that two entities have the same score s_e, then we compute **the highest occurrence of this property with the subject entity** $e_{i,j1}$ and the entity $e_{i,j2}$. The highest confidence score $HcfSr_e$ of this entity increases by **1**. In this case, the entity has the highest score $HcfSr_e = s_e + 1$. We conclude here, that there are **mutual influences** between **CEA** and **CPA** tasks to resolve ambiguities entities.

The SPARQL query (Listing 5.2) retrieves an object e_{ij} for the content of the column "http://dbpedia.org/ontology/birthPlace" (Property/Predicate) and the subject-entity of the first row "http://dbpedia.org/resource/barack_Obama" from DBpedia KG, where the cell entity e_{ij} (object) retrieved by this query (Listing 5.2) is "http://dbpedia.org/resource/Honolulu" ((Figure. 5.9, section(c))).

```
1  {
2  PREFIX dbr: <http://dbpedia.org/resource/>
3        SELECT ?object
4         WHERE
5            { <http://dbpedia.org/resource/barack_Obama>
6              <http://dbpedia.org/ontology/birthPlace>
7              ?object
8            }
9  }
```

Listing 5.2: SPARQL query to retrieve a cell entity e_{ij} (Object).

The example of query returns ambiguous entities. Our algorithm calculates the scores of the entities $s_e(Honolulu) = 1$ and $s_e(Honolulu) = 1$, thus it computes the highest occurrence of this property ("http://dbpedia.org/ontology/birthPlace") with the subject-entity $e_{i,j1} = "barack_Obama"$ and the entity $e_{i,j2} = "Honolulu"$, increased by **1**. Hence, the highest confidence score is $HcfSr_e(Honolulu) = 1+1 = 2$ of the entity "Honolulu". This phase is provided an automatic data repair feature to add and complete missing values, which meets the requirement identified (**R5**) to support the **SINATRA** approach.

3. During **the Data augmentation** phase, the system allows the user to extend the annotated dataset with **new information by adding new columns** (Figure 5.6).

http://dbpedia.org/ontology/President	http://dbpedia.org/ontology/birthPlace
http://dbpedia.org/resource/Barack_Obama	http://dbpedia.org/resource/Honolulu_
http://dbpedia.org/resource/George_W._Bush	http://dbpedia.org/resource/New_Haven,_Connecticut_

http://dbpedia.org/ontology/President	http://dbpedia.org/ontology/birthPlace	http://dbpedia.org/ontology/birthDate
http://dbpedia.org/resource/Barack_Obama	http://dbpedia.org/resource/Honolulu_	1961-08-04
http://dbpedia.org/resource/George_W._Bush	http://dbpedia.org/resource/New_Haven,_Connecticut_	1946-07-06

Figure 5.6: Example of *data augmentation* by adding new column "http://dbpedia.org/ontology/birthDate" (light blue).

We adopte a service as DBpedia Endpoint in this phase to obtain set of relations (or predicates). Denote, the relations between two columns the subject column and the new column $R(m_{j1}, m_{jNew})$, that contain the user's word, where $r \in R$. The user simply enters a word as the title (**CEA**) of the additional column "new-column" (Listing 5.3) to choose a property r for the additional column m_{jNew}) **CPA** from DBpedia KG (URI of the new column) in the proposal list of this approach.

For the same word (e,.g, new-column = "birth"), there can be several URIs (CPA) that appear in this list, such as: "http://dbpedia.org/ontology/birthDate" and "http://dbpedia.org/ontology/birthDeath". The user chooses the one **CPA** (r is a property) from the list and SINATRA approach will be added as a new column to the dataset. She/he can enter the name of the column exactly as "birth-

Date". The system provides the user with the most suitable property containing exactly this word. Therefore, it allows the user to add the chosen CPA property r is "http://dbpedia.org/ontology/birthDate", if this new column is not already in this annotated dataset (Figure. 5.9, section(d)).

The system has created a list of a set of CPA proposals, where, each time the SPARQL query (Listing 5.3) returns a CPA (**Predicate has an rdf:property**), which contains a word proposed by the user, it stores it in this list.

```
{
PREFIX dbr: <http://dbpedia.org/resource/>
      SELECT ?predicate
       WHERE {
                ?predicate a rdf:Property
                FILTER ( REGEX ( STR (?predicate), http://dbpedia.org/ontology/, i) )
                FILTER ( REGEX ( STR (?predicate), " + new–Column + ", i ) )
              }
ORDER BY ?predicate
}
```

Listing 5.3: Generic query to detect predicates from a SPARQL endpoint to add column.

Once the user chooses a CPA (property), the system creates a new empty column (m_{jNew}) and then applies the same SPARQL query of the *data repair* phase to fulfill the corresponding cell entities of the newly added column.

5.3.2 ODSAG Tool

In this section, we improve **the SINATRA approach** by implementing the **ODSAG tool**, which is **user-friendly chrome extension** that can be easily interoperable with any OD portal and **allows generating graphs from the annotated data**. It aims to improve data analysts' understanding of the open dataset. **ODSAG tool** is an enhancement of the SINATRA approach to incorporate the identified requirements. It consists of four phases (Figure 5.7):

1) **Data selection**, which takes as input a URL to access a dataset on any Open Data portal or on the user's computer, which is Facilitate data selection (**R1**);
2) **Data pre-processing**, which removes some exceptional numeric columns (**R2**),
3) Data Annotation, which is the second phase of the SINATRA approach to automatically annotate Tabular Data (TD) with **Wikidata KG** (**R4**);
4) **Graph generation**, which aims to create two graphs (minimal graph and full graph) from the annotated data (**R8**).

Figure 5.7 presents an overview of the prototype composed of these steps and the following paragraphs describe each step in more detail.

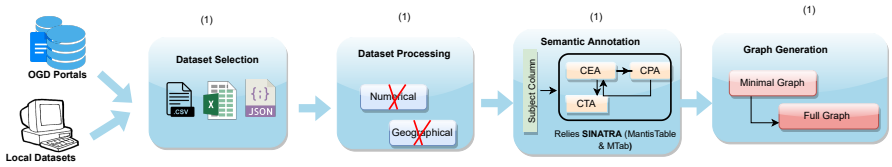


Figure 5.7: An overview of the ODSAG.

1. During the **dataset selection** phase, ODSAG takes as input an URL to access a dataset in Excel, CSV or JSON format on any open data portal (e.g., Data.gov, ODWB, Portal data of Namur, CKAN, OpenDataSoft, Socrata, DKAN) or on the user's computer (Figure 5.10). It allows users to interact with any Open Data portal. The user copy the URL link of the desired open dataset and paste it into the tool's URL field or he can load the file from his computer.

2. During the **data pre-processing** phase, ODSAG removes same numeric columns (except columns containing *postal code*, *NSI code*, etc., which have a match in Wikidata), as well as geographic columns, because currently most numeric and geographic values in datasets available on OD portals (e.g., the number of cases column, *limite communale* or *geo_point_2d* in the COVID data) do not have a match in the used KG: **Wikidata** (Figure 5.10).

2. The **semantic annotation/data annotation** phase aims at enabling users to automatically match open dataset elements into Wikidata concepts. This phase is divided into two sub-steps: **(i) the detection of the subject column** and **(ii) the generation of CEA, CPA and CTA As for the annotation** (See Figure 5.8 and Figure 5.10).

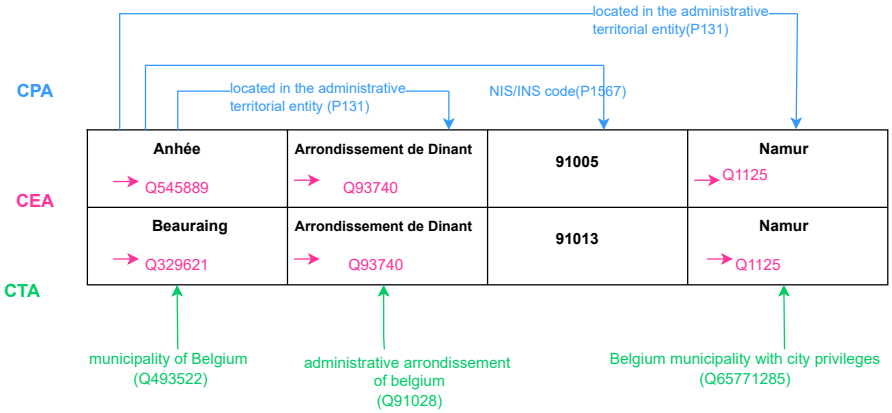


Figure 5.8: Data Annotation. Tabular Data (black) is annotated with the properties (blue), entities (magenta), and types (green) from Wikidata as asked in the CPA, CEA, and CTA tasks respectively.

For the detection of subject columns, we use the same procedure as described in the first phase of SINATRA using the MantisTable approach. More details on the

subject column detection can be found in [17, 46, 48]. Once the subject column detection step is complete, we rely on Mtab [145] to automatically generate the three semantic annotation tasks including CEA, CPA and CTA. Unlike the previous system MTab (DBpedia matching) based on the joint probability distribution of table element signals. This work focuses **on improving the entity search capability** by **using Wikidata item history editing to enrich the Wikidata dump to improve matching performance**, using **edit distance to build a new fuzzy entity lookup** to handle the high level of spelling errors in table cells and by building a novel fuzzy statement search that could find relevant statements from two cells (cell ambiguity) taken from a table. This searcher reduces the number of relevant candidates for ambiguous queries. The mapping process in Mtab (Wikidata matching) is done in 3 steps.

(i) **Step 1** involves generating Wikidata resources using the Wikidata entity dump and history revisions, as well as creating two indexes for fuzzy entity search and fuzzy statement search.

- Extracting Wikidata items and these statements: They extract this information from the Wikidata 29 June 2020 entity dump. Note that this entity dump is not available on the Wikidata server now, since they only keep the two months' latest dumps.
- Getting history revision of Wikidata items: They crawl Wikidata item history revisions as a statement form (Subject - Predicate - Object) from 1 March to 29 June 2020 [10] .They use those revisions to enrich the Wikidata dump statements. Since they do not know when the challenge's tabular data is generated, they only use the adding operation for all history statements of Wikidata items. In other scenarios, where they know the exact time point of Wikidata, they can reconstruct Wikidata items at this time point by using adding, removing, modifying operations.
- Building an entity index: They build this index using a hash table with the entity dump data using multilingual labels, aliases, and identifiers (the identifiers to other sources). In total, the fuzzy entity search index about 150 million entity names for 87 million items of Wikidata.
- Building a statement index: They use a sparse matrix to build an index for Wikidata statements. To be simple, we only encode the item statements (item - property - item) and ignore the statements' property information. In the general case, they can also index the literal statements and properties into the sparse matrix. Overall, they index around 500 million statements.

(ii) In **Step 2**, the fuzzy entity search is used to find relevant entity candidates for each cell in the table. They perform a fuzzy entity search on the entity index. This searcher returns a ranking list of relevant entities ranked with an edit distance, specifically Levenshtein distance. They first start with two edits; if the searcher does not have answers, the number of edits is gradually increased

[10] Revision history: https://www.mediawiki.org/wiki/API:Revisions.

to a maximum of six edits. For example, queries as two misspelling queries: " *AnhÃ©e*", and "*BiÃ¨vre*". This fuzzy search returns the correct answers for the two misspelling queries as Q545889 ("*AnhÃ©e*") and Q715371 ("*BiÃ¨vre*"). This searcher achieves coverage of 99.89% on average when performing entity lookup for table cells on SemTab 2020 datasets.
Fuzzy statement search is used to handle the ambiguity of the table cells and consists of using the values of two cells in the same row to determine if there is a statement (relation) between them. Therefore, they perform the two cells' search on the target cells in the same rows. The first cell is in the core attribute column, and the second cell is in the remaining columns. They ignore the rows do not have enough two target cells. This idea is inspired by their novel idea of entity-entity column matching in MTab [142]. Unlike the original idea in MTab that performs the two cells matching in the step 3, this work adapts this idea into the lookup step and makes the step 3 more efficient. Using statement search in the lookup step (Step 2) is also efficient because of the sparse matrix searching. This idea also works effectively, and robust on DBpedia [143]. For example in Figure 5.8: two queries: the two cells of row 1 "*AnhÃ©e*", "Arrondissement of Dinant", and row 2 "*BiÃ¨uraing*", " Arrondissement of Namur". They first search one cell then check all combinations between the two respond lists, whereas the statements are available. In the end, only the entity candidates for which there is a relationship (statement) between the cells are retained. For each cell that need to be annotated, they priority select the entity candidates from two cells search. If they do not have candidates in the "two cells" searches, they select the candidates from a cell search .

(iii) In **Step 3**, the value matching module performs context similarity calculation between candidate statements and table row values then ranks the entity candidates based on these similarities. For each retained entity candidate, the system calculates a value matching (which depends on the statement similarities of each candidate and the other cells in that row in the post processing step) and keeps only the entity candidate with the highest value. An example of value matching for a row in table. Recall the cell "*AnhÃ©e*" has two entity candidates with the ID of Q545889, and Q21770072. They calculate the statement similarities of each candidate and the remaining cells in this row. For example, they found three similar statements to the cell values in candidate 1, while only found one statement similar in candidate 2. As a result, candidate 1 gets higher context similarity than candidate 2. Once all CEA annotations are assigned, CPA annotations are retrieved by aggregating all properties of statement candidates in the same rows, and then using majority voting to select the CPA annotations. For CTA annotations, we get the direct types from the CEA annotations in a column and vote for the majority types to get the CTA annotations.

In the CPA task and CTA tasks, if the system returns many answers for one target annotation, they randomly select one answer as the final annotation.

More details on the subject column detection can be found in [145–147].

3. During the **graph generation** phase, this system **automatically creates two graphs** (minimal graph and full graph) from the annotated data. This phases address identified requirements such as: (R8), which **allows users to visually understand their datasets before analyzing or applying them**. In this phase, we aim to improve and support the state-of-the-art data annotation approach to provide the user with a more easily understood representation (see, Figure 5.10, section(d)).

(i) In **the minimal graph**, we only depict the CTA and CPA are taken into account. CTA (class) are represented as nodes and CPA (properties) ($R((m_{j1}, m_{jn+1}))$) as links connecting the subject column (the main column) (m_{j1}) to other nodes (items). This graph represents a triplet format $< item, property, item >$, which is $< c_{i1,j1}, R(m_{j1}, m_{jn+1}), c_{i1,jn+1} >$, where:
- i_1 is the index of the first line of data annotated to represent the CTA;
- j_1 is the index of the first column of data annotated to represent the subject column;
- m_{j1} is the sub_col;
- m_{jn+1} is the next columns, where, n is the number of columns, $n = 1$ and $n = n + 1$.

$c_{i1,j1}$ is the first cell of the sub_col, $R(m_{j1}, m_{jn+1})$ are the properties (CPA) and $c_{i1,jn+1}$ (CTA) are the other columns. For example, $< descr_fr(Q493522)$, *located in the administrative territorial entity (P131)*, $tx_adm_desc_fr(Q91028) >$. The figure 5.10 (section(c)) represents the minimal graph. This graph allows users to visualize the relations (CPA) between the subject column and other columns (CTA).

(ii) The **full graph**, on the other hand, is an extended version of the minimal graph and includes all the information of the annotated data: CEA, CTA and CPA. CTA and CEA are represented as nodes (classes and instances) and CPA (properties) as links connecting the subject column (resp. CEA of subject column) with other CTA (resp. CEA of other CTA). This graph represents a triplet format which contains CTA, CPA as the minimal graph and extends this graph by representing the CEA. The triplets of CEA represent such as: $< item, property, item >$, which is $< c_{i+1,j1}, R(m_{j1}, m_{jn+1}), c_{i+1,jn+1} >$, then $< c_{i3,j1}, R(m_{j1}, m_{j3}), c_{i3,j3} >$. Where:
- $i + 1$ is the index of the second line of data annotated to represent the cell value (CEA);
- j_1 is the index of the first column of data annotated to represent the subject column;
- m_{j1} is the sub_col;
- m_{jn+1} is the next columns, where, n is the number of columns, $n = 1$ and $n = n + 1$.

$c_{i+1,j1}$ is the second cell value of the sub_col, $R(m_{j1}, m_{jn+1})$ are the properties (CPA) between these two columns and $c_{i1+1,jn+1}$ (CEA) are the cell values in the other next columns. For example, < *Anhée (Q545889), located in the administrative territorial entity (P131), Arrondissement of Dinant (Q93740))*>.

The figure 5.9 (section(d)) represents the full graph. This graph allows users to visualize the relations between data content items and column names and to discover hidden relations between them.

Both generated graphs are **interactive**, so users can move the nodes and **discover relations between elements in a more readable way**. The nodes in the graph are also clickable, allowing users to get more details about each entity if needed. Unlike some previous studies presented in Chapter 3 that generate only the minimal graph. **ODSAG generates a full graph containing classes, properties and instances, that helps users to better explore and discover all relationships between cell values (CEA)**.

5.4 Implementation and Demonstration

In this section, we briefly explain **the preliminary implementation of SINATRA approach and ODSAG tool** (both source code of which are available on GitHub [11] for future research) and show use cases of this two proposals.

Regarding **the implementation of the SINATRA approach**, to provide a graphical user interface easy to install on a computer and show the phases of this approach. This tool supports multiple operating systems like Windows, Mac, Linux. We use Selenium WebDriver [12] to build an API to extract annotated information from the web (API ExtarctHtmlLMtab). It is a web framework that permits you to execute cross-browser tests. This tool is used for automating web-based application testing to verify that it performs expectedly. We choose this tools it allows us to choose a programming language to create our scripts. It provides compatibility with a range of languages including Python, Java, Perl, Ruby, etc. We develop the interface using *the python library Tkinter* [13]. Visually, it is less pretty than other extensions, but it is better to check the frequency of updates of their source code before choosing one, and its license is more flexible. One of the advantages of Tkinter is its portability to the most widely used OS.

Figure 5.9 depicts the graphical interface of SINATRA approach and focuses on data annotation (a), data repair (b), and data augmentation (c) features when using the the president's list [14], which is extracted from the T2Dv2 Gold Standard to demonstrate each phase of this contribution.

First, the user clicks on the "Load" button to load a tabular data from his computer (**Section(a)**). Then, in order to annotate the dataset, the user clicks on the "Load URI" button. So our system automatically starts to prepare the data, detect the subject column is the first column which contains the names of presidents and start to annotate the file. The user receives an annotated data file (**Section (b)**).

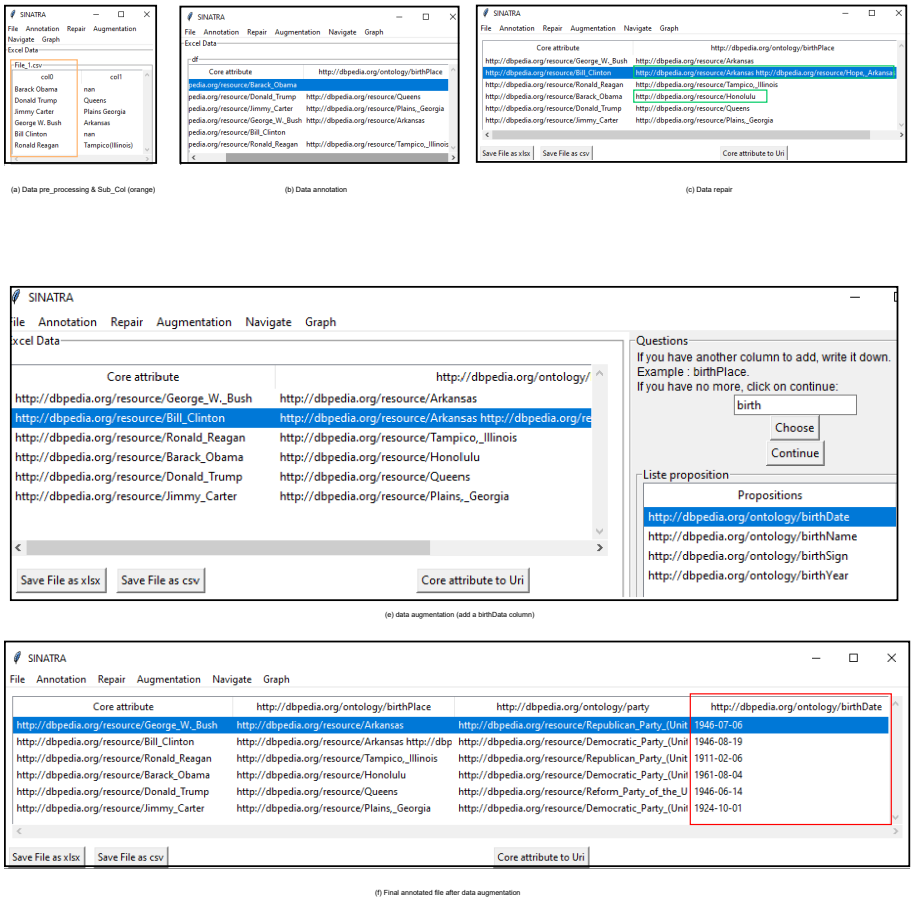
We observe that there are cells in this file that have hidden values, so the user clicks on the "Repair" button and automatically our system adds the missing values

[11] https://github.com/123rabida123/SINATRA-Annotation-Repair-Augmentation
[12] https://www.selenium.dev/documentation/webdriver/
[13] https://docs.python.org/fr/3/library/tkinter.html
[14] https://tinyurl.com/4hrx6s48

(**Section (c)**). Our approach is able to recover the birth place of "Barack Obama" is "http://dbpedia.org/resource/Honolulu" and the birth place of "Bill Clinton" is "http://dbpedia.org/resource/Arkansas".

Finally, if the user wants to extend the annotated file and add an additional column. He writes a request in the text field that contains the name of the new column and clicks on the "Add" button. Our system automatically add the new column is " http://dbpedia.org/ontology/birthDate" and then fills in the missing values (**Section (d)**). According to the user's request, this system capable to create more than column.

SINATRA approach provides users **to save the annotated datasets** in a local folder and **can be exported** in Excel(XLSX) and CSV format (**Section(f)**). If there are still datasets to annotate, SINATRA restarts its phases again to meet the needs of data analysts.



(a) Data pre_processing & Sub_Col (orange)

(b) Data annotation

(c) Data repair

(e) data augmentation (add a birthData column)

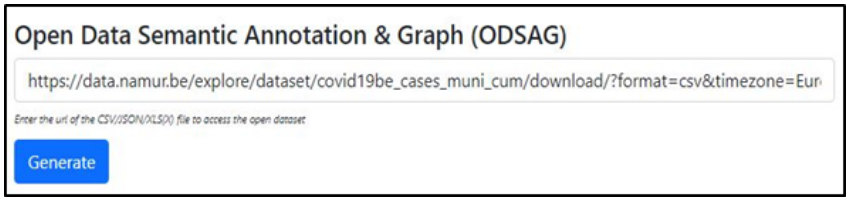
(f) Final annotated file after data augmentation

Figure 5.9: Screenshots of the features of SINATRA approach using President's list.

Regarding **the implementation of ODSAG**, in order to provide a tool that is easy to install and use and that is interoperable with any Open Data portal (**R7**), we chose to implement: (1) a chrome extension that users can interact with (frontend) and (2) a django web application [15] that is used to interact between the chrome extension and SINATRA tool (data annotation and data repair phases) which can be hosted online or locally (backend). In this section, we try **to demonstrate our chrome extension on a real use case available on the Namur (Belgium) portal**. This portal was chosen as it is the most advanced portal in Wallonia (Belgium) and access with key stakeholders of this portal was possible (useful to evaluate later the direct integration of the tool to a portal).

Figure 5.10 illustrates **an example of annotation and graphs generated by ODSAG** when using the COVID[16] open dataset. In order to annotate a dataset, the user must first load the extension in the chrome extensions (only for the first run). Then, as shown in section (a) of Figure 5.10, he/she has to go to any open data portal, copy the URL link of the desired open dataset and paste it into the ODSAG URL field, then click on "Generate" button. The system removes the numeric columns ("$Nbredecas$", "$Nombredecasminimum$") and geographic columns ("$limitecommunale$", "geo_point_2d") from the selected dataset (section (b) of Figure 5.10). A few seconds later, the system generates the annotated data and returns it as a table and graphs. Section (c) of Figure 5.10 shows the annotated data in table form where the type row includes the semantic tags associated with the name of each column (CTA). The property row includes the semantic tags associated with the relation between the subject column (tx_descr_fr) and the other columns. The entity annotations are in red and located below the table cell values. For example, the cell values "$Anhée$" and "Arrondissement de Dinant" have been respectively mapped to the entities Q545889 "$Anhée$" and Q93740 "$ArrondissementofDinant$" (CEA), the column tx_descr_fr was associated with the class Q493522 "Municipality of Belgium" (CTA) and the system detects a property P131 "located in the administrative territory entity" between the column tx_descr_fr and $tx_adm_dstr_descr_fr$. Section (d) of Figure 5.10 shows the minimal graph. The subject column in this graph is represented by a light blue color, the CTA are represented by a dark blue color and the CPA are represented by the black links. Section (e) of Figure 5.10 shows the full graph. The subject column is represented by an orange color, the CTA are represented by a light blue color, the CEA are represented by a dark blue color and the CPA are represented by the black links.

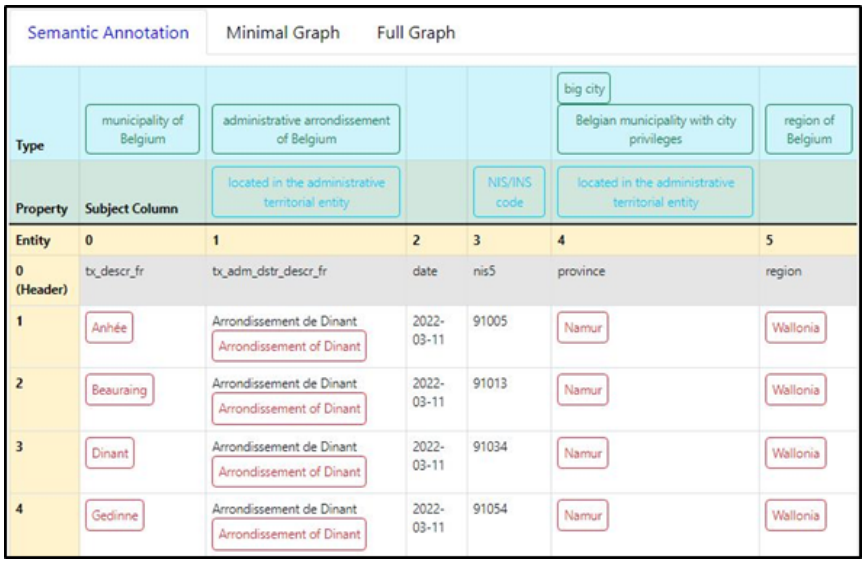
[15] https://www.djangoproject.com/
[16] https://rb.gy/xigrww



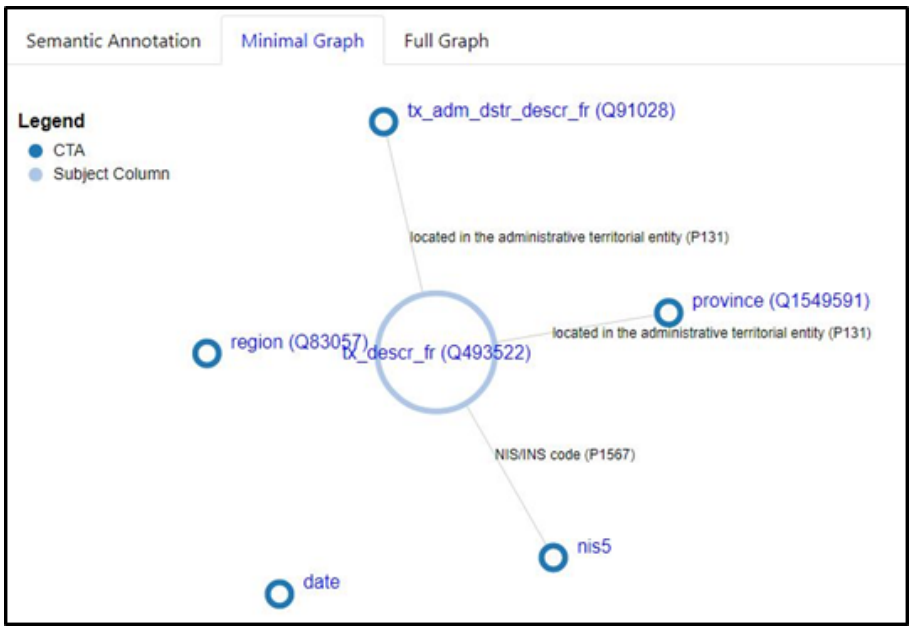
(a) Data Selection: Provide URL to access open data.

Commune	Nbre de cas	Nombre de cas minim...	Arrondissement	date	NIS5	Limite Communale	PROVINCE	REGION	geo_point_2d
Anhée	871	871	Arrondissement de Di...	2021-06-30	91005	'{"type": "Polygon", "...	Namur	Wallonia	50.3187837412, 4.809...
Dinant	1437	1437	Arrondissement de Di...	2021-06-30	91034	'{"type": "Polygon", "...	Namur	Wallonia	50.2494906716, 4.943...
Hamois	1026	1026	Arrondissement de Di...	2021-06-30	91059	'{"type": "Polygon", "...	Namur	Wallonia	50.3370324998, 5.139...
Rochefort	1512	1512	Arrondissement de Di...	2021-06-30	91114	'{"type": "Polygon", "...	Namur	Wallonia	50.1569870969, 5.169...
Andenne	3547	3547	Arrondissement de Na...	2021-06-30	92003	'{"type": "Polygon", "...	Namur	Wallonia	50.4867614568, 5.057...
Fosses-la-Ville	1412	1412	Arrondissement de Na...	2021-06-30	92048	'{"type": "Polygon", "...	Namur	Wallonia	50.3931993971, 4.681...

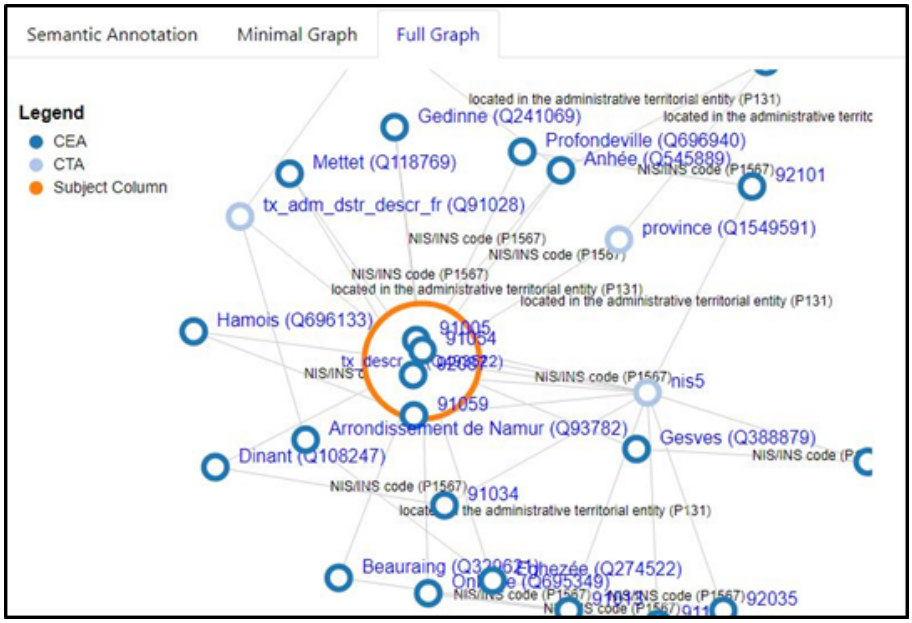
(b) Data pre-processing: Remove numeric and geographic columns circled in red.



(c) Semantic Annotation: Type row represents the CTA. Property row represents CPA. CEA are in red and located below the table cell values.



(d) Graph generation (Minimal Graph: CTA and CPA are taken into account).



(d) Graph generation (Full Graph: CTA , CPA and CEA are taken into account).

Figure 5.10: Screenshots of ODSAG using a COVID Dataset from the Namur portal.

5.5 Evaluation Results of SINATRA approach

In this section, we evaluate and demonstrate **the result of SINATRA approach**, in which we presented more details about benchmark datasets, ground truths, and evaluation metrics in Section 2.6.3. This evaluation **aims to measure the performance of the data repair and data augmentation features of the SINATRA approach**.

5.5.1 Datasets, Ground truths and Measures

Datasets and Ground truths

We evaluate SINATRA approach **by using the datasets and the ground truths proposed by the SemTab 2019 and 2020 competitions of the "Tabular Data to Knowledge Graph Matching"** [90, 91], where, this challenge came with a different set of tables and the three targets (CEA-targets [17], CPA-targets [18] and CTA-targets [19]) of matching each annotation task (CEA, CTA, and CPA) with DBpedia KG [90, 91].For more explication about the datasets and the grounds truths of the SemTab competition, you can find in Section 2.6.3. For more information about the datasets, we have made all datasets available here[20]. To evaluate this proposal **using randomized datasets** [21] **from different round of SemTab 2019 and 2020**.

In Table 5.5, we present the datasets used in our evaluation: ID, Dataset, Round, #Col, #Rows, and Names of columns.

Evaluation metrics

To measure the efficiency of the data repair and data augmentation features of SINATRA approach, we use the following metrics proposed in the challenge of SemTab 2019 and 2020 [90]: **Precision**, **Recall** and **F-measure(F1)**. More information about the measures, you will find in Chapter 3 Section 2.6.3.

$Precision$ (P) of the mapping between the data table and the KG is calculated using the following formula:

$$P = \frac{(\#perfect\ annotations)}{(\#submitted\ annotations)}$$

where a #perfect annotation referred to the annotation returned by our approach, which corresponded to the annotations of ground truths. A #submitted annotation referred to the annotation returned by our approach.

$Recall$ (R) is calculated as follows:

$$R = \frac{(\#perfect\ annotations)}{(\#ground\ truth\ annotations)}$$

[17] https://www.aicrowd.com/challenges/semtab-2020/problems/cell-entity-annotation-cea-challenge

[18] https://www.aicrowd.com/challenges/semtab-2020/problems/columns-property-annotation-cpa-challenge

[19] https://www.aicrowd.com/challenges/semtab-2020/problems/column-type-annotation-cta-challenge

[20] https://github.com/123rabida123/Datasets-and-Results-of-evaluation-SDA

[21] https://zenodo.org/record/3518531#.Y1PRbXZBwuU

Table 5.5: The characteristics of the datasets were evaluated by SINATRA approach

ID	Datasets	Round	#Col	#Rows	Name of Columns
D1	211	2	4	56	Col1:University of UEA Col2:President Col3:number of staff Col4:Surface Area
D2	212	2	4	15	Col1:University Col2: Post holde Col3:Number of students Col4:Number of staff
D3	274	3	5	17	Col1:Name of the animal Col2:Family of the animal Col3:(unnamed) Col4:Location Col5:(unnamed)
D4	275	3	5	65	Col1:Name of the animal Col2:Family of the animal Col3:Place Col4: (Unnamed) Col5:(Unnamed)
D5	308	4	4	78	Col1:Group event Col2:County Col3:Company Col4:(Unnamed)
D6	309	4	6	44	Col1:Group event Col2:Second team Col3:Third team Col4: Second Driver Country Col5:Third Driver Country Col6:(Unnamed)

where the number of #ground truth annotations corresponded to the number of annotations in the Target Tables.

To calculate the $F-measure$ ($F1$), we combined the predefined measures, which represented the harmonic mean between P and R.

$$F1 = \frac{(2 * P * R)}{(P + R)}$$

5.5.2 Evaluation and demonstration results of SINATRA approach

This section evaluates and demonstrates **the performance of the SINATRA approach's features**. For more details on the results of the evaluation, consulting our Github[22].

[22]https://github.com/123rabida123/Datasets-and-Results-of-evaluation-SDA

Regarding the evaluation of the **data annotation** feature, **this phase of SINATRA is based on the MTab approach. Therefore, it automatically has the same performance as MTab**. Table 5.6 below shows the results of the evaluation of the data annotation phase by the MTab approach [142].

Table 5.6: Evaluation results of the data annotation feature by MTab approach.

Feature	Data annotation (MTab approach)					
Tasks / Datasets	CEA		CPA		CTA	
Measures	F1	P	F1	P	F1	P
D1	1.0	1.0	0.881	0.929	0.850	0.852
D2	1.0	1.0	0.877	0.929	0.850	0.850
D3	0.983	0.983	0.844	0.845	0.833	0.833
D4	0.970	0.970	0.832	0.832	0.825	0.825
D5	1.0	1.0	0.987	0.975	0.929	0.933
D6	1.0	1.0	0.965	0.991	0.970	0.970

Our goal in this evaluation is to compare the results of the **data repair** and **data augmentation** phases (Table 5.7 and Table 5.8) with the results of the **data annotation** phase (Table 5.6) **to show that they can correctly add the data (entities) and the missing columns**.

Table 5.7: Evaluation results of the data repair feature.

Feature	Data repair											
Factors	Remove cell values (1)						Add missing cell values (2)					
Tasks	CEA		CPA		CTA		CEA		CPA		CTA	
Measures/ Datasets	F1	P	F1	P	F1	P	F1	P	F1	P	F1	P
D1	0.854	0.845	0.881	0.929	0.850	0.852	1.0	1.0	0.881	0.929	0.850	0.852
D2	0.852	0.874	0.877	0.929	0.850	0.850	1.0	1.0	0.877	0.929	0.850	0.852
D3	0.832	0.832	0.844	0.845	0.833	0.833	0.877	0.877	0.844	0.845	0.833	0.833
D4	0.812	0.813	0.832	0.832	0.825	0.825	0.834	0.836	0.832	0.832	0.825	0.825
D5	0.911	0.911	0.987	0.975	0.929	0.933	0.983	0.983	0.987	0.975	0.929	0.933
D6	0.943	0.945	0.965	0.991	0.970	0.970	0.987	0.975	0.965	0.991	0.970	0.970

Regarding the evaluation of the **data repair** feature, **we re-based on the same datasets as above (Table 5.5)**. In this phase, the evaluation is based on two factors to compare results with the results of MTab. The first factor (1): **we removed some values from those datasets (Table 5.5) and calculated the performance of this phase**. The second factor (2): **we added the missing cell values into these datasets during the data repair phase**. Table 5.7 below shows the performance results of the data repair phase based on the two factors mentioned. From the results of Table 5.7, we notice the results of the CEA task are reduced in the factor (1) because (R) is reduced (the removed URIs (entities) are in the CEA-targets). Based on the factor (2), we highlight that this phase can add missing data very nicely, where the CEA task has $F1 = \mathbf{1}$ of the datasets (D1 and D2). They have the same results as the **data annotation** feature. The CEA results are represented by the yellow color in Table 5.7.

For the datasets (D5 and D6), the results of the CEA task have been reduced a little bit (from $F1 = \mathbf{1}$ in Data annotation to $F1 = 0.987$ in Data repair), because some URIs were not perfect or were not available in the CEA-targets. The CPA task is represented by magenta color and the CTA task is represented by cyan color, which have no variation in both factors. **They have the same results as the data annotation feature in Table 5.6**.

Table 5.8: Evaluation results of the data augmentation feature.

Feature	Data augmentation											
Factors	Remove Column (1)						Add missing Column (2)					
Tasks	CEA		CPA		CTA		CEA		CPA		CTA	
Measures/ Datasets	F1	P	F1	P	F1	P	F1	P	F1	P	F1	P
D1	0.750	0.753	0.706	0.738	0.754	0.70	1.0	1.0	0.881	0.929	0.850	0.852
D2	0.751	0.754	0.706	0.738	0.736	0.739	1.0	1.0	0.881	0.929	0.850	0.852
D3	0.606	0.638	0.754	0.700	0.729	0.781	0.981	0.981	0.844	0.845	0.833	0.833
D4	0.632	0.634	0.707	0.717	0.729	0.781	0.943	0.945	0.844	0.845	0.833	0.833
D5	0.913	0.915	0.846	0.855	0.860	0.878	1.0	1.0	0.987	0.975	0.929	0.934
D6	0.845	0.845	0.846	0.855	0.833	0.835	0.996	0.997	0.939	0.987	0.956	0.956

Regarding the evaluation of the **data augmentation** feature, **we re-used the same datasets as above (Table 5.5)**. The evaluation of the data augmentation feature is based on two factors: In the first factor (1), **we removed every second column from those datasets (Table 5.5) and calculated the performance of this phase (without the second columns)**. In the second factor (2), **we added the missing columns into these datasets**. Table 5.8 above shows the performance results of this phase based on the two factors mentioned: whether this proposal **is able to add exactly the deleted column in each dataset**. From the results of the factor (1) in Table 5.8, we notice that the results of the CEA, CPA, and CTA tasks are more reduced because (R) is reduced (the removed URIs (entities) are in the targets). In addition, we notice from the results of the factor (2) in Table 5.8, that this feature is able to add the missing column very well, where the CEA, CPA, and CTA tasks of the datasets (D1, D2, and D5) have the same results as the **data annotation** feature in Table 5.6 are represented by the yellow color. The magenta color represents the results of the CPA task, and the CTA task is represented by the cyan color of the datasets (D1, D2, D3, D4, and D5). They also have the same results as the **data annotation** feature. Thus, the **data augmentation** feature is perfectly able to add missing columns to the datasets. For the datasets (D3, D4, and D6), the results of the CEA task were slightly reduced, because some URIs were not perfect or were not available in the CEA targets.

5.6 Discussion and Limitations of SINATRA approach

Regarding research question RQ2.1 and RQ2.2, we developed an all-in-one approach called SINATRA, that is tacked these questions. This proposal is based on some of the requirements identified in Section 5.2.

In Chapter 3, we synthesized the approach and tools to identified the best methods of the Semantic Table Annotation (STA). Thus, SINATRA approach improved

the quality and the interoperability of the Tabular data. In the first two phases of SINATRA, we combined **the strenghts of the MantisTable and MTab** to perform both phases (sub_col detection and annotation data). These approaches have had the highest ranking in the SemTab competition during 3 years. **MTab** did not offer a sub_col detection phase but **has excellent results in annotating data**; and **MantisTable** did not offer excellent results like MTab but allowed **sub_col detection**.

According to the findings of the evaluation, the SINATRA approach was able to address the **data quality** issues. Mtab solved the misspellings errors. The aim of the subject step column was to identify the relationships between columns in order to identify the metadata of file. The data annotation phase addressed the issue of missing **metadata**. We concentrated on three tasks (CTA, CPA, and CEA) to enhance the metadata of the datasets. In the evaluation of this phase, we found the same result of the MTab 2019 edition, because we based on this solution. In SINATRA approach we assigned the data elements to concepts in **DBpedia** KG. In ODSAG tool, we have matched the data elements to **Wikidata** KG (Mtab 2020 edition).

During our analysis in chapter 3, we noticed that the STA tools were not able to add new information. To support the STA process, we have provided additional phases such as **data repair and data augmentation** that can both define semantic annotations and provide new information based on semantic annotations are of high interest nowadays [181]. To evaluate these phases, we compared our results with the results of the MTAB annotation, because in the literature, the results of these phases are not provided in the literature. We defined an evaluation method to verify that SINATRA's feautres were able to select the appropriate entity. We evaluated the effectiveness of SINATRA features using benchmark datasets and the measurements (F-1 and Precision).

According to the results we have achieved, We deployed an extension of the SINATRA approach called ODSAG, which was able to interoperate with all data portals. **This contribution is used to answer the research question RQ.2.2**, because the majority of the existing tools in Chapter 3 were not able to **automate** the process and to be both **interoperable** with open data portals. Based on the evaluation of the ODSAG tool, it may allow publishers or analysts to automatically clean and annotate their datasets found on OGD to improve their discovering and understanding before using them.

These research efforts have contributed to the theory by proposing a list of requirements that should be implemented in an approach and tool to facilitate semantic annotation of open data and provide new information (see table 3.6. They also provided a comparative table highlighting the strengths and weaknesses of some existing semantic annotation approaches and tools used in the literature(See Chapter 3). **These contributions also contributed to the practice by implementing the identified requirements and providing source codes. These can be used as a starting point for developers to create their tool or to improve them**.

However, these research **has certain limitations** that will need to be addressed in future work:

- Our aim is **to propose an analysis functionality** that can analyze data quality in public datasets before the pre-processing feature, as most data scientists

spend most of their time studying and manipulating data. And yet, there is a lack of tools to assist in data analysis.

- We evaluate our contributions by other datasets and other stakeholders (Government user).
- We aim to compare this approach with other existing approaches and tools.
- We plan **to improve the contribution with additional features** such as: (1) improving the pre-processing phase to adapt it to all Open Data portals by integrating the abbreviation of certain cells (e.,g, quarter code, F, M, *etc.*), (2) integrating annotation of numerical and geographical columns, (3) integrating additional knowledge graphs such as **LOV** and **Geonames** to improve the annotation.
- We intend to extend these works by combining the annotated datasets, which have the same context. This functionality can be used to discover relatedness in datasets.
- We aim to (1) make documentation and tutorials of our application available to users to better understand the workflow and save time, (2) **create APIs** for these features **to allow the potential to use public features and data**, and (3) generate a RDF file of the annotated Open dataset for use in Linked Open Data.

5.7 Conclusion

In this chapter we present two solutions that help to bridge the gaps of data quality by providing the following features: data annotation (to address misspelled and incomplete metadata issues); data repair (to address missing values (data) issues); data augmentation (to dynamically add meaningful columns and corresponding cell values to data); and graph generation (to generate an annotated dataset visualization graph). These contributions aim to improve the usability of Tabular Data (TD) and make OD easier to understand and discover. Additional, Its goal is to establish effective interoperability with Open Data portals.

To achieve that goal, we first identified a list of requirements that need to be implemented in two contributions, which were based on two scientific publications covering three research questions (see Section 5.1.1). (1) **SINATRA** (SemantIc aNnotation AugmentaTion and RepAir), which is an all-in-one and fully automatic approach. (2) **ODSAG** tool that is extension of SINATRA. It is a user-friendly chrome extension that can be easily interperable with any OD portal. These research contribute to the practice by implementing the identified requirements into an approach and tool and providing the source code on GitHub for the use of the stakeholder community.

To evaluate our research, we used some of Benckmark's datasets from the 2019 and 2020 SemTab competition to measure the effectiveness of the SINATRA approach through metrics. The evaluation results show that it was effective in annotating, repairing and augmenting structured data. In addition, we also evaluated the capability of ODSAG with COVID data from open data portals. However, this research had some limitations that will need to be addressed in the next Chapter 6.

CHAPTER 6

SINATRA WEB APPLICATION

6.1 Introduction

The analysis carried out in Chapter 3 and the limitations in Chapter 5 allows us to conclude that the development of an overall framework called **Sinatra Web Application (SinatraAPP)** can be useful to improve SINATRA approach in Chapter 5 and **make it more effective and useable for all users who are aware of and operate in an Open Data context**.

This Chapter presents: **(i) the most requested requirements of the framework**; **(ii) the analysis approach used prior to data preparation, there is a lack of tools to assist in data analysis**; **(iii) the measures operations used to find datasets that have the similar context for merging**, that allow users collect and join similar data into a single dataset in order to obtain a comprehensive view of a particular topic or phenomenon; and **(iv) the architecture of the implemented framework**, which consists of two main elements: a **user interface** (UI) to support user interaction, and a **backend** to handle all the functionalities and interface the API services for these functionalities. **It aims to enable Open Data analysts and publishers to increase Open Data training, find potential features to extract, analyze, annotate, repair, and find joinable or linkable tables to merge the Open datasets**.

In this context, **Data analysts spend most of their time reviewing and handling data but need tools to help them analyze it**. The first part of **this chapter identifies recurring data quality problems in public datasets**. It presents a catalog of faults in Open Data that can be used to indicate early signs of technical problems or liabilities in pre-processing systems. In addition, we **develop an analytical tool that utilizes a graphical interface** to visually represent the quantity of errors present in each dataset.

Numerous contemporary data analysis applications build on Open Data. There

is a need to build Open Data search capabilities into data analysis environments so that **data analysts can find and join tables, schemas, and datasets applicable to their tasks**. The second part of this chapter is about coming up with ways to look for and handle data on the Open Data platform, **to allow scientists to augment training data, find potential features to find joinable annotated tables**. **SinatraAPP** focuses on meeting the requirements of both users and systems.

6.1.1 Research question

The content of this chapter is based on the following questions that cover these contributions:

(i) RQ3. "How to build a tool to help data analysts analyze open data to reduce time spent quickly pointing out signs of open data issues?"
   - RQ3.1. "what is the recurrent data quality issues that appear in Open datasets?"

(ii) RQ4. "How can we implement a simpler technique and application to help users discover and combine related annotated datasets to get an overview of a particular topic?"

6.2 Requirements Analysis

The previous analysis of related approaches and the state-of-the-art revealed requirements that should be met by implementing this framework to fully encapsulate the experience of data analysis, semantic annotation, data augmentation, data merging, and data publishing:

- It should involve identifying recurrent data quality problems in Open public databases. Indeed, data analysts spend most of their time studying and manipulating data, yet there are no tools to assist them in data analysis. In addition, a pre-analysis of the data is required and is typically the first step in an effective open data system.
- It should provide functionalities that facilitate the user in locating the most relevant information, comprehending the interconnections among the data, and integrating multiple sources of data or products (for example, merge the Geo-names file with other datasets that have no location or position column, whereas this file contains all countries with longitude and postal code.)
- It should engage the user in the pre-processing and merging features of the process through interactive choices so that the results of the automatic algorithms can be better monitored and, if necessary, corrected.
- It should provide API services to application developers that can be easily added to form an ecosystem of services available to end users. They can enable analysts to increase potential services to extract, clean, enrich, repair, augment data, and find linkable tables.
- It should be possible to use the functionalities on a mobile application at any time, via its APIs.

The detailed requirements of the contribution are presented in the following sections.

6.2.1 User Requirements

User requirements are frameworks or characteristics of products that developers need to implement to enable users to perform their tasks. The following list describes the identified user requirements:

- The user should be able to handle tables with basic operations: download, and delete one or more data sets in different formats (e.g., CSV, EXCEL, and JSON), search datasets by ID name, collect or analyze information about tables (datasets) (e.g., number of rows and columns, number of duplicates per row and cell, number of particular missing values, number of binary missing values, and unknown unit of measure), save and reload tables to replay working sessions, join related datasets, and export tables in different formats (e.g., the W3C standard annotation format).
- The system should provide users with a more straightforward interface with the steps of selecting portal and dataset data, analyzing the data before the preprocessing step, enriching the data, repairing it by adding missing values, augmenting the data by adding meaningful columns or data (Completennes the data), joining linkable datasets, and transforming the annotated data to RDF format for export.
- The system should provide users with a more straightforward interface with the steps of selecting portal and dataset data, analyzing the data before the preprocessing step, enriching the data, repairing it by adding missing values, augmenting the data by adding meaningful columns or data, joining linkable datasets, and transforming the annotated data to RDF format for export.
- The system must allow the user to select the column type or properties to add a new column, change the selection, or manually add a correct entity to the candidate list.

6.2.2 System Requirements

System requirements are the constraints placed on the system. They address issues such as ease of use and maintainability. The following list describes the major system requirements:

- The system should be able to select datasets from any Open Data portals or form local. The user interface must let user analyze and visualize any dataset, navigate such elements and the user can navigate between the steps of the framework.
- The system should be able to select datasets from any portals data or from locally. The user interface must let the user analyze and visualize any dataset, navigate such elements, and the user can navigate between the steps of the framework.

- The agile development approach requires that the user interface and backend be modular to allow for the introduction of new features, revision or replacement of existing features by simply changing or replacing components. The system automatically reloads if the developer changes any of the source components.
- The system must be able to interact with external services and exchange annotation, repair, and augmentation data.
- A vital feature of the system must be the ability to include current and future loading/analysis/annotation/repair/augmentation/Visualizing/exporting services with minimal effort. The backend must provide input and output transformation components that can be specialized to interact with external services.

6.3 Open Data Analysis

This section **identifies recurring data quality issues in open datasets** and **implements an approach to facilitate data analysis**. The majority of the publication frameworks and Semantic Tabular Annotation (STA) approaches start the first step as a pre-processing phase that aims to clean and unify the datasets (More details about this phase in Chapter 4 and Chapter 5). However, **the first step should be data analysis, customers can discover the faults of Open Datasets before processing them**. Data analysts spend most of their time studying and manipulating data, **but until today, there have yet to be tools to help them analyze it** [32, 85]. Once the dataset is selected from the data portal, the critical part of the work is to analyze and reprocess it to make it suitable.

This section provides a study of the issues of Open Data quality that can be useful to the user and implements a framework for analysis.

6.3.1 Data selection phase

In this methodology, we first select the open datasets from many different data portals and other public data sources (OpenDataSoft, OpenDataNamur-Capitale, OpenDataBrussels, OpenDataLiege, and Kaggle [1]). The datasets are shortlisted **based on the inclusion criterion** in Table 6.1. Sample data includes only CSV, Tabular Data, EXCEL, HTML Table, and other files less than 1GB in size to facilitate analysis on a personal notebook computer. Structured datasets are more common on the data portal. **Practitioners and academics frequently work with structured datasets often used for educational purposes**. For this reason, we have invested our efforts in analyzing structured datasets to make our work relevant to a broader audience. The **25 datasets** selected for this study are listed in Table 6.2. This table also includes additional information.

In this context, there is a considerable amount of literature dealing with data-related problems [67, 169]. However, many different terms are used to refer to data-

[1] https://www.kaggle.com/datasets

Table 6.1: Inclusion criterion for datasets

Criteria	Description
C1	Dataset is of CSV, Tabular Data, Excel, and HTML format.
C2	Dataset is smaller than 1GB in size.
C3	Dataset contains structured data.
C4	Dataset primarily contains numerical and categorical features.

related problems, such as "dirty data", "data error", "data defect", "data anomaly", or "data quality problem".

6.3.2 Detection phase of Open Data quality faults

We applied the Python programming language [2] along with the data analysis package Pandas [3] to perform the analysis. This phase involves identifying datasets' characteristics that indicate poor-quality categories. **It is to conduct a preliminary analysis of the datasets listed below. This is a standard list of checks made by data scientists when making sense of the data in the CRISP-DM data mining model** [169].

(i) The analysis of their heads and tails, in their fullness and on a feature-by-feature basis.
(ii) Examine column headers and data types for relevant metadata, such as the desired schema of the dataset.
(iii) Verification of missing values and duplicate rows or columns.

This study analyzed Open Datasets from **5 different data portals**, **finding 6 data faults**, were discovered. The next section presents an overview of all data faults of each dataset.

Duplicate examples

This study identified problems with datasets that contain duplicate rows. Duplicate examples in a dataset are defined as two or more rows that refer to the same entity. They are not helpful and can be removed from the dataset, making their presence a redundancy problem in the dataset (see these dataset[4], [5]). Duplicate examples make a dataset too voluminous. They do not provide any information during the data annotation phase. Furthermore, the annotation model with a dataset containing duplicate examples may hinder the model's performance on new data.

Unknown unit of measure

Several datasets contain several numerical features such as *area, currency, density,electric current, energy, flow rate, force, frequency, fuel efficiency, information unit, length, linear mass density, mass, numbers, population density, power, pressure,*

[2] https://www.python.org/
[3] https://pandas.pydata.org/
[4] https://tinyurl.com/yc4buk7u
[5] https://tinyurl.com/34hnh24k

speed, *temperature*, *time*, *torque*, *voltage*, and *volume*. However, the names of the columns and the documentation do not mention the unit in which the characteristics were measured (see, these datasets [6] and [7]). Numerical data or data of type int or float are the most common data types in structured datasets. **Understanding the distribution and trends within numeric features is the core of the data analysis stage**. The lack of a standard unit of measurement for all feature observations is an early indicator of problems during the data annotation phase. **Mixed units** can also lead to incorrect results during data annotation. Several pre-processing techniques (Chapter 5) are used to normalize a dataset's numerical characteristics before the data annotation step. But such a transformation is only useful if the observations are not recorded with the same unit of measurement.

Null type missing values

A common problem observed in the analyzed datasets is **the absence of data** that may indicate potential problems in downstream processes early. We noticed some abnormalities when analyzing the open datasets. Subsequently, the missing data check revealed that the datasets had large amounts of missing values. Missing values are usually ignored by data analysis tools when performing statistical calculations (such as descriptive analysis), **leading to inaccurate and misleading conclusions** [142, 169]. Some structured data in the data portals have empty rows, which also means missing values (See this datasets [8]). **This problem can also reduce the potential value of data, as well as the performance of automatic annotation models**.

Special missing values

The null type is the most common data type used to represent missing values, memorable characters, and words such as "*?*", "*nil*," "*null*," and "*nan*" are also used. Using special characters and numbers to represent missing values - especially when they are not documented - is a source of problems in downstream steps (Check these datasets [9] and [10]). Data analysis tools often contain embedded functionality to check for missing values by detecting null data types in the dataset. Unless otherwise noted, the use of special characters or numbers hinders the ability of data analysis tools to accurately detect missing values [169]. **The data must be analyzed manually to identify the unique character or number used to represent the missing values, wasting a lot of time and effort that could have been spent on other productive tasks**.

[6] https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data
[7] https://tinyurl.com/mrx4ve28
[8] https://tinyurl.com/msunrnkh
[9] https://tinyurl.com/msunrnkh
[10] https://www.kaggle.com/datasets/uciml/adult-census-income

Strings with special characters

The missing values in the dataset are represented as *HTML tags, stop words, punctuation marks, white-spaces* and some other characters (i.e., ", ', @, ©, ?, (), []) (See these datasets [11] and dataset Covid [12]). Another example can be found in the datasets, which is the categorical characteristic of gender where the value can be '*M*', '*F*' or '*I*'. However, due to the presence of white spaces, data analysis tools may consider values such as '*M*', '*F*', '*I*' as valid and separate from each other (see dataset abalone [13]). **The presence of these errors in the structured data is a potential problem in the data analysis phase**. **The presence of redundant white space and special characters in a large dataset can make the data-cleaning process both tedious and time-consuming**. In addition, they can also **lead to incorrect data annotation and reduce model performance**. The information must always be manually checked and handled case-by-case to transform, which wastes time and effort that could have been used for other productive tasks. It must always use data pre-processing methods (Chapter 5) to reduce these faults by unifying and transforming a dataset before the following steps. Where, this is a standard task performed by data analysts, and the most popular data processing tools provide integrated solutions (Chapter 5).

Binary missing values

When analyzing the data portal, **we identified over 40% non-missing values (No understanding value)**. However, these values included abbreviated strings such as "Y," a common abbreviation for "yes," "NN," a common abbreviation for "Naninne", "BZ", a common abbreviation for "Beez", (See this dataset "*Namur - Limites de 46 quartiers*" [14]). These indicates that the missing values had an implied meaning, were not actually missing values, that we can found in this dataset [15]. The presence of large amounts of missing data primarily in one column - rather than distributed across rows and columns - may indicate that the data are not lacking. In this case, the missing values may have the implicit meaning of a negative binary response. If the non-missing data indicates a positive response such as "t, Ttrue" or "y, Yyes", the missing data may indicate a negative response. **No understanding value can indicate relevant information to lend implicit meaning to the context of the dataset**. In software engineering, replacing an answer or a negative result with a null type, such as None in Python and null in Java is common. **It is not the case in data science, as null types are widely used to represent missing data**.

6.3.3 THREATS TO VALIDITY

This study opted for a superficial analysis of the datasets. The analysis phase consisted of a specific set of steps that were performed (as outlined in Section 6.3.2) for

[11] https://tinyurl.com/mrx4ve28
[12] https://tinyurl.com/ys47xjjt
[13] https://tinyurl.com/25zph4sb
[14] https://tinyurl.com/52ebhedy
[15] https://tinyurl.com/52ebhedy

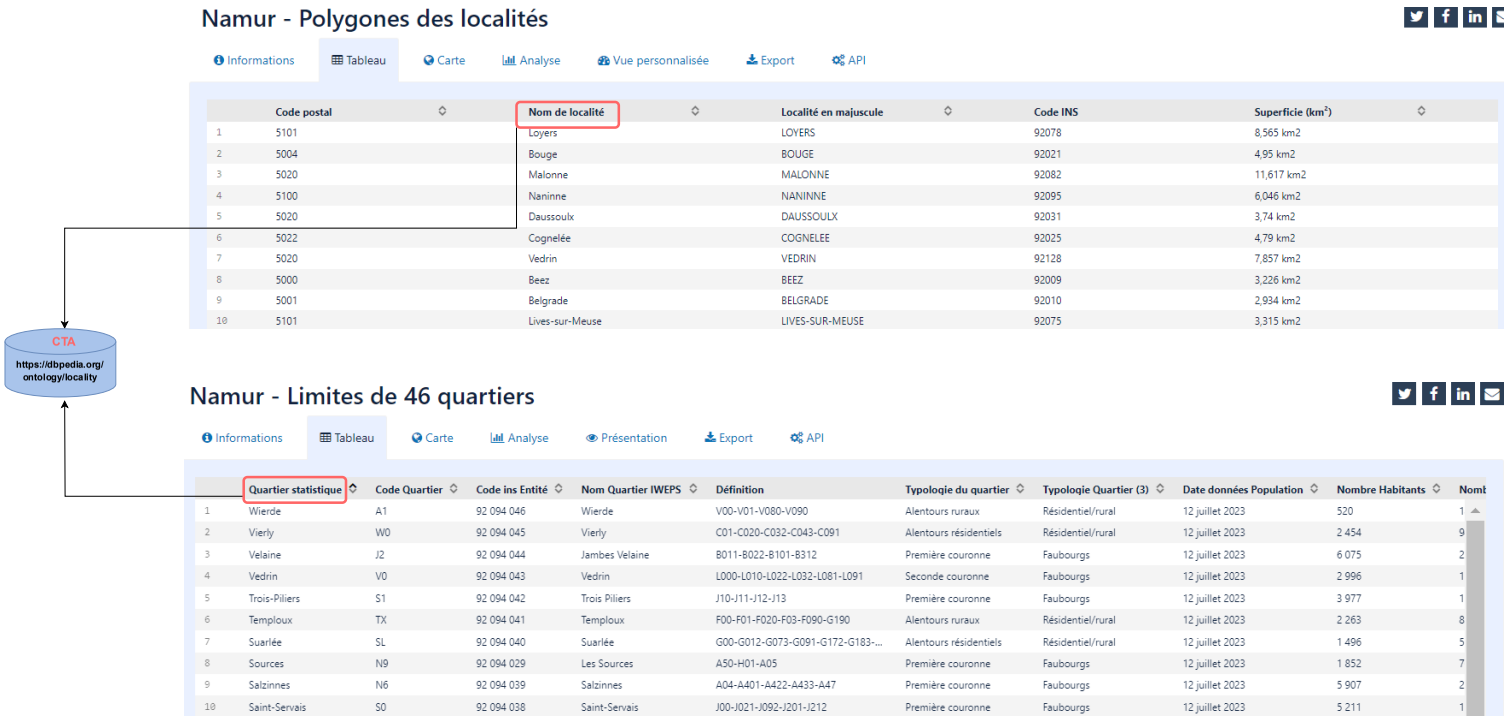
each dataset. This judgmental decision facilitates the extent and reproduction of the analysis stages on open datasets. Further analysis of each data item may show other potential faults. In addition, we believe that the generalization of defects decreases as the depth of the analysis increases. In other words, these examples would only be relevant in the context of the problem domain. The faults discovered by this research are associated with the Open Data used for the analysis.
Additionally, this research provides numerous Open Data issues intended to help the analysts develop a practical approach to dealing with these issues and preparing the data before use. For example, the data from the Namur portals and other portals such as Kaggle, Liège, and Brussels contain the datasets for the analysis. While we are aware of this problem, unfortunately, it is the nature of all data science issues, which we have gathered in many forums and events. The validation process implies analytical methods, which are highly experimental and time-consuming. This issue of the data is subject to the interpretation of the data analysts, the operator who performs the analysis. This problem of substance is also present in code faults. Only some methods are excellent and good exists in open-source repositories.

6.4 Merging Related Open Tabular Data

Open Data offers excellent benefits in terms of data access. **Still, it only sometimes allows the user to find the most relevant data, understand the relationships between data products, and join heterogeneous data sources**. An important issue emerging from publishing data using multiple formats is **that the users need help merging multiple datasets and initiating a data analysis process**. For example, assume that a user needs to find the number of schools located in a particular area and the road traffic in that area in the last year so that he can find out which school is located nearby and the rate of traffic during normal and peak hours on those roads which go to that school. To achieve this, the user will need to open two datasets to complete the analysis. In addition, **this function becomes more complex when performing analyses with more than two data sheets**. The **merge functionality** is particularly beneficial for **analysts, or businesses** that must analyze data from multiple contexts, regions, or countries to make informed decisions. In addition, Open Data was developed to promote the reuse of data (and associated workflows) - but if **users are not aware of what is available** or **do not find complete data that meets their needs**, and if **they cannot trust what they find, they end up reinventing their schemas, import processes, and analysis processes**. It leads **not only to inefficiencies and redundant work but also to consistency in data processing, irregular data representation, maintainability, and usability challenges**.

Toward the overall goal, **we seek to help users find and merge semantically related Open datasets to facilitate Data analysis tasks**. Our work focuses on interactive, "human-in-the-loop" environments in which data analysts undertake data discovery, processing, exploration, cleaning, and analysis tasks. We develop a framework for supporting multiple measures of Open Table relatedness and build on previous techniques developed to find unionable [139] and joinable tables [57, 139].

By merging data feature, **users can analyze large volumes of information to identify trends, patterns, and relationships that are not apparent by examining individual datasets**. In this concrete example, Figure 6.1 shows that these datasets contain the same semantic links and need to merge these related Tabular Data, that exist on the Namur portal, where dataset-1 contains postal code information and dataset-2 does not. Analysts could merge dataset-1 with dataset-2, which contains the number of residents, in order to plan the future population for each commune and make the final decision at a certain point in time. **This task can be complicated if the data analyst is merging more than two tables**.



	Code postal	Nom de localité	Localité en majuscule	Code INS	Superficie (km²)
1	5101	Loyers	LOYERS	92078	8,565 km2
2	5004	Bouge	BOUGE	92021	4,95 km2
3	5020	Malonne	MALONNE	92082	11,617 km2
4	5100	Naninne	NANINNE	92095	6,046 km2
5	5020	Daussoulx	DAUSSOULX	92031	3,74 km2
6	5022	Cognelée	COGNELEE	92025	4,79 km2
7	5020	Vedrin	VEDRIN	92128	7,857 km2
8	5000	Beez	BEEZ	92009	3,226 km2
9	5001	Belgrade	BELGRADE	92010	2,934 km2
10	5101	Lives-sur-Meuse	LIVES-SUR-MEUSE	92075	3,315 km2

	Quartier statistique	Code Quartier	Code ins Entité	Nom Quartier IWEPS	Définition	Typologie du quartier	Typologie Quartier (3)	Date données Population	Nombre Habitants	Nomb
1	Wierde	A1	92 094 046	Wierde	V00-V01-V080-V090	Alentours ruraux	Résidentiel/rural	12 juillet 2023	520	1
2	Vierly	W0	92 094 045	Vierly	C01-C020-C032-C043-C091	Alentours résidentiels	Résidentiel/rural	12 juillet 2023	2 454	9
3	Velaine	J2	92 094 044	Jambes Velaine	B011-B022-B101-B312	Première couronne	Faubourgs	12 juillet 2023	6 075	2
4	Vedrin	V0	92 094 043	Vedrin	L000-L010-L022-L032-L081-L091	Seconde couronne	Faubourgs	12 juillet 2023	2 996	1
5	Trois-Piliers	S1	92 094 042	Trois Piliers	J10-J11-J12-J13	Première couronne	Faubourgs	12 juillet 2023	3 977	1
6	Temploux	TX	92 094 041	Temploux	F00-F01-F020-F03-F090-G190	Alentours ruraux	Résidentiel/rural	12 juillet 2023	2 263	8
7	Suarlée	SL	92 094 040	Suarlée	G00-G012-G073-G091-G172-G183-...	Alentours résidentiels	Résidentiel/rural	12 juillet 2023	1 496	5
8	Sources	N9	92 094 029	Les Sources	A50-H01-A05	Première couronne	Faubourgs	12 juillet 2023	1 852	7
9	Salzinnes	N6	92 094 039	Salzinnes	A04-A401-A422-A433-A47	Première couronne	Faubourgs	12 juillet 2023	5 907	2
10	Saint-Servais	S0	92 094 038	Saint-Servais	J00-J021-J092-J201-J212	Première couronne	Faubourgs	12 juillet 2023	5 211	1

Figure 6.1: A data merging example: merge two tables to complete missing information for a "Polygones des localités" table.

6.4.1 Algorithm For Linking Relatedness Tables

In this section, we present in detail the steps of **the merging algorithm** and **the method for linking and joining the correlated annotated tables**.

Detecting key mapping (K)

This step is an important task in merging phase. It consists **of identifying and finding the key mapping (K) between two tables** with the specific type of join operation "Equijoin". It is a type of **join operations in relational databases** where **records are matched based on the equality of values** in the specified columns (k_A and k_B, in this case). The key mapping (K) consists of two components, k_A **and** k_B, which are likely keys in the two tables being joined. In this example 6.2, we have in Table (A), the k_A is *President* and in Table (B), the k_B is *President*. There is **same CTAs of the sub-col**, Therefore we have chosen the CTA of the first column of the first table (Identify the main table). If there **is no same CTAs of the Sub-col**, we found the matching key (K) between two tables with the join operation where **the identified key** that **can maximize table overlap** would mean **to ensure that the greatest number of rows in the first table (Main Table) match the rows in the second table**, therefore, we have chosen (K) is **the CTA of first column (Sub-col) of the first table (Identify the main table)**.

The aim is to identify **the k matching of Sub-col (**k_A **,** k_B **)** to identify **the main table**, which the k is intended to do to ensure that the two tables have as many points in common or as many matching records as practical after the join operation.

Merging annotated tabular data

The merging data was based on three previous tasks (CEA, CPA, CTA), which we have presented in Section 5.3.1, that created the annotated structured data.

In this section, we explain our algorithm on two annotated tables, but it can work on more than two tables. Our Algorithm merges Table (A) with Table (B) once the key mapping has been detected. The merging data is established **on three conditions to combine the Annotated Tabular Data (ATD)**: **(1) The first condition is that they have common key mapping CTAs** (The same CTAs of Sub-col (Key matching)), **(2) The second condition is that they have no common key mapping CTAs (CTAs of Sub-col)** , but they have common CEAs (The same entities in the key matching columns), and **(3) the third condition is that there are no common CTAs and no common CEAs**, which brings it back to the annotation phase.

Once the system has detected the key mapping (K) to identify the main table. **(1) First condition**: the CTAs are common, so the system maps the CEAs of the key mapping, therefore, the system maps the CEAs of the key mapping, it applies a **similarity metric of String** (Jaccard similarity) to check the similarity between the CEAs entities. The system applies **UNION operation** between the two Tabular Data (TD) to add the columns and the rows of the secondary tables in the main table (see, fig.6.3). If there is **no similarity between these CEAs** of the key mapping (sub-col),

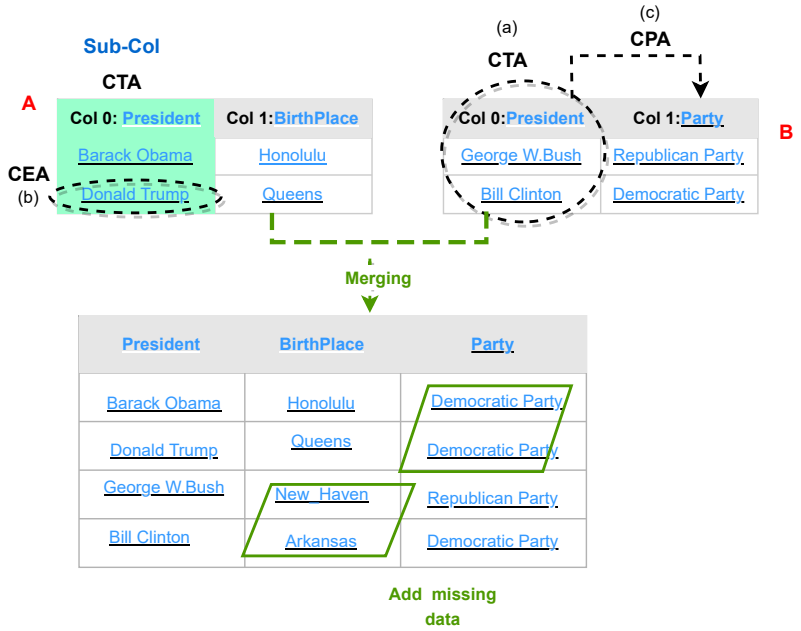


Figure 6.2: Merging Datasets.

the system does not apply the UNION technique and continues to apply **SPARQL queries** on the entities to fulfill the columns with the cell values.
(2) the second condition: the CTAs of the key mapping are no common, the algorithm applies a **similarity metric of String** (Jaccard similarity) to check the similarity between the CEAs entities. It applies **the JOIN operation** to join all the corresponding between the CEAs of the key mapping of the main table and all the rows of the secondary tables. Therefore, the system applies **SPARQL queries** to fill the entities associated with the columns of the main table (See Figure 6.3).
(3) the third condition: the CTAs of key mapping and the CEAs cells of the two tables are not exist (there is no correspondence between the columns and the rows), the system does not apply these operations to the merge and may encourage users to complete each table separately by adding missing values or adding new columns.

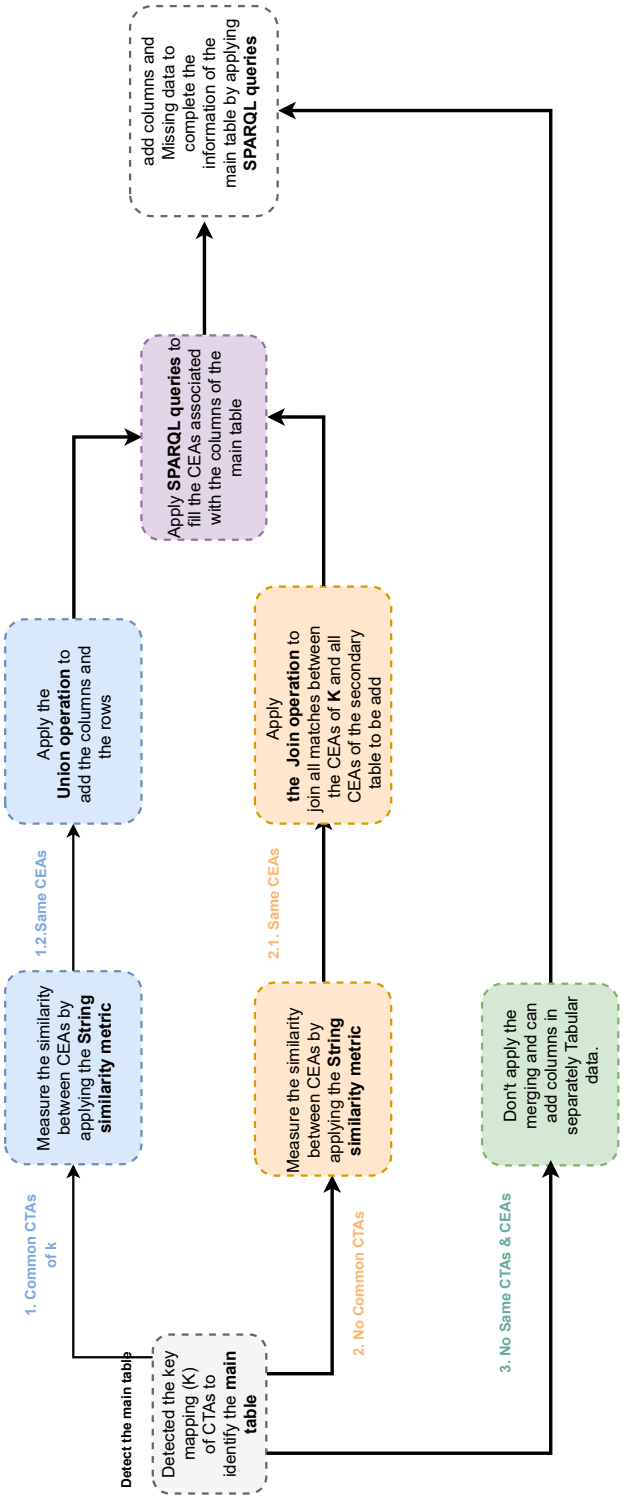


Figure 6.3: Steps in the table-merging algorithm.

6.5 SinatraAPP Implementation (SinatraAPI and SinatraUI)

In this section, we describe the architecture of the implemented framework. It **microservice-oriented architecture** (**SOA**: Service-Oriented Architecture), which is composed of **two major components**: (i) *a user interface* to facilitate interaction with the user (*front-end*), and (ii) a *back-end* to manage the endpoint of each feature and external services. It defines both **a type of architecture** and **an approach to software development** that consists in breaking down applications into simple, independent components (**component-oriented**). This application and its script code are available via this URL [16].

6.5.1 Framework Architecture

Sinatra Web Application (SinatraApp) combines the power of **a component-based back-end server with the flexibility of a web-based front-end application**. It is made up of a multitude of **autonomous components**, each of which performs a function within the application. It contains **8** components: **(1) loading open dataset**, **(2) analytics open dataset**, **(3) pre-process data**, **(4) annotation data**, **(5) repair data**, **(6) merging data**, **(7) augmentation data**, **(8) visualizing data**. This application aims to combine and develop all the phases of the previous approaches. Moreover, it is a decentralized approach, where all components are not centrally controlled. Each component is focused on the realization of a well-defined functionality. Our application is a product in its own right: **it may be produced by a different team, and have a specific technology, life-cycle, and/or deployment**.

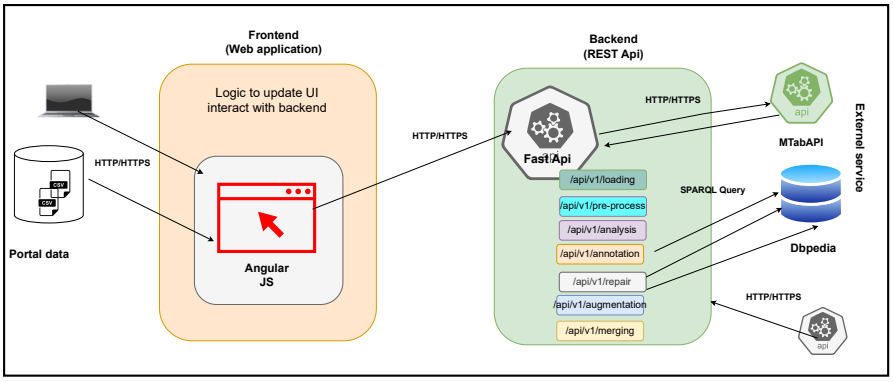


Figure 6.4: General architecture of SinatraApp.

The architecture of this framework is shown in figure 6.4 . It is designed to satisfy the presented requirements, and it is divided into two main components:

(i) **A front-end web application** to manage end-user interaction and provide the functionality needed to exploit Open Data. Each feature has sub-features to perform the tasks required for data. It is a web application composed of

[16] https://github.com/123rabida123/SinatraAPP/tree/master

sub-layers that manage the application logic, communicate with the back-end server via REST APIs, and update the UI components according to the changes made to the internal state by user interaction. This web application is built using **Angular**, **JQuery**, and **D3** based on the **JavaScript framework**.

(ii) **A back-end server is a web framework** designed to manage integration and communication with integrated external services (enrichment, knowledge bases, Knowledge graphs, SPARQL Query). It consists of **new services**: loading, analyzing, repairing, augmenting, merging, and visualizing as well as other services that added by writing *semantic enrichment* (Mtab API) by receiving requests from another API or from the front-end, querying external services and providing responses in a standard format. **The back-end** also manages the retrieval of annotated files for users. All functionalities are exposed as **REST APIs** to support communication with the front-end. It is built using **FastAPI** based on **Pydantic** and **the python programming language**.

6.5.2 Demonstration of Sinatra Web Application (SinatraApp)

In this section, we present the main operations triggered by user actions, to show how framework components (**SinatraAPI** and **SinatraUI**) interact with each other **to satisfy user requests**:

SintraAPI

This framework provides and displays **API documentation users** using *Swagger Editor* to help them use these **REST APIs** and their services. It allows users to interact with their endpoints. *Swagger Editor* is **a browser-based tool** that lets users write and edit their API specifications in *YAML* or *JSON format*. Figure 6.5 shows the SinatraAPI interface.

SintraAPI is composed of **7** microservices that communicate with each other via **network protocols**, each exposing **an API**, and which can also access a data source.

- **Loading API**: allows users to load data from a remote or local site. Users can upload their file from any Open Data portal by entering the *url* and CSV file type, and clicking on *Execute* to load the data.
- **Analytics API**: allows users to analyse their dataset from a remote or local site and clicking on *Execute* and displays the number of faults in dataset.
- **Pre-Process API**: allows users to pre-process data from a local or remote site and display the content after pre-proces.
- **Annotation API**: allows users to enrich their data from local or remote site. This endpoint communicates with the *Mtab API* and queries **the external knowledge graphs** to improve the annotation result. Users enter dataset, and the endpoint displays a set of annotated data in **JSON format** or the users **can download** the annotated data in CSV format.
- **Merging API**: allows users to enter two annotated datasets from a local or remote site and click on *Execute* to display the final annotated dataset or to save this data.

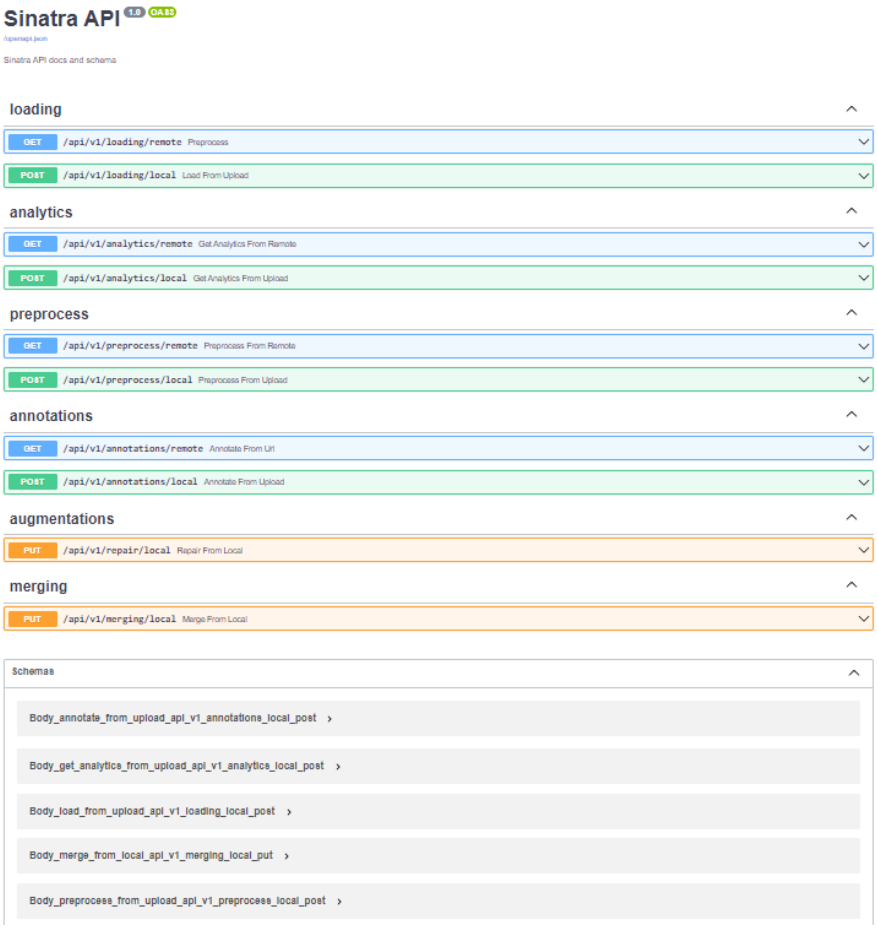


Figure 6.5: Screenshot of the SintraAPI Interface.

- **Repair API**: allows the users to add an input annotated dataset and display the results in JSON format. This Endpoint communicates with externel source (DBpedia/Wikidata), and queries and sends back the SPARQL query to fill in the missing values.
- **Augmentation API**: allows users to enter a keyword to select a new column to add. It communicates with external sources (DBpedia) by applying a SPARQL query to request a required column name.

SinatraUI

SintraUI is a user interface that **facilitates interaction with the user**. It offers a compact visualization of services, **allowing the user to visualize and navigate to each phase**. SinatraUI is built up of **7** modules, which connect to the back-end via network protocols (Figure 6.6).

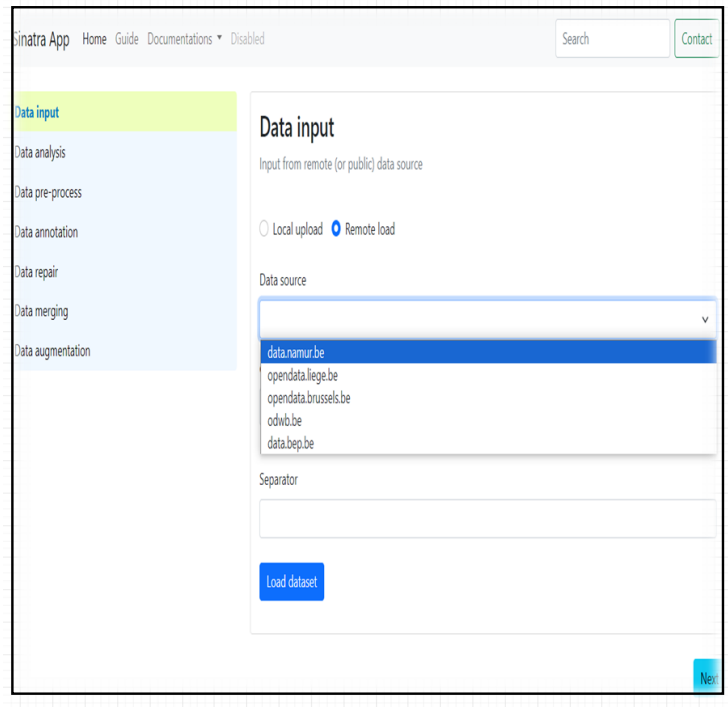


Figure 6.6: Screenshot of SinatraUI.

- **Data input**: the user can load data from a local or remote site. He/she must select the data portal, indicate the *dataset ID*, the file type and then click on *Load dataset* to load the file before moving on to the Next service. Figure 6.7 illustrates an example of how the user can load a dataset from the Namur portal or any other portal.

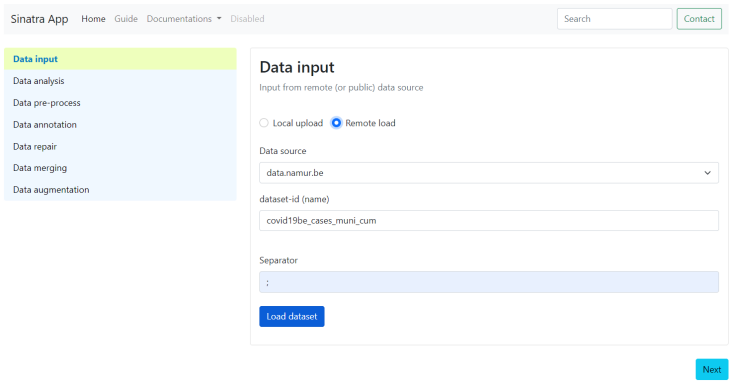


Figure 6.7: Screenshot of loading dataset phase.

- **Data analysis**: the user **automatically visualizes the percentage and number**

of faults in the selected dataset, which are displayed in charts and graphs of the percentage. Figure 6.8 shows an example of data analysis by displaying the number of faults in this dataset.

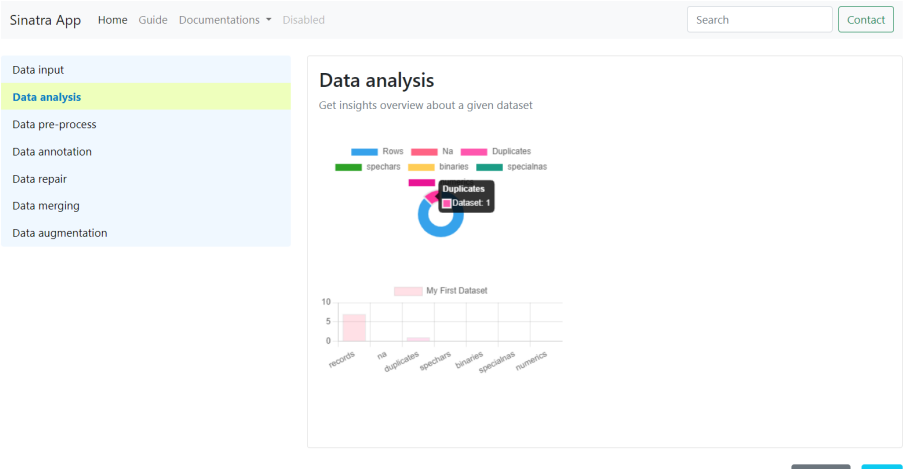


Figure 6.8: Screenshot of data Analysis phase.

- **Data pre-process**: the user visualizes clear, unified data (delete the duplicate rows or special characters, transform to uppercase,...), and can perform actions on the dataset: edit the cellular dataset, add rows/columns, delete rows/columns and send a request to the back-end to save the dataset. The two figures 6.9, and 6.10 illustrate the dataset before and after the pre-processing module, which transform uppercase letters into lowercase and remove duplicate columns.

  The two figures 6.11, and 6.12 show the dataset before and after the pre-processing module, which assigns the abbreviation for each cell in the column entitled *Abbrevation*.
- **Data annotation**: the user obtains annotated data automatically during this phase and can perform actions to improve the enrichment of certain table cells. Figure 6.13 provides an annotated table, which the user can modify manually if he/she knows the correct annotation in certain cases.

- **Data repair**: the user can enter annotated dataset, and the system automatically sends a request to the back-end to complete the missing value. Finally, the user can obtain complete information.
- **Data augmentation**: the user can enter the annotated dataset and the name column needed to add or complete the dataset, and the system automatically sends a request to the back-end to obtain the name of the new column and complete its content. Finally, the user can obtain a dataset with the new column added.
- **Data merging**: the user can merge two annotated datasets. The user must

Index	Commune	cas	minimum	Arrondissement	date	NIS5	PROVINCE	REGION
0	Biévre	1091	1091	Arrondissement de Dinant	28/02/2022	91015	Namur	Wallonia
1	Gedinne	1449	1449	Arrondissement de Dinant	28/02/2022	91054	Namur	Wallonia
2	Hamois	2781	2781	Arrondissement de Dinant	28/02/2022	91059	Namur	Wallonia
3	Onhaye	1050	1050	Arrondissement de Dinant	28/02/2022	91103	Namur	Wallonia
4	Biévre	1091	1091	Arrondissement de Dinant	28/02/2022	91015	Namur	Wallonia
5	Assesse	1449	1474	Arrondissement de Namur	28/02/2022	92006	Namur	Wallonia

Figure 6.9: Screenshot of before pre-processing.

Add column

Index	commune	nbre de cas	nombre de cas minimum	arrondissement	date	nis5	province	region
0	biévre	1091	1091	arrondissement de dinant	28/02/2022	91015	namur	wallonia
1	gedinne	1449	1449	arrondissement de dinant	28/02/2022	91054	namur	wallonia
2	hamois	2781	2781	arrondissement de dinant	28/02/2022	91059	namur	wallonia
3	onhaye	1050	1050	arrondissement de dinant	28/02/2022	91103	namur	wallonia
4	assesse	1449	1474	arrondissement de namur	28/02/2022	92006	namur	wallonia
5	anhée	3781	1747	arrondissement de dinant	28/02/2022	91059	namur	wallonia

Figure 6.10: Screenshot of after pre-processing.

enter either two local annotated datasets or two remote datasets. The key mapping (sub-col) in this step is by default the first column of the main table, and the user can insert any sub-col he or she prefers. The system shows the user the final dataset after merging. Figure 6.14 shows the merging interface, where the user has chosen the **Key mapping** (sub-col) to merge two datasets "*https://dbpedia.org/ontology/Place*". Figure 6.14 displays a complete table by merging two related datasets. Users can download datasets in standard

Load dataset

Data preview

View the dataset as table

Index	Entité	Localité	Code ins Localité	Code postal	Nombre de Voiries	Abréviation	Total populatio
0	NAMUR	Loyers				LO	
1	NAMUR	Bouge				BG	
2	NAMUR	Malonne				ML	
3	NAMUR	Naninne				NN	

Next

Figure 6.11: Screenshot of before pre-processing.

Choisir un fichier namur-limite-des-localite.csv

Separator

;

Load dataset

Data preview

View the dataset as table

Index	entité	localité	code ins localité	code postal	nombre de voiries	abréviation	total population de
0	namur	loyers				loyers	
1	namur	bouge				bouge	
2	namur	malonne				malonne	
3	namur	naninne				naninne	

Figure 6.12: Screenshot of After pre-processing.

format. For more details on SinatraApp, you can visit the Github [17].

[17] https://github.com/123rabida123/SinatraAPP/tree/master

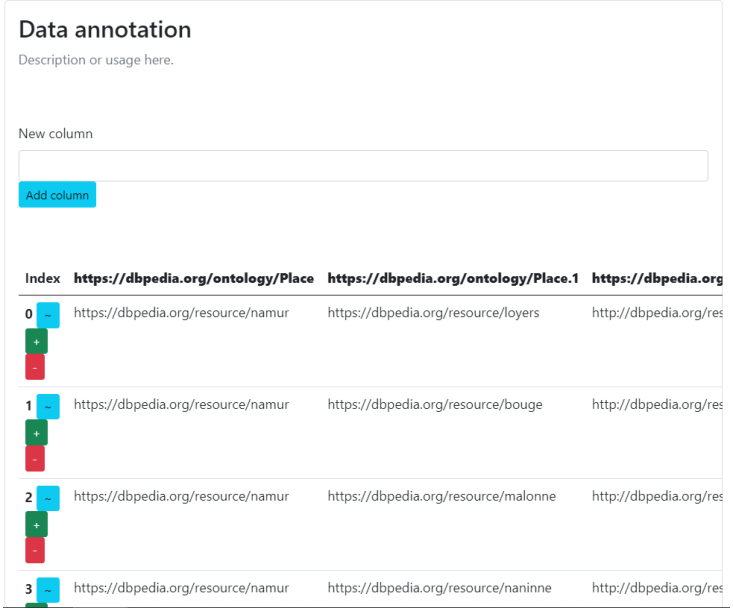


Figure 6.13: Screenshot of annotation data phase.

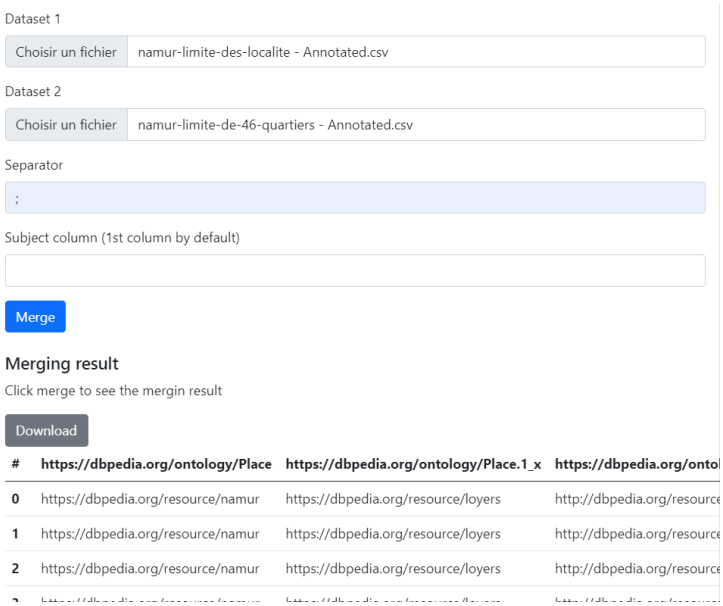


Figure 6.14: Screenshot of merging data phase.

6.6 Conclusion

In this Chapter, we describe two solutions to address certain issues of the barriers of Open Data and make SINATRA's contribution more efficient.

The first part of this chapter presented **the most requested components** to implement **SinatraApp** contribution, which **the requirements analysis composed of two specific requirements** (user and system).

Afterwards, the second part was **the data analysis functionality**, which aimed to help analysts identify and examine quality issues in Open Data and presented a catalog of defects in public datasets. It also showed **a 25 real use case** to demonstrate the performance of its features. In Chapter 7, we evaluate this contribution by applying a bootstrap method to estimate a classifier's Recall, Precision, construct corresponding confidence intervals, and standard error **to estimate its performance**. We intend our **estimate to be as close as possible to the "true" Precision and Recall**.

Next, the third part was **merging related Open Tabular Data (TD)**, which focused on finding similar datasets, which allowed data analysts to collect and combine datasets into a single dataset in order to get an comprehensive perspective on a certain issue or subject matter. In Chapter 7, we demonstrate this approach on real use cases available on the Namur data portal to prove the capability of the fusion approach.

The last part described **the architecture, implementing the identified requirements into SinatraApp**, demonstrating its effectiveness by applying the real-life case studies available on the Namur portal (Belgium), and making the source code available on GitHub for the stakeholder community to access. SinatraApp has a UI that effectively engages **data publishers and data analysts** by suggesting tools for analysis, interpretation with the data portal and other features that can be used without specialized knowledge. In Chapter 7, we describe the evaluation of the application and the results of experiments conducted with **20 participants** to confirm **the foreseen user-friendliness and usability** of SinatraApp Web.

Table 6.2: Datasets selected for the data analysis module.

Dataset	Data Portals	Description	Size	Downlaods
V'Lille - Disponibilité en temps réel	OpenDataSoft	Availability of bicycles and self-service bicycle spaces in real time	44K	6 859 809
Parkings-moto	Data Portal of liége	Geolocations and addresses of parking lots for motorcycles	4K	390
Inventaris overheidsopdrachten 2021	Data Portal of Brussels	Public contracts awarded by the City of Brussels in 2020	1082K	8
Vbx-open-data	Data Portals of Wallonia	VBX open data	298K	12
Stationnement pour personnes handicapées	Data Portal of liége	Geolocated parking spaces for disabled people on the territory of the City of Liege	198K	96
Public toilets	Data Portal of Brussels	Location of the public toilets on the territory of the City of Brussels	41K	343
Insurance	Kaggle	Medical Cost Personal (Insurance Forecast)	58 K	181101
Abalone	Kaggle	Predicting age of abalone from physical measurments	191K	15302
Maisons-des-enfants	Data Portal of Brussels	location of the Children's Houses for the young people of Brussels (6-12 years old)	3 K	332
Adult	Kaggle	Predicting whether income exceeds $ 50K/yr based on census data	461K	49266
Limites de 46 quartiers	Data Portals of Namur	The territory of Namur is divided into 46 districts.	281K	473
COVID-19 Pandémie	Data Portals of Namur	Number of confirmed cases per municipality	650K	290
Permit	Kaggle	San Francisco building permits	96M	18591
Points verts de l'ADEPS	Data Portal of Wallonia	"Points Verts" walks, marked out walks are proposed in about twenty sites spread all over the territory of Wallonia and Brussels.	1025K	207
Iris	Kaggle	Classify iris plants into three species	4K	339777
Position des bus en circulation sur le réseau STAR en temps réel	OpenDatasoft	List of the positions of the buses used on the STAR network in real time, the line and the possible destination on which they circulate.	35K	13 741 233
Shop&Drive + Kiss&Ride Real Time	Data Portal of Liege	Shop&Drive is a free parking offer (maximum 30 minutes)	341K	102 739
Prochains passages des lignes de bus du réseau STAR en temps réel	Open Datasoft	List of the next bus stops on the STAR network in real time, including departure and arrival times.	41M	6 099 849
Structure de la population par d'âges quinquennaux	Portal data of Namur	Distribution of the male and female population in the municipality of Namur, by age group from 0 to more than 100 years.	14K	56
Covid-vaccine	Kaggle	Daily and total vaccination for COVID-19	12M	93587
COVID-19 Pandémie- Nouvelles contaminations par commune	Portal Data of Namur	Number of new confirmed COVID-19 cases per municipality per day provided by the public institution Sciensano	44M	76
Breast Cancer	Kaggle	Predict whether the cancer is benign or malignant	50M	284262

Part IV

Evaluation and Discussion

CHAPTER 7

EVALUATION AND DISCUSSION

"It's not that we use technology, we live technology."

— ***Godfrey Reggio, Film Director***

7.1 Introduction

In Chapter 4 , we described and specified the **Pub_LOGD framework** and the **SINATRA framework**. Once the Pub_LOGD framework for the integration and publication of **LOD** is implemented, it is necessary to assess **the feasibility and usability** of this framework by applying **real use cases** (Population, Workers, Housing and Covid-19) from Belgium's local catalog to this framework **in order to help data publishers obtain data published in the interconnected RDF format**. In addition, we implemented an extension to this framework (Pub_LOGD Extension) to meet user requirements.

In Section 7.2, we measure user satisfaction and the ease of use of these proposals, we engage **8 participants** with different skills to test these frameworks and complete the post-survey with their experience to gather their feedback for calculating them. In addition, we discuss the limits of **Pub_LOGD** to show its weakness .

In Chapter 6.3, we developed a data analysis solution to help users detect data errors in open datasets. We used **25 datasets** to select these malicious data. In this Chapter, we seek to estimate its performance by using *precision* and *recall* measures

and creating bootstrap confidence intervals **to estimate the average numbers of the system's performance in detecting data faults on datasets**.

In addition, we demonstrate the potential of **the annotated data fusion phase** by illustrating an example of merging annotated datasets using the two datasets from the Namur data portal.

In Chapter 5, we began by evaluating the performance of Sinatra approach. We used the 2019 and 2020 Benckmark datasets to evaluate annotated data quality and data accuracy by calculating *Precision* and *Recall* measures. We also used a real-life case study based on data from the Namur portal to show and demonstrate the ODSAG tool's capabilities. Furthermore, we have shown the limits of the **Sinatra** approach to demonstrate its weaknesses.

In this chapter, we aim to evaluate the usability and effectiveness of **SinatraApp** to answer the following question:

(i) RQ5. "What is the performance, usability, and usefulness of the developed solutions?"

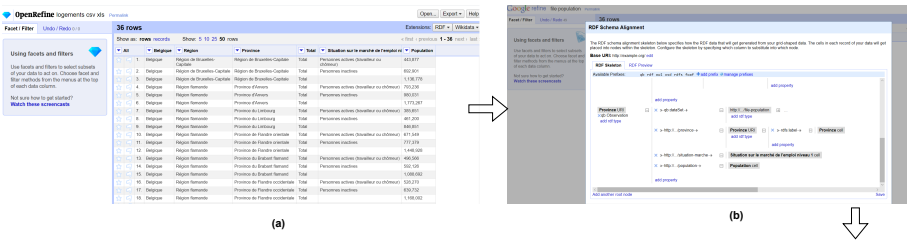
A task-based user evaluation proved that the **Sinatra Web Application** (SinatraAPP) is **understandable, usable, and capable** of loading the data, analyzing the data, processing the data, annotating the data, repairing the data, merging the data, and augmenting the data with **little effort and time, thanks to user tests involving people with different skills**.

7.2 Assessment of Pub_LOGD and Pub_LOGD Extension

In this section, we present the evaluation efforts made to demonstrate the feasibility of our approaches (**Pub_LOGD** and **Pub_LOGD Extension**). To this end, we first describe **the experiment results of our framework**. Second, we present a description of the real-life case study on which we applied our **Pub_LOGD** in the section 7.2.2. Then in the section 7.2.3, **we present an evaluation description by engaging the users to evaluate Pub_LOGD extension the usability of this contribution by completing a survey based on Technology Acceptance Model (TAM)**. In the section 7.2.1, we discuss the use of these approaches and their benefits.

7.2.1 Experimental results

To evaluate our proposal, we apply the aforementioned catalog to the *Pub_LOGD* framework in order to assist end-users in achieving published data in the interlinked RDF format in an easy and transparent manner.



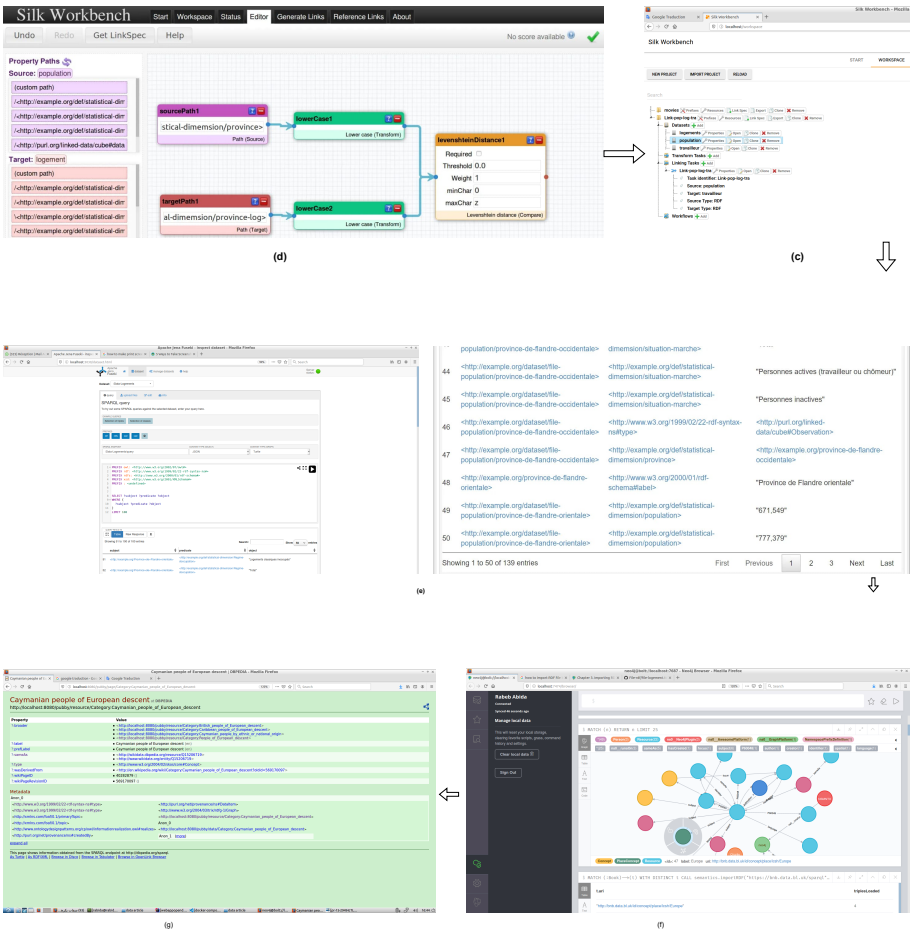


Figure 7.1: User Interaction with the publishing Framework.

At the beginning, the user is provided with **guidelines** that explain the manner in which to utilize the framework and what tools are involved to ensure the publishing process. Besides, he is supplied with some reliable data portals and tutorials to facilitate his tasks. Once the process is clarified, the user can launch the publishing steps. As shown in Figure 7.1, in the framework **Pub_LOGD**, the user uses at first *Open Refine* to preview the data, clean and cluster statistical data according to its needs (Figure 7.1 (a)) and builds the RDF skeleton using the RDF Data Cube Vocabulary (Figure 7.1 (b)). At the first step of the (Pub_LOGD Extension), the user utilizes *Open Refine* to preview the data, clean, and cluster statistical data according to its needs (Figure 7.1 (a)), and then applies a Sparqlify tool to create a SPARQL query to build the RDF format (See Annexes B). The following steps are the same as those of **Pub_LOGD**. Then, an interlinking step to discover the relationships between data elements in different linked data sources and improve knowledge with related external data sources is followed through *SILK* (Figure 7.1 (c)). For example, in Figure 7.1 (c), we get the specific link between the population RDF file and the

housing RDF file (Figure 7.1 (d)). Afterwards, the user stores his data via *Apache Jena Fuseki*, which provides him an easy user interface for server monitoring and administration (Figure 7.1(e)). It is given a page that can be used as a training area to practice and test SPARQL queries to get the *subject*, *predicate* and *object* of the dataset. Next, with the integration of *neo4j* (Figure 7.1 (f)), the user may have a holistic and complete view of his data as well as the semantic relationships that exist among them. From this stage, the user can be sure if these data meet their requirements and expectations. If it is the case, he continues to properly publish the according data using *Pubby* (Figure 7.1 (g)). We note that at each step, the user is guided with instructions and is aware of the state of his advancement in the publishing process. Moreover, he couldn't proceed to the next step unless the previous one had been correctly performed. More details about our framework, its source code, and implementation aspects are available online at [1].

According to our data, the user which can be a public organization can obtain statistical data that he desires which improve his decision-making. In such context, multiple combinations of needs can be formulated via combining different facets of every data category. For instance, the public institution, in charge of helping jobless persons in finding jobs such as *ONEM* in Belgium, is interested to obtain statistics of workers per place of residence in order to strengthen its efforts in regions with low statistics more than other regions. Besides, the public organization that deals with social housing, such as *CPAS* in Belgium, is interested in knowing the proportion of beneficiaries in relation to the size of the population or the number of workers in the same family of these beneficiaries. The Institute for Health Measures statistics, such as the *IHME*, wanted to obtain the number of new daily deaths attributable to COVID-19 in Belgium by age and sex in terms of the population size in the same region. The purpose is to adapt and regularize their decisions compared to statistics reflecting reality.

Our end-to-end framework **Pub_LOGD** includes a user-friendly interface that can be executed either from a web browser or through a standardized *RESTful web API* in order to enable easy integration into more complex workflows. It permits to eliminate searches for ineffective tools and thus reduces the significant time it takes. Furthermore, it allows eliminating the problem of software tools dependencies and thus significantly reducing time in terms of tools research and combination.

7.2.2 Dataset Description

We have offered users the datasets needed to operate this framework. The dataset consists of a local Belgium catalog [2] which consists of statistical data related to four data collections: (1) population, (2) workers, (3) housing, and (4) COVID-19. These data exactly concern the Wallonia region in Belgium, where we aim to study the population statistics by taking into account a set of major facets per collection. The descriptions of these facets are presented below.

[1] https://github.com/123rabida123/WebAppOpenData
[2] https://statbel.fgov.be/en

Table 7.1: Dataset Description.

Datasets	Category	Facets	Size
collection 1	population	residence	139,000 rows
		sex	
		position household	
		size of the locality	
		labor market situation	
collection 2	workes	place of residence	184,000 rows
		sex	
		position household	
		professional status	
		country of citizenship	
collection 3	housing	geographical level	163,000 rows
		occupancy regime	
		type of building	
collection4	COVID-19	number of deaths	128,000 rows
		sex	
		age	
		region	
		province	

As shown in Table 7.1, the catalog includes three datasets: (1) collection1: referencing to the population category and including five facets; place of residence, sex, position in the household, size of the locality, and labor market situation (139,000 rows), (2) collection2: denoting the workers' category involving four facets i.e., place of residence, sex, position in the household, and professional status and country of citizenship (184,000 rows), (3) collection3: Typical housing with taking into account three facets i.e., geographical level (municipality), occupancy regime and type of building (163,000 rows), and (4) collection4: referenced the number of deaths by COVID-19 with regard to five facets i.e., number of deaths, sex, age, region and province (128,000 rows).

7.2.3 Evaluation of Pub_LOGD Framework and Pub_LOGD Extension

In this section, we present the evaluation efforts conducted **to demonstrate the practicality and usability** of our **Pub_LOGD** and **Pub_LOGD Extension** proposals. Our main goals are to provide a mechanism **to promote user engagement with governments and the reuse of government information** (e.g., the users used the datasets in Section. 7.2.2) in order to enable data consumers to access a publishing solution, allowing them to produce and publish high-quality data on the web.

User Study

We used two post-surveys [3] [4] (See Appendix B) that were realized to gather feedback from participation by engaging participants with different skills at the University of Namur (Belgium), the University of Madrid, and the University of Paris-Dauphine. **Eight participants**, who cover the relevant occupation profiles, have taken part in the experiment; the profiles of participants are summarized in Table 7.2.

Table 7.2: The relevant professional profiles of the participants.

#	Occupation	Education	Technical skills	Semantic Web	Similar tools
P1	R&D engineer	Post-Doc	7	7	5
P2	R&D engineer	PhD	6	5	5
P3	Data Scientist	Master Degree	6	1	1
P4	Data Scientist	Master Degree	6	2	2
P5	Data Scientist	Master Degree	6	1	1
P6	Data Scientist	Master Degree	6	2	2
P7	Application Developer	Master Degree	6	2	3
P8	Application Developer	Bachelor Degree	5	1	2

A brief introduction to the topic and the platform was presented to the participants, accompanied by real-life case studies (https://statbel.fgov.be/en). Then, they were invited to complete a profile survey. This survey asks users about their occupation and education level and asks them to rate their technical skills such as programming and query languages, knowledge of semantic web technologies, and familiarity with similar platforms or tools on a Likert scale (i.e., 1 for "not familiar at all ", 5 for "strongly familiar").

Once the first step was completed, we organized these participants into **two groups** with the same technical skills. Therefore, these participants are categorized into two groups: **Group-1** is composed of 4 participants (2 data scientists, a R&D and a developer) to evaluate the **Pub_LOGD** and **Group-2** is composed of the same to evaluate the **Pub_LOGD Extension**. Then, they were asked to complete a post-survey about their experience with the framework to measure *user satisfaction*.

The post-surveys included two types of questions: questions with a 7-point Likert scale (from “Strongly Disagree” to “Strongly Agree”) based on the TAM [55] to assess two aspects: perceived usefulness and perceived ease of use (Table 7.3). Verbatim questions are used to justify the scores given, to comment on what they liked and disliked about the framework (Table 7.4) and to provide any feedback they deem important. The results of the post-surveys are presented in the following Table 7.5.

Table 7.3 summarizes the corresponding questions for each construct and explain the reason of our choice.

After collecting participants' feedback, a set of metrics, i.e., **median**, **mean** and **standard deviation**, are computed for the questions aforementioned to evaluate the following aspects: (**A1**) ease of use of each framework feature; (**A2**) overall ease of use of the framework; (**A3**) usefulness of each framework feature; and (**A4**) overall

[3] https://tinyurl.com/3zpkwce8
[4] https://tinyurl.com/2s4kucx7

Table 7.3: The corresponding questions of survey.

Aspects	Question	Comments
(1)Perceived Usefulness (PU) (2)Perceived ease of use (PEU)	12 questions of the TAM questionnaire: Q1 to Q6 mean (PU) Q7 to Q12 mean (PEU)	Tam was used as it is suitable to measure the usability of a system in a standalone.

Table 7.4: The feedback given by the users.

What did you like about the framework?	**Participant**
"Easy to learn"	P1, P2
"Nice framework"	P4
"Very comfortable to use"	P2
"Possibility to see the original framework"	P1
"The interaction between the tools in process work well"	P4, P6
"Modeling in RDF format is easy"	P1, P5, P6
"Very good Plateform, there are guidelines that explain how to use the framework and the necessary tools"	P8
"Very good strategy to use a feature to generate links between data"	P1, P2, P4
"Framework is really useful for storing RDF data"	P2, P3
"It is very attractive to visualize the data and the relationship between the different concepts in the RDF data"	P1, P2, P3, P7
"This is an interesting framework for publishing RDF data"	P1, P2, P4, P6
"It is very important for our work (e,g., unifying and cleaning tasks)"	P8, P5, P7
"It is very easy to install this framework in one line"	P3, P5, P8
What didn't you like about the framework?	**Participant**
"Should be remove the iterative mode, because the user couldn't pass to the next step without the previous step being correctly done"	P3, P8, P5
"Defining an ontological model to annotate data is a task a little bit complex"	P8,P5
"Generating links between RDF graphs is not easy for a non-expert user"	P8, P3
"The content of the dataset should be modeled, it can provide pertinent information, I think"	P1
"Must remove the iterative method to allow user to choose the tools he/she wants to use"	P2
"The use of these tools always requires the involvement of the end user"	P1
"It can be confusing to clean up the dataset before analyzing it"	P2
"The user spends a lot of time doing the tasks, which is why it is necessary to use automatic tools"	P1
"The portal data icon must be placed before the workspace icon, because the user first selects the dataset and then starts working"	P4, P6

usefulness of the framework to improve data quality and facilitate the integration and publication of statistical LOGD. The A1 (respectively A3) questions are used as a reference to understand the answers to the A2 (respectively A4) questions. These statistical measures are chosen because they are the most appropriate for analyzing Likert data and for having a central tendency measure [31]. Verbal thoughts and responses collected from the free text questions are coded using short sentences to retain context and conceptual relations.

Results of the Pub_LOGD Framework Assessment

Table 7.5 presents the median, mean, and standard deviation of the questions with a 7-point Likert scale regarding the two aspects (perceived usefulness and perceived ease of use) evaluated for Pub_LOGD and Pub_LOGD Extension (consult annexes B). For more details on the calculation, you can consult the evaluation results on this URL [5].

[5] https://github.com/123rabida123/WebAppOpenData

In **Group-1**, most of the participants notice that the proposed framework was easy to use (median and mean ≥ 5 for A2 with a low standard deviation around 1), useful to enhance the quality of open statistical data and easy to publish linked open statistical data (median and mean ≥ 5 for A4 with a low standard deviation around 1). More specifically, the data scientists note that the framework met their expectations, made their work easier and was user-friendly for features such as cleaning and unifying datasets and enriching and discovering datasets. The participants found this framework **very useful and helpful in their applications**. Indeed, the means of participants' scores on the ease of use (A1) and usefulness (A3) questions ranged between 3 and 5 for these features.

Table 7.5: Median, mean and standard deviation (SD) of survey scores.

	Group-1 (Pub_LOGD) (N=4)		**Group-2(Pub_LOGD Extension) (N=4)**	
	A2.Perceived ease of use	**A4.Perceived usefulness**	**A2.Perceived ease of use**	**A4.Perceived usefulness**
Median	5.5	5	6	5.7 (0.82)
Mean (SD)	5.25 (1.18)	5.67 (1.14)	6	6 (1.41)

Table 7.5 presents the median, mean, and standard deviation of the questions with a 7-point Likert scale regarding the two aspects (perceived usefulness and perceived ease of use) evaluated for Pub_LOGD and Pub_LOGD Extension (consult annexes B). For more details on the calculation, you can consult the evaluation results on [6].

In **Group-1**, most of the participants notice that the proposed framework was easy to use (median and mean ≥ 5 for A2 with a low standard deviation around 1), useful to enhance the quality of open statistical data and easy to publish linked open statistical data (median and mean ≥ 5 for A4 with a low standard deviation around 1). More specifically, the data scientists note that the framework met their expectations, made their work easier and was user-friendly for features such as cleaning and unifying datasets and enriching and discovering datasets. The participants found this framework very useful and helpful in their applications. Indeed, the means of participants' scores on the ease of use (A1) and usefulness (A3) questions ranged between 3 and 5 for these features.

7.3 Limitations of Pub_LOGD framework And Discussion

In this section, we discuss some limitations of Pub_LOGD. This framework is **a semi-automated version of Linked Open Government Data Publishing**, where expert user intervention is absolutely necessary for modeling and interlinking the data. This work requires preliminary work to raise users' awareness of Open Statistical Data and data publication, for example, **sharing a survey to increase awareness and utility of OGD**. The **data extraction** step of this process was based on a National Statistical Portal for Belgium, which requires us to extract data from international portals to evaluate our framework. In some cases, data transformation requires the use of **the Open Refine's expression** (GREL) functions for data cleaning, data

[6] https://github.com/123rabida123/WebAppOpenData

unification, cell and column editing, and data transposition. This step **is not easy for users**, as they have to read a lot of documentation to master the syntax of the expressions and how they work.

For an expert-user to build an ontological model using *RDF Extesion* of *Open Refine*, they need to go through the *data modeling step*. This lets them find the *URI* entities, share the data, and add notes to it. In addition, the modeling by *Sparqlify* **is not easier for users who are not familiar with SPARQL queries** and simply annotates the header of the dataset and the relationship between the headers. This step must be directly mapped to the RDF format in order to facilitate data modeling for users.

The **data interlinking** is based on *Silk*, which is a framework for integrating heterogeneous data sources, has been discussed, but there **is no details on how it has been enforced**. Further investigation is required to understand the relationship among the data models and political priorities [97].

The methodology of this work is not yet sufficient, and some tools are not yet available; they have completely disappeared. In addition, the limitation of this evaluation is **the type of collection**, as in this context we focused purely on statistical data, perhaps **we should use other types of data collection to increase the performance and usefulness of the Pub_LOGD framework**.

Moreover, other limitation, **the number of participants to evaluate the ease of using our approach may be small**, because when we sent these post-surveys, we expected the response from the largest number of participants, but unfortunately, we received just **8 responses**, but referring to previous studies [148] and [64], the use of **at least 5 participants for usability tests is a good baseline**. However, to increase this representativeness of the participants, we suggest using other communication channels or collecting data on-site in universities or public places. In this study, this was not feasible due to the COVID-19 situation.

7.4 Data analysis Results

The most recurrent data quality issues are presented first in Section 6.3.2. In this context, we invest our efforts in the analysis of real-life use cases. We used more than **25 datasets** for this evaluation, which are listed in Section 6.3.2.

7.4.1 Duplicate examples

Duplicate examples in a dataset are defined as two or more rows that refer to the same entity. These problems make larger datasets and data pretreatment more difficult to continue working, and result in results that are not accurate. In this cas, these datasets *D1*[7], *D2*[8], *D3*[9], *D4*[10], *D23*[11], and *D22*[12], that contained duplicate rows. Table 7.6 indicates that the total number of problems is **34** rows.

[7] https://www.kaggle.com/datasets/mirichoi0218/insurance
[8] https://www.kaggle.com/datasets/uciml/iris
[9] https://tinyurl.com/yrew8a2e
[10] https://tinyurl.com/ys47xjjt
[11] https://github.com/123rabida123/Datasets-Analysis
[12] https://tinyurl.com/5n97f7uc

7.4.2 Unknown unit of measure

When the unit of measurement is not clearly defined or unknown, it becomes difficult to interpret the data correctly, to perform reliable calculations, aggregations, or comparisons between different data sets, and to integrate and merge these data. Standardization of units requires additional effort, and differences may introduce errors or inconsistencies into the integrated dataset. Different analysts may make incorrect statements about the data's size, context, or scale, leading to erroneous interpretations and findings. Furthermore, in the absence of a known unit of measurement, it becomes difficult to reproduce or replicate analytical processes, which can lead to divergent results and affect the scientific accuracy of the research. In this case, we analyzed this problem and found the following datasets: *D7*[13], *D8*[14],*D9*[15], *D11*[16],*D23*[17], *D13*[18], *D19*[19], and *D6*[20], which contained specific columns, did not indicate the unit of measurement. Table 7.6 indicates that the total number of this problem is **27931** cells.

7.4.3 Null type missing values

In this context, we've noticed that some datasets on data portals contain empty cells, which can reduce the data's potential value and especially the performance of data annotation systems. This problem can lead to incomplete datasets, limiting the completeness and reliability of the information made available to users. It can also make it difficult to interpret the data correctly. Users may find it difficult to determine whether missing values are due to data collection errors, non-response, or other reasons. To overcome these drawbacks, data providers must seek to minimize missing values in open public data through robust data retrieval and repair operations. We have proposed a data repair feature in Chapter 5 and a data repair API in Chapter 6 to reduce this shortcoming.

In this case, we analyzed the problem and found the following datasets, which you can find here: [21] : *D1*, *D12*,*D13*, *D11*,*D14*, *D15*, *D116*,*D17*,*D18*, *D7*, *D23*, *D24* and *D25*, which contained certain columns, did not indicate the unit of measurement. Table 7.6 indicates that the total number of this problem is **2285285** cells.

7.4.4 Special missing values

Special codes or symbols are used to represent missing data. However, the meaning of these special missing values may not be clearly defined. This ambiguity can lead to confusion and misinterpretation during data analysis, as different analysts may

[13]https://tinyurl.com/ys6svxe6
[14]https://tinyurl.com/uxhpwmwd
[15]https://tinyurl.com/y6jj27ef
[16]https://tinyurl.com/42c7tymh
[17]https://github.com/123rabida123/Datasets-Analysis/blob/main/adult.csv
[18]https://tinyurl.com/yw8m3hwx
[19]https://tinyurl.com/3fwtxfc3
[20]https://tinyurl.com/52ebhedy
[21]https://github.com/123rabida123/Datasets-Analysis/tree/main

attribute different meanings or interpretations to these special missing values. It can also reduce the transparency of open government data. This lack of transparency makes it difficult to assess the quality and reliability of data, making it harder to make informed decisions. At the same time, data users may need help understanding its meaning, origin, or implications. We suggested a new data pre-processing function for a specific treatment to minimize this drawback. In this case, we analyzed this problem, where we found the following datasets, which you can find here [22] : *D1*, *D22*,*D10*, *D9*, and *D24*, which contained certain columns did not indicate the unit of measurement. Table 7.6 indicates that the total number of this problem is **4370** cells.

7.4.5 Strings with special characters

Special character strings can cause data quality problems. They may result from input errors, encoding problems, or inconsistencies in data formats. Special characters can complicate data transformation, aggregation, search, or filtering, especially when combining data from different sources or systems. It can also make reconciling or merging data sets for complete analysis and exploitation difficult. To mitigate these drawbacks, data providers must ensure coherent encoding, validation, and data entry practices, ensure data users know the potential difficulties associated with special characters, and use appropriate data cleansing, transformation, and encoding techniques to ensure precise interpretation and analysis. We found the following datasets, which you can find here: [23] : *D22*, *D1*,*D14*, *D24*,*D23*, and *D25*, which contained certain columns that did not indicate the unit of measurement. Table 7.6 indicates that the total number of this problem is **22097** cells.

7.4.6 Binary missing values

The use of a binary missing value leads to the loss of valuable information about the data that is lacking. This loss of information can impact the accuracy and reliability of data analysis and decision-making. It may not adequately represent the scope of missing data in the dataset. This problem can complicate data integration when combining multiple datasets. If different datasets use different missing binary values to represent the missing data, it becomes difficult to align and merge the data with any accuracy. To address these limitations, we have developed a data pre-processing function (Chapter 6) to adopt more comprehensive binary missing data handling strategies in open public data to improve the reliability and interpretability of this data. We can see the following datasets, which can be found here: *D1*, *D6*,*D14*, *D21*,*D11*, *D24*, *D7*, and *D19*, which contained certain columns and did not indicate the unit of measurement. Table 7.6 indicates that the total number of this problem is **15,551443** cells.

[22] https://github.com/123rabida123/Datasets-Analysis/tree/main
[23] https://github.com/123rabida123/Datasets-Analysis/tree/main

Table 7.6: Quantity of data faults

Data faults	Datasets		Count
P1: Duplicate examples	- Insurance (D1) - Iris (D2) - V'Lille - Disponibilité en temps réel (D3) - COVID-19 Pandémie-Nouvelles (D4) - WinData(D23) - Adulte (D22)	3 1 3 1 2 24	34 Rows
P2: Unknown unit of measure	- Breast Cancer (D7) - Points verts de l'ADEPS (D8) - Maisons-des-enfants (D9) - Vbx-open-data (D11) - WinData (D23) - Stationnement pour personnes handicapées(D13) -Abalone(D19) - Limites 46 quartiers (D6)	11673 302 9 1269 39 2016 12531 92	27931 Cells
P3: Null type missing values	- Insurance (D1) - Parkings-moto (D12) - Stationnement pour personnes handicapées (D13) - Vbx-open-data (D11) - Permit (D14) - Position des bus en circulation (D15) - Shop and Drive + Kiss and Ride (D16) - Prochains passages des lignes (D17) - Covid-vaccine (D18) - Breast Cancer (D7) - WinData (D23) - Spec (D24) - JCDecaux Bike Stations Data (25)	2 3 2951 1741 2245941 734 7 33242 9 569 8 78 96	2285381 Cells
P4: Special missing values	- Insurance (D1) - Adulte (D22) - Public toilets (D10) - Points verts de l'ADEPS (D8) - Spec (D24)	2 4262 26 78 2	4370
P5: Strings with special characters	- Adulte(D22) - Insurance(D1) - Permit (D14) - Spec (D24) - WinData(D23) - JCDecaux Bike Stations Data (25)	232 1 21836 26 2 861	22958
P6: Binary missing values	- Insurance (D1) - Limites de 46 quartiers (D6) - Permit (D14) - Structure de la population par age (D21) - Vbx-open-data (D11) - Spec (D24) - Breast Cancer (D7) - Abalone(D19)	4 20 151245 208 613 6 212 2835	155143

We developed **data analytic model** to help users to discover the data faults in the Open Data. Figure 7.2 shows an example of a special dataset containing numerous types of faults.

Figure 7.3 shows a screenshot of the **Data Analysis API**. It provides two **APIs** that allow the user to load and analyze data remotely (from the Web) or locally (from the desktop). The user can load a URL or file, define the file format type and then run it to display the results.

Figure 7.4 shows a screenshot of the **Sinatra Web application (SinatraApp)** . The user loads the data, then clicks on the button to have the system automatically analyze the data. After a few seconds, the interface shows users the results of the

Data preview

View the dataset as table

Index	Povince	Abrevation	Région	Nb hôpitaux remontant leurs données	Nb hospitalisés	nb h
0	Povince	Abrevation	Région	Nb hôpitaux remontant leurs données	Nb hospitalisés	nb h
1	Oostvlaanderen	OV	Flanders	1	0	0
2	Brussels	BL	Brussels	4	4	0
3	Lià¨ge	LG	Wallonia	2	0	0
4	Namur		Wallonia	1	0	0
5	BrabantWallon	BW	Wallonia	3	0	0
6	Brussels	BL	Brussels	9	3	0
7	Luxembourg	LX	Wallonia	2	0	0

Figure 7.2: Special dataset contains many faults.

data analysis, with the number of percentages of each type of data defect. In this example, the system counts the number of erroneous data, displays the number of records and lines on the screen, and detects erroneous data, which are the strings with special characters, duplicate rows, binary missing values, null type missing values, special missing values, and unknown units of measure.

7.4.7 Are we confident of the precision of our model's recall?

In this section, we build a data analysis model and wish to estimate its performance using some metrics, such as precision and recall. We want our **estimate to be as close as possible to the "true" precision and recall**. In such contexts, we apply the bootstrap method to estimate a classifier's recall, precision, corresponding confidence intervals, and standard error. Find out more about our results here [24].

A concrete example

We would like to find the average numbers of the system's performance to detect the data faults on the datasets. To do that, **we calculate the precision and the recall from 1,000 samples and find that they are 1.00 for precision and 0.952 for recall**. However, we would like to get an estimate of how "reliable" or "accurate" this average is—or in other words, we would like to estimate the standard error] of the average we found.

[24] https://github.com/123rabida123/Datasets-Analysis/tree/main/Results

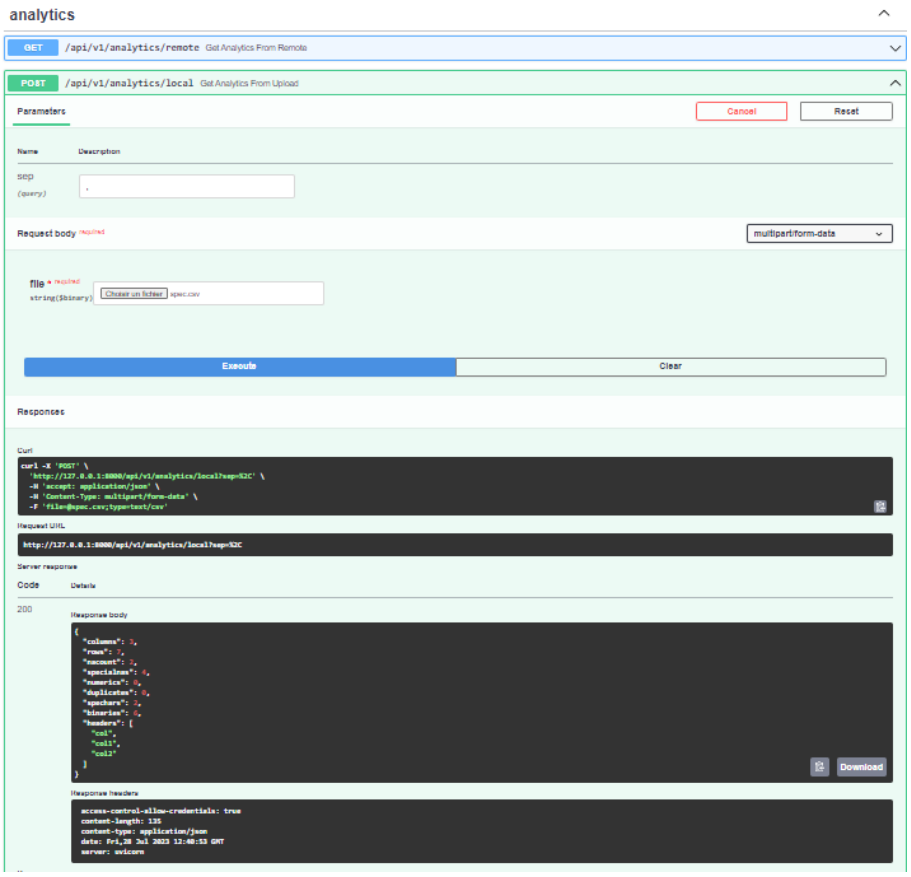


Figure 7.3: API interface for data analysis model.

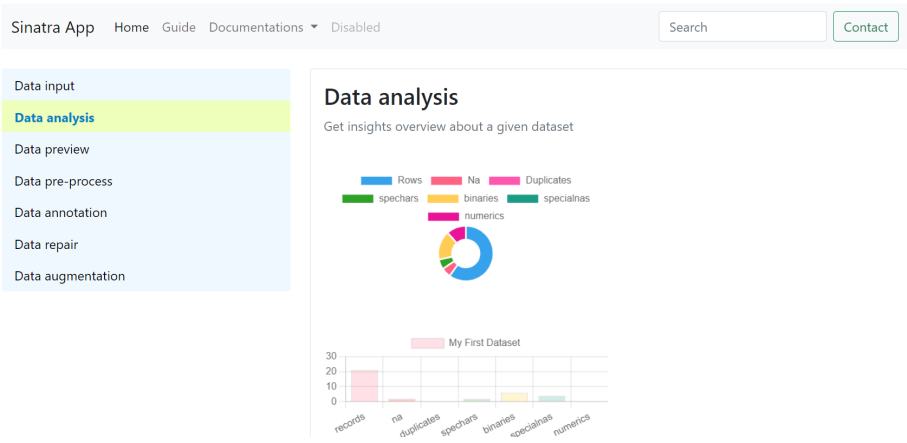


Figure 7.4: User Interface for data analysis model.

Bootstrap method?

The bootstrap is a widely applicable and extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method [87].

In general terms, bootstrapping is a resampling technique applied to estimate the sampling distribution of a statistic. It involves randomly sampling the original dataset with replacements to create multiple bootstrap samples. The statistic of interest is calculated for each bootstrap sample, resulting in a distribution of bootstrap estimates. To do this, we use our existing datasets to create new bootstrap datasets by drawing from the original datasets with replacement, i.e., data points can appear more than once in a bootstrap-generated dataset.

What is the reason for using a bootstrap method ?

If we can gather more data, that's great. Still, often we only have a certain volume of data, and we have to be satisfied with that - we can't get more data from the original population due to time constraints, budget or other reasons. We'd like to use all available data and extract as much information as possible, as we will do in the next section. Using bootstrap methods allows us to do this (in some situations).

How to estimate the performance of the data analysis model?

In the performance of the **data analysis model**, we start to create many such bootstrap datasets and use them to calculate our statistics of interest. This process enables us to obtain a **bootstrap distribution**, which can be used to estimate the distribution of the sample statistic. This development of the process was inspired by *Ron Itzikovitch* [132] . To accomplish this, we apply a bootstrapping procedure as follows:

- Sample (with replacement): 1,000 data points from the datasets.
- Calculate the precision and recall of models based on this sample and calculate the average.

We repeat these two steps 1,000 times — each time resampling data points from the **true population values** (Precision and Recall) of the datasets.

The distribution of average estimates was obtained by generating **1000 bootstrap samples by random resampling from the actual values in the population**. It calculates precision and recall on each bootstrap sample and computes average bootstrap estimates for both precision and recall.

We notice in Table 7.7 that the estimates obtained using bootstrap estimates are very close to the values we were trying to estimate:

We currently have 1,000 averages obtained from bootstrap samples. The means of those averages are 1.00 for Bootstrap Precision and 0.952 for Average Bootstrap Recall; these are our **bootstrap estimates for the sample means** (the average number of Precision and Recall (data faults) in our datasets) and are **exactly** to the means of the true Precision 1.00 and Recall estimates 0.952. We provide here in detail (Annexes B) a code script that performs bootstrapping to estimate precision and recall values

Table 7.7: The estimates obtained using the bootstrap estimates

Measure	True population		Bootstrap	
	Precision	Recall	Precision	Recall
Average	1.000	0.9528	1.000	0.9528
Median	1.000	0.9527	1.000	0.95279
Standard Deviation	0.0	0.0058	0.0	0.005841

from the true population values for calculating the average, median, and standard deviation bootstrapping estimates by using the functions in *NumPy* python.

We further calculate the standard deviations for precision and recall of these 1,000 bootstrap estimates and obtain 0.0–0.005—these are our **estimate of the standard error of the sample means**, i.e., the estimate of the standard error of the average number of the detection of the data faults we calculated using our data analysis models. **In other terms, we can anticipate that the original sample means (1.00 for precision and 0.952 for recall) will differ from the true mean by 0.005 on average.**

The *standard error* we have found can be used to construct a *confidence interval*. If we suppose a normal distribution of the sample means, **the estimates obtained using the bootstrap method allow us to state that there is a 95% probability that the precision confidence interval is [1.00, 1.00] and the recall confidence interval is [0.941, 0.964] covers (contains) the true means** - the average number of recall values of the system to detect correctly data faults by data analytics models.

7.4.8 Bootstrap confidence intervals

Confidence intervals have now become a familiar friend in the applied statistician's collection of data analysis tools. They blend point estimation and hypothesis testing into a single deductive statement with great intuitive attraction [59].

The reasons for using bootstrap confidence intervals?

Bootstrap confidence intervals are **a tool used to estimate the uncertainty associated with a parameter or statistic based on a limited sample of data**. Here are a few reasons why bootstrap confidence intervals are useful:

- Bootstrap confidence intervals are non-parametric, meaning that they do not assume a specific distribution for the original data. They are therefore more polyvalent and applicable to a wide range of situations, even when the underlying data do not necessarily follow a known distribution [37].

- They are derived directly from observed data. They do not rely on theoretical assumptions about population distribution. Bootstrap confidence intervals are therefore well suited to situations where the assumptions of traditional statistical methods may not be observed [37].

- Bootstrap confidence intervals can be used with complex models for which analytical methods of deriving confidence intervals may be difficult or unavailable. They can be applied to any statistic that can be calculated from the data, whatever the complexity of the underlying model [37].

Bootstrap confidence intervals can be constructed using **non-parametric sampling**, which basically means we don't need to make any hypotheses about the distribution of our parameter of interest (or of our data for that matter). There are several methods for constructing bootstrap confidence intervals, each with its own approach and its own hypotheses, such as the percentile method, the bias-corrected and accelerated (BCa) method, and Studentized Bootstrap Method [37, 72].

In this context, we have used the *percentile method* of the bootstrap estimate distribution to construct confidence intervals. The justification for constructing confidence intervals based on bootstrap percentiles is that we can use the percentiles of the distribution of bootstrap estimates since they will likely be sufficiently similar to the distribution of the parameter we are trying to estimate.

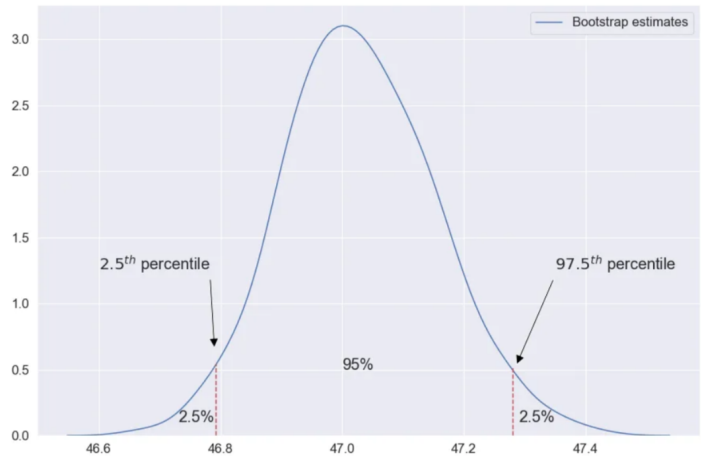


Figure 7.5: Percentiles of the distribution of bootstrap estimates [37].

As an example we will construct a 95% confidence interval by using the distribution of bootstrap estimates (figure 7.5):

(i) The lower bound of the confidence interval will be the 2.5th percentile of the distribution (46.8) — the value below which 2.5% of the bootstrap estimates are found.

(ii) The upper bound of the confidence interval will be the 97.5th percentile of the distribution (47.3) — the value below which 97.5% of the bootstrap estimates are found.

As Figure 7.5 shows, choosing the lower and upper bounds as we have done means that 95% of bootstrap estimates fall within the confidence interval, and we say that there is a 95% probability that the confidence interval [46.8, 47.3] covers the parameter we are trying to estimate using a bootstrap method. If we're trying

to estimate the **recall of a classifier**, for example, we could say that there's a 95% probability that the interval [46.8, 47.3] covers (contains) the **true recall** of our classifier. In the following section, we will interpret the results of the confidence interval to estimate the estimation method.

Interpretation of the results

Based on the results provided by the confidence interval script code (Annexes C) let's interpret each part:

(i) The precision_std and precision-std-error values of 0.0 indicate that the precision values are not varying much across the 1000 samples. The Bootstrap Precision Confidence Interval [1.000, 1.000] suggests that the precision is consistently perfect (100%) across all the samples.

(ii) The recall_std value of 0.005849 suggests a small amount of variation in recall values across the 1000 samples. The recallstd_error of 0.000185 is the standard error as sociated with the recall values. The Bootstrap Recall Confidence Interval [0.941, 0.964] suggests that the recall values range between 94.1% **and** 96.4%**, with a** 95% **confidence level**.

(iii) The true_precison_std and true_precision_stderror values of 0.0 indicate that the true precision values are not varying much across the 1000 samples. The Bootstrap True Precision Confidence Interval [1.000, 1.000] suggests that the true precision is consistently perfect (100%) across all the samples.

(iv) The true_recall_std value of 0.01389 indicates some variation in true recall values across the 1000 samples. The true_recall_stderror of 0.00694 is the standard error associated with the true recall values. The Bootstrap True Recall Confidence Interval [0.938, 0.982] suggests that the true recall values range between 92.8% and 98.2%, with a 95% confidence level.

In summary, the bootstrap precision is consistently perfect (100%) across all the 1000 samples, while the bootstrap recall value shows some variation but generally fall within high ranges 94.1% **to** 96.4% **with a** 95% **confidence level**.

In this evaluation, We suppose construct a 95% confidence interval for the recall values and obtain the interval [0.941, 0.964]. **So, there is a 95% probability that the interval [0.941, 0.964] covers (contains) the true recall**. When examining frequentist confidence intervals (the most common type of confidence interval), the true parameter (we focus on recall in our case, as the precision is 100%) is considered a fixed number, not a random variable, and these values are either inside or outside the confidence interval. They have no associated probability. The 95% probability refers to the method used to construct the interval. The method guarantees that the interval will cover the true parameter 95% of the time (on average) (see the code in Annexes C bootstrap estimate confidence interval). Consider an **example** of the following situation: **There is a 95% probability that the interval [0.941, 0.964] covers (contains) the true Recall, and there is a 95% probability that this true Recall is in this interval**.

In our specific case, we calculate recall 1,000 **times** (each time on the basis of a different sample test set) and construct 1,000 **confidence intervals** at a 95% confidence level, while the **true recall** of 0.960 falls within the confidence interval of [0.941, 0.964]. This means that if the same experiment were conducted multiple times with different sample test sets and confidence intervals calculated each time, approximately 95% of those intervals would **contain the true recall value**, which is 0.960 in this case. **This result indicates that we have a high degree of confidence (95% confidence level) that the true recall value lies within the interval [0.941, 0.964] based on the 1000 calculated confidence intervals**. The fact that the true recall value of **0.960** falls within this interval suggests that **our estimation method is performing well and providing accurate estimates of the recall**. For more details on the datasets and results, see the following website [25].

7.5 Demonstration of Merging Annotated datasets

In this section, we aim to demonstrate the potential of this phase. Figure 7.6 and Figure 7.7 illustrate an example of merging annotated datasets using the two datasets from the Namur data portal (*Namur-limites-de-46-quartires*[26] (File1) and *Limites-des-localities*[27] (File2)). In order to merge these datasets and complete the missing information in these datasets, for example, the *zip code of each commune* and *the number of inhabitants*, this information can help data analysts calculate the number of residents in each commune with the zip code in order **to plan the future population** and **make the final decision**. Once the data analyst has finished annotating the data sets, He or she starts loading more than two annotated datasets in tabular or JSON formats. Then, as shown in figure 7.6 of **SinatraApi**, he must load two or more annotated datasets, indicate the file type, and then click on "merge". The system **selects the key mapping**, **detects the master dataset**, and **starts joining the datasets**. A few seconds later, the system generates the merged dataset and returns it as **a final file in JSON or Tabluar format**. The user can then load this final file to continue working. Figure 7.6 shows the final set of annotated data merged in JSON format, where we can see that all the information exists in a single file, making it possible to estimate the number of inhabitants according to their needs without having to spend a lot of time looking for information.

Similarly, as shown in figure 7.7 from **SinatraApp**, the system provides a user interface for the annotated data merge phase, in which the end-user loads two annotated datasets in CSV format, indicates the file type separator, may also indicates the name (Url) of the sub-column kep mapping, and then clicks on the "merge" button. Within a few minutes, the system displays the final merge annotated data in tabular form, and the user can then clicks on the "Download" button to save the final file on his local computer. On the basis of the results obtained, we can therefore demonstrate that the data fusion model is capable of bringing together numerous datasets.

[25] https://github.com/123rabida123/Datasets-Analysis
[26] https://urlz.fr/okXB
[27] https://urlz.fr/okXx

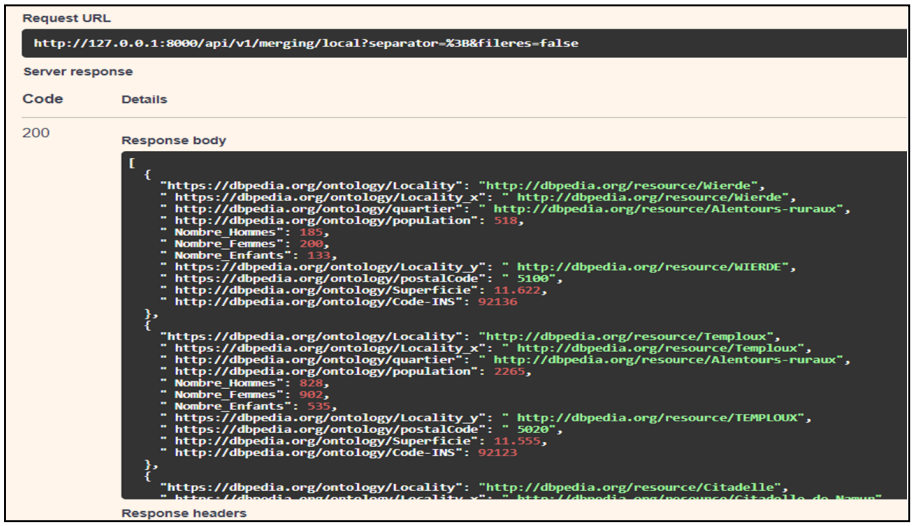


Figure 7.6: Screenshot of the annotated data fusion model (MergingAPI).

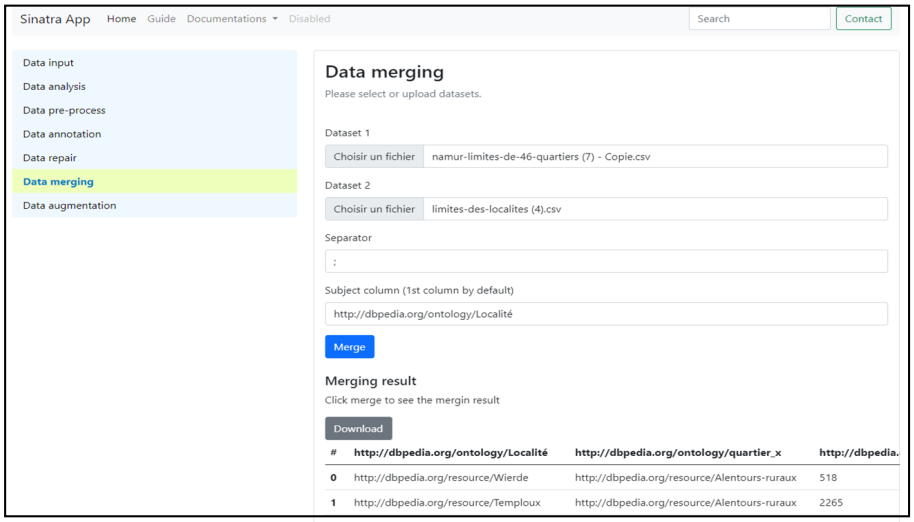


Figure 7.7: Screenshot of the annotated data fusion model (MergingUI).

In the following section, we describe **the efficiency and usability of SinatraAPP** through user tests involving people with different competencies in this field.

7.6 Efficiency and Usability of the SinatraAPP

In this section, a task-oriented user evaluation shows that **the SINATRA process** was understandable, usable, and capable, with little effort and time, of (1) loading datasets from several data portals, (2) analyzing datasets and discovering errors in

Table 7.8: Types of participants for assessing the SINATRA process

Non-Expert Users	Expert Users
No experience of :	Aware of :
- Semantics	- Semantics
- Linked Open Data	- Related topics (LOD, Ontology, SQL, SPARQL)
- knowledge graphs	- Data Analysis
- Data integration	- Data enrichment
- Data analysis	- Data aggregation

datasets, (3) cleaning and unifying this data, (4) enriching and annotating open datasets, (5) repairing empty cells with relevant information, (6) adding a column according to the user's needs, and (7) attempting to merge the associated datasets. The system's ease of use and effectiveness were tested using a standard evaluation process consisting of guided interviews, followed by a user experience questionnaire completed by the participants.

In a usability testing session, the evaluator asked a participant to perform a series of phases involving SINATRA's functionalities. As the participant attempted to complete the tasks described, the evaluator observed the participant's behavior, recording the time taken to complete a task and listening to comments.

The test experiment was conducted to:

- Identify problems in the conception of product features and components.
- Expose and discover potential improvements to the current prototype.
- Learn how users typically interact with the application, thus enabling analysts and developers to adapt future changes so that user behaviour is taken into consideration.
- Make data consumers (users) aware of Open Data, and data providers more aware of the existence of linked Open Government Data.

We engaged two types of participants to conduct the evaluation process and represent a sample of the intended population (Table 7.8).

A total of 20 participants were involved in the assessment, including 10 non-expert users and 10 more experienced users. Participants were classified as expert or non-expert users based on what they declared in the questionnaire concerning their knowledge of semantics, knowledge graphs, and data reconciliation or integration. The experiment was run by deploying a remote version of the SINATRA application to ensure that any requests to external services are executed in the same time frame, so that any external variables can be excluded from the network delay between the user's client and the back-end server.

7.6.1 The User Experience Questionnaire (UEQ)

To evaluate user experience, we have adopted a standard and reliable user experience questionnaire called UEQ[28] (See Annexes C). The principal objective of the UEQ is to measure user experience quickly and directly. The questionnaire has been designed

[28]https://www.ueq-online.org/

to be used as part of a normal usability test but also as an online survey [166]. This analysis resulted in a survey of 26 items grouped into six scales. Each item is marked by two terms with opposite meanings and an evaluation scale between them. The order of the terms is purposely randomized among the items so that inconsistencies and suspicious data answered by users are easier to detect.

Examples:

```
Not understandable o o o o o o o Understandable
Efficient o o o o o o o Inefficient
```

Each item can be evaluated on a 7-point Likert scale. Responses to an item thus range from -3 (strongly agree with the negative term) to +3 (strongly agree with the positive term), with 0 being a neutral response. The consistency of the UEQ measures and their validity (i.e., whether the scales actually measure what they are intended to measure) were tested in 11 usability tests with a total of 144 participants and in an online survey with 722 participants [166].

The UEQ consists of 6 scales and 26 items:

- **Attractiveness**: Overall impression of the product. Do users like it or not? Is it attractive, pleasant or enjoyable?
  6 items: annoying / enjoyable, good / bad, unlikable / pleasing, unpleasant / pleasant, attractive / unattractive, friendly / unfriendly.
- **Perspicacity**: How easy is it to become familiar with the product? Is it easy to learn? Is the product easy to understand and clear?
  4 items: not understandable / understandable, easy to learn / difficult to learn, complicated / easy, clear / confusing.
- **Efficiency**: Can users accomplish their tasks without unnecessary effort? Is interaction fast and efficient? Does the product react quickly to user input?
  4 items: fast / slow, inefficient / efficient, impractical / practical, organized / cluttered.
- **Reliability**: Does the user feel in control of the interaction? Can they predict the system's behavior? Does the user feel safe with the product?
  4 items: unpredictable / predictable, obstructive / supportive, secure / not secure, meets expectations / does not meet expectations.
- **Stimulation**: Is the product an exciting and motivating experience to use? Is it fun to use?
  4 items: valuable / inferior, boring / exciting, not interesting / interesting, motivating / demotivating.
- **Novelty**: Is the product innovative and creative? Does it capture the attention?
  4 items: creative / dull, inventive / conventional, usual / leadingedge, conservative / innovative.

Attractiveness is a pure valence dimension. *Perspicacity*, *efficiency* and *reliability* are Pragmatic (goal-oriented) Qualitative Aspects, while *Stimulation* and *Novelty* are Hedonic (non-goal-oriented) Qualitative Aspects.

The attractiveness scale has 6 items, while all the other scales have 4. Figure 7.8 illustrates the assumed structure of the UEQ scale and the English items per scale.

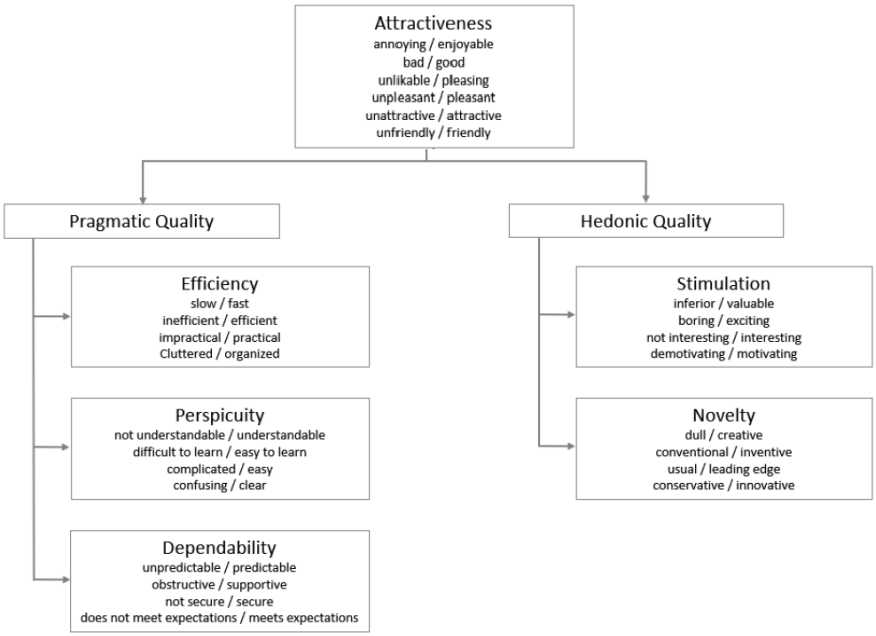


Figure 7.8: Assumed scale structure of the UEQ [166].

7.6.2 Results of Sinatra Web Application (SinatraApp)

The results were collected from **20** participants, mostly data scientists, application developers, researchers, and a few professors. The number of participants was chosen according to UEQ specifications so that the scale means of the survey responses could be interpreted with some precision.

Table 7.10 illustrates the approximate *sample size* estimate based on the **UEQ benchmark** and the **standard deviation** of the samples collected (Table 7.9). The results of this experiment are highlighted in yellow. Each item in the table indicates an estimate of the sample size, such that the specific scale can be measured with a **specific precision and probability of error, 0.5 and 0.05** in our case.

Table 7.9: Results for standard deviation of scale value.

Scale	**Standard Deviation**
Attractiveness	1,21
Perspicuity	0,93
Efficiency	0,74
Dependability	0,67
Stimulation	0,96
Novelty	0,84

Table 7.10: Results of assessment tests.

Condition	Attractiveness	Perspicuity	Efficiency	Dependability	Stimulation	Novelty
Precision=0.5, Err.Prob.=0.1	16	10	6	5	10	8
Precision=0.5, Err.Prob.=0.05	23	13	8	7	14	11
Precision=0.5, Err.Prob.=0.01	39	23	14	12	25	19
Precision=0.25, Err.Prob.=0.1	64	38	24	20	40	31
Precision=0.25, Err.Prob.=0.05	90	54	33	28	57	43
Precision=0.25, Err.Prob.=0.01	156	93	58	48	98	75
Precision=0.1, Err.Prob.=0.1	400	238	147	123	251	193
Precision=0.1, Err.Prob.=0.05	564	335	208	174	355	272
Precision=0.1, Err.Prob.=0.01	977	581	360	302	614	471

Efficiency and Usability

The UEQ survey measures **the usability and user experience of a framework using six different scales**, as described above. For each questionnaire item and scale, averages have been calculated with **95%** confidence intervals.

Table 7.11 presents the results for each element of the questionnaire. In compliance with UEQ specifications, items with a mean value between -0.8 and 0.8 represent a more or less neutral evaluation in relation to the relevant scale; values **higher than 0.8 indicate a positive evaluation**, and **values lower than -0.8 represent a negative evaluation**. The scale ranges from -3 (**very bad**) to +3 (**excellent**). The results demonstrate that **23/26** items received a positive evaluation. The only items with a neutral evaluation are item 3 corresponding to *creative / dull* belonging to the *Novelty* scale, item 8 corresponding to *unpredictable / predictable* belonging to the *Dependability* scale, and item 26 corresponding to *conservative / innovative* belonging to the *Novelty* scale. For more details on the **SinatraApp** evaluation, you can consult Github [29] for full information.

These items can then be grouped into the six user experience scales already mentioned. Table 7.12 displays the means, standard deviations and confidence intervals for each scale.

In addition, the UEQ is complemented by a benchmark based on data from 21175 people participating in 468 studies of different digital products, including business software, web pages, online stores, and social networks. Figure 7.9 on the benchmark chart shows means by scale. The benchmark classifies a product into five categories and can be interpreted as follows:

- **Excellent**: In the range of the highest 10% of results.
- **Good**: 10% of results in the benchmark sample set are better, and 75% of results are worse.
- **Above average**: 25% of the benchmark results are better than the results of the product being evaluated, 50% of the results are worse.
- **Below average**: 50% of the benchmark results are better than the results of the product being evaluated, 25% of the results are worse.
- **Bad**: Within the range of the 25% worst results.

In Figure 7.9 the measured scale means are set in relation to the existing values from the benchmark. The results demonstrate that **SinatraApp** is positioned in the

[29] https://github.com/123rabida123/SinatraAPP/tree/master

Table 7.11: Mean for each item on the survey. Green indicates a positive rating, and yellow a neutral rating.

Confidence interval (p=0.05) per item							
Item	**Mean**	**Variance**	**Std. Dev.**	**Confidence**	**Confidence interval**		**Scale**
1	1,800	1,9	1,361	0,597	1,203	2,397	Attractiveness
2	2,050	1,3	1,146	0,502	1,548	2,552	Perspicuity
3	0,650	2,6	1,599	0,701	-0,051	1,351	Novelty
4	1,300	3,9	1,976	0,866	0,434	2,166	Perspicuity
5	1,650	1,5	1,226	0,537	1,113	2,187	Stimulation
6	1,100	1,3	1,119	0,491	0,609	1,591	Stimulation
7	1,300	1,4	1,174	0,515	0,785	1,815	Stimulation
8	0,700	1,9	1,380	0,605	0,095	1,305	Dependability
9	1,400	1,9	1,392	0,610	0,790	2,010	Efficiency
10	0,800	1,3	1,152	0,505	0,295	1,305	Novelty
11	1,250	1,6	1,251	0,548	0,702	1,798	Dependability
12	1,850	2,0	1,424	0,624	1,226	2,474	Attractiveness
13	1,800	1,1	1,056	0,463	1,337	2,263	Perspicuity
14	1,500	2,5	1,573	0,689	0,811	2,189	Attractiveness
15	1,000	2,3	1,522	0,667	0,333	1,667	Novelty
16	1,600	2,3	1,501	0,658	0,942	2,258	Attractiveness
17	2,100	0,9	0,968	0,424	1,676	2,524	Dependability
18	0,800	1,7	1,322	0,579	0,221	1,379	Stimulation
19	2,050	0,7	0,826	0,362	1,688	2,412	Dependability
20	2,150	0,9	0,933	0,409	1,741	2,559	Efficiency
21	2,050	0,6	0,759	0,333	1,717	2,383	Perspicuity
22	2,200	0,6	0,768	0,336	1,864	2,536	Efficiency
23	1,900	1,6	1,252	0,549	1,351	2,449	Efficiency
24	1,450	2,1	1,432	0,627	0,823	2,077	Attractiveness
25	1,900	1,1	1,071	0,469	1,431	2,369	Attractiveness
26	0,500	3,5	1,878	0,823	-0,323	1,323	Novelty

Table 7.12: Mean for each scale in the survey. Green means positive evaluation, and yellow neutral.

Confidence intervals (p=0.05) per scale					
Scale	**Mean**	**Std. Dev.**	**Confidence**	**Confidence interval**	
Attractiveness	1,683	1,211	0,531	1,152	2,214
Perspicuity	1,800	0,934	0,409	1,391	2,209
Efficiency	1,913	0,736	0,322	1,590	2,235
Dependability	1,525	0,673	0,295	1,230	1,820
Stimulation	1,213	0,961	0,421	0,791	1,634
Novelty	0,738	0,841	0,369	0,369	1,106

excellent range for *Efficiency*, in the **good** range for *Attractiveness*, *Perspicuity*, and *Dependability*, and above average for *Stimulation* and *Novelty*. Table 7.13 details and interprets the results of SinatraUI's comparison with the benchmark.

The UEQ scales can be grouped into two categories: pragmatic quality (perspicacity, efficiency, and dependability) and hedonic quality (stimulation, novelty). *Pragmatic quality* describes task-related aspects of quality, while *hedonic quality*

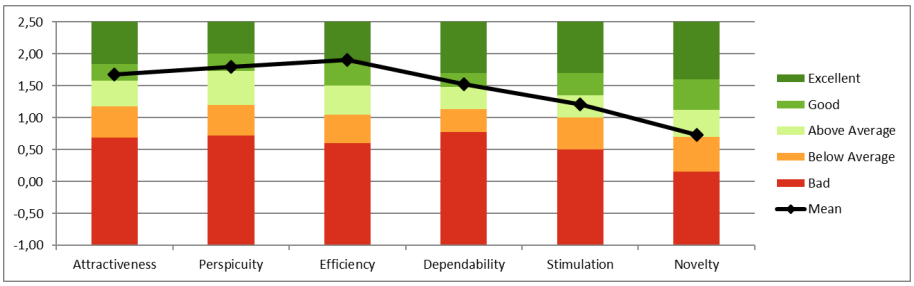


Figure 7.9: Comparison of SinatraUI UEQ results in relation to the UEQ benchmark.

Table 7.13: Results of comparison with benchmark.

Scale	Mean	Comparisson to benchmark	Interpretation
Attractiveness	1,68	Good	10% of results better, 75% of results worse
Perspicuity	1,80	Good	10% of results better, 75% of results worse
Efficiency	1,91	Excellent	In the range of the 10% best results
Dependability	1,53	Good	10% of results better, 75% of results worse
Stimulation	1,21	Above Average	25% of results better, 50% of results worse
Novelty	0,74	Above Average	25% of results better, 50% of results worse

describes non-task-related aspects of quality. The average of the three aspects of pragmatic and hedonic quality is calculated and presented below in Figure 7.10 and Table 7.14.

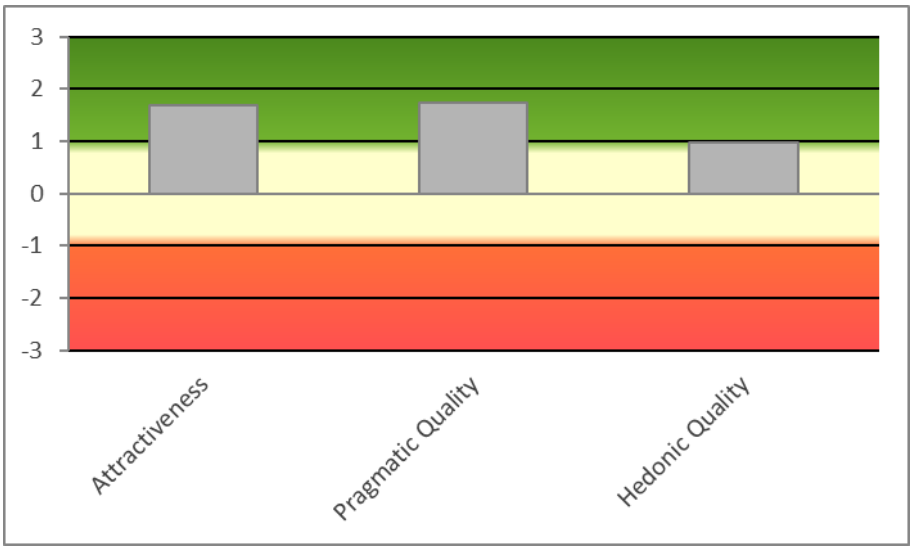


Figure 7.10: Pragmatic and Hedonic Quality.

Table 7.14: Average of the three aspects: Pragmatic and Hedonic quality.

Pragmatic and Hedonic Quality	
Attractiveness	1,68
Pragmatic Quality	1,75
Hedonic Quality	0,98

7.7 KPIs to assess SinatraApp's performance

In this section, we use **KPIs** (Key Performance Indicators) to assess SINATRA' s performance in achieving **our specific objectives**. Our aim is to present a systematic approach to the selection of **key performance indicators** to demonstrate the effectiveness of our system. Our objectives are to increase **the accessibility** and **discoverability of open data**, **the quality of the data/metadata**, **the ease of use of this application** and **the ability to easily interpret it with any OD portal**.

Since we have received **positive feedback from users** in terms of **efficiency**, **attractiveness**, **insight**, **reliability**, **stimulation** and **novelty**, we consider the **KPIs that relate to these qualities**. For each scale, we show **the distribution of answers to the various items**. This gives us deeper insights into which aspects of **SinatraApp** are perceived as very positive by participants (See Annexes C.2.2).

- **Attractiveness**: To gauge attractiveness, KPIs involve user satisfaction ratings, the number of positive design and user experience reviews (Table 7.15).

Table 7.15: Distribution of answers to the various items.

Attractiveness	
Good user experience	**Bad user experience**
Enjoyed 18	Boring 2
Good 17	Bad 3
Pleasing 15	Unlikable 5
Pleasant 5	Unpleasant 6
Fast 14	Slow 6
Friendly 18	Unfriendly 2

- **Insight**: Is measured by KPIs such as the number of usable information derived, and the percentage of successful problem solving (Table 7.16).
- **Efficiency**: Key performance indicators include the measurement of functionality usage to ensure optimum efficiency (Table 7.17).
- **Dependability**: KPIs for reliability can include whether this system meets their requirements, supports them, and can be a reliable application for them

Table 7.16: Distribution of answers to the various items.

Perspicuity	
Good user experience	**Bad user experience**
Understandable 18	Not understandable 2
Easy to learn 13	Not easy to learn 7
Easy 18	Complicated 2
Clear 50	Confusing 0

Table 7.17: Distribution of answers to the various items.

Efficiency	
Good user experience	**Bad user experience**
Fast 18	Slow 2
Efficient 20	Inefficient 0
Practical 20	Impractical 0
Organized 19	Cluttered 1

(Table 7.18).

Table 7.18: Distribution of answers to the various items.

Dependability	
Good user experience	**Bad user experience**
Predictable 16	Unpredictable 4
Supportive 19	Obstructive 1
Secure 19	Not secure 1
Meets expectations 17	Does not meet expectations 3

- **Stimulation**: Performance indicators linked to user interaction frequency and interest in the application(Table 7.19).
- **Novelty**: KPIs reflecting novelty could involve tracking the percentage of users adopting newly introduced elements (Table 7.20).

Table 7.21 represents the importance scores for each scale. In addition, Figure 7.11 demonstrates the importance of *UEQ scales* for participants.

Table 7.19: Distribution of answers to the various items.

Stimulation	
Good user experience	**Bad user experience**
Valuable 16	Inferior 4
Exciting 19	Boring 1
Interesting 19	Not interesting 1
Motivating 17	Demotivating 3

Table 7.20: Distribution of answers to the various items.

Novelty	
Good user experience	**Bad user experience**
Creative 17	Dull 3
Inventive 18	Conventional 2
Leading edge 18	Usual 2
Innovative 16	Conservative 4

Table 7.21: Importance Ratings: Mean and Standard Deviation.

Scale	Mean	Std.Dev.	N	Confidence	Confidence Interval	
Attractiveness	5,64	1,21	255	0,15	5,49	5,79
Perspicuity	6,15	1,02	255	0,13	6,02	6,27
Efficiency	6,44	0,79	255	0,10	6,34	6,54
Dependability	6,22	0,94	255	0,12	6,11	6,34
Stimulation	5,09	1,43	255	0,17	4,91	5,26
Novelty	5,32	1,24	255	0,15	5,17	5,47

7.8 Limitation of SinatraApp

The **completeness** problem arises when data repair or augmentation functions are unable to fill missing information required by the user. Hence, the completeness of Knowledge Graphs (KGs) is a significant issue in building comprehensive Open Datasets. KGs are inherently limited by their ability to collect and update data from diverse sources, which can result in missing or outdated information. Additionally, the ever-expanding nature of human knowledge makes it challenging to keep KGs up-to-date with the latest developments. Furthermore, some information, particularly niche or proprietary data, may not be publicly available and therefore cannot be included in a KG. These limitations highlight the constant need for curation, data

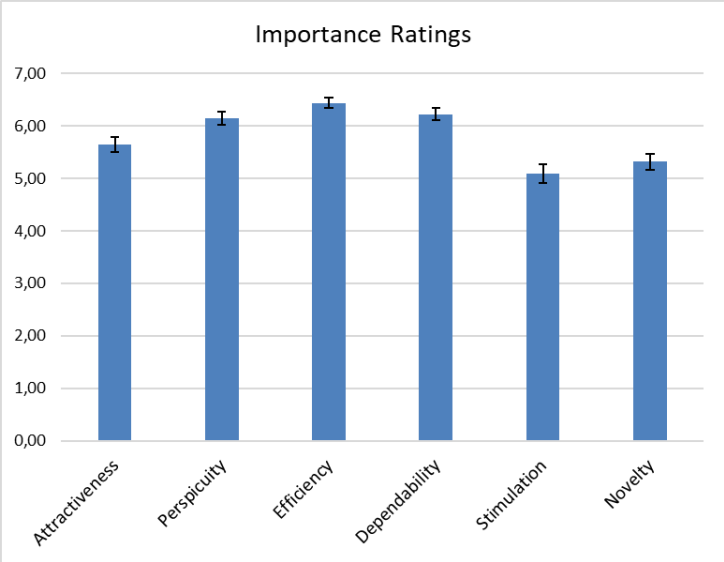


Figure 7.11: Importance Ratings.

sources, and maintenance to strive for completeness in KGs, while recognizing that perfect completeness may remain an elusive goal.

The **scalability** challenge is a critical consideration when implementing these functions, including analysis, annotation, repair, augmentation, and fusion, on large datasets. When applied to substantial volumes of data, computational and resource constraints can become major obstacles. To address this issue, it is essential to design adaptable frameworks that can efficiently process data in smaller batches and maintain the capability to re-apply the same phases in the future when dealing with similar data structures. By doing so, we not only improve the overall performance of these data processing tasks but also provide SinatraApp with the capacity to manage and extract valuable insights from vast and constantly evolving open data sources. Scalability is pivotal for accommodating the dynamic and ever-growing nature of data, enabling us to harness the full potential of these datasets for various analytical and knowledge discovery purposes.

The **specific domain to data-driven** challenge is presented when the loading process needs to focus on **the efficient collection and ingestion of data** from a variety of sources, including databases, APIs, and external datasets. It can also involve real-time data flow to keep information up-to-date.

7.9 Conclusion

In this chapter, we have presented an assessment of the results of our contributions. We illustrated the application of the **Pub_LOGD framework** in a particular Belgian government catalog to assess the feasibility and ease of use of our solution by inviting users to complete two **TAM-based surveys**. The results showed that Pub_LOGD and

Pub_LOGD Extension were **easy to use**, **useful** and **met the expectations** of data analysts. We highlighted some of the limitations of these contributions, which could be addressed in future work.

According to the evaluation of results in Chapter 5, the SINATRA approach **was able** to solve data quality problems. It obtained **good results** in terms of precision and recall when **correctly adding missing data and new columns**. In addition, ODSAG was able to **automate** the process and be **interoperable** with Open Data portals.

In addition, we have estimated the performance of the data analysis model, finding that bootstrap precision is always perfect (**100%**) for all **1,000 samples**, while the bootstrap recall value shows some variation but is generally in the high range of **94.1 to 96.4% with a 95% confidence level**. We also demonstrated the potential of **the annotated dataset fusion phase** using the two datasets from the Namur data portal, where the results showed that **this contribution is capable of correctly joining** more than two annotated tabular data or Json formats.

Furthermore, the results showed that **SinatraAPP** was in the **excellent** range for effectiveness, in the **good** range for attractiveness, insight, and reliability, and **above average** for stimulation and novelty. We proved its performance in this evaluation by selecting key performance indicators (**KPIs**) from user feedback and taking into account **the indicators** that relate to **these qualities**. We have outlined some of the limitations of SinatraAPP, which could be the subject of further research in the future.

Part V

Postface

CHAPTER 8

CONCLUSION AND FUTURE DIRECTIONS

"The end ... is just the beginning."

*— **Geopolitical analyst, Author: Peter Zeihan***

8.1 Summary of the Contributions

This thesis aims to tackle some of **the barriers to Open Data (OD)** faced by most **data analysts and publishers**, as well as provide **solutions to improve practical approaches by realizing an automated and all-in-one approach** that can be embedded in any service and Open Data portal. These barriers were selected on the basis of a literature review, identifying the most important issues in OD, namely **data heterogeneity**; **data quality**, **ambiguity**, and **completeness**; **data misuse** or **misinterpretation**; and **data discoverability**, and **comprehensibility**. In this context, each barrier and issue is related to research questions and challenges for implementation.

RQ1.How can we supply a standardized solution to ease the use and publication of LOGD?

Our first research question, highlighted in Chapter 3, sought to explore the most key barriers to Open Data adoption and the shortcomings of Linked Open Data (LOD) approaches. In Chapter 3, we have started to use the systematic mapping method **to identify work supporting and publishing LOD**. Using the research

methodology, we looked to find the best methodologies and approaches to **LOD publication** that can be recommended to cover all phases of data publication and to engage the user community with government, as well as the type of quality process implicated in practices to deal with these obstacles. Despite intensive efforts by engineers to obtain high-quality data, existing general practice guidelines may not be sufficient to make processes consistently replicable. Furthermore, collecting, analyzing, exploiting, enriching, and integrating data from different sources is an essentially manual activity, requiring the skills of users (**data analysts and data publishers**), and is costly in terms of effort, time, and money. On the whole, this necessitates a combination of standardization efforts, data integration techniques, powerful data processing and analysis software. A great deal of **limited work exists to support the process of publishing LOD in the context of e-government and to increase transparency and foster innovation**. Clearly, integrated, automatically-supported solutions are **lacking that combine appropriate technologies and tools to help non-expert users**, in particular to manipulate datasets efficiently, **from extraction OD to publication LOD**, **track progress in this process and visualize data semantically**, and **involve users with governments to reuse data** to deliver public services and applications.

Despite these shortcomings, **limited work exists in the R&D community to support the end-to-end process of LOGD production and publishing**. Chapter 4 presents two works designed to put the previous research question into practice. The **first objective** is to propose a **unified process covering all phases of the LOGD life-cycle** that is iterative, reusable, and interactive with domain experts and application developers. It also ensures data distribution, support, and consolidation of process steps in a **coherent manner**. This develops a pipeline-based framework to **publish OGD**. This **framework Pub_LOGD** aims at shifting the burden of converting and publishing LOGD to data consumers rather than the government. It consists of an end-to-end solution that covers five main data analysis steps. (1) data transformation; (2) data interlinking; (3) data storage; (4) data visualization and (5) data publishing. This approach is compared with other existing approaches in the literature, using a number of qualitative measures.

In Section 7.2, an evaluation of our proposals is done through a **case study of Belgium's statistical catalog**, based on a local population dataset used by the community's users. The **second objective** is **to increase the transparency of governance data** by extending and simplifying **the data modeling step** to make it even more user-friendly for the community. This step is based on the set of requirements that are feedback on participation, which they completed in the pre-survey. The main goals of this work are to transform the raw data into a high-quality government dataset published on the web. In parallel, it aims to provide a mechanism to promote user engagement with governments, reuse of government information and access to a publishing solution. In Chapter 7.2, the evaluation and demonstration of the usability and ease of use of the proposals is done by engaging users to complete **Pub_LOGD** and **Pub_LOGD Extension** post-surveys based on *TAM* questionnaire.

Finally, the results show that **these contributions are highly relevant and useful for the participants' operations**. Although we have good results, the methodology

of this work is not yet sufficient, so it requires **automated phases** that are easy for non-expert users and experimentation to further evaluate the effectiveness of our proposals, and to measure data quality and completeness.

RQ2. How can we design an approach and tool that supports semantic interpretation of Open Data OD?

Chapter 5 contributes **all-in-one and automatic approach** to dealing with data quality issues and supporting annotation systems that can be easily integrated into any OD portal. It therefore intends to remedy the shortcomings of the contributions in the fourth chapter and to address **OD quality issues** such as misspellings, incomplete meta-data, ambiguity data, misinterpretation, and a lack of relevant value and meaningful information in existing datasets. In addition, this chapter aims to improve **the usability**, **discoverability**, and **understanding** of OD to ensure that data can be accessed and interpreted accurately and efficiently.

To conduct the research, Chapter 3 presents and describes a methodology to select and identify the work related to the different tasks and methods that can answer those research questions. Moreover, Chapter 5 identifies a list of requirements across selected existing tools that should be included in a tool-based approach to automatically and correctly annotate open datasets, deal with data quality issues and meet user requests.

Based on the comparison provided in Chapter 5 with the proposed approaches and tools, the majority of **existing works are unable to support users in overcoming data quality issues**, and **none are capable of interoperating with Open Data portals**, automatically annotating data, or generating graphs from annotated data. To summarize, there is still (i) no all-in-one approach that can handle these problems, (ii) no other features that can further support Semantic Annotation task, (iii) a lack of semantic annotation tools that integrate well with OD portals.

To address the gaps, we have implemented the requirements in two contributions, called **SINATRA** and **ODSAG** tool. The SINATRA approach (**SemantIc aNnotation AugmentaTion and RepAir**) is intended to **solve data quality difficulties and improve the usability of tabular data**. It has extended and integrated the best methods in the field of Semantic Annotation (SA) with its five features: **(1) data pre-processing** and **(2) subject column (sub_col) detection**, **(3) data annotation**, **(4) data repair**, and **(5) data augmentation**. This approach is based on associating Tabular Data with the **DBpedia knowledge graph**, unlike the ODSAG tool, which is based on mapping a dataset to the **Wikidata knowledge graph**. We have measured the performance and effectiveness of the SINATRA approach using the *SemTab challenge benchmark* and evaluation metrics: *precision*, *recall* and *F-measure*.

According to the results of the evaluation, the SINATRA approach **has solved certain problems associated with data quality and the lack of metadata**, and has made it **perfectly suitable for adding missing values** and **improving the usability** of Tabular Data by adding meaningful columns to the existing structured data. Assessment results in Section 5.5 show that it is effective for each of its phases. To meet and implement all customer requirements, we have **improved this approach**

by implementing the **ODSAG tool** (**Open Data Semantic Annotation and Graph**), which is a **user-friendly chrome extension** that can be **easily interoperable with any OGD portal** and **automatically generate interactive graphs from the annotated data**. **ODSAG** consists of four phases: **(1) data selection**, **(2) data pre-processing**, **(3) data Annotation**, and **(4) graph generation**. This tool provides **two graphs** to help users understand metadata, and makes it easier for data analysts to interpret unannotated datasets before using them. We have used a real *COVID-19* dataset available on the Namur, Belgium portal to show **the capability of ODSAG features**. It is capable of integrating with any OD portal, and generating a graph for metadata and a graph for dataset content to help users understand and navigate the big data landscape more effectively. As a result, these contributions: SINATRA approach and ODSAG tool can serve as a starting point for application developers to create or enhance their own services.

However, these contributions**still require further improvement in order to be better adapted to the data portal** and to be more effective and applicable to people with different skills and experiences. **The quality of the data annotation result is greatly influenced by the quality of the input data**, necessitating **a pre-analysis step** to examine the data prior to pre-processing. It may be the first step in an effective interpretation system. Furthermore, the data analysts spend most of their time reviewing and handling data. There are **no tools to help them automatically analyze open government data**. Open Data search capabilities must be integrated so that data analysts and data publishers **can easily discover links and combine tables, and datasets** that are useful to their task. It should **deliver API services that can be easily added to form an ecosystem of services** available to application developers.

RQ.3: How to build a tool to help data analysts examine and analyze public data to reduce time spent quickly pointing out signs of open data issues?

Chapter 6 describes a framework called *SinatraAPP* (**Sinatra Web Application**), which is based on the implementation of user and system requirements. This approach is founded on the **SINATRA** approach (Chapter 5), which we have sought to extend **by enriching it with analysis and merging functions and improving its performance and convenience of use**. To develop this contribution, we first established **the requirements of the user and the system**. User requirements are frameworks or specifications that developers must apply to allow users to accomplish their tasks. System requirements are the constraints imposed on the system. They cover questions such as usability and maintainability.

We have selected **25 public datasets from different portals and other public data sources** to identify recurring data quality issues. We have identified **six data fragrances**, which have been discovered: duplicate examples, unknown unit of measure, null-type missing values, special missing values, strings with special characters, and binary missing values. In Chapter 7, we applied a bootstrap method to estimate a classifier's recall, precision, construct corresponding confidence intervals, and standard error to estimate its performance. We intended our **estimate to be as close as possible to the "true" precision and recall**. We therefore discovered the

bootstrap estimates for the sample averages (the average number of precision and recall (data faults) in our datasets) and they are exactly the averages of the actual precision (**1.00**) and recall (**0.952**) estimates. Our result indicated that we have a high degree of confidence (**95% confidence level**) that the true recall value lies within the interval **[0.941, 0.964]** based on the **1,000 calculated confidence intervals**. The fact that the **true recall value of 0.960** falls within this interval suggests that **our estimation method is performing well and providing accurate estimates of the recall**.

RQ.4. How can we implement a simpler technique and application to help users discover and combine related annotated datasets to get an overview of a particular topic?

Open Data does **not generally allow the user to find the most relevant data, understand the relationships between data products and join heterogeneous data sources or products**. Furthermore, if users don't know what's available, if they **can't find comprehensive data for their needs**, and if they **can't trust what they do find**, they end up reinventing their own schemes or business processes. To achieve the overall goal, this feature allows users who collect and combine similar data in multiple contexts to combine these datasets into a single dataset in order to obtain a comprehensive view of a particular topic or phenomenon. In Chapter 6, we sought **to help users find and merge semantically related datasets to facilitate ongoing data analysis tasks**. Section 6.4.1 presents in detail the steps of our fusion algorithm to join the related tables. The first step is to detect key mapping in order to select the main table for merging related annotated tables. The second step is based on the key mapping and the three tasks (CEA, CPA, and CTA) that created the annotated structured data. This algorithm is established under three conditions to merge tabular data. (1) The first condition is that they have common CTAs. (2) The second condition is that they have common CEAs and (3) the third condition is that there are no common CTAs or CEAs, which brings it back to the annotation phase.

In Section 7.5, we have demonstrated the ability of **merging data phase** by applying **a real case study available** available on the **Namur** portal (Belgium). Moreover, the results of the evaluation showed that it is perfectly capable of discovering related datasets and merging them.

To fulfill user and system requirements, we have implemented the **framework SinatraApp**, a **microservice-oriented architecture** consisting of two main elements: (i) a user interface to facilitate interaction with the user (SinatraUI) and (ii) a backend to manage the endpoints of each feature and external service (SinatraAPI). It contains **8** components: (1) loading, (2) analytics, (3) pre-processing, (4) annotation, (5) repair, (6) merging, (7) augmentation, (8) visualizing. Moreover, It is a decentralized approach, where all components are not centrally controlled and each component is focused on the realization of a well-defined functionality or service. This framework provided *API documentation* to users using the *Swagger editor* to help them use these *REST APIs* and their services. It enabled users to interact with their endpoints. **SintraUI** offered a compact visualization of services, allowing the user to view and navigate to each phase. SinatraApp has a User Interface that effec-

tively engages **non-expert users** by suggesting tools for analysis, interpretation with the data portal and other features **that can be used without specialized knowledge**.

In Chapter 7, we described the evaluation of the application and the results of experiments conducted with **20 participants** to confirm **SinatraApp's user-friendliness and usability**.

RQ5. What is the performance, usability, and usefulness of the developed solutions?

A task-oriented user evaluation that shows that the **SinatraApp** was understandable, usable, and capable with little effort and time of (1) loading datasets from several data portals, (2) analyzing datasets and discovering errors in public datasets, (3) cleaning and unifying this data, (4) enriching and annotating Open Data, (5)repairing empty cells with relevant information, (6) adding a column according to the user's needs, (7) attempting to merge the associated datasets, and (8) visualizing the metadata and content of the datasets.

The system's ease of use and effectiveness were tested using a standard evaluation process involving guided interviews, followed by a user experience questionnaire. The participants were classified as **expert** or **non-expert users** based on what they declared in the questionnaire concerning their knowledge of semantics, knowledge graphs, and data reconciliation or integration. To evaluate the user experience, we have adopted a standard and reliable questionnaire called **UEQ**, which aims to measure attractiveness, perspicacity, efficiency, reliability, stimulation, and novelty. We therefore employed the **Key Performance Indicator** (KPI) to demonstrate Sinatra's performance in achieving our objectives, where we have considered the **KPIs** associated with these qualities. Consequently, SinatraApp's results compared with the **UEQ benchmark** indicate that it is in the **excellent** range for *Efficiency*, in the **good** range for *Attractiveness*, *Perspicuity*, and *Dependability*, and **above average** for *Stimulation* and *Novelty*.

8.2 Future Directions

8.2.1 Extend the Sinatra framework approach by integrating other features

The **Sinatra approach** is currently supported by these features: loading, analyzing, pre-processing, interpreting, repairing, augmenting, merging, and visualizing. By integrating search engines such as *Elasticsearch* into this framework, can enhance **its capabilities**, and improve **accessibility** and **searchability**. For example, it's possible to index large volumes of data and perform fast searches on Open Data. This function allows users to explore data according to different attributes, narrowing down search results based on specific criteria, such as categories or data sources. Moreover, the integration of **geospatial elasticsearch capabilities** with **SinatraApp** can also improve the annotation of geographic information, which can be exploited to enable location-based searches and analysis. The combination of SinatraApp and

Elasticsearch's powerful search function **fosters innovation** and **creativity**, enabling researchers and companies to discover new ideas and develop innovative solutions based on available data.

8.2.2 Big Data Scalability and Improve the Performance

The challenge of scalability still remains when applying these functions—analysis, annotation, repair, augmentation, and fusion to large volumes of datasets and tables. While these functions may work efficiently on a reasonably sized dataset, they can encounter several obstacles when dealing with big data. The performances shown have been achieved on a single machine with a reasonably sized dataset. They may not be sufficient for processing huge datasets. There are several development possibilities that **can adopt distributed computing frameworks such as Apache Hadoop or Apache Spark**. These frameworks will improve performance and eventually enable SinatraApp to handle large volumes of Open Data.

8.2.3 Integration of other Knowledge Graphs (KGs)

The current version of the framework does not integrate other knowledge graphs, such as *LOV* (Linked Open Vocabulary) and *Geonames* to improve annotation and data augmentation. It will be necessary to further develop these tasks and extend the framework.

8.2.4 Iterative improvement

Iterative improvement in the context of the Sinatra framework means that this process of Open Data is not a one-time event but a continuous and cyclical process. It involves learning from previous analyses, making improvements, and refining approaches based on new insights and changing circumstances. This iterative nature enables governments to stay adaptive, responsive, and up-to-date in their governance practices, leading to more effective policies, solutions for the benefit of society, and impactful decision-making.

8.2.5 Increasing public engagement and participation

Through data interpretation, citizens can assess the real-world impact of government policies. This understanding enables constructive feedback, leading to policy improvements and more effective governance. However, citizens are more likely to engage with information that is visually appealing and easy to comprehend. When data is presented in interactive dashboards, people can explore the information at their own pace, deepening their understanding of different policy areas. This increased engagement can lead to more active participation in public discussions, town hall meetings, and other democratic processes. Visualization tools and data dashboards can help present complex information in a user-friendly manner, making it easier for citizens to grasp key findings. These tools can encourage citizen

engagement and collaboration. Citizens, as well as researchers and civil society organizations, can actively participate in exploring data, generating ideas, and proposing solutions to societal challenges.

8.2.6 Towards the Sustainable Development Goals

This framework fosters **openness**, **collaboration**, and **data-driven decision-making**, which are essential to advancing sustainable development efforts. By leveraging Open Data and applying analytical techniques to gain valuable insights, organizations and policymakers can make informed decisions, develop sustainable practices, and track progress towards the Sustainable Development Goals. In addition, public data promotes transparency and inclusion, enabling diverse stakeholders, including citizens, researchers, and non-governmental organizations, to actively participate in the sustainability discourse and contribute their expertise and insights. This collaborative approach can lead to more effective, more widely accepted, and more equitable sustainable development solutions.

Part VI

Appendix

APPENDIX A

ACRONYMS

ABox - Assertion Component
AG - Automatic generated dataset
AI - Artificial Intelligence
STA - Semantic Table Annotation
ATD - Annotated Tabular Data
STI - Semantic Table Interpretation
C - Columns
CIA - CIA World Factbook
CSV - Comma Separated Values
CPA - Column Property Annotation
CEA - Column Entity Annotation
CTA - Column Type Annotation
CKAN - Open Source Data Management System
DA - Data Annotation
DCAT -Data Catalog Vocabulary
Eurostat - European Commission Statistical
HTTP - Hypertext Transfer Protocol
HTML - HyperText Markup Language
INK - knowledge graph embeddings for node classification
JSON - JavaScript Object Notation
KG - knowledge Graph
KB - knowledge Base
KML - Keyhole Markup Language
KPIs - Key Performance Indicators
LamAPI - Learning Analytics Machine API
L-column - Literal column

LD - Linked Data
LOD - Linked Open Data
LOGD - Linked Open Government Data
LOSD - Linked Open Statistical Data
LOV - Linked Open Vocabularies
MOOC - Massive Open Online Course
ML - Machine Learning
N3 - Notation 3
NLP - Natural Language Processing
NE-column - Named-Entity column
OD - Open Data
OGD - Open Government Data
OSD - Open Statistical Data
ODSAG - (Open Data Semantic Annotation and Graph
ODSAG - Open Data Semantic Annotation Government
OECD - Organisation for Economic Co operation and Development
Pandas - Python Data Analysis Library
Pub-LOGD - Publishing Linked Open Data
Pub-LOGD Extension - Extension of Publishing Linked Open Data
PDF - Portable Document Format
QB - RDF Data Cube
R - Rows
RDF - Resource Description Framework
RML - RDF Mapping language
R&D - Research and Development
RDBMS - relational database management system
RDFLib - Python library for working with RDF
REGEX - Regular Expression
SLR - Systematic Literature Review
SemTab - Semantic Web Challenge on mapping tabular data to knowledge graphs
SINATRA - SemantIc aNnotation AugmentaTion and RepAir
SinatraAPP - Sinatra Application
SinatraAPI - Sinatra Application Programming Interface
SinatraUI - Sinatra User Interface
Sub_col - Subject Column (Main Column)
SW - Semantic Web
SKOS - Simple Knowledge Organization System
SPARQL - SPARQL Protocol and RDF Query Language
STATBel - Statistics Belgium
STIB -Portal data STIB of Brussels
SML - Sparqlification Mapping Language
SQL - Structured query language
SOA - Service Oriented Architecture
T - Table
TD - Tabular Data

TSV - Tab separated values
TBox - Terminology Component
2T - Tough Tables
URI - Uniform Resource Identifier
URL - Uniform Resource Locator
UEQ - User Experience Questionnaire
UNAMUR - University of NAMUR
W3C - World Wide Web Consortium
WYSIWYM - What You See Is What You Mean
XLS - Microsoft Excel Spreadsheet file

APPENDIX B

PUB_LOGD FRAMEWORK ASSESSMENTS

B.1 Pre-questionnaire for the development of an open data framework

The aim of the questionnaire is to survey analysts and data scientists who have used data portals, to gather their opinions in order to improve our contributions.

Q1.What is the application you have developed?

Q2. Did you retrieve data from which portals data or from where exactly, and why?

- O Data portal of Namur
- O Data Portal of Wallonie Bruxelles
- O Data portal of liege
- O Statistical data portal

Q3. What difficulties have you encountered with data portals (e.g., missing or

incomplete metadata, lack of description, accessibility of data, filtering, missing data, data not available, extract data, and/or other)?

Q4. What type of open data did you retrieve?

- O Geographic
- O Statistical
- O Housing
- O Health

"Other"?

Q5. What formats type of open data do you use and why?

O Tabular Data
O CSV
O EXCEL
O JSON

Q6. Would you like to provide further explanation of your answers to the above questions?

Q7. Did you encounter any difficulties with the open datasets (e.g., duplicate data, data quality, cell ambiguity, missing cell values, abbreviations, and/or other)?

Q8. Did you have any difficulties in using some of the data treatments (e.g., split columns, combining datasets, linking datasets, storing datasets, and/or other)?

Q9. Did you use the Api of portal data?

O Yes
O No
"Other"?

Q10. Are the APIs usable for your applications or do you have other API proposals to make your tasks easier?

O Yes
O No
"Other"?

Q11. Would you like to provide further explanation of your answers to the above questions?

Q12. Do you know of any tools you use to clean, unify, analyze, link and publish open data (e.g. Open Refine, MTab, Silk, Pubby, Neo4j (Graph Visualization), Fuseki Sparql query, and/or other), which ones?

Q13. Do you know of any framework you have used to clean, unify, interpret, analyze, link and publish open data, which ones?

Q14. Do you have any other comments and / or suggestions regarding the data portals (open data)? If so, can you write them down?

Q15. What is your level of education?

O High School
O High Education
O PhD

"Other"?

Q16. What is your field in computer science?

O Data analyst
O Data scientist
O IT specialist
O Researcher
"Other"?

Q17. What is your type of organization

O Public
O Private
O University

B.2 questionnaire for Evaluation of Pub_LOGD Framework

The purpose of these questions was to evaluate the Pub_LOGD Framework.

Q1.What are your general impressions of PubLOGD? Does it meet your expectations?

Q2.Would you like to give some further explanation of your answers to the above questions?

Perceived Usefulness of this platform

Q3.To what extent do you agree with the following statements? This platform is useful:
- This platform is useful to clean and unify a dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to clean and unify a dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to enrich a dataset with external Services (KGs).
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to design an annotated dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to understand an annotated datasets.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to store an annotated dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to merge an annotated dataset.

Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to publish an annotated dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- Generally, I think this Pub-LOGD platform useful for enhancing the quality data and the interpretation of datasets.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree

Perceived Ease-of-Use

Q4.To what extent do you agree with the following statements?
- Learning to operate the Pub-LOGD would be easy for me
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- I would find it easy to get this Pub-LOGD to do what I want it to do
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- My interaction with this platform would be clear and understandable
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- I would find this Pub-LOGD would be clear and understandable
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- It would be easy for me to become skillful at using this Pub-LOGD
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- I would find this tool easy to install
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- Generally, I think this framework is easy to use
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree

Q5.Would you like to provide further explanation of your answers to the above questions?

Q6.Do you have any other comments and / or suggestions regarding publishing platform? If so, can you write them down?

Q7.Do you know other data integration/interpretation platform/ application, which ones?

Q8. What is your Education level?
- O Higher Education
- O Secondary School
- O PhD

Q9. What is your field in computer science?

- O Data analyst
- O Data scientist
- O IT specialist
- O Researcher

"Other"?

Q10. What is your type of organization

O Public
O Private
O University

B.3 Survey of Framework Pub_LOGD Extension Evaluation

The purpose of these questions was to evaluate the Pub_LOGD Extension Framework.

Q1.What are your general impressions of Pub-LOGD Extension? Does it meet your expectations?

Q2.Would you like to give some further explanation of your answers to the above questions?

Perceived Usefulness of this platform

Q3.To what extent do you agree with the following statements? This platform is useful:
- This platform is useful to clean and unify a dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to clean and unify a dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to enrich a dataset with external Services (KGs).
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to design an annotated dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to understand an annotated datasets.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to store an annotated dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to merge an annotated dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- This platform is useful to publish an annotated dataset.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- Generally, I think this Pub-LOGD Extension platform useful for enhancing the quality data and the interpretation of datasets.
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree

Perceived Ease-of-Use

Q4.To what extent do you agree with the following statements?
- Learning to operate the Pub-LOGD would be easy for me
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- I would find it easy to get this Pub-LOGD to do what I want it to do
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- My interaction with this platform would be clear and understandable
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- I would find this Pub-LOGD would be clear and understandable
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- It would be easy for me to become skillful at using this Pub-LOGD
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- I would find this tool easy to install
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree
- Generally, I think this framework is easy to use
Totally Disagree ☐ ☐ ☐ ☐ ☐ ☐ ☐ Totally Agree

Q5.Would you like to provide further explanation of your answers to the above questions?

Q6.Do you have any other comments and / or suggestions regarding publishing platform? If so, can you write them down?

Q7.Do you know other data integration/interpretation platform/ application, which ones?

Q8. What is your Education level?
- O Higher Education
- O Secondary School
- O PhD

Q9. What is your field in computer science?

- O Data analyst
- O Data scientist
- O IT specialist
- O Researcher

"Other"?

Q10. What is your type of organization

- O Public
- O Private
- O University

B.4 RDF's representation of the population dataset of Belgium.

```
1  @prefix qb: <http://purl.org/linked-data/cube#> .
2  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3  @prefix owl: <http://www.w3.org/2002/07/owl#> .
4  @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
5  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
6  @prefix foaf: <http://xmlns.com/foaf/0.1/> .
7  <http://example.org/dataset/file-population> a qb:DataSet ;
8    rdfs:label "Census data grouped by stabel population" .
9  _:node1depk0gsjx1 a qb:DataStructureDefinition .
10 _:node1depk0gsjx2 a qb:ComponentSpecification .
11 <http://example.org/def/statistical-dimemsion/provincepop> a qb:MeasureProperty ;
12   rdfs:label "Province" .
13 _:node1depk0gsjx2 qb:dimension <http://example.org/def/statistical-dimemsion/provincepop
       ↪ > .
14 _:node1depk0gsjx1 qb:component _:node1depk0gsjx2 .
15 _:node1depk0gsjx3 a qb:ComponentSpecification .
16 <http://example.org/def/statistical-dimemsion/situation-marche> a qb:MeasureProperty ;
17   rdfs:label "Situation-marche" .
18 _:node1depk0gsjx3 qb:dimension <http://example.org/def/statistical-dimemsion/situation-
       ↪ marche> .
19 _:node1depk0gsjx1 qb:component _:node1depk0gsjx3 .
20 _:node1depk0gsjx4 a qb:ComponentSpecification .
21 <http://example.org/def/statistical-dimemsion/population> a qb:MeasureProperty ;
22   rdfs:label "Population" .
23 _:node1depk0gsjx4 qb:measure <http://example.org/def/statistical-dimemsion/population> .
24 _:node1depk0gsjx1 qb:component _:node1depk0gsjx4 .
25 <http://example.org/dataset/file-population> qb:structure _:node1depk0gsjx1 .
26 <http://example.org/dataset/file-population/rgion-de-bruxelles-capitale> a qb:Observation ;
27   qb:dataSet <http://example.org/dataset/file-population> .
28 <http://example.org/region-de-bruxelles-capitale> rdfs:label "Region de Bruxelles-Capitale" .
29 <http://example.org/dataset/file-population/rgion-de-bruxelles-capitale> <http://example.
       ↪ org/def/statistical-dimemsion/province> <http://example.org/rgion-de-bruxelles-
       ↪ capitale> ;
30   <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes actives (
         ↪ travailleur ou chomeur)" ;
31   <http://example.org/def/statistical-dimemsion/population> "443,877" ;
32   <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes inactives" ;
33   <http://example.org/def/statistical-dimemsion/population> "692,901" ;
34   <http://example.org/def/statistical-dimemsion/situation-marche> "Total" ;
35   <http://example.org/def/statistical-dimemsion/population> "1,136,778" .
36 <http://example.org/dataset/file-population/province-danvers> a qb:Observation ;
37   qb:dataSet <http://example.org/dataset/file-population> .
38 <http://example.org/province-danvers> rdfs:label "Province d'Anvers" .
39 <http://example.org/dataset/file-population/province-danvers> <http://example.org/def/
       ↪ statistical-dimemsion/province> <http://example.org/province-danvers> ;
40   <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes actives (
         ↪ travailleur ou chomeur)" ;
41   <http://example.org/def/statistical-dimemsion/population> "793,236" ;
42   <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes inactives" ;
43   <http://example.org/def/statistical-dimemsion/population> "980,031" ;
44   <http://example.org/def/statistical-dimemsion/situation-marche> "Total" ;
45   <http://example.org/def/statistical-dimemsion/population> "1,773,267" .
46 <http://example.org/dataset/file-population/province-du-limbourg> a qb:Observation ;
47   qb:dataSet <http://example.org/dataset/file-population> .
```

```
48  <http://example.org/province-du-limbourg> rdfs:label "Province du Limbourg" .
49  <http://example.org/dataset/file-population/province-du-limbourg> <http://example.org/def
        ↪ /statistical-dimemsion/province> <http://example.org/province-du-limbourg> ;
50    <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes actives (
          ↪ travailleur ou chomeur)" ;
51    <http://example.org/def/statistical-dimemsion/population> "385,651" ;
52    <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes inactives" ;
53    <http://example.org/def/statistical-dimemsion/population> "461,200" ;
54    <http://example.org/def/statistical-dimemsion/situation-marche> "Total" ;
55    <http://example.org/def/statistical-dimemsion/population> "846,851" .
56  <http://example.org/dataset/file-population/province-de-flandre-orientale> a qb:
        ↪ Observation ;
57    qb:dataSet <http://example.org/dataset/file-population> .
58  <http://example.org/province-de-flandre-orientale> rdfs:label "Province de Flandre orientale" .
        ↪
59  <http://example.org/dataset/file-population/province-de-flandre-orientale> <http://example.
        ↪ org/def/statistical-dimemsion/province> <http://example.org/province-de-flandre-
        ↪ orientale> ;
60    <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes actives (
          ↪ travailleur ou chomeur)" ;
61    <http://example.org/def/statistical-dimemsion/population> "671,549" ;
62    <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes inactives" ;
63    <http://example.org/def/statistical-dimemsion/population> "777,379" ;
64    <http://example.org/def/statistical-dimemsion/situation-marche> "Total" ;
65    <http://example.org/def/statistical-dimemsion/population> "1,448,928" .
66
67  <http://example.org/dataset/file-population/province-du-brabant-flamand> a qb:
        ↪ Observation ;
68    qb:dataSet <http://example.org/dataset/file-population> .
69  <http://example.org/province-du-brabant-flamand> rdfs:label "Province du Brabant flamand"
        ↪ .
70  <http://example.org/dataset/file-population/province-du-brabant-flamand> <http://example
        ↪ .org/def/statistical-dimemsion/province> <http://example.org/province-du-brabant
        ↪ -flamand> ;
71    <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes actives (
          ↪ travailleur ou chomeur)" ;
72    <http://example.org/def/statistical-dimemsion/population> "496,566" ;
73    <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes inactives" ;
74    <http://example.org/def/statistical-dimemsion/population> "592,126" ;
75    <http://example.org/def/statistical-dimemsion/situation-marche> "Total" ;
76    <http://example.org/def/statistical-dimemsion/population> "1,088,692" .
77  <http://example.org/dataset/file-population/province-de-flandre-occidentale> a qb:
        ↪ Observation ;
78    qb:dataSet <http://example.org/dataset/file-population> .
79  <http://example.org/province-de-flandre-occidentale> rdfs:label "Province de Flandre
        ↪ occidentale" .
80  <http://example.org/dataset/file-population/province-de-flandre-occidentale> <http://
        ↪ example.org/def/statistical-dimemsion/province> <http://example.org/province-de-
        ↪ flandre-occidentale> ;
81    <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes actives (
          ↪ travailleur ou chomeur)" ;
82    <http://example.org/def/statistical-dimemsion/population> "528,270" ;
83    <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes inactives" ;
84    <http://example.org/def/statistical-dimemsion/population> "639,732" ;
85    <http://example.org/def/statistical-dimemsion/situation-marche> "Total" ;
86    <http://example.org/def/statistical-dimemsion/population> "1,168,002" .
87  <http://example.org/dataset/file-population/province-du-brabant-wallon> a qb:Observation ;
```

```
88    qb:dataSet <http://example.org/dataset/file–population> .
89
90  <http://example.org/province–du–brabant–wallon> rdfs:label "Province du Brabant wallon" .
91  <http://example.org/dataset/file–population/province–du–brabant–wallon> <http://example.
        ↪ org/def/statistical–dimemsion/province> <http://example.org/province–du–brabant–
        ↪ wallon> ;
92    <http://example.org/def/statistical–dimemsion/situation–marche> "Personnes actives (
          ↪ travailleur ou chomeur)" ;
93    <http://example.org/def/statistical–dimemsion/population> "169,270" ;
94    <http://example.org/def/statistical–dimemsion/situation–marche> "Personnes inactives" ;
95    <http://example.org/def/statistical–dimemsion/population> "214,378" ;
96    <http://example.org/def/statistical–dimemsion/situation–marche> "Total" ;
97    <http://example.org/def/statistical–dimemsion/population> "383,648" .
98  <http://example.org/dataset/file–population/province–du–hainaut> a qb:Observation ;
99    qb:dataSet <http://example.org/dataset/file–population> .
100 <http://example.org/province–du–hainaut> rdfs:label "Province du Hainaut" .
101 <http://example.org/dataset/file–population/province–du–hainaut> <http://example.org/def/
        ↪ statistical–dimemsion/province> <http://example.org/province–du–hainaut> ;
102   <http://example.org/def/statistical–dimemsion/situation–marche> "Personnes actives (
          ↪ travailleur ou chomeur)" ;
103   <http://example.org/def/statistical–dimemsion/population> "550,932" ;
104   <http://example.org/def/statistical–dimemsion/situation–marche> "Personnes inactives" ;
105   <http://example.org/def/statistical–dimemsion/population> "768,813" ;
106   <http://example.org/def/statistical–dimemsion/situation–marche> "Total" ;
107   <http://example.org/def/statistical–dimemsion/population> "1,319,745" .
108 <http://example.org/dataset/file–population/province–de–lige> a qb:Observation ; qb:dataSet
        ↪ <http://example.org/dataset/file–population> .
109 <http://example.org/province–de–lige> rdfs:label "Province de Lige" .
110 <http://example.org/dataset/file–population/province–de–lige> <http://example.org/def/
        ↪ statistical–dimemsion/province> <http://example.org/province–de–lige> ;
111   <http://example.org/def/statistical–dimemsion/situation–marche> "Personnes actives (
          ↪ travailleur ou chmeur)" ;
112   <http://example.org/def/statistical–dimemsion/population> "462,803" ;
113   <http://example.org/def/statistical–dimemsion/situation–marche> "Personnes inactives" ;
114   <http://example.org/def/statistical–dimemsion/population> "619,333" ;
115   <http://example.org/def/statistical–dimemsion/situation–marche> "Total" ;
116   <http://example.org/def/statistical–dimemsion/population> "1,082,136" .
117
118 <http://example.org/dataset/file–population/province–de–luxembourg> a qb:Observation ;
119   qb:dataSet <http://example.org/dataset/file–population> .
120
121 <http://example.org/province–de–luxembourg> rdfs:label "Province de Luxembourg" .
122
123 <http://example.org/dataset/file–population/province–de–luxembourg> <http://example.org/
        ↪ def/statistical–dimemsion/province> <http://example.org/province–de–luxembourg>
        ↪ ;
124   <http://example.org/def/statistical–dimemsion/situation–marche> "Personnes actives (
          ↪ travailleur ou chmeur)" ;
125   <http://example.org/def/statistical–dimemsion/population> "119,693" ;
126   <http://example.org/def/statistical–dimemsion/situation–marche> "Personnes inactives" ;
127   <http://example.org/def/statistical–dimemsion/population> "153,512" ;
128   <http://example.org/def/statistical–dimemsion/situation–marche> "Total" ;
129   <http://example.org/def/statistical–dimemsion/population> "273,205" .
130 <http://example.org/dataset/file–population/province–de–namur> a qb:Observation ;
131   qb:dataSet <http://example.org/dataset/file–population> .
132 <http://example.org/province–de–namur> rdfs:label "Province de Namur" .
```

```
<http://example.org/dataset/file-population/province-de-namur> <http://example.org/def/statistical-dimemsion/province> <http://example.org/province-de-namur> ;
  <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes actives (travailleur ou chmeur)" ;
  <http://example.org/def/statistical-dimemsion/population> "212,213" ;
  <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes inactives" ;
  <http://example.org/def/statistical-dimemsion/population> "267,173" ;
  <http://example.org/def/statistical-dimemsion/situation-marche> "Total" ;
  <http://example.org/def/statistical-dimemsion/population> "479,386" .
<http://example.org/dataset/file-population/belgique> a qb:Observation ;
  qb:dataSet <http://example.org/dataset/file-population> .
<http://example.org/belgique> rdfs:label "Belgique" .
<http://example.org/dataset/file-population/belgique> <http://example.org/def/statistical-dimemsion/province> <http://example.org/belgique> ;
  <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes actives (travailleur ou chomeur)" ;
  <http://example.org/def/statistical-dimemsion/population> "4,834,060" ;
  <http://example.org/def/statistical-dimemsion/situation-marche> "Personnes inactives" ;
  <http://example.org/def/statistical-dimemsion/population> "6,166,578" ;
  <http://example.org/def/statistical-dimemsion/situation-marche> "Total" ;
  <http://example.org/def/statistical-dimemsion/population> "11,000,638" .
```

Listing B.1: The ontology of the population in Belgium.

B.5 RDF representation for the population of Belgium by NOE4J.

```
neo4j$ CALL n10s.rdf.import.fetch("file:///Users/edilton/Downloads/file-population.ttl","Turtle");
```

terminationStatus	triplesLoaded	triplesParsed	namespaces	extraInfo	callParams
"OK"	139	139	{ns0: "http://purl.org/linked-data/cube#", rdfs: "http://www.w3.org/2000/01/rdf-schema#", ns1: "http://example.org/def/statistical-dimemsion/"}	""	{}

Figure B.1: Population ontology information.

Figure B.2: Data visualization by NEO4J.

APPENDIX C

SINATRAAPP ASSESSMENTS

C.1 Estimating data analysis performance

C.1.1 Bootstrap Resampling for Precision and Recall Estimation

```
import numpy as np
from sklearn.utils import resample
from sklearn.metrics import precision_score, recall_score
# Define precision and recall values
precision_values = np.array([1, 1, 1, 1])
recall_values = np.array([0.955, 0.978, 0.959, 0.948])
# Define a function for bootstrap resampling
def bootstrap(data, n_iterations=1000, sample_size=500):
    bootstrap_samples = np.zeros(n_iterations)
    for i in range(n_iterations):
        # Resample the data with replacement
        resampled_data = resample(data, replace=True, n_samples=sample_size)
        # Calculate the mean of the resampled data and store it
        bootstrap_samples[i] = np.mean(resampled_data)
    return bootstrap_samples
# Perform bootstrap resampling for precision values
precision_samples = bootstrap(precision_values)
# Perform bootstrap resampling for recall values
recall_samples = bootstrap(recall_values)
# Calculate the mean estimates for precision and recall
precision_estimate = np.mean(precision_samples)
recall_estimate = np.mean(recall_samples)
# Calculate the confidence intervals for precision and recall
precision_ci = np.percentile(precision_samples, [2.5, 97.5])
recall_ci = np.percentile(recall_samples, [2.5, 97.5])
# Display the precision samples
print("precision_samples: ", precision_samples)
# Display the recall samples
print("recall_samples: ", recall_samples)
# Display the precision estimate
print("Precision Bootstrap Estimate: ", precision_estimate)
# Display the precision confidence intervals
```

```
print("Precision Confidence Intervals: ", precision_ci)
# Display the recall estimate
print("Recall Bootstrap Estimate: ", recall_estimate)
```

Listing C.1: *Bootstrap Resampling for Precision and Recall Estimation (Data analysis).*

C.1.2 Calculation of confidence intervals for both measurements

```
import numpy as np
import statistics
from scipy.stats import norm
from scipy import stats
# True population values
true_precision = [1, 1, 1, 1]
true_recall = [0.955, 0.978, 0.959, 0.948]
# Bootstrap estimates ( Replace with bootstrap precison and recall values)
bootstrap_precision = precision_samples
bootstrap_recall = recall_samples
# Calculate bootstrap confidence intervals
confidence_level = 0.95
alpha = 1 - confidence_level
# Precision confidence interval
precision_mean = np.mean(bootstrap_precision)
precision_std = statistics.stdev(bootstrap_precision)
print("precison_std", precision_std)
precision_standard_error = precision_std / np.sqrt(len(bootstrap_precision))
print("precision_std-error", precision_standard_error)
precision_z = np.abs(stats.norm.ppf(alpha / 2))
precision_lower_bound = precision_mean - (precision_z * precision_std)
precision_upper_bound = precision_mean + (precision_z * precision_std)
# Recall confidence interval
recall_mean = np.mean(bootstrap_recall)
recall_std = statistics.stdev(bootstrap_recall)
print("recall_std", recall_std)
recall_standard_error = recall_std / np.sqrt(len(bootstrap_recall))
print("recall_std-error", recall_standard_error)
recall_z = np.abs(stats.norm.ppf(alpha / 2))
recall_lower_bound = recall_mean - (recall_z * recall_std)
recall_upper_bound = recall_mean + (recall_z * recall_std)
# Print the results
print("Bootstrap Precision Confidence Interval: [{:.3f}, {:.3f}]".format(
    ↪ precision_lower_bound, precision_upper_bound))
print("Bootstrap Recall Confidence Interval: [{:.3f}, {:.3f}]".format(recall_lower_bound,
    ↪ recall_upper_bound))
 # True_Precision_confidence_interval
std_true_precision = np.std(true_precision)
avg_true_precesion = np.mean(true_precision)
print("true_precison_std", std_true_precision)
true_precision_standard_error = std_true_precision/ np.sqrt(len(true_precision))
print("true_precision_std-error", true_precision_standard_error)
true_precision_z = np.abs(stats.norm.ppf(alpha / 2))
true_precision_lower_bound = avg_true_precesion - (true_precision_z * std_true_precision)
true_precision_upper_bound = avg_true_precesion + (true_precision_z * std_true_precision)
print("Bootstrap True Precision Confidence Interval: [{:.3f}, {:.3f}]".format(
    ↪ true_precision_lower_bound, true_precision_upper_bound))
# True_Recall_confidence_interval
std_true_recall = np.std(true_recall)
avg_true_recall = np.mean(true_recall)
print("true_recall_std", std_true_recall)
true_recall_standard_error = std_true_recall/ np.sqrt(len(true_recall))
print("true_recall_std-error", true_recall_standard_error)
true_recall_z = np.abs(stats.norm.ppf(alpha / 2))
true_recall_lower_bound = avg_true_recall - (true_recall_z * std_true_recall)
true_recall_upper_bound = avg_true_recall + (true_recall_z * std_true_recall)
```

```
54 print("Bootstrap True Recall Confidence Interval: [{:.3f}, {:.3f}]".format(
       ↪ true_recall_lower_bound, true_recall_upper_bound))
```

Listing C.2: *Calculation of confidence intervals for both measurements (Data analysis).*

C.2 Survey of Framework SinatraAPP

The purpose of these questions was to evaluate the SinatraApp.

annoying	☐	☐	☐	☐	☐	☐	☐	enjoyable
not understandable understandable	☐	☐	☐	☐	☐	☐	☐	
creative	☐	☐	☐	☐	☐	☐	☐	dull
easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn
easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn
valuable	☐	☐	☐	☐	☐	☐	☐	inferior
boring	☐	☐	☐	☐	☐	☐	☐	exciting
not interesting	☐	☐	☐	☐	☐	☐	☐	interesting
unpredictable	☐	☐	☐	☐	☐	☐	☐	predictable
fast	☐	☐	☐	☐	☐	☐	☐	slow
inventive	☐	☐	☐	☐	☐	☐	☐	conventional
obstructive	☐	☐	☐	☐	☐	☐	☐	supportive
good	☐	☐	☐	☐	☐	☐	☐	bad
easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn
complicated	☐	☐	☐	☐	☐	☐	☐	easy
unlikable	☐	☐	☐	☐	☐	☐	☐	pleasing
usual	☐	☐	☐	☐	☐	☐	☐	leading edge

unpleasant	☐	☐	☐	☐	☐	☐	☐	pleasant
secure	☐	☐	☐	☐	☐	☐	☐	not secure
motivating	☐	☐	☐	☐	☐	☐	☐	demotivating
meets expectations	☐	☐	☐	☐	☐	☐	☐	does not meet expectations
inefficient	☐	☐	☐	☐	☐	☐	☐	efficient
clear	☐	☐	☐	☐	☐	☐	☐	confusing
impractical	☐	☐	☐	☐	☐	☐	☐	practical
organized	☐	☐	☐	☐	☐	☐	☐	cluttered
attractive	☐	☐	☐	☐	☐	☐	☐	unattractive
friendly	☐	☐	☐	☐	☐	☐	☐	unfriendly
easy to learn	☐	☐	☐	☐	☐	☐	☐	difficult to learn
conservative	☐	☐	☐	☐	☐	☐	☐	innovative

Q2. Can you give us any other remarks or comments?

C.2.1 Correlations of the items per scale and reliability coefficients

In this step, the Cronbachs Alpha-Coefficient (Cronbach, 1951 [49]) is used. It is a measure for the consistency of a scale.

C.2.2 Distribution of Answers per Item

The figure C.1 presents distribution of answers to the single items by scales.

The figure C.2 presents distribution of answers to the single items.

C.2.3 Comparison of SinatraUI UEQ results in relation to the UEQ benchmark

Attractiveness	
Items	Correlation
1, 12	0,85
1, 14	0,89
1, 16	0,86
1, 24	0,48
1, 25	0,60
12, 14	0,83
12, 16	0,88
12, 24	0,32
12, 25	0,58
14, 16	0,89
14, 24	0,62
14, 25	0,84
16, 24	0,43
16, 25	0,69
24, 25	0,75
Average	0,70
Alpha	0,93
Conf. Int. Alpha (5%)	0,86
	0,97

Perspicuity	
Items	Correlation
2, 4	0,46
2, 13	0,40
2, 21	0,36
4, 13	0,38
4 , 21	0,20
13, 21	0,74
Average	0,42
Alpha	0,75
Conf. Int. Alpha (5%)	0,46
	0,88

Efficiency	
Items	Correlation
9, 20	0,19
9, 22	0,32
9, 23	0,05
20, 22	0,32
20, 23	0,42
22, 23	0,46
Average	0,29
Alpha	0,63
Conf. Int. Alpha (5%)	0,20
	0,82

Dependability	
Items	Correlation
8, 11	-0,08
8, 17	-0,02
8, 19	0,01
11, 17	0,37
11, 19	0,19
17, 19	0,72
Average	0,20
Alpha	0,50
Conf. Int. Alpha (5%)	-0,06
	0,76

Stimulation	
Items	Correlation
5, 6	0,18
5, 7	0,33
5, 18	0,41
6, 7	0,78
6, 18	0,69
7, 18	0,65
Average	0,51
Alpha	0,80
Conf. Int. Alpha (5%)	0,58
	0,91

Novelty	
Items	Correlation
3, 10	0,27
3, 15	-0,19
3, 26	0,04
10, 15	-0,30
10, 26	0,12
15, 26	0,31
Average	0,04
Alpha	0,15
Conf. Int. Alpha (5%)	-0,80
	0,60

Nr	Item	1	2	3	4	5	6	7	Scale
1	annoying/enjoyable	0	1	1	0	4	7	7	Attractiveness
2	not understandable/understandable	0	0	1	1	3	6	9	Perspicuity
3	dull/creative	0	3	0	8	2	4	3	Novelty
4	difficult to learn/easy to learn	1	2	2	0	2	6	7	Perspicuity
5	inferior/valuable	0	0	1	3	4	6	6	Stimulation
6	boring/exciting	0	0	1	6	5	6	2	Stimulation
7	not interesting/interesting	0	1	0	3	6	8	2	Stimulation
8	unpredictable/predictable	0	1	4	3	5	6	1	Dependability
9	slow/fast	0	1	1	3	3	8	4	Efficiency
10	conventional/inventive	0	1	1	5	8	4	1	Novelty
11	obstructive/supportive	0	1	0	4	6	6	3	Dependability
12	bad/good	1	0	0	1	3	8	7	Attractiveness
13	complicated/easy	0	0	0	2	7	4	7	Perspicuity
14	unlikable/pleasing	1	0	1	2	4	6	6	Attractiveness
15	usual/leading edge	1	0	1	5	6	3	4	Novelty
16	unpleasant/pleasant	1	0	1	0	6	6	6	Attractiveness
17	not secure/secure	0	0	1	0	2	10	7	Dependability
18	demotivating/motivating	0	1	2	5	6	4	2	Stimulation
19	does not meet expectations/meets expectations	0	0	0	1	3	10	6	Dependability
20	inefficient/efficient	0	0	0	1	4	6	9	Efficiency
21	confusing/clear	0	0	0	0	5	9	6	Perspicuity
22	impractical/practical	0	0	0	1	1	11	7	Efficiency
23	cluttered/organized	0	1	0	1	3	8	7	Efficiency
24	unattractive/attractive	0	1	0	5	3	5	6	Attractiveness
25	unfriendly/friendly	0	0	0	2	6	4	8	Attractiveness
26	conservative/innovative	3	0	1	6	2	6	2	Novelty

Figure C.1: Distribution of Answers per Item.

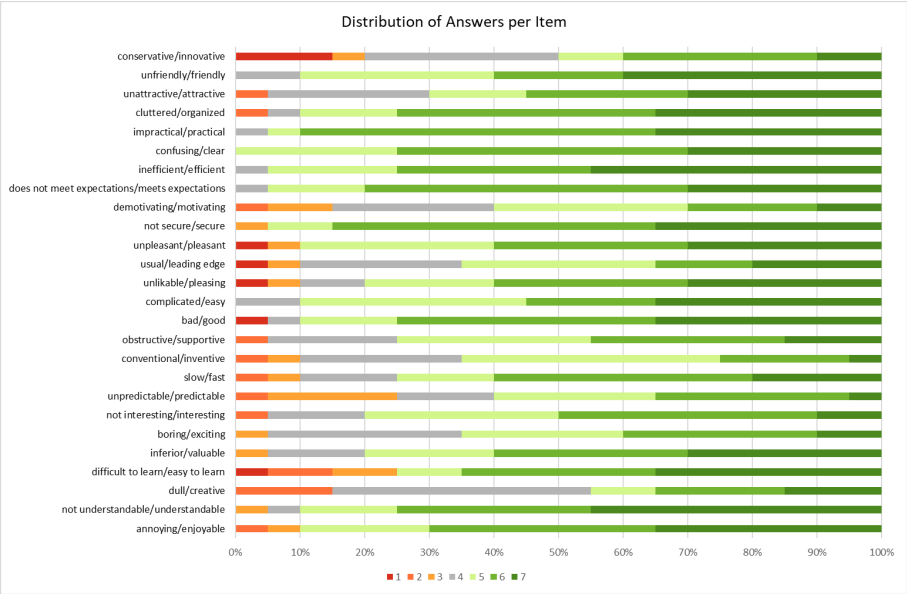


Figure C.2: Distribution of Answers per Item.

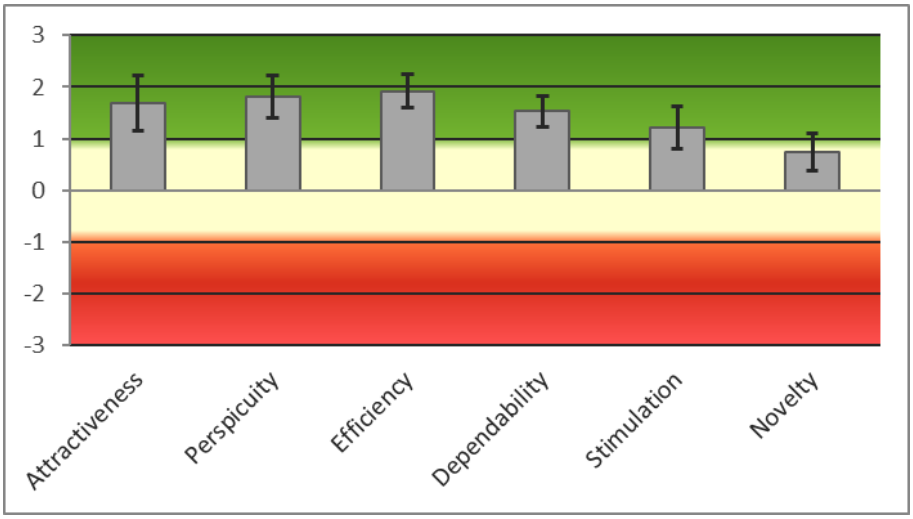


Figure C.3: Comparison of SinatraUI UEQ results in relation to the UEQ benchmark.

BIBLIOGRAPHY

[1] https://openrefine.org/, 2010. [last visited on October 21, 2019].

[2] http://silkframework.org/, 2014. [last visited on October 24, 2019].

[3] https://opendefinition.org/, 2019. [last visited on November 8, 2019].

[4] Nora Abdelmageed and Sirko Schindler. Jentab meets semtab 2021's new challenges. In **SemTab@ ISWC**, pages 42–53, 2021.

[5] Rabeb Abida, Emna Hachicha Belghith, and Anthony Cleve. An end-to-end framework for integrating and publishing linked open government data. In **2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)**, pages 257–262. IEEE, 2020.

[6] Rabeb Abida, Emna Hachicha Belghith, and Anthony Cleve. Towards transparent governance by publishing open statistical data. In **Digital Technologies and Applications: Proceedings of ICDTA'23, Fez, Morocco, Volume 1**, pages 355–365. Springer, 2023.

[7] Rabeb Abida and Anthony Cleve. Improving the usability of tabular data through data annotation, repair and augmentation. In **Nordic Artificial Intelligence Research and Development: 4th Symposium of the Norwegian AI Society, NAIS 2022, Oslo, Norway, May 31–June 1, 2022, Revised Selected Papers**, pages 65–77. Springer International Publishing Cham, 2023.

[8] Rabeb Abida, Anthony Cleve, and Emna Hachicha Belghith. Supporting semantics-aware analysis in linked open data-enabled information systems. 2019.

[9] Rabeb Abida, Emna Hachicha, and Anthony Cleve. Supporting semantic-aware analysis in linked open data-enabled government systems. 2019.

[10] Harith Alani, David Dupplaw, John Sheridan, Kieron O'Hara, John Darlington, Nigel Shadbolt, and Carol Tullo. Unlocking the potential of public sector information with semantic web technology. In **The semantic web**, pages 708–721. Springer, 2007.

[11] Afnan M AlSukhayri, Muhammad Ahtisham Aslam, Sachi Arafat, and Naif Radi Aljohani. Leveraging the saudi linked open government data: a framework and potential benefits. **International Journal of Modern Education and Computer Science**, 11(7):14, 2019.

[12] Ester Appelgren and Gunnar Nygren. Data journalism in sweden: Introducing new methods and genres of journalism into "old" organizations. **Digital journalism**, 2(3):394–405, 2014.

[13] Judie Attard, Fabrizio Orlandi, Simon Scerri, and Sören Auer. A systematic review of open government data initiatives. **Government information quarterly**, 32(4):399–418, 2015.

[14] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In **The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings**, pages 722–735. Springer, 2007.

[15] Sören Auer, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N Mendes, Bert Van Nuffelen, et al. Managing the life-cycle of linked data with the lod2 stack. In **The Semantic Web–ISWC 2012: 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part II 11**, pages 1–16. Springer, 2012.

[16] Sören Auer and Jens Lehmann. Creating knowledge out of interlinked data. **Semantic Web**, 1(1-2):97–104, 2010.

[17] Roberto Avogadro and Marco Cremaschi. Mantistable v: A novel and efficient approach to semantic table interpretation. In **SemTab@ ISWC**, pages 79–91, 2021.

[18] Hiteshwar kumar Azad, Akshay Deepak, and Amisha Azad. Lod search engine: A semantic search over linked data. **Journal of Intelligent Information Systems**, pages 1–21, 2021.

[19] Rabia Azzi, Gayo Diallo, E Jiménez-Ruiz, O Hassanzadeh, V Efthymiou, J Chen, and K Srinivas. Amalgam: making tabular dataset explicit with knowledge graph. In **SemTab@ ISWC**, pages 9–16, 2020.

[20] Molood Barati, Quan Bai, and Qing Liu. Mining semantic association rules from rdf data. **Knowledge-Based Systems**, 133:183–196, 2017.

[21] Florian Bauer and Martin Kaltenböck. Linked open data: The essentials. **Edition mono/monochrom, Vienna**, 710:21, 2011.

[22] José Rodolfo Beluzo, Bruno Dias, Gisele Silva Craveiro, and Renata Araújo. Re-public: workflow to publish and reuse linked open government data. In **XVII Brazilian Symposium on Information Systems**, pages 1–7, 2021.

[23] Fabio Benedetti, Sonia Bergamaschi, and Laura Po. Online index extraction from linked open data sources. In **Second International Workshop on Linked Data for Information Extraction (LD4IE 2014)**, volume 1267, pages 9–20. DEU, 2014.

[24] Martin Beno, Kathrin Figl, Jürgen Umbrich, and Axel Polleres. Open data hopes and fears: determining the barriers of open data. In **2017 Conference for E-Democracy and Open Government (CeDEM)**, pages 69–81. IEEE, 2017.

[25] Martin Beno, Kathrin Figl, Jürgen Umbrich, and Axel Polleres. Perception of key barriers in using and publishing open data. **JeDEM-eJournal of eDemocracy and Open Government**, 9(2):134–165, 2017.

[26] Martin Beno, Erwin Filtz, Sabrina Kirrane, and Axel Polleres. Doc2rdfa: semantic annotation for web documents. In **Proceedings of the Posters and Demos Track of the 15th International Conference on Semantic Systems (SEMANTiCS 2019)**, page 1, 2019.

[27] Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In **Proceedings of the 3rd international semantic web user interaction workshop**, volume 2006, page 159. Athens, Georgia, 2006.

[28] Christian Bizer. Tom health, tim berners-lee, linked data-the story so far. **International Journal on Semantic Web and Information Systems (IJSWIS). Citado na pág**, 12:13, 2009.

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

[30] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In **Proceedings of the 2008 ACM SIGMOD international conference on Management of data**, pages 1247–1250, 2008.

[31] Harry N Boone and Deborah A Boone. Analyzing likert data. **Journal of extension**, 50(2):1–5, 2012.

[32] Jan Bosch, Helena Holmström Olsson, and Ivica Crnkovic. Engineering ai systems: A research agenda. **Artificial Intelligence Paradigms for Smart Cyber-Physical Systems**, pages 1–19, 2021.

[33] Katrin Braunschweig, Julian Eberius, Maik Thiele, and Wolfgang Lehner. The state of open data: Limits of current open data platforms categories and subject descriptors. In **Conference Proceedings World Wide Web Conference, WWW**, 2012.

[34] Jérôme Brugger, Marianne Fraefel, Reinhard Riedl, Hansjakob Fehr, Daniel Schöeneck, and Christoph Stäehli Weissbrod. Current barriers to open government data use and visualization by political intermediaries. In **2016 Conference for E-Democracy and Open Government (CeDEM)**, pages 219–229. IEEE, 2016.

[35] Marut Buranarach, Pattama Krataithong, Sirinaree Hinsheranan, Somchoke Ruengittinun, and Thepchai Supnithi. A scalable framework for creating open government data services from open government data catalog. In **Proceedings of the 9th International Conference on Management of Digital EcoSystems**, pages 1–5, 2017.

[36] Sarven Capadisli, Sören Auer, and Axel-Cyrille Ngonga Ngomo. Linked sdmx data. **Semantic Web**, 6(2):105–112, 2015.

[37] James Carpenter and John Bithell. Bootstrap confidence intervals: when, which, what? a practical guide for medical statisticians. **Statistics in medicine**, 19(9):1141–1164, 2000.

[38] Pompeu Casanovas. Legal crowdsourcing and relational law: What the semantic web can do for legal education. **Journal of Australian Law Teachers Association**, 5(1/2):159–176, 2012.

[39] Yannis Charalabidis, Charalampos Alexopoulos, and Euripidis Loukis. A taxonomy of open government data research areas and topics. **Journal of Organizational Computing and Electronic Commerce**, 26(1-2):41–63, 2016.

[40] Yannis Charalabidis, Anneke Zuiderwijk, Charalampos Alexopoulos, Marijn Janssen, Thomas Lampoltshammer, Enrico Ferro, Yannis Charalabidis, Anneke Zuiderwijk, Charalampos Alexopoulos, Marijn Janssen, et al. The multiple life cycles of open data creation and use. **The World of Open Data: Concepts, Methods, Tools and Experiences**, pages 11–31, 2018.

[41] Shuang Chen, Alperen Karaoglu, Carina Negreanu, Tingting Ma, Jin-Ge Yao, Jack Williams, Andy Gordon, and Chin-Yew Lin. Linkingpark: An integrated approach for semantic table interpretation. In **SemTab@ ISWC**, pages 65–74, 2020.

[42] Abiola Paterne Chokki, Rabeb Abida, Benoît nay é Fr, Benoît Vanderose, and Anthony Cleve. Odsag: Enhancing open data discoverability and understanding through semantic annotation. **EGOV-CeDEM-ePart 2022**, page 120, 2022.

[43] European Commission. Directive 2003/98/EC of the European Parliament and of the council of 17 November 2003 on the re-use of public sector information. http://ec.europa.eu/information_society/policy/psi/rules/eu/index_en.htm, 2003. [Online; Accessed Dec 12 2012].

[44] European Commission. Digital agenda: Turning government data into gold. Brussels: European Commission, 2011.

[45] Peter Conradie and Sunil Choenni. On the barriers for local government releasing open data. **Government information quarterly**, 31:S10–S17, 2014.

[46] Marco Cremaschi, Roberto Avogadro, Andrea Barazzetti, David Chieregato, E Jiménez-Ruiz, O Hassanzadeh, V Efthymiou, J Chen, and K Srinivas. Mantistable se: an efficient approach for the semantic table interpretation. In **SemTab@ ISWC**, pages 75–85, 2020.

[47] Marco Cremaschi, Roberto Avogadro, and David Chieregato. Mantistable: an automatic approach for the semantic table interpretation. **SemTab@ ISWC**, 2019:15–24, 2019.

[48] Marco Cremaschi, Flavio De Paoli, Anisa Rula, and Blerina Spahiu. A fully automated approach to a complete semantic table interpretation. **Future Generation Computer Systems**, 112:478–500, 2020.

[49] Lee J Cronbach. Coefficient alpha and the internal structure of tests. **psychometrika**, 16(3):297–334, 1951.

[50] Vincenzo Cutrona, Federico Bianchi, Ernesto Jiménez-Ruiz, and Matteo Palmonari. Tough tables: Carefully evaluating entity linking for tabular data. In **The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II 19**, pages 328–343. Springer, 2020.

[51] Vincenzo Cutrona, Jiaoyan Chen, Vasilis Efthymiou, Oktie Hassanzadeh, Ernesto Jiménez-Ruiz, Juan Sequeda, Kavitha Srinivas, Nora Abdelmageed, Madelon Hulsebos, Daniela Oliveira, et al. Results of semtab 2021. **Proceedings of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching**, 3103:1–12, 2022.

[52] Vincenzo Cutrona, Michele Ciavotta, Flavio De Paoli, and Matteo Palmonari. Asia: a tool for assisted semantic interpretation and annotation of tabular data. In **ISWC (Satellites)**, pages 209–212, 2019.

[53] Richard Cyganiak, Michael Hausenblas, and Eoin McCuirc. Official statistics and the practice of data fidelity. In **Linking government data**, pages 135–151. Springer, 2011.

[54] Richard Cyganiak, Dave Reynolds, and Jeni Tennison. The rdf data cube vocabulary. **World Wide Web Consortium**, page 134, 2014.

[55] Fred D Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. **MIS quarterly**, pages 319–340, 1989.

[56] Elisabete M de Gouveia Dal Pino and Grzegorz Kowal. Particle acceleration by magnetic reconnection. **Magnetic Fields in Diffuse Media**, pages 373–398, 2015.

[57] Dong Deng, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed K Elmagarmid, Ihab F Ilyas, Samuel Madden, Mourad Ouzzani, and Nan Tang. The data civilizer system. In **Cidr**, 2017.

[58] Diego Dermeval, Jéssyka Vilela, Ig Ibert Bittencourt, Jaelson Castro, Seiji Isotani, and Patrick Brito. A systematic review on the use of ontologies in requirements engineering. In **2014 brazilian symposium on software engineering**, pages 1–10. IEEE, 2014.

[59] Thomas J DiCiccio and Bradley Efron. Bootstrap confidence intervals. **Statistical science**, 11(3):189–228, 1996.

[60] Julian Eberius, Katrin Braunschweig, Maik Thiele, and Wolfgang Lehner. Identifying and weighting integration hypotheses on open data platforms. In **Proceedings of the First International Workshop on Open Data**, pages 22–29, 2012.

[61] Ivan Ermilov, Michael Martin, Jens Lehmann, and Sören Auer. Linked open data statistics: Collection and exploitation. In **International Conference on Knowledge Engineering and the Semantic Web**, pages 242–249. Springer, 2013.

[62] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. Introducing wikidata to the linked data web. In **International semantic web conference**, pages 50–65. Springer, 2014.

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

[64] Laura Faulkner. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. **Behavior Research Methods, Instruments, & Computers**, 35:379–383, 2003.

[65] Nosheen Fayyaz, Irfan Ullah, and Shah Khusro. On the current state of linked open data: Issues, challenges, and future directions. **International Journal on Semantic Web and Information Systems (IJSWIS)**, 14(4):110–128, 2018.

[66] Rita Fleiner. Linking of open government data. In **2018 ieee 12th international symposium on applied computational intelligence and informatics (saci)**, pages 1–5. IEEE, 2018.

[67] Harald Foidl, Michael Felderer, and Rudolf Ramler. Data smells: Categories, causes and consequences, and detection of suspicious data in ai-based systems. In **Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI**, pages 229–239, 2022.

[68] International Organization for Standardization, Software Technical Committee ISO/IEC JTC 1, Information technology. Subcommittee SC 7, and systems

engineering. **Systems and Software Engineering: Systems and Software Quality Requirements and Evaluation (SQuaRE): System and Software Quality Models**. ISO, 2011.

[69] André Freitas, Edward Curry, João Gabriel Oliveira, and Sean O'Riain. Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends. **IEEE Internet Computing**, 16(1):24–33, 2011.

[70] A-SIT Zentrum für sichere Informationstechnologie, Andrea Kö, Christine Leitner, Alexander Prosser, et al. Technology-enabled innovation for democracy, government and governance. 2013.

[71] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? **International journal of human-computer studies**, 43(5-6):907–928, 1995.

[72] Peter Hall. On the bootstrap and confidence intervals. **The Annals of Statistics**, pages 1431–1452, 1986.

[73] Kelli Ham. Openrefine (version 2.5). http://openrefine. org. free, open-source tool for cleaning and transforming data. **Journal of the Medical Library Association: JMLA**, 101(3):233, 2013.

[74] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. Openke: An open toolkit for knowledge embedding. In **Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations**, pages 139–144, 2018.

[75] Michael Hausenblas. Exploiting linked data to build web applications. **IEEE Internet Computing**, 13(4):68–73, 2009.

[76] Arto Hellas, Petri Ihantola, Andrew Petersen, Vangel V Ajanovski, Mirela Gutica, Timo Hynninen, Antti Knutas, Juho Leinonen, Chris Messom, and Soohyun Nam Liao. Predicting academic performance: a systematic literature review. In **Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education**, pages 175–199, 2018.

[77] Yusniel Hidalgo-Delgado, Reina Estrada-Nelson, Bin Xu, Boris Villazon-Terrazas, Amed Leiva-Mederos, and Andrés Tello. Methodological guidelines for publishing library data as linked data. In **2017 International Conference on Information Systems and Computer Science (INCISCOS)**, pages 241–246. IEEE, 2017.

[78] Anders Hjalmarsson, Paul Johannesson, Gustaf Juell-Skielse, and Daniel Rudmark. Beyond innovation contests: A framework of barriers to open innovation of digital services. In **ECIS 14 22nd European Conference on Information Systems, Tel Aviv, 5-13 June 2014**, 2014.

[79] Anders Hjalmarsson, Niklas Johansson, and Daniel Rudmark. Mind the gap: exploring stakeholders' value with open data assessment. In **2015 48th Hawaii international conference on system sciences**, pages 1314–1323. IEEE, 2015.

[80] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. **ACM Computing Surveys (CSUR)**, 54(4):1–37, 2021.

[81] White House. Memorandum on transparency and open government. Washington,DC:WhiteHouse(Retrievedfromhttp://edocket.access.gpo.gov/2009/pdf/E9-1777.pdf, 2009. [2009, January 21].

[82] White House. Memorandum on transparency and open government. Washington,DC:WhiteHouse(Retrievedfromhttp://edocket.access.gpo.gov/2009/pdf/E9-1777.pdf, 2009. [2009, January 21].

[83] Ruhua Huang, Tong Lai, and Lihong Zhou. Proposing a framework of barriers to opening government data in china: A critical literature review. **Library Hi Tech**, 35(3):421–438, 2017.

[84] Viet-Phi Huynh, Jixiong Liu, Yoan Chabot, Frédéric Deuzé, Thomas Labbé, Pierre Monnin, and Raphaël Troncy. Dagobah: Table and graph contexts for efficient semantic annotation of tabular data. 2021.

[85] Nick Hynes, D Sculley, and Michael Terry. The data linter: Lightweight, automated sanity checking for ml data sets. In **NIPS MLSys Workshop**, volume 1, page 5, 2017.

[86] Ali Inan, Murat Kantarcioglu, Elisa Bertino, and Monica Scannapieco. A hybrid approach to private record linkage. In **2008 IEEE 24th International Conference on Data Engineering**, pages 496–505. IEEE, 2008.

[87] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. **An introduction to statistical learning**, volume 112. Springer, 2013.

[88] Marijn Janssen, Yannis Charalabidis, and Anneke Zuiderwijk. Benefits, adoption barriers and myths of open data and open government. **Information systems management**, 29(4):258–268, 2012.

[89] Marijn Janssen and Jeroen van den Hoven. Big and open linked data (bold) in government: A challenge to transparency and privacy?, 2015.

[90] Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, and Kavitha Srinivas. Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In **The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17**, pages 514–530. Springer, 2020.

[91] Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, Kavitha Srinivas, and Vincenzo Cutrona. Results of semtab 2020. In **CEUR Workshop Proceedings**, volume 2775, pages 1–8, 2020.

[92] Evangelos Kalampokis, Efthimios Tambouris, Areti Karamanou, and Konstantinos Tarabanis. Open statistics: The rise of a new era for open data? In **International Conference on Electronic Government**, pages 31–43. Springer, 2016.

[93] Evangelos Kalampokis, Efthimios Tambouris, and Konstantinos Tarabanis. Linked open government data analytics. In **Electronic Government: 12th IFIP WG 8.5 International Conference, EGOV 2013, Koblenz, Germany, September 16-19, 2013. Proceedings 12**, pages 99–110. Springer, 2013.

[94] Evangelos Kalampokis, Efthimios Tambouris, and Konstantinos Tarabanis. On publishing linked open government data. In **Proceedings of the 17th Panhellenic Conference on Informatics**, pages 25–32, 2013.

[95] Jānis Kampars, Jelena Zdravkovic, Janis Stirna, and Jānis Grabis. Extending organizational capabilities with open data to support sustainable and dynamic business ecosystems. **Software and Systems Modeling**, 19(2):371–398, 2020.

[96] Maxat Kassen. Open data in kazakhstan: Incentives, implementation and challenges. **Information Technology & People**, 2017.

[97] Paramjeet Kaur and Parma Nand. Towards transparent governance by unifying open data. **IAENG International Journal of Computer Science**, 48(4), 2021.

[98] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering–a systematic literature review. **Information and software technology**, 51(1):7–15, 2009.

[99] Fotis Kitsios, Nikolaos Papachristos, and Maria Kamariotou. Business models for open data ecosystem: Challenges and motivations for entrepreneurship and innovation. In **2017 IEEE 19th Conference on Business Informatics (CBI)**, volume 1, pages 398–407. IEEE, 2017.

[100] Tomás Knap. Towards odalic, a semantic table interpretation tool in the adequate project. In **LD4IE@ ISWC**, pages 26–37, 2017.

[101] Pavel Krasikov, Timo Obrecht, Christine Legner, and Markus Eurich. Is open data ready for use by enterprises? learnings from corporate registers. In **DATA**, pages 109–120, 2020.

[102] Pattama Krataithong, Marut Buranarach, and Thepchai Supnithi. Rdf dataset management framework for data. go. th. In **Recent Advances and Future Prospects in Knowledge, Information and Creativity Support Systems: Selected Revised Papers from the Tenth International Conference on Knowledge, Information and Creativity Support Systems (KICSS 2015), 12-14 November 2015, Phuket, Thailand 10**, pages 42–53. Springer, 2018.

[103] Petr Křemen and Martin Nečaskỳ. Improving discoverability of open government data with rich metadata descriptions using semantic government vocabulary. **Journal of Web Semantics**, 55:1–20, 2019.

[104] R Krimmer, T Kalvet, M Toots, and K McBride. Opengovintelligence: Fostering innovation and creativity in europe through public administration modernization towards supplying and exploiting linked open statistical data, 2016.

[105] Sylvain Kubler, Jerermy Robert, Sebastian Neumaier, Jürgen Umbrich, and Yves Le Traon. Comparison of metadata quality in open data portals using the analytic hierarchy process. **Government Information Quarterly**, 35(1):13–29, 2018.

[106] Jan Kucera and Dusan Chlapek. Benefits and risks of open government data. **Journal of Systems Integration**, 5(1):30, 2014.

[107] Jan Kucera, Dusan Chlapek, Jakub Klímek, and Martin Necaskỳ. Methodologies and best practices for open data publication. In **DATESO**, pages 52–64, 2015.

[108] Matt Laessig, Bryon Jacob, and Carla AbouZahr. Opening data for global health. **The Palgrave Handbook of Global Health Data Methods for Policy and Practice**, pages 451–468, 2019.

[109] Gongjin Lan, Ting Liu, Xu Wang, Xueli Pan, and Zhisheng Huang. A semantic web technology index. **Scientific reports**, 12(1):3672, 2022.

[110] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. A large public corpus of web tables containing time and context metadata. In **Proceedings of the 25th international conference companion on world wide web**, pages 75–76, 2016.

[111] Kristina Lerman, Anon Plangprasopchock, and Craig A Knoblock. Semantic labeling of online information sources. **International Journal on Semantic Web and Information Systems (IJSWIS)**, 3(3):36–56, 2007.

[112] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and searching web tables using entities, types and relationships. **Proceedings of the VLDB Endowment**, 3(1-2):1338–1347, 2010.

[113] Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. **Journal of Web Semantics**, page 100761, 2022.

[114] Jixiong Liu and Raphaël Troncy. Dagobah: an end-to-end context-free tabular data semantic annotation system. **SemTab@ ISWC**, 2019.

[115] Qing Liu, Quan Bai, Li Ding, Huong Pho, Yun Chen, Corné Kloppers, Deborah L McGuinness, David Lemon, Paulo de Souza, Peter Fitch, et al. Linking australian government data for sustainability science-a case study. In **Linking government data**, pages 181–204. Springer, 2011.

[116] B Farias Loscio, C Burle, and N Calegari. W3c data on the web best practices, 2017.

[117] Bernadette Farias Lóscio, Caroline Burle, and Newton Calegari. Data on the web best practices. w3c recommendation. **World Wide Web Consortium**, 2017.

[118] Fadi Maali, Richard Cyganiak, and Vassilios Peristeras. A publishing pipeline for linked government data. In **Extended Semantic Web Conference**, pages 778–792. Springer, 2012.

[119] Sarah B Macfarlane, Carla AbouZahr, et al. **The Palgrave Handbook of Global Health Data Methods for Policy and Practice**. Springer, 2019.

[120] Renáta Máchová, Miloslav Hub, and Martin Lnenicka. Usability evaluation of open data portals: Evaluating data discoverability, accessibility, and reusability from a stakeholders' perspective. **Aslib Journal of Information Management**, 2018.

[121] SM Hasan Mahmud, Md Altab Hossin, Md Rezwan Hasan, Hosney Jahan, Sheak Rashed Haider Noori, and Md Razu Ahmed. Publishing csv data as linked data on the web. In **Proceedings of ICETIT 2019: Emerging Trends in Information Technology**, pages 805–817. Springer, 2020.

[122] SM Hasan Mahmud, Md Altab Hossin, Hosney Jahan, Sheak Rashed Haider Noori, and Touhid Bhuiyan. Csv-annotate: Generate annotated tables from csv file. In **2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)**, pages 71–75. IEEE, 2018.

[123] Chris Martin. Barriers to the open government data agenda: Taking a multi-level perspective. **Policy & Internet**, 6(3):217–240, 2014.

[124] Sébastien Martin. Risk analysis to overcome barriers to open data. **Electronic Journal of e-Government**, 11(2):pp348–359, 2013.

[125] Luiz CB Martins, Márcio C Victorino, Maristela Holanda, George Ghinea, and Tor-Morten Grønli. Unbgold: Unb government open linked data: semantic enrichment of open data tool. In **Proceedings of the 10th International Conference on Management of Digital EcoSystems**, pages 1–6, 2018.

[126] Ricardo Matheus and Marijn Janssen. An evaluation framework for linked open statistical data in government. In **Electronic Government: 16th IFIP WG 8.5 International Conference, EGOV 2017, St. Petersburg, Russia, September 4-7, 2017, Proceedings 16**, pages 255–263. Springer, 2017.

[127] Andrea Maurino, Anisa Rula, Bjørn Marius von, Mauricio Soto Gomez, Brian Elvesæter, and Dumitru Roman. Modelling and linking company data in the eubusinessgraph platform. In **Proceedings of the 5th Workshop on Data Science for Macro-modeling with Financial and Economic Datasets**, pages 1–6, 2019.

[128] Suvodeep Mazumdar and Ziqi Zhang. A tool for creating and visualizing semantic annotations on relational tables. CEUR Workshop Proceedings, 2016.

[129] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In **Proceedings of the 7th international conference on semantic systems**, pages 1–8, 2011.

[130] Alistair Miles and Sean Bechhofer. Skos simple knowledge organization system reference. **W3C recommendation**, 2009.

[131] Raimondo Iemma Minotas and Federico Morando. Open data quality measurement framework: Definition and application to open government data. 2020.

[132] Are We Confident Our Model's. Are we confident our model's recall is precise?

[133] Benjamin Moreau and Patricia Serrano-Alvarado. Ensuring license compliance in linked data with query relaxation. In **Transactions on Large-Scale Data-and Knowledge-Centered Systems XLIX**, pages 97–129. Springer, 2021.

[134] Benjamin Moreau, Patricia Serrano-Alvarado, Matthieu Perrin, and Emmanuel Desmontils. Modéliser la compatibilité entre les licences. In **31es Journées francophones d'Ingénierie des Connaissances (IC)**, 2020.

[135] Benjamin Moreau, Nicolas Terpolilli, and Patricia Serrano-Alvarado. A semi-automatic tool for linked data integration. In **18th International Semantic Web Conference (ISWC2019)**, 2019.

[136] Lorraine Morgan and Patrick Finnegan. Beyond free software: An exploration of the business value of strategic open source. **The Journal of Strategic Information Systems**, 23(3):226–238, 2014.

[137] Varish Mulwad, Tim Finin, and Anupam Joshi. Semantic message passing for generating linked data from tables. In **International Semantic Web Conference**, pages 363–378. Springer, 2013.

[138] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Linked open data-based explanations for transparent recommender systems. **International Journal of Human-Computer Studies**, 121:93–107, 2019.

[139] Fatemeh Nargesian, Erkang Zhu, Ken Q Pu, and Renée J Miller. Table union search on open data. **Proceedings of the VLDB Endowment**, 11(7):813–825, 2018.

[140] Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. Automated quality assessment of metadata across open data portals. **Journal of Data and Information Quality (JDIQ)**, 8(1):1–29, 2016.

[141] Axel-Cyrille Ngonga Ngomo, Sören Auer, Jens Lehmann, and Amrapali Zaveri. Introduction to linked data and its lifecycle on the web. **Reasoning Web. Reasoning on the Web in the Big Data Era: 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings 10**, pages 1–99, 2014.

[142] Phuc Nguyen, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. Mtab: matching tabular data to knowledge graph using probability models. **arXiv preprint arXiv:1910.00246**, 2019.

[143] Phuc Nguyen, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. Tabeano: table to knowledge graph entity annotation. **arXiv preprint arXiv:2010.01829**, 2020.

[144] Phuc Nguyen, Khai Nguyen, Ryutaro Ichise, and Hideaki Takeda. Embnum: Semantic labeling for numerical values with deep metric learning. In **Joint International Semantic Technology Conference**, pages 119–135. Springer, 2018.

[145] Phuc Nguyen, Ikuya Yamada, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. Mtab4wikidata at semtab 2020: Tabular data annotation with wikidata. **SemTab@ ISWC**, 2775:86–95, 2020.

[146] Phuc Nguyen, Ikuya Yamada, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. Semtab 2021: Tabular data annotation with mtab tool. In **SemTab@ ISWC**, pages 92–101, 2021.

[147] Phuc Nguyen, Ikuya Yamada, and Hideaki Takeda. Mtabes: Entity search with keyword search, fuzzy search, and entity popularities. In **35 (2021)**, pages 1N4IS1a02–1N4IS1a02. , 2021.

[148] Jakob Nielsen. **Designing web usability: The practice of simplicity**. New riders publishing, 1999.

[149] Nikola Nikolić, Goran Savić, Milan Segedinac, Stevan Gostojić, and Zora Konjović. Rdf stores performance test on servers with average specification. In **Proceedings of the 5th International Conference on Information Science and Technology (ICIST)**, pages 67–72, 2015.

[150] Javier Nogueras-Iso, Javier Lacasta, Manuel Antonio Ureña-Cámara, and Francisco Javier Ariza-López. Quality of metadata in open data portals. **IEEE Access**, 9:60364–60382, 2021.

[151] Femke Ongenae. Magic: Mining an augmented graph using ink, starting from a csv. 2021.

[152] OpenDataSoft. Create the best data experiences, 2011.

[153] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and software technology**, 64:1–18, 2015.

[154] Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, and Peter F Patel-Schneider. **Reasoning Web. Semantic Technologies for the Web of Data: 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures**, volume 6848. Springer Science & Business Media, 2011.

[155] Gianluca Quercini and Chantal Reynaud. Entity discovery and annotation in tables. In **Proceedings of the 16th International Conference on Extending Database Technology**, pages 693–704, 2013.

[156] Lara Quijano-Sánchez, Iván Cantador, María E Cortés-Cediel, and Olga Gil. Recommender systems for smart cities. **Information systems**, 92:101545, 2020.

[157] Filip Radulovic, María Poveda-Villalón, Daniel Vila-Suero, Víctor Rodríguez-Doncel, Raúl García-Castro, and Asunción Gómez-Pérez. Guidelines for linked data generation and publication: An example in building energy consumption. **Automation in Construction**, 57:178–187, 2015.

[158] Razlini Mohd Ramli. E-government implementation challenges in malaysia and south korea: a comparative study. **The Electronic Journal of Information Systems in Developing Countries**, 80(1):1–26, 2017.

[159] Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching html tables to dbpedia. In **Proceedings of the 5th international conference on web intelligence, mining and semantics**, pages 1–6, 2015.

[160] Dumitru Roman, Nikolay Nikolov, Antoine Putlier, Dina Sukhobok, Brian Elvesæter, Arne Berre, Xianglin Ye, Marin Dimitrov, Alex Simov, Momchill Zarev, et al. Datagraft: One-stop-shop for open data management. **Semantic Web**, 9(4):393–411, 2018.

[161] Erna Ruijer, Francoise Détienne, Michael Baker, Jonathan Groff, and Albert J Meijer. The politics of open government data: Understanding organizational responses to pressure for more transparency. **The American review of public administration**, 50(3):260–274, 2020.

[162] Alejandro Saez Martin, Arturo Haro De Rosario, and María Del Carmen Caba Pérez. An international analysis of the quality of open government data portals. **Social science computer review**, 34(3):298–311, 2016.

[163] Percy Salas, José Viterbo, Karin Breitman, and Marco Antonio Casanova. Stdtrip: Promoting the reuse of standard vocabularies in open government data. **Linking government data**, pages 113–133, 2011.

[164] Arup Sarkar, Ujjal Marjit, and Utpal Biswas. Linked data generation for the university data from legacy database. **International Journal of Web & Semantic Technology**, 2(3):21–31, 2011.

[165] Stuti Saxena and Irfan Muhammad. The impact of open government data on accountability and transparency. **Journal of Economic and Administrative Sciences**, 2018.

[166] Martin Schrepp, Jorg Thomaschewski, and Andreas Hinderks. Construction of a benchmark for the user experience questionnaire (ueq). 2017.

[167] Hanane Sebbaq and Nour-eddine El Faddouli. Moocs semantic interoperability: Towards unified and pedagogically enriched model for building a linked data repository. In **Digital Technologies and Applications: Proceedings of ICDTA 21, Fez, Morocco**, pages 621–631. Springer, 2021.

[168] Kumar Sharma, Ujjal Marjit, and Utpal Biswas. Automatically converting tabular data to rdf: An ontological approach. **Int J Web Semant Technol**, 2015.

[169] Arumoy Shome, Luis Cruz, and Arie Van Deursen. Data smells in public datasets. In **Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI**, pages 205–216, 2022.

[170] Barbara Slibar, Dijana Oreski, and Bozidar Klicek. Aspects of open data and illustrative quality metrics: literature review. **Economic and Social Development: Book of Proceedings**, pages 90–99, 2018.

[171] Blerina Spahiu, Riccardo Porrini, Matteo Palmonari, Anisa Rula, and Andrea Maurino. Abstat: ontology-driven linked data summaries with pattern minimalization. In **The Semantic Web: ESWC 2016 Satellite Events, Heraklion, Crete, Greece, May 29–June 2, 2016, Revised Selected Papers 13**, pages 381–395. Springer, 2016.

[172] Klaas-Jan Stol and Muhammad Ali Babar. Challenges in using open source software in product development: a review of the literature. In **Proceedings of the 3rd international workshop on emerging trends in free/libre/open source software research and development**, pages 17–22, 2010.

[173] Iryna Susha, Åke Grönlund, and Marijn Janssen. Driving factors of service innovation using open government data: An exploratory study of entrepreneurs in two countries. **Information polity**, 20(1):19–34, 2015.

[174] Iryna Susha, Åke Grönlund, and Marijn Janssen. Organizational measures to stimulate user engagement with open data. **Transforming Government: People, Process and Policy**, 2015.

[175] Efthimios Tambouris. Multidimensional open government data. **JeDEM-eJournal of eDemocracy and Open Government**, 8(3):1–11, 2016.

[176] Efthimios Tambouris, Evangelos Kalampokis, and Konstantinos Tarabanis. Visualizing linked open statistical data to support public administration. In **Proceedings of the 18th Annual International Conference on Digital Government Research**, pages 149–154, 2017.

[177] Jeremy Tandy, Ivan Herman, and Gregg Kellogg. Generating rdf from tabular data on the web, w3c recommendation 17 december 2015. **World Wide Web Consortium. Available at https://www. w3. org/TR/2015/REC-csv2rdf-20151217**, 2015.

[178] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig. ma: Live views on the web of data. In **Proceedings of the 19th international conference on World wide web**, pages 1301–1304, 2010.

[179] Jürgen Umbrich, Sebastian Neumaier, and Axel Polleres. Quality assessment and evolution of open data portals. In **2015 3rd international conference on future internet of things and cloud**, pages 404–411. IEEE, 2015.

[180] Anne Fleur Van Veenstra and Tijs Van Den Broek. **A community-driven open data lifecycle model based on literature and practice**. Springer, 2015.

[181] Gilles Vandewiele, Bram Steenwinckel, Pieter Bonte, Michael Weyns, Heiko Paulheim, Petar Ristoski, Filip De Turck, and Femke Ongenae. Walk extraction strategies for node embeddings with rdf2vec in knowledge graphs. **arXiv preprint arXiv:2009.04404**, 2020.

[182] Nataša Veljković, Sanja Bogdanović-Dinić, and Leonid Stoimenov. Benchmarking open government: An open data perspective. **Government Information Quarterly**, 31(2):278–290, 2014.

[183] Petros Venetis, Alon Y Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, and Gengxin Miao. Recovering semantics of tables on the web. 2011.

[184] Antonio Vetrò, Lorenzo Canova, Marco Torchiano, Camilo Orozco Minotas, Raimondo Iemma, and Federico Morando. Open data quality measurement framework: Definition and application to open government data. **Government Information Quarterly**, 33(2):325–337, 2016.

[185] Boris Villazon Terrazas, Andrés Tello, Amed Abei Leiva Mederos, Yusniel Hidalgo Delgado, Nelson Estrada, and Bin Xu. Methodological guidelines for publishing library data as linked data. 2017.

[186] Boris Villazón-Terrazas, Luis M Vilches-Blázquez, Oscar Corcho, and Asunción Gómez-Pérez. Methodological guidelines for publishing government linked data. **Linking government data**, pages 27–49, 2011.

[187] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk-a link discovery framework for the web of data. **Ldow**, 538:53, 2009.

[188] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. **Communications of the ACM**, 57(10):78–85, 2014.

[189] Zhu Wenjing, Xia Cuijuan, and Liu Wei. Analysis of silk linkage discovery framework. **Data Analysis and Knowledge Discovery**, (4):18–24, 2013.

[190] Jędrzej Wieczorkowski. Barriers to using open government data. In **Proceedings of the 2019 3rd International Conference on E-commerce, E-Business and E-Government**, pages 15–20, 2019.

[191] David Wood, Marsha Zaidman, Luke Ruth, and Michael Hausenblas. **Linked Data**. Manning publications co., 2014.

[192] Usha Yadav, Neelam Duhan, and Komal Kumar Bhatia. Dealing with pure new user cold-start problem in recommendation system based on linked open data and social network features. **Mobile Information Systems**, 2020:1–20, 2020.

[193] Tung-Mou Yang, Jin Lo, and Jing Shiang. To open or not to open? determinants of open government data. **Journal of Information Science**, 41(5):596–612, 2015.

[194] Tung-Mou Yang and Yi-Jung Wu. Examining the socio-technical determinants influencing government agencies' open data publication: A study in taiwan. **Government Information Quarterly**, 33(3):378–392, 2016.

[195] Ruojing Zhang, Marta Indulska, and Shazia Sadiq. Discovering data quality problems. **Business & Information Systems Engineering**, 61(5):575–593, 2019.

[196] Ziqi Zhang. Effective and efficient semantic table interpretation using tableminer+. **Semantic Web**, 8(6):921–957, 2017.

[197] Anneke Zuiderwijk and Marijn Janssen. Barriers and development directions for the publication and usage of open data: A socio-technical view. In **Open government**, pages 115–135. Springer, 2014.

[198] Anneke Zuiderwijk and Marijn Janssen. Open data policies, their implementation and impact: A framework for comparison. **Government information quarterly**, 31(1):17–29, 2014.

[199] Anneke Zuiderwijk and Marijn Janssen. Participation and data quality in open data use: Open data infrastructures evaluated. In **Proceedings of the 15th European Conference on e-Government**, pages 351–359, 2015.

[200] Anneke Zuiderwijk and Marijn Janssen. Towards decision support for disclosing data: Closed or open data? **Information Polity**, 20(2-3):103–117, 2015.

[201] Anneke Zuiderwijk, Marijn Janssen, Sunil Choenni, Ronald Meijer, and Roexsana Sheikh Alibaks. Socio-technical impediments of open data. **Electronic Journal of e-Government**, 10(2):pp156–172, 2012.

[202] Anneke Zuiderwijk, Keith Jeffery, and Marijn Janssen. The potential of metadata for linked open data and its value for users and publishers. **JeDEM-eJournal of eDemocracy and Open Government**, 4(2):222–244, 2012.