SRH Hochschule Heidelberg

Master Thesis

Recommendation of Annotations for optimal F.A.I.R.ification

Author:
Nikhil Gaikwad
Matriculation Num: 11010865

Supervisor:
Dr. Johan van Soest
Prof. Dr. Gerd Moeckel

Recommendation of Annotations for optimal F.A.I.R.ification

A thesis submitted in fulfillment of the requirements
for the degree of Masters of Applied Computer Science

in the

Department of Applied Computer Science

March 2020

Thesis Advisory Committee:

_____(Internal Supervisor)

Prof. Dr. Gerd Moeckel

SRH Hochschule Heidelberg, Germany

_____(External Supervisor)

Dr. Johan van Soest

Maastricht University, Netherlands

Declaration of Authorship

I, Nikhil Gaikwad, declare that this thesis titled, "Recommendation of Annotations for optimal F.A.I.R.ification" and the work presented in it are my own. I confirm that:

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

Signed:

Date:

SRH HOCHSCHULE HEIDELBERG

Abstract

Department of Applied Computer Science

Masters of Applied Computer Science

Recommendation of Annotations for optimal F.A.I.R.ification

by Nikhil Gaikwad

Researchers want and need to make their data, metadata and data management infrastructure F.A.I.R. (Findable, Accessible, Interoperable, Resusable). Using the semantic web stack, our work aims to build a framework which simplifies the FAIR conversion process. This framework converts the relational and CSV data into RDF triples using custom triple schema ontology built on top of table descriptions. Using the custom triple schema ontology, we demonstrate manual annotations using OWL-2 based reasoning rules for terminology binding. Furthermore, to ease the annotation process, we are trying to predict annotation using historical, lexical and contextual matching algorithms on the database labels against the concept labels from a given standardized ontology. To apply our methodology, we have built a web application for the annotation process. We perform terminology binding by reuse of annotations and semi-automated selection of prediction for annotations. This process indirectly targets "Interoperable" aspect of FAIR. Using this approach, we get moderate outcomes for lexical and contextual matches. Improving this quality of recommendations will be a significant part of future works.

Keywords: FAIR, annotations, linked data, Natural Language Processing.

Acknowledgements

I wish to express sincere gratitude to my supervisor, Dr Johan van Soest. He convincingly guided and encouraged me to be professional and do the right thing. His persistent guidance and advice helped me realize the goals of this thesis.

I would also like to thank Prof. Dr Gerd Moeckel, who has motivated me for pursuing this thesis. He has helped me coordinate my work and manage my timelines. With his passionate participation and valuable inputs, he helped me to enhance my thesis.

I am also indebted to Prof. Dr Andre Dekker for giving me various opportunities to explore domain knowledge. He supported me with his valuable comments and helped me to attend multiple distinct initiatives.

Also, I would like to thank my colleagues and friends at Maastro and at college for their continuous support and motivation for writing. They have helped me with their innovative perspectives and willingly supported me at every step.

Finally, I whole-heartedly appreciate my very profound gratitude to my family for great love, continuous encouragement throughout my years of study and providing me with endless support. It was impossible to complete this goal without them.

Thank you.

Contents

List of Figures

List of Tables

List of Abbreviations

AUC	**A**rea **U**nder **C**urve
CBOW	**C**ontinuous **B**ag **O**f **W**ords
DSS	**D**ecision **S**upport **S**ystems
FAIR	**F**indable, **A**ccessible, **I**nteroperable, **R**eusable,
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
IRI	**I**nternationalized **R**esource **I**dentifier
NLP	**N**atural **L**anguage **P**rocessing
OWL	**W**eb **O**ntology **L**anguage
R2RML	**R**DB **2**to **R**DF **M**apping **L**anguage
RDB	**R**elational **D**ata **B**ase
RDF	**R**source **D**escription **F**ramework
RDFS	**R**source **D**escription **F**ramework **S**chema
TP	**T**rue **P**ositive
TN	**T**rue **N**egative
URI	**U**niform **R**esource **I**dentifier

Dedicated to all who supported me in this journey...

Chapter 1

Introduction

1.1 Data is the new Oil

|| विद्या धनं सर्वधनप्रधानम् ||

"The wealth of knowledge is supreme among all forms of wealth."

In the era of digitalization, the age-old Sanskrit verse still seems to be apt not only for individuals but also for current market tech giants. Data is an imminent part of knowledge discovery and inferencing. Modern devices and internet generate ubiquitous data. Tech companies, since they own root hold technologies for digital devices, capture this raw data and use it for knowledge inferences which can benefit their user base as well business. Indirectly, data seems as valuable raw material for the Tech companies. Metaphorically as these companies process the raw data and derive valuable knowledge, they function as distilleries processing the new oil - Data. ("The world's most valuable resource is no longer oil, but data")

The abundance of data changes the nature of the analysis, which has a network effect. More the data, more scope to enhance predictions which attracts more users and in return, more data. A study in Poznan University estimates that emission of greenhouse gases would reduce by 40% to 60% if we shift to autonomous vehicles. As modern computing platforms can be mobile, we can gather the required petabytes of data and improvise machine learning solutions for self-driving. It will also help us to improve road safety and reduce unnecessary congestions (Igliński and Babiak, 2017). Similarly, farmers not only need to harvest the Crop but also the parameters affecting the quality of the harvest (e.g. climate, soil fertility). To make it a profitable business, crop schedule can be planned based on historical data of market demand, supply and cost.

1.2 Data Stewardship

With massive amounts of data comes the responsibility of maintaining it. This maintenance includes data availability, structural and schematic constraints and sourcing heterogeneous and homogeneous sources. For knowledge integration and reuse of the scholarly data, sound data management is a crucial factor. Data warehousing and curation are achievable on the individual, organizational levels. However, publicly generated datasets end up being orphans since no one owns them. They are maintained by researchers and open source communities until these data sets are their focus of interest. Later on, these data sets are not actively updated and maintained with required levels of detailing. There are many peer initiatives, agencies heavily investing for building an ecosystem which can manage such datasets for longer term. For facilitating knowledge discovery on legacy as well as incoming new data which can also be used later by downstream investigations, we do not have the necessary constituting principles defined.

Next to data stewardship for human curation and reuse of data, we need it for computational agents as well. Computational agents make the data retrieval and analysis on our behalf also need assistance for knowledge discovery. There are special-purpose, well-maintained repositories which strive continuously to curate the existing scholarly data and capture newly generated high-value datasets (For example GenBank (Benson et al., 2013), World Wide Protein Data Bank (Berman, Henrick, and Nakamura, 2003)). Even though these repositories are capable of serving Humans as well as machine agents, not all data types or datasets can be included or submitted in them (e.g. datasets derived from bench science). In response to this, there is a rise for general-purpose databases which accept a comprehensive range of datasets (For example Dataverse (Crosas, 2011) and Datahub (`https://datahub.io/`)). As general-purpose repositories do not harmonize deposited data or have any constraints for metadata quality, these repositories move away from centralization, making them more diverse in terms of standard for curation.

We need an ecosystem which ensures that quality and the impact of published dataset is the same what data publisher means and is easily accessible by the other researchers as well as machines. Defining the goals of data management, which can meet the constraints and requirements of data-intensive science can be an excellent utility for the data publisher (Wilkinson et al., 2016).

1.3 Clinical Data Science

Various types of data sets like clinical notes (semi-structured or free text), medical images (CT scan and radiomics) and Electronic Health records are generated during treatments. They are directly linked to the health status of the patient. Implementation of latest data science techniques on the clinical data is called as Clinical data science. Clinical data science covers the range from getting data, making a meaningful model, and evaluate and implement it in practice. (Kubben, 2019)

At the International Symposium on Biomedical Imaging (ISBI), students of Harvard medical school published a report for automated detection of metastatic breast cancer. They scored 0.925 for the area under the receiver operating curve (AUC). Area under the curve is the measure of separability. On the scale of 0 to 1, higher the value of AUC, higher the is the capability of model to classify. While with the same dataset, the pathologist who manually did this task scored 0.966. Later on, when both these results were combined, the algorithm scored AUC of 0.995, which is almost 85% improvement rectifying the human error. (Wang et al., 2016)

However, in actual scenarios, gaining access over clinical data is a hurdle. In the health care sector, because the patient data is privacy sensitive, researchers have to cross many hurdles to analyze this data. Due to the quantity and type, health care data generated is quite different. Opposed to the use case of recommending advertisements against the browsing data for users, we need a different approach in the health care sector. Advertisement data does not need to be validated before practise, while the accuracy of medical data is of utmost importance. Therefore, predicting treatment outcomes on the data which has many implicit meanings becomes a challenging task.

Also, health care data can be either be structured or unstructured form. For structured data health institutions and hospitals try to organize data in various data standards for both schema of data and terminology services to associate terms. However, due to the large variation in standards and formats, data is structured and stored differently by different vendors/organizations. Next to the data structure (syntax), there is also a significant gap in the underlying meaning (semantics) of data. This dilemma is a major hurdle affecting exchange for primary and secondary purposes of medical data (Eriksson and Helgesson, 2005). In this setting, the primary purpose is for the exchange of healthcare information to inform medical professionals in delivering medical care. The secondary purpose is to perform an analysis of the routinely collected

medical data. This analysis can be used for operational optimization within the institute, or additional scientific research about observations and diagnostics, treatments and their respective outcomes. This process of collecting and analyzing the data is an eternal part of clinical data science.

Semantics is a challenging task in both primary and secondary use of data. Some things which are obvious for the data provider, may not be noticeable for the data consumer. For example, a column with header "cT" might mean a CT scan or a clinical tumour staging category. In addition to this, there are variations in clinical data generated by hospitals due to geographical and cultural differences, or due to the way healthcare is organized in specific countries.

A huge amount of data resides in hospital-wide information systems, departmental information systems, or even local user/group information systems. These systems range from well-tested and maintained software packages to local spreadsheets of additional data that cannot be stored in the existing information systems. Since a single hospital does not have enough data to make reliable results, this data needs to be shared. However, due to the lack of syntactic and semantic interoperability, the data from one institution is not straightforward to understand by the other institution. There are syntactic and semantic standards (e.g. FHIR, OMAP) standards in market, but they do not match the plurality of data formats within the hospital information systems (Choudhury et al., 2019). Therefore, it cannot be leveraged for improving patient care.

While precision is of paramount importance in health care, for precise analytics the basic rule of thumb in statistical regression is to have more amount of predictive variables in your dataset (eg.One in ten rule (Peduzzi et al., 1996)). This makes estimation easier while reducing the risk of overfitting (*Regression modelling strategies for improved prognostic prediction - Harrell - 1984 - Statistics in Medicine - Wiley Online Library*). Furthermore, the intricacies of Personalised healthcare (Ankolekar et al., 2018) for dosage calculations, medications and treatment planning are data driven. Also there are "Rapid Learning Systems" which focus on integration of real-time clinical data (Abernethy et al., 2010) for process of scientific discovery. Hence, every bit of data becomes important.

1.4 About Organisation

The amount of people getting affected by cancer is significantly increasing each year. Data science brings a promise of early detection and survival prediction in cancer treatment. The idea of personalized treatment for each individual for achieving the best outcome is very innovative and interesting as every individual can have different features like age, weight, etc. which would matter when predicting the outcome of a treatment. Also, personalized treatments are necessary to patients as size, shape, and type of tumour vary. Furthermore, along with apparent features, consideration of non-apparent features like genomics are a must for such treatments.

Maastro Clinic - located in Maastricht, The Netherlands - is the only radiotherapy center in Limburg region which treats national patients with a wide variety of cancer treatments. A Company of around 285 employees offers traditional radiotherapy and Proton therapy the latest in tumour treatment. Maastro aims to provide treatments with the highest possibility of a cure with the least side effects. Due to the patient-centric approach at Maastro, the patient plays an essential role in planning the trajectory of treatment. Maastro treats around 4500 treatments on patients annually (*Over Maastro*).

Next to the clinical patient care and treatment, Maastro is investing in several research divisions, namely, clinical, physics, lab, and data science research. Maastro Lab aims to understand the cell biology of cancer and it's treatments, to improve existing and investigate into upcoming treatments options. The Maastro Clinical Research team carries out various clinical trials. These trials help improving the efficiency of treatments and reduce the physical and mental side effects. The physics research team mainly focuses on radiotherapy physics for medical imaging, brachytherapy dose calculation, and particle beam radiotherapy.

The Data Science Research team mainly focuses on :

- Decision support for individualized radiotherapy.
- Development of prediction models for cancer prediction.
- Build a global data-sharing infrastructure.

Decision support systems (DSS) aim to provide context in the medical information, and can be aimed for patients and/or physicians. As the medical knowledge and perspective for both user groups is different, DSSs are tailored for these user groups,

respectively. For instance, Radiotherapy can treat a patient's tumour; however, the higher the radiation dose to the tumour, the higher the dose to surrounding tissues and subsequently the chance on side effects of radiotherapy. Different forms of radiotherapy (e.g. dose, or treatment delivery method) can reduce these chances of side effects; however, it could also reduce the effectiveness of the treatment. Treatment planning is an intricate procedure since it involves various dependencies like patient characteristics, treatment efficacy, and side-effects. Taking all these information elements into consideration when deciding which treatment is best for a specific patient is a task beyond the computational power of humans. Therefore we need decision support systems to inform patients and to decide which treatment option they prefer while accepting risks in (adverse) treatment effects (Ankolekar et al., 2018). The cancer prediction team uses various types of medical images as inputs for machine learning models. In several cancer treatment procedures, these models are used as a helping hand.

When building these prediction models to support clinical decisions, privacy and semantics are the main challenges. The main requirement of privacy preservation is to anonymize the identity of patients. The passing of data becomes an even more challenging task when it comes to sharing across country borders. So instead of feeding data to algorithms, containerized algorithms can be sent to hospitals collecting data, with the request to execute this algorithm, and only send back the output. The Clinical Data Science group uses this alternative method to data analysis in the concept of the Personal Health Train (Choudhury et al., 2019) (Deist et al., 2017) (Deist et al., 2020).

Hospitals have a massive amount of data in various structures and formats. To have more reliable prognostication results by the research community, this kind of data must be easily exchangeable while retaining its underlying core meaning. To promote open science and better (meta)data, Maastro is heavily investing in FAIR (meta)data descriptions. This makes Maastro a robust partner for collaboration. Maastro is hosting data endpoints using Semantic Web technologies. This makes it easier for syntax agreement with other institutes. Consumption of this data for machine learning and analytics becomes easier inter- as well as intra-organization.

1.5 Objective and Research Questions

1.5.1 Objectives

Hospitals' databases, schemas and data sources are not static. They are constantly adapting to new medical terms, and are different across hospitals. Compared to the traditional column row approach, adapting to these challenges becomes easier using technologies supporting semantics and linked data. Although these semantic web technologies are perceived as the way forward, implementation is an elaborate task.

Adding the semantics, which is a manual and labour intensive work, requires a combination of competences from different specializations ranging from the IT department to the medical expert. Considering the skill sets generally available at Hospitals, it is time consuming and resource intensive to reap the benefits of FAIR data. Hence, our aim is to provide a framework and software tooling where users can easily make their data (more) FAIR, while reducing the amount of information and software engineering skills needed to get started.

1.5.2 Research Questions

1. What is the optimal way of converting relational data into RDF?
2. What is the most competent practice? to materialize the data OR provide a virtual endpoint mimicking the RDF data at run-time?
3. What will be an optimal methodology which will not over-fit for building a recommendation system on annotated data for triplication?
4. Is the annotation of data sources faster for hospitals to make their data FAIR, compared to existing methods?

1.6 Outline

The Thesis is structured mainly in 4 parts. In chapter 2 we evaluate various methods used for making data FAIR. Chapter 3 describes different algorithms used for predicting the column labels , including the architectural explanation for storing these annotation. In Chapter 4 we discuss the outcomes and evaluate the results for the methods we described in Chapter 3. The Web application development and embedding the

algorithms to predict labels in this application is discussed in Chapter 5. Chapter 6 and 7 discusses our main findings in relation to the literature and provides a general conclusion.

Chapter 2

Background

2.1 Radiotherapy and Clinical Data

Uncontrolled cell division which invades local and surrounding tissues and which can spread throughout the human body is known as cancer. It was the second leading cause of death worldwide in 2012, with 8 million deaths. It is estimated that this rate will rise to 23 million cases per year in the next decade (Bray et al., 2018). All forms of cancer share following 6 characteristic capabilities on the cellular level:

1. stimulating cell division
2. de-activating cell division inhibitors
3. evading apoptosis (cell death)
4. self-sufficiency in growth signals
5. insensitive to growth-inhibiting signals
6. invasion in other tissues and metastases

With a better understanding of these six Hallmarks defined by Hanahan and Weinberg (Hanahan and Weinberg, 2000) and with the help of recent advancements in early detection technologies and treatment modalities, the majority of cancers have become more curable.

Radiation is a physical agent, which destroy cancer cells by depositing high physical energy. Ionizing radiation (ions formed from electrically charged particles) when passing through the tissues and reaches the affected cancerous cell causes genetic changes resulting in cancer cell death. Damage to DNA (deoxyribonucleic acid) of cancerous cells blocks their ability to divide and propagate further. Although healthy cells are

also damaged, which are in the trajectory of radiation, radiotherapy aims to maximize the absorption by abnormal cancer cell with minimal exposure to healthy cells. Cancer cells are not as efficient as normal cells in restoring the damage caused by radiation treatment, which results in differential cancer cell killing (Baskar et al., 2012).

The study which encompasses all factors of research in treating cancer using radiation is known as radiation oncology. Numerous findings in molecular and cellular radiation biology, radiation physics and technology are published in in this field (*Radiation Oncology*).

Controlled delivery of radiation beams for delivering precise tumoricidal dose in the irradiated region is the basic principle in Radiation therapy. Modern radiation treatment systems rely heavily on computerized information for tailoring and meeting the setup parameter objectives. Delineating the tumour contours using the CT scan images, calculating the best treatment plan and even planning actual treatment parameter for linear accelerators to adjust beam direction, intensity, size and duration of radiation requires substantial computational power with human moderation. Hence we can state that radiation oncology is IT-driven task. Since the majority of medical systems are digitalized (like Electronic Health Record System), it becomes quite easy to bring in meaningful insights utilizing the aforementioned clinical data for better specificity (Kalet and Austin-Seymour, 1997).

2.2 Need for Interoperability

Good data stewardship is rapidly becoming an essential part of data science. In the research field, along with the vast amount of data collected, comes the task of maintaining and making it reusable by individuals. This data should also be findable so that machines could consume it for analytics. Proper data management is not just a goal but can help to make new inferences and discoveries on which the community can do further research by utilizing this published data. Unfortunately, even though many peer initiatives are trying to solve this urgent need, they are not able to reap maximum benefits out of these ecosystems (Wilkinson et al., 2016).

2.3 FAIR initiative

In 2014, considering this need, a workshop named "Jointly Designing a Data Fairport" was held in Leiden, Netherlands. A diverse group of researchers and private stakeholders gathered for meeting the goal of data discovery and semantic interoperability. A notion emerged that the community should have a minimal set of principles which help discover, access, appropriately reuse and cite enormous amounts of data generated by modern data-intensive science.

A draft was formulated with a set of foundational principles which stated that all the research objects should be:

- Findable
- Accessible
- Interoperable
- Reusable

(referred as FAIR guiding principles) both for machines and for people. Subsequently, to elaborate and fine-tune the principles, a FAIR group was formed by group of members called "FORCE 11".

The main aim of these principles is to increase machine-actionability with none or minimal human intervention. These principles make sure computers can "understand" the data, and can use it accordingly (e.g. for computational purposes). Below are the 15 principles which help to describe your data F.A.I.R:

2.3.1 Findable

Machine-readable metadata is essential for the automatic discovery of dataset and services. Find-ability has the following sub-principles:

- **F1.** (Meta)data are assigned a globally unique and persistent identifier (Wilkinson et al., 2016).
- **F2.** Data are described with rich metadata (defined by R1 below) (Wilkinson et al., 2016).
- **F3.** Metadata clearly and explicitly include the identifier of the data they describe (Wilkinson et al., 2016).

- **F4.** (Meta)data are registered or indexed in a search-able resource (Wilkinson et al., 2016).

2.3.2 Accessible

Once the user finds the required data, she/he needs to know how they can be accessed, possibly including authentication and authorization.

- **A1**. (Meta)data are retrievable by their identifier using a standardized communications protocol (Wilkinson et al., 2016).
  - **A1.1** The protocol is open, free, and universally implementable (Wilkinson et al., 2016).
  - **A1.2** The protocol allows for an authentication and authorization procedure, where necessary (Wilkinson et al., 2016).
- **A2.** Metadata are accessible, even when the data are no longer available (Wilkinson et al., 2016).

2.3.3 Interoperable

The data usually needs to be integrated with other data. In addition, the data needs to inter-operate with applications or workflows for analysis, storage, and processing.

- I1. (Meta)data uses a formal, accessible, shared, and broadly applicable language for knowledge representation (Wilkinson et al., 2016).
- I2. (Meta)data uses vocabulary that follows FAIR principles (Wilkinson et al., 2016).
- I3. (Meta)data includes qualified references to other (meta)data (Wilkinson et al., 2016).

The obligation of interoperability has been a significant domain of research for the medical community. For instance, SNOMED-CT, a health care terminology service, is in development for the past forty years. SNOMED-CT caters to the majority of the concepts and terms which a health care system would require. Irrespective of its long existence and continuous evolution, globally, many health care systems rely on this service (Cornet and Keizer, 2008).

2.3.4 Reusable

Reusability has been one of the significant hurdles within the scientific community. The reproducibility of data reduces due to the lack of rich meta-data. The ultimate goal of FAIR is to better enable the reuse of data. To achieve this, metadata and data should be well-described so that they can be replicated and/or combined in different settings.

- **R1.** Meta(data) are richly described with a plurality of accurate and relevant attributes (Wilkinson et al., 2016).
  - **R1.1.** (Meta)data are released with a clear and accessible data usage license (Wilkinson et al., 2016).
  - **R1.2.** (Meta)data are associated with detailed provenance (Wilkinson et al., 2016).
  - **R1.3.** (Meta)data meet domain-relevant community standards (Wilkinson et al., 2016).

Using these 15 principles, we can convert three types of entities into a FAIR data set

- data
- metadata
- infrastructure

Funding agencies, such as the European Union's Horizon 2020 and the Dutch Research Council (NWO) (*Open (FAIR) data*), require researchers to put effort in making their data FAIR (which includes the standardization of their data) and require them to include their methodology in a Data Management Plan (DMP) when researchers apply for a grant. The rationale is that FAIR Data accelerates innovation due to primary use *and* secondary (re)use of data. For example, to reuse clinical data to develop and make decisions for opting proton therapy centers use the previous use case and scenarios.

2.4 FAIRification Process

The process of making data FAIR (FAIRification) consists of seven main steps (*FAIR-ification Process*), as shown in Figure 2.1

1. **Retrieve non-FAIR data:** The first step is to gain access to the data set to convert.

2. **Analyze the retrieved data:** For converting the non-FAIR data, we must analyze the structure and the contents. Here we try to determine the concepts, relationships, and distribution. For instance, for a relational dataset, we can determine the structure, links, and cardinality from the data definitions of the schema.

3. **Define the semantic model:** After the identification of required concepts and terms, we should get a set of entities that unambiguously define them. A vocabulary is a set that includes all the concepts related to a particular domain. A terminology is a structured vocabulary including the definitions of the concepts, while an Ontology is a set of relations between these concepts in a hierarchical format (Keizer, Abu-Hanna, and Zwetsloot-Schonk, 2000) . The concepts and relationships in these ontology files help us define the Semantic model, which describes the entities and make them machine-readable. This ontology describes the schema "contract" between data provider and consumer (Keizer, Abu-Hanna, and Zwetsloot-Schonk, 2000).

4. **Make data linkable:** linking between the dataset and opted Semantic model (Ontology) is currently done by using Semantic Web and Linked Data technologies (described below). This linking facilitates interoperability, reuse, and integration of data with other datasets. However, it might not be possible to link every type of data for e.g. in case of video and audio data, annotation requires special infrastructure.

5. **Assign license:** To highlight the importance, license association is a separate process in FAIR data descriptions, even though it is a part of meta-data.

6. **Define metadata for the dataset:** Defining the rich-meta is one of the pillars upon which FAIR principles rely.

7. **Deploy FAIR data resource:** Here, we publish the FAIR metadata, optionally with the actual data, and license so that indexed search engines can display them in search results. Irrespective of this, we can set up an authentication process on top of this if required for limiting the access. (*FAIRification Process*)

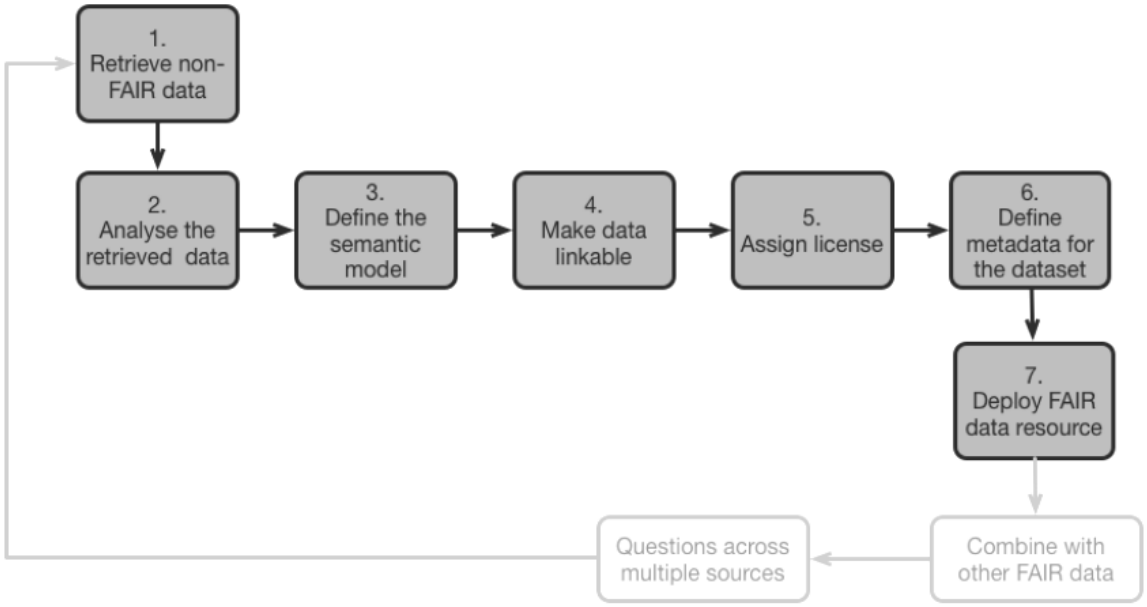

FIGURE 2.1: Overview of FAIRification process.

2.5 Semantic Web

The Semantic Web is a technology stack that enables computers to make meaningful interpretations similar to the way humans process information to achieve their goals. It is an extension of the World Wide Web through standards by the World Wide Web Consortium (W3C). The term "Semantic Web" was coined by Tim Berners-Lee, director of the World Wide Web Consortium ("W3C") and the pioneer of the World Wide Web. He defined the Semantic Web as "a web of data that can be processed directly and indirectly by machines" (*The next web*). Promoting the inclusion of semantic content in web pages, the Semantic Web aims to convert current unstructured and semi-structured web documents (HTML) into a "web of data". While in FAIR approach, we could go from FAIR data to structured or unstructured web.

Many technologies like IRI, URI and RDF proposed by W3c were already existing before being included in the Semantic stack. They were already used in a context where information/data sharing was necessary for a well defined limited domain (webpages).

"The Semantic Web is a web of connections between different forms of data that allow a machine to do something it wasn't able to do directly."

(*Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its inventor*)

2.5.1 Semantic Web Stack

A Semantic Web stack is a architecture of technologies and standards used to implement the Semantic Web. It includes various hypertext, markup and query languages indicating it as an extension of classic hypertext web.

To uniquely identify the Semantic Web resources, the bottom layer uses International Resource Identifiers (IRIs), a generalization of URI. This gives the top layer resources capabilities to extend the intended data. Documents with semi-structured data can be handled using XML. They enable the Semantic Web to associate meaning to the semi-structured data.

The middle layer technology serves the purpose of semantics. The Resource Description Framework (RDF) represents the semantic resource in a graph structure. Furthermore, Resource Description Framework Schema (RDFS) gives the basic vocabulary to make classes and hierarchy structure. Web ontology language (OWL) gives advanced capabilities for reasoning rules, restrictions of values, cardinality and characteristics of properties such as transitivity (for example, if Heidelberg is in Baden-Württemberg and Baden-Württemberg lies in Germany, then Heidelberg is in Germany) and symmetry (e.g. Freddie Mercury and Farrokh Bulsara are same person, lead vocalist of the rock band Queen.)(*OWL Web Ontology Language Reference*).

The top layer for the Semantic Web is not standardized yet or it just has ideas to be realized. The Top layer is designed to serve the security and user interface to access semantic resources. For serving the security needs existing architectures like HTTP/HTTPS can be borrowed. However, more advanced security depends on the use case. Primarily, this layer tries to address security, authentication and trust.

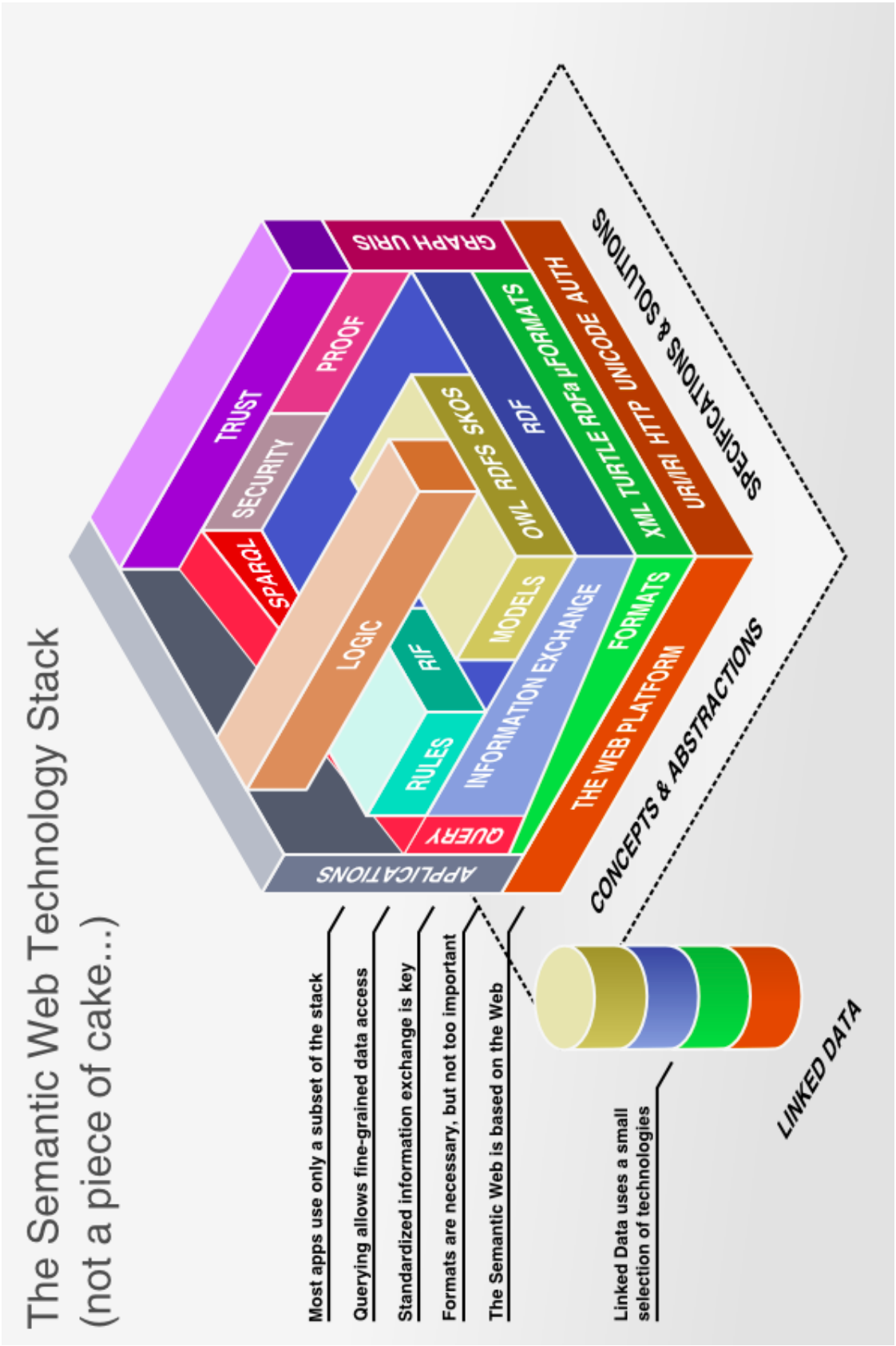

FIGURE 2.2: Technologies in Semantic Web stack

2.5.2 Semantic Web serving FAIR principles

World-wide-web has shown it's flexible enough to scale, and that the adoption rate as an online exchange or request/response mechanism is ubiquitous. Semantic Web brings all these advantages to structured data on the web, which can be enriched by using the F.A.I.R principles. It is a major proponent for sharing data and not just a web document. URI is a string of characters that help to identify an entity universally. Using this identifier we can access the designated resource. In this way, URI is serving Findability and Accessibility.

RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a "triple")

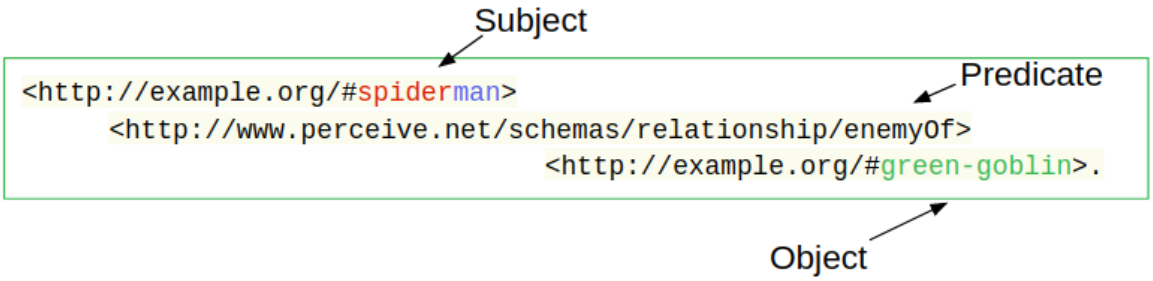

FIGURE 2.3: Basics of triplet formation

This forms as a meta-structure RDF. It can be extended to any kind of data as well as meta-data. Using this concept, combined as a graph structure, any kind of data structure can be modelled, and data can be easily merged (as there is only one meta-structure with only 3 columns). RDF and RDFS standardize the data as well as the metadata. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed. Associating classes and subclasses from the opted schema standardizes meta-data. Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications. From the opted ontology we can associate the transitive classes to the triples of converted input data. Therefore RDF and RDFS, and the accompanying query language SPARQL take care of Accessibility and Interoperability.

Linked data is a method of publishing structured data on the web so that it can be interlinked and queried using a formal semantic query language (Berners-Lee, 2000). Linked Open Data is linked data that is covered under an open-source license so that

it can be reused by third parties. Linked data builds upon standard Web technologies, e.g., HTTP, RDF (Resource Description Framework) and URIs (Uniform resource identifiers). Since these data-sharing protocols come under open source licenses they become easily Reusable.

Since the semantic web is serving the majority of the FAIR principles, it seems to be an optimal choice. With further research, we can expect the technological stack to meet the shortcomings.

2.6 RDF

RDF which stands for ”Resource Description Framework” is a standard model for data interchange on the web. It is a data model designed to represent both the data structure and terminology (ontology), while also being able to represent the actual data itself. The collection of RDF statements represents a labelled directed multigraph. Therefore theoretically RDF makes a better data model for storing knowledge graphs compared to relational databases.

RDF uses a linking structure for establishing relationships between classes and/or instances, which are defined by URIs. Its concept is to make statements about the resource in the form of subject-predicate-object. This linking structure forms a directed, labelled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations.

For example, we can break down the statement ”Nikhil works at Maastro” can be represented in RDF triple form as follows:

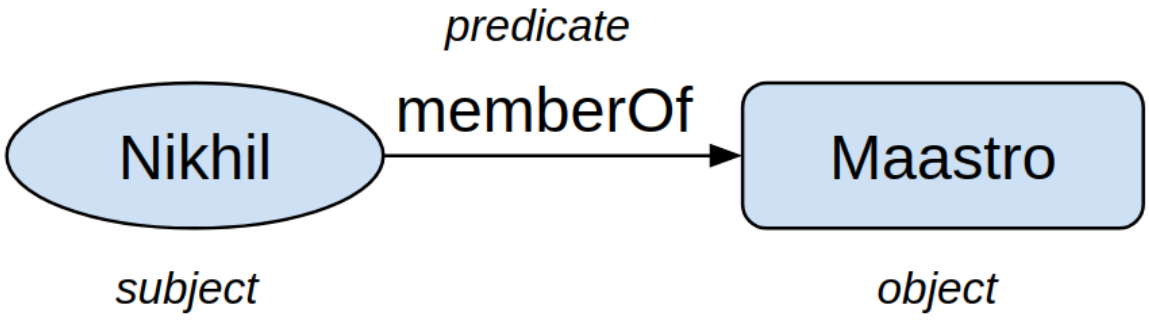

Figure 2.4: Concept of Triples

Later to uniquely identify them we include URIs in this.

```
<http://www.example.org/#Nikhil>
            <https://www.w3.org/ns/org#memberOf>
                                                <http://www.maastro.nl>.
```

Listing 2.1: Triples Example w/o prefix

In this structure, we stored the converted RDF triples in the triple store. But works to make this triple more human-readable, we can prefix the URI and add the necessary extension.

```
@prefix eg: <http://www.example.org/#>
@prefix org: <https://www.w3.org/ns/org#>

eg:Nikhil   org:memberOf    <http://www.maastro.nl>.
```

Listing 2.2: Triples Example with prefix

To model the RDF data in a structured way we need a library that holds the syntax. We need a mechanism that describes groups of related resources and the relationships between them. Also in the above example we can substitute the organization URL with a literal or a label. For all these requirements, we define classes and objects in RDF schema using RDF itself.

2.7 RDFS (RDF schema)

Structuring of classes and objects in RDF schema is very similar to structures in object-oriented programming. RDF schema is a base set of properties used to construct elaborate vocabulary, also known as an Ontology. The RDF schema can potentially be stored in a triple store, as this whole set is written in RDF. It is a semantic extension of RDF which helps define relationships between groups of similar related resources. Further these defined resources can be used to describe the characteristics of other resources. RDF schema sets the domain and range (the direction) of properties and relates the RDF classes and properties into taxonomies using the RDFS vocabulary. RDF has the following constructs, and are explained below :

- Classes
- Associative Properties

- Utility Properties

2.7.1 Classes

All resources identified by URIs, can be classified as a class. Resources, when further divided into groups, form a class. Property rdf:type is used to define a resource as a Class. In summary, class is a meaningful way of grouping resources (*What Are Classes And Individuals?*).

For example, Herbivorous can be defined as a class which includes all the living beings which eat plants. Neoplasm can be defined as a class which includes all types of cancer. Automobiles can be defined as a class which includes all types of mode of transport which use roads.

RDF differentiates class and instance of the class. A class and an instance of a class may not be the same. Also, two different classes may contain the same set of instances. For example instance, age can be used to define the property of class patient as well as class tumour (*RDF Schema RDFS - Introduction to ontologies and semantic web - tutorial*). To define the hierarchy of classes, we use Subclasses. Organised in the tree structure, the most general classes are at the top while more specific classes are at the bottom.

2.7.2 Associative Property

The relation between subject and object is called Property i.e, predicate. All properties may have a defined domain and range. The domain of a property states that any resource that has a given property is an instance of the class. The range of a property states that the values of a given property is/are instance(s) of the class. If multiple classes are defined as the domain and range then the intersection of these classes is used. Note: Even though predicates are sub-properties of a specific type, during definition they are in place of subject, but in usage they are predicate.

2.7.3 Utility Property

Utility Property indicates a resource that might provide additional information about the subject resource. It is used to describe an RDF resource. For example :

```
1 @prefix eg: <http://www.example.org/#>
2 @prefix foaf: <http://xmlns.com/foaf/0.1/>
```

```
3  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5  @prefix ncit: <http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#>
6
7  eg:Nikhil rdf:type foaf:Person ; #associate URI to class foaf:Person.
8            foaf:name "Nikhil" . #associate URI to class with string.
9
10 :Maastro rdf:type ncit:C19326 ; #associate URI to class Hospital.
11          rdf:resource   <http://www.maastro.nl> . #associate URI to
       Hospital URL.
12
13 :worksAtHospital rdf:type rdf:predicate ; # defining tag :
       worksAtHospital
14                  rdfs:range foaf:Person ; # defining range
15                  rdfs:domain ncit:C19326 . # defining domain
16
17 eg:Nikhil :worksAtHospital :Maastro . # demonstrating use predicate.
```

Listing 2.3: Triples Example with prefix

Here in the example, we illustrate the usage of RDF schema literals subClassOf, range, and domain. In lines 7 and 8 we declare the necessary subjects to define the class they belong to and the necessary additional property for `eg:Nikhil`. Similarly we define Hospital and, its url on line 10 and 11. Line 13 defines a predicate "`worksAtHospital`" and specifies the domain and range it caters to. Then we associate the subjects, `eg:Nikhil` and `:Maastro`, with predicate `worksAtHospital` on line 16.

2.8 SPARQL

Simple Protocol and RDF query language (SPARQL) is a query language for RDF is designed to meet the use cases and requirements (*RDF Schema RDFS - Introduction to ontologies and semantic web - tutorial*) identified by the RDF Data Access Working Group. This query language can be used to query across diverse data groups. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

SPARQL allows querying loosely bound key-value pairs that follow the RDF format. Therefore SPARQL treats the whole database in the ”subject - predicate - object ” structure. This is analogous to NoSQL graph databases.

```
@prefix ncit: <http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#>

SELECT ?patient
WHERE {
    ?patient a ncit:C16960.
}
```

Listing 2.4: Triples Example with prefix

Listing 2.4 illustrates a simple select query in which we get all the subjects who have rdfs:type as rdfs:Patient. Identifier always begins with a “?”. We also have capabilities like filtering, aggregating, insertion,selection,deletion, and construct operations similar to SQL.

2.9 OWL Ontology

The philosophical study of a domain which applies neutrally for being everything which seems to be real is an Ontology.(*Ontology | metaphysics | Britannica*) It defines the terms used in this domain and the inter-relation between them to represent the knowledge. The ontology describes Artifacts and their different degree of structures. They can be of type taxonomies (Yahoo hierarchy) or metadata schemes (Dublin core) or logical theories. Particularly Users, databases or applications and machines use ontology when there is a requirement of precise domain knowledge sharing. It enable communities of a particular interest to collect and standardize concepts for reuse (*OWL Web Ontology Language Use Cases and Requirements*).

The semantic web aims to achieve integration of explicit meaning to web data, making it easier for the machine to act on it. The Web Ontology Language (OWL) by W3C is a Semantic Web language designed to the consistent, accurate and meaningful distinction of terms. It is part of the W3C’s Semantic Web technology stack. They define descriptions of the following concepts:

- Classes (general things) of a particular domain
- The relationships between things

- The properties (or attributes) of things

This precise description enables software agents to exploit the knowledge expressed for search, retrieve, get explicit as well as implicit meaning out. OWL makes an open-world assumption. The resources of a single ontology can be extended across multiple ontologies. They can be published over the world wide web and referred. It also helps to gather the scattered knowledge.

Although XML, DTD and XML schemas are used for data exchanges, due to lack of semantics machine agents cannot get the context of terms. Same terms may have a different context, or in contrast, different terms might be describing the same entity. RDF and RDF schema allow us to create classes, subclasses, properties and sub properties, domain and range of these properties. By interpretations and rules it indirectly becomes a simple ontology. However to achieve interoprations on automous schemas we need richer semantics. (*OWL Web Ontology Language Use Cases and Requirements*)

2.9.1 Namespaces

A typical OWL file begins with a namespace declaration. Under `rdf:RDF` tag we specify all vocabularies which will be used in defining the ontology. It provides us with unambiguous identifiers for other ontologies and represents the rules in a much clear and human-readable form.

```
1 <rdf:RDF
2     xmlns     ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
3     xmlns:vin ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
4     xml:base  ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
5     xmlns:food="http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#"
6     xmlns:owl ="http://www.w3.org/2002/07/owl#"
7     xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
8     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

Listing 2.5: Namespace example in OWL

The first two declarations identify the namespace, declaring it as default. The line 3 adds prefix :vin for current ontology. While line 5 identifies the prefix for food ontology with prefix :food and URI. And as a conventional ontology file, we include :owl prefix to introduce OWL vocabulary on line 6. Since OWL depends on RDF and RDFS schema, we also include them in declarations on line 7 and 8. (*OWL Web Ontology Language Guide*)

2.9.2 Simple Named Class

The most basic domain corresponds to a base class forming the root of the taxonomy tree. Every instance of OWL world is of class `owl:Thing`. By declaring a named class, we define Domain-specific root classes.

```
<owl:Class rdf:ID="Winery"/>
<owl:Class rdf:ID="Region"/>
<owl:Class rdf:ID="ConsumableThing"/>
```

Listing 2.6: Simple Named Class example in OWL

Despite familiar English labels, we still don't know anything about above declare classes apart from their existence. This declaration helps us to use Region instead of the whole URL in the current OWL file.

Elemental tag for a taxonomic constructor is `rdfs:subclassOf`. Used for creating specific classes compared to generic ones, this constructor is transitive. If X is a subclass of Y, and Y is a subclass of Z, then Z is also a subclass of X. (*OWL Web Ontology Language Guide*)

```
<owl:Class rdf:ID="PotableLiquid">
  <rdfs:subClassOf rdf:resource="#ConsumableThing" />
  ...
</owl:Class>
```

Listing 2.7: Sub Class example in OWL

Here we define `ConsumableThing` as a subclass of `PotableLiquid`

2.9.3 Properties

Properties are used for defining general attributes of the class and state the specific facts about instances created. It is a binary relationship which has two properties :

1. Data type properties for declaring the relationship between the class and datatypes(RDF literal, XML schema datatypes)

2. Object properties for declaring the relation between the instance of two classes.

For having control over associations properties relations, we define range and domain. A property can also be defined as sub-property of a generic class.

Data Type Property

```
<owl:Class rdf:ID="VintageYear" />

<owl:DatatypeProperty rdf:ID="yearValue">
  <rdfs:domain rdf:resource="#VintageYear" />
  <rdfs:range  rdf:resource="&xsd;positiveInteger"/>
</owl:DatatypeProperty>
```

Listing 2.8: Data type Property

In listing 2.8 we define a data type property `yearValue` which is a property of resource VintageYear and expects a positive integer.

Object Property

```
<owl:ObjectProperty rdf:ID="madeFromGrape">
  <rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="#WineGrape"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="course">
  <rdfs:domain rdf:resource="#Meal" />
  <rdfs:range rdf:resource="#MealCourse" />
</owl:ObjectProperty>
```

Listing 2.9: Object Property

In listing 2.9 we define object property `madeFromGroup` where in expects a Wine resource as domain and WineGrape as range.

Property Characteristics

To associate the inference rules and embed them into triples we have following Characteristics of properties:

TransitiveProperty If for property P,
P(A,B) and P(A,C)
implies
B == C

```
<owl:ObjectProperty rdf:ID="locatedIn">
  <rdf:type rdf:resource="&owl;TransitiveProperty" />
```

```
  <rdfs:domain rdf:resource="&owl;Thing" />
  <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>

<Region rdf:ID="HeidelbergRegion">
  <locatedIn rdf:resource="#BadenWürttembergRegion" />
</Region>

<Region rdf:ID="#BadenWürttembergRegion">
  <locatedIn rdf:resource="#Deutschland" />
</Region>
```

Listing 2.10: Object Property

Here in listing 2.10 since Heidelberg Region is located in Baden Württemberg Region and Baden Württemberg is located in Deutschland, Heidelberg is located in Deutschland.

SymmetricProperty For a property tagged as Symmetric
If P(A,B) = P(B,A)

```
<owl:ObjectProperty rdf:ID="adjacentRegion">
  <rdf:type rdf:resource="&owl;SymmetricProperty" />
  <rdfs:domain rdf:resource="#Region" />
  <rdfs:range rdf:resource="#Region" />
</owl:ObjectProperty>

<Region rdf:ID="BavariaRegion">
  <locatedIn rdf:resource="#Deutschland" />
  <adjacentRegion rdf:resource="#BadenWürttembergRegion" />
</Region>
```

Listing 2.11: Object Property

Listing 2.11 utilises Symmetric property for Bavarian region being adjacent to Baden Württemberg, which implies both are in Deutschland (*OWL Web Ontology Language Guide*).

2.9.4 Property Restrictions

To assign specific context and limit the range of property we use Property restrictions. The tag `owl:OnProperty` under owl:restriction tag indicates the property is restricted.

Tags `allValuesFrom` and `someValuesFrom`, help limiting the class type being associated in property.

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid" />
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker" />
      <owl:allValuesFrom rdf:resource="#Winery" />
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```

LISTING 2.12: Object Property

Listing 2.12 allows us to limit the property `hasMaker` of wine class to have values from `Winery` only and not from cheese factory.

Similary if we replace `allValuesFrom` tag with `someValuesFrom`, it implies atleast one property of `hasMaker` must point to winery (*OWL Web Ontology Language Guide*).

2.10 R2RML conversion script

R2RML is a scripting language used for converting relational data into RDF data. Through this, we view the relational data in triple structure with the target vocabulary of the author's choice. R2RML is also written in the RDF form, hence making it interpretable by the same technologies (e.g. SPARQL queries, or RDF interpreters) as the data it produces.

The output is an RDF dataset that uses predicates and types from the target vocabulary (incorporated in the R2RML description triples). The mapping is conceptual; R2RML processors are free to materialize the output data (in RDF data files), or to offer virtual access through the standardized SPARQL API interface that queries the underlying database.

Libraries like rdflib give the capability of RDF tuple creation based on the schema defined. But the drawback of this process is that you need you need to specify yourself the structure. It is suitable for the new creation of data. For converting valuable legacy data, we need to rely on R2rml which only an abstraction layer between so that we don't have to code it ourselves. Also, another significant advantage of R2RML

script is its modularity. For adapting a schema change in R2RML, we don't have to edit anything apart from triple schema definitions. The code for scripting language remains the same. This improves the maintainability of code. In contrast, while using RDF data creation libraries, not only the schema definition has to be changed, but we also have to change the data creation calls.

Since the R2RML file is an independent plugin file to convert data, when it comes to adapt the changes in schema, we do not need to recompile and test the application in contrast to applications relying on class creation using libraries. Also, R2RML gives you a brief overview of triples and the their underlying triple schema which are created (*A Direct Mapping of Relational Data to RDF*).

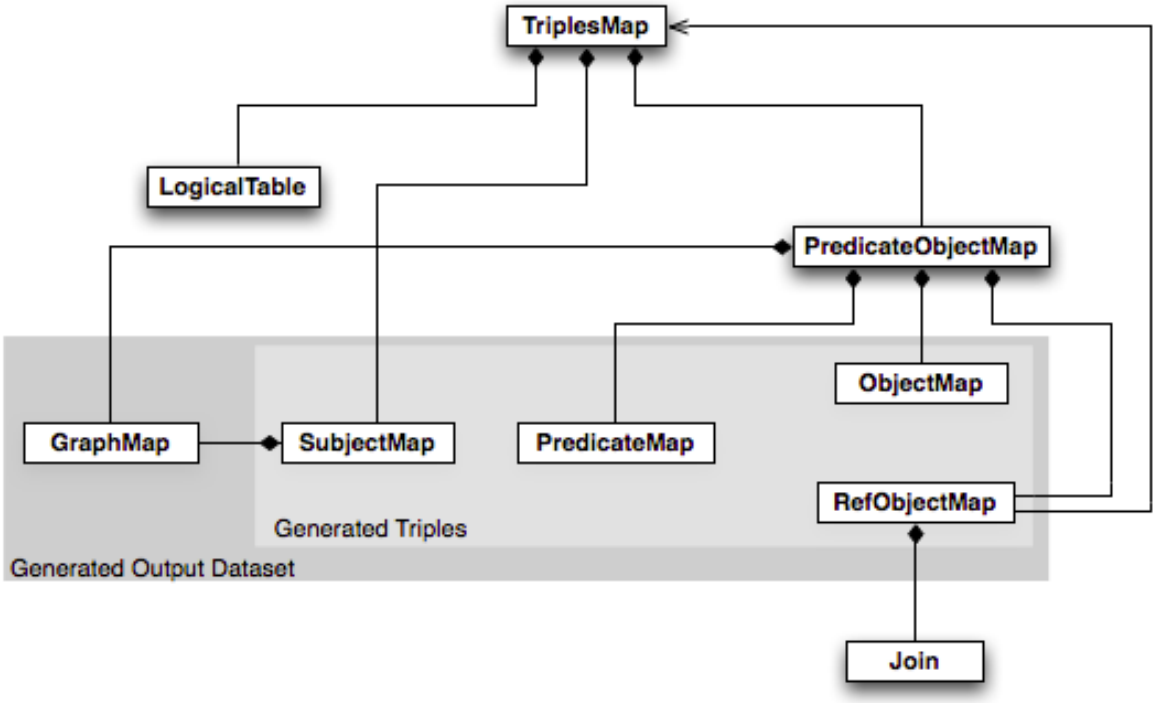

FIGURE 2.5: Overview of Triples

Figure 2.5 shows a typical structure of an R2RML script. To begin with, we refer to the logical table where we retrieve the data from the database. It can be either a :

- A Table
- A view

- A valid SQL query

This logical table is later connected to the *TriplesMap* which also consists of a subject and the predicate map. The triple map is a set of rules for each row in the logical table to RDF triples. It consists of two entities which generate triples :

1. A subject map which generates all the subject of RDF triples with IRI.
2. Multiple predicate-object maps with all predicates and objects.

Furthermore, triples are generated using this subject-predicate-object map for each row in a logical table.Hence we need to write a customized mapping for every dataset.

```
1  @prefix rr: <http://www.w3.org/ns/R2RML#>.
2  @prefix ex: <http://example.com/ns#>.
3
4  <#TriplesMap1>
5      rr:logicalTable [ rr:tableName "EMP" ];
6      rr:subjectMap [
7          rr:template "http://data.example.com/employee/{EMPNO}";
8          rr:class ex:Employee;
9      ];
10     rr:predicateObjectMap [
11         rr:predicate ex:name;
12         rr:objectMap [ rr:column "ENAME" ];
13     ].
14 }
```

Listing 2.13: R2RML conversion script using table name

In the above RDF triples (*A Direct Mapping of Relational Data to RDF*), we declare a logical table on line 5 from the relational table "EMP". Then at line 6, we have the subject map with IRI (line 7) and class declaration (line 8). For the predicate object map, we define the predicate as `ex:name` (line 11) while the object as the value in column "ENAME" from the "EMP" table (line 12). A sample converted triple would look like this:

```
1
2  <http://data.example.com/employee/7369> rdf:type ex:Employee.
3  <http://data.example.com/employee/7369> ex:name  "SMITH".
```

```
}
```

Listing 2.14: sample converted output triples

We can get similar output triples by using Listing 2.15 R2RML conversion script. The only difference here is on line 5, which describes a SQL query instead of table name. This gives the control of column selection from the the table.

```
@prefix rr: <http://www.w3.org/ns/R2RML#>.
@prefix ex: <http://example.com/ns#>.

<#TriplesMap1>
    rr:logicalTable [ rr:sqlQuery """SELECT EMPNO as EMPNO, ENAME as
    ENAME FROM EMP;""" ];
    rr:subjectMap [
        rr:template "http://data.example.com/employee/{EMPNO}";
        rr:class ex:Employee;
    ];
    rr:predicateObjectMap [
        rr:predicate ex:name;
        rr:objectMap [ rr:column "ENAME" ];
    ].
}
```

Listing 2.15: R2RML conversion script using SQL select query

2.10.1 R2RML conversion Tool

The tool to materialize the relational data into RDF triples is available on GitHub (https://github.com/maastroclinic/DataFAIRifier). The tool has been updated and modified promptly. Apart from the python script, custom R2RML scripts were made as per each Hospital's requirements. The hospitals had total control over this script. The following projects used R2RML conversion tool :

- ProTrait (proton-based radiation therapy decentralized national registry)
- PoleCat (Poland)
- Sage (Maastricht - Rome collaboration on rectal cancer treatment investigation)

- VWdata (standardizing and performing federated analyses on vertically-partitioned data)
- Head and neck cancer (Maastro)
- atomCAT (Norway/UK).

Based on the requirement, hospitals had to plan the RDF schema based on the columns in database columns. Also, the SQL query which served as input had to be changed as per the hospital's specific database schema.

Chapter 3

Methods

3.1 Automated Triple Conversion

In R2RML Direct Mapping, we need a diverse range of competencies. It requires Medical specialist who have a conceptual understanding of medical terms to map and business intelligence trained data managers. Considering this requirement of bridging the organizational hurdles we have built a tool which aims to minimize the additional information needed for the FAIRification process. It helps reserachers to easily get started with conversion of the tabular data in RDF triples. We define set rules based on Data Definition Language (DDL) which help us automate the process preparing R2RML scripts. Furthermore, we show that the ontology extracted from existing databases can provide a description framework to describe and annotate existing data sources. This annotation process would target mostly the "Interoperable" aspect of FAIR. (*Annotation of existing databases using Semantic Web technologies: making data more FAIR*)

3.1.1 General Workflow

This framework expects a comma-separated file or relational database connection as input. Using this connection, we extract the database schema which fetches the table definitions (table names and column names) and table details of unique key constraints. We utilize this information to build an ontology which creates classes . This ontology maps every table class definition as `dbo:TableRow` and every column is instance of class `dbo:ColumnCell`. Every individual column cell is assigned to its unique table row to which they belong.

Using this schema, we make an ontology file through which we materialize the triples. With restructuring to enhance the underlying schema for triples, we upload

the script as well as the materialized triples. For binding the materialized triples with a standardized set of terminologies, owl-inference statements are applied on top. Hence materialization is automated while binding is done manually. We can publish the ontology file and annotation rules as the description of the materialized dataset. Figure 3.1 illustrates overview of this workflow. Green elements are automated processes, orange elements are manual tasks and Blue elements are outcomes of processes.

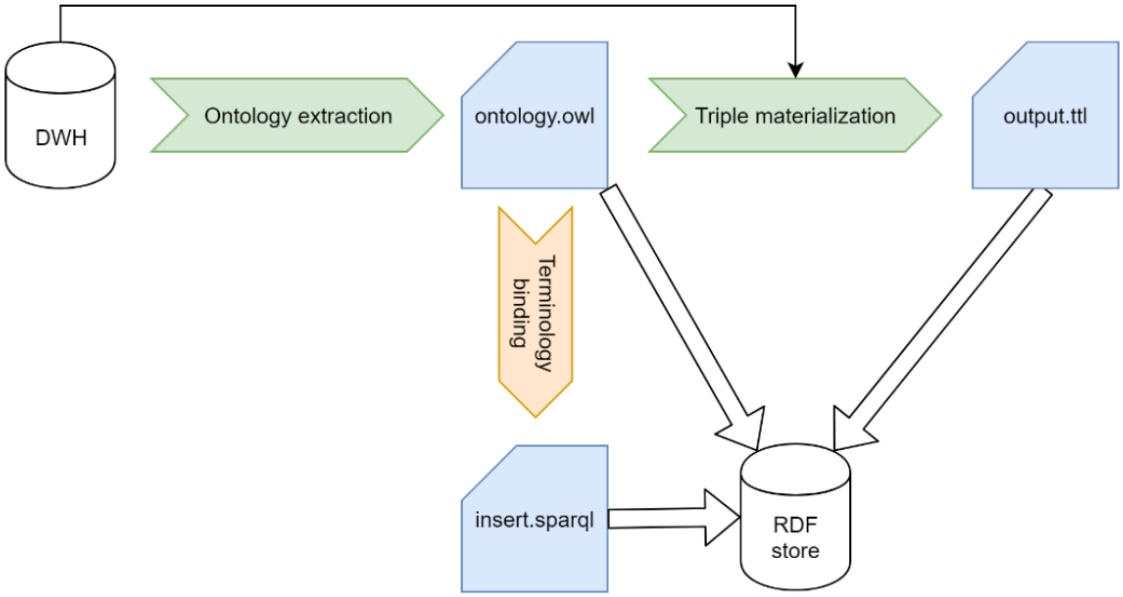

FIGURE 3.1: Overview of Triplifier workflow

3.1.2 Materialization Schema

Converting every row and column into triples requires an ontology file which utilizes the database structure. Using the basic building blocks of database like a table, row, column and key constraints, we have built a generic ontology which transforms any database schema into RDF triples. The schema of this ontology has the following fundamental mapping rules :

- every database row will be an instance of the subclass of `dbo:TableRow`
- every cell for this row will be an instance of a subclass of `dbo:ColumnCell`.
- The actual cell values are literals connected to the instance of `dbo:ColumnCell`.

- For uniquely identifying the `dbo:TableRow`, URI will be formed by the combination of table name and primary key column(s). This URI will also be used as base of `dbo:ColumnCell` URI.
- Instances of `dbo:TableRow` and `dbo:ColumnCell` will be associated with `dbo:hasColumn` relation.
- The cell value (literals) and `dbo:ColumnCell` will be associated with $\texttt{dbo:has}_{v}aluerelation.\texttt{dbo:PrimaryKey}and\texttt{dbo:ForeignKey}willbeassociatedaspertheconstra$ $subClassOf$ to`dbo:ColumnCell`
- For foreign key relationships `dbo:ColumnReference` will be created between `dbo:TableRow` and `dbo:ColumnCell`

Figure 3.2 summarizes these rules.

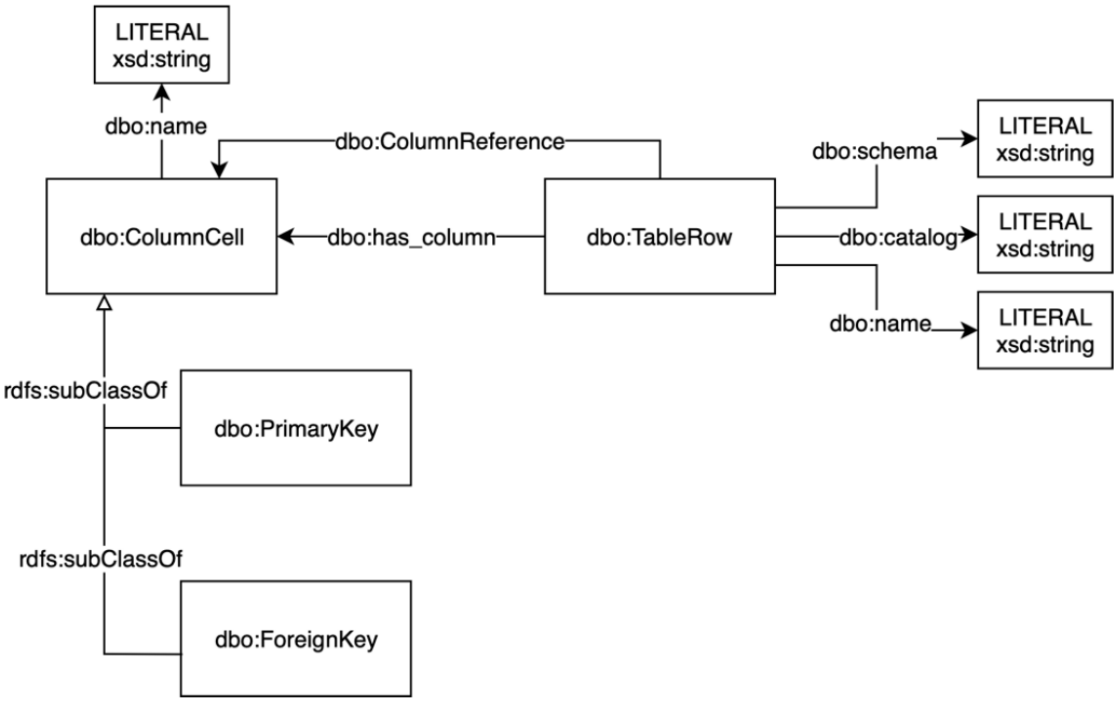

FIGURE 3.2: Overview of general schema structure

Using the sample table 3.1 we get following structure Figure 3.3 of triples post materialization by triplifier framework. The elements in orange are outcome of database schema while the elements in grey are outcomes of DDL schema ontology file.

TABLE 3.1: Sample data for materialization

id	age
123	23

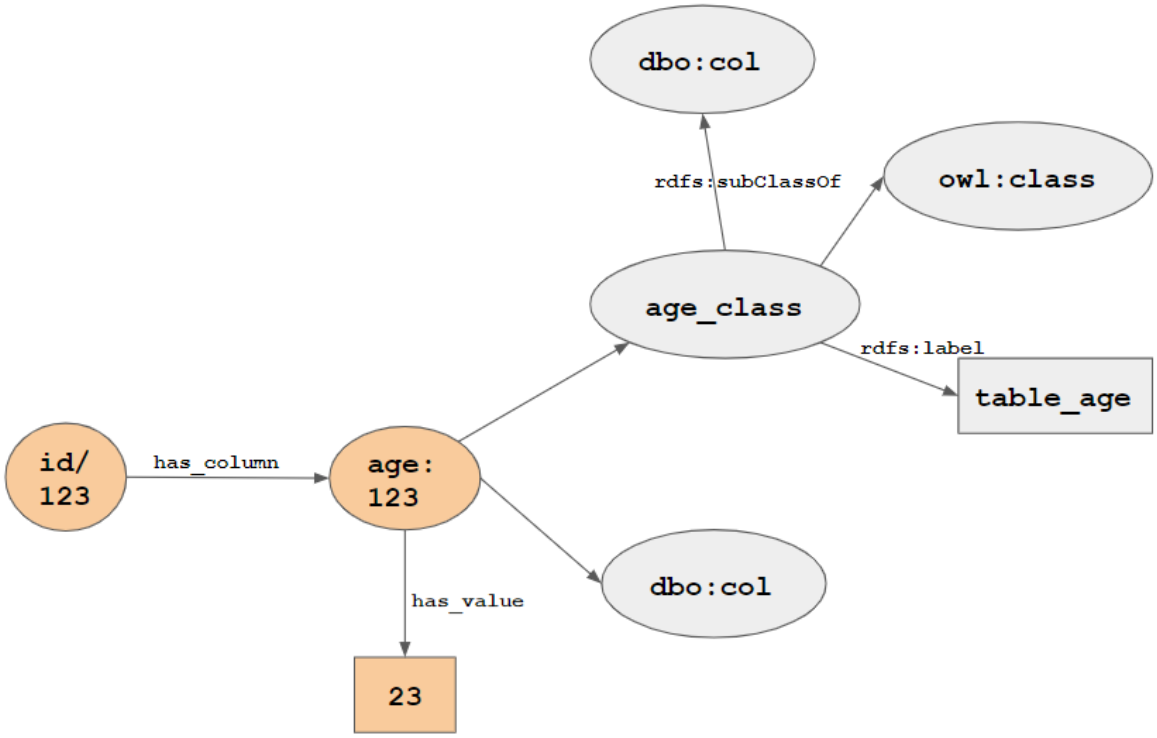

FIGURE 3.3: Schema structure of converted triples

For binding terms from the standardized ontology, we can use owl inference rules. We annotate the subclasses of `dbo:ColumnCell` with `rdfs:equivalentClass` axioms for numeric values. In Figure 3.4 the green element denotes added equivalent class on `age_class` from actual standardised the NCI thesaurus. Similarly, annotation of other columns can be implemented. Initially all the instances of `dbo:col` must be identified. Further, relevant term from the ontology should be identified. Using appropriate insert query the identified relevant concept from the ontology can be associated with column class. The reasoning rule is added on the DDL schema ontology. As the DDL ontology file can be shared, re-usability of the inference rules increases.

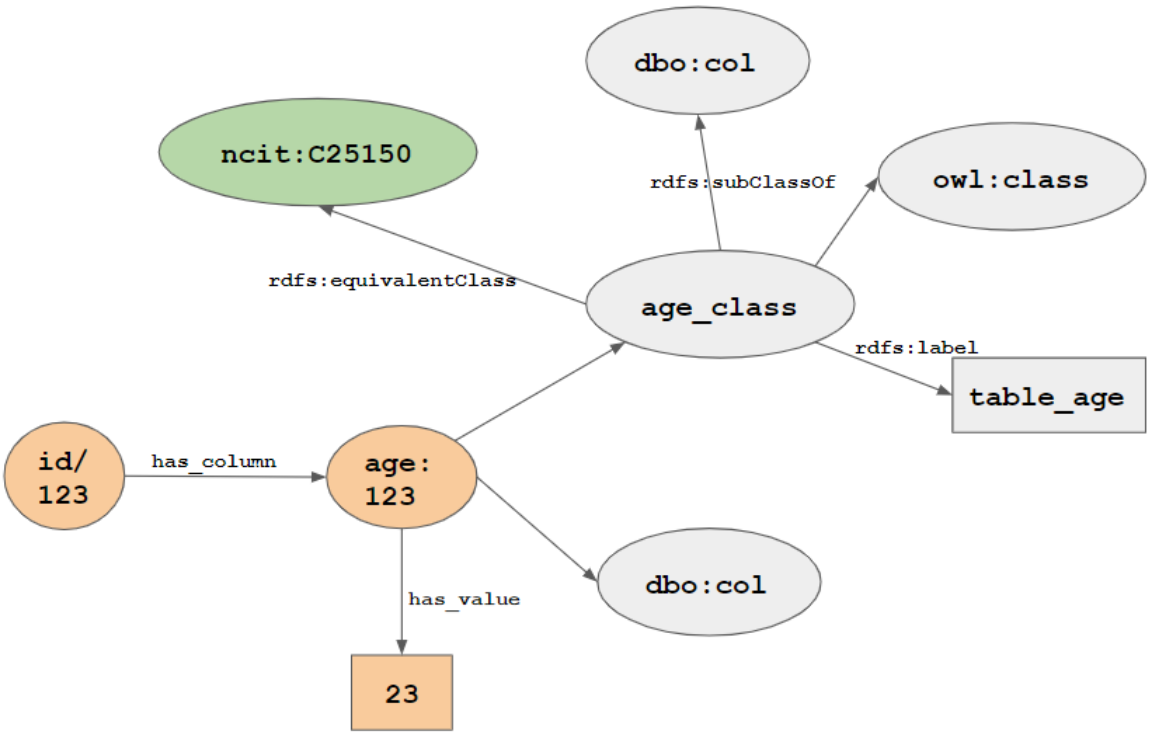

FIGURE 3.4: Schema structure of inference added triples

3.1.3 Specifications and Performance

With concepts explained above, we built a framework which reads the database schema to build the ontology. The database rows and cells are later materialized based on the extracted ontology file. For performance evaluation we will be measuring time required for the intermidieate activities in converstion process:

- Ontology extraction of DDL
- Materialization of triples
- Uploading DDL ontology to RDF store
- Uploading materialized triples

3.1.4 Major advantages

In contrast to direct R2RML mapping, this tool takes a different approach for materialization (*A Direct Mapping of Relational Data to RDF*). This deviation has the following advantages:

1. R2RML is an intermediate description format which is also based on RDF for defining the triples schema in the materialization process. As the triplifier framework has the inbuilt capability to represent the DDL scehma in RDF, the end-user does not need to understand additional R2RML format.

2. Based on the schema of the input dataset, in R2RML, we define a custom SQL query as an add-on preprocessing step. In the triplifier framework, only the queries which fetch all entries from a particular table are executed to bring the data, which is most likely to run on any RDBMS database. This helps us skip the effort of writing SQL joins.

3. As per R2RML specification, we have to define cell value as a direct literal predicate of the row object, for instance of a row. As this limits the inferencing capabilities of the equivalent class, we implement every cell as an instance of column class and apply inference rules on it.

3.1.5 Reasoning rules on triplified data

Post triplification, the user can add inference rules as annotations. Using the triple schema to the advantage, annotation can be done as shown in Figure 3.4. Taking this further, here the hybrid recommendation system is integrated to automate/ease the process of annotation.

3.2 Annotation Recommendation

Standardized taxonomies are vast and continue to grow as new concepts are always on the rise. It becomes a challenging task even for business users with domain knowledge to keep track of these new concepts and utilize them as they were intended to be. Opposed to simple lexical search we need an efficient mechanism which would help in this conceptual discovery.

As an aid to manual task of binding terms from opted standardized ontology to materialized DDL-based ontology in the workflow of Automated Triplication (Figure 3.1),

we have built an Annotation Recommendation System. Along with string similarity matching, we also use historical matches and conceptual comparison (matching using NLP) of words to make the most appropriate recommendation. A person with better understanding of domain knowledge should do the annotation for data improving its semantics, interoperability and usability. To cater this requirement, we are splitting recommendation in a manual and semi-automatic process. By structuring the system in this way, we aim to reduce the extent of variations in annotation.

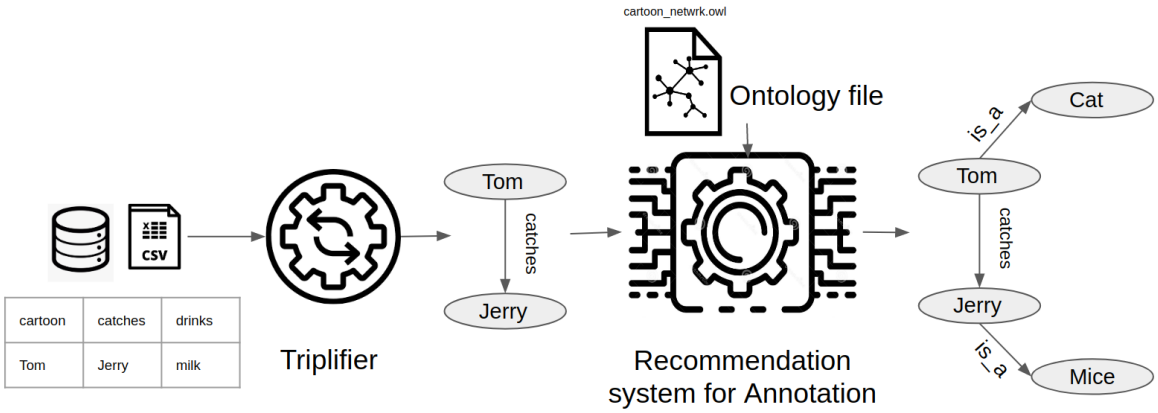

FIGURE 3.5: Overview of Annotation Recommendation workflow

3.2.1 Workflow Overview

The ontology file generated in the process of DDL extraction by triplifier as mentioned in section 3.1.2 is serving as input for this recommendation system. The system can also be customized to give recommendations directly on the list terms (which can be a list of column labels from a table). Apart from the list of labels on which prediction will be made, the system also expects the standardized ontology from which the terms will be matched. By comparing the input label and terms from the ontology, the system will give a recommendation of the best match in the form of matched label and IRI of the term from the ontology. Figure 3.5 illustrates the workflow of Recommendation System.

3.2.2 Flow chart of algorithms

For finding the perfect match, we are relying on three systems. Firstly, to save computational time and power, we are searching for a hard match in the history of previously

opted recommendations of the input label. If we do not have any history for the input label, we are doing string comparisons using Levenshtein and Jaro Winkler algorithm.

We have many other lexical matching algorithms, and Rules for string similarity may differ from case to case. Therefore for the third stage, we switch to Natural language processing for getting contextual matched results. The opted ontology may not be a rich ontology which has all the required standardized terms. In that case user manually select in the ontology for appropriate term through user interface.

Also, the recommended term might be related to the input term but maybe not correctly defining it. To overcome this ambiguity, we give the user a choice to select an outcome from any of these stages. Figure 3.6 explains the stage of algorithms used.

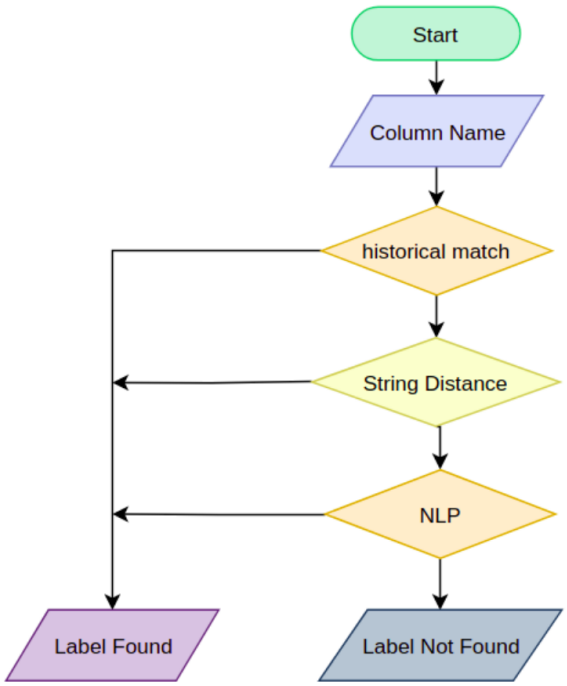

FIGURE 3.6: Overview of Automated Annotation algorithm flowchart

3.2.3 Historical Match algorithm

"I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail." (Maslow, 1966)

Instead of directly relying on computationally intensive algorithms, we have maintained a history of opted IRI of standardized term and the input label chosen by the

researcher. Along with inserts of annotation to the materialized triples, we also make an entry IRI and input label pair in the historical annotations repository. For future searches, by simple hard string match on input label and chosen label column, we aggregate results on counts of distinct IRIs.

To demonstrate the capability and promote the usage of graph databases, we are storing this history on http://sparql.cancerdata.org/, which is a Blazegraph endpoint. By generating a unique id (uid) and appending it to a base URI, we create the historical annotations repository in the triple schema as demonstrated in Figure 3.7

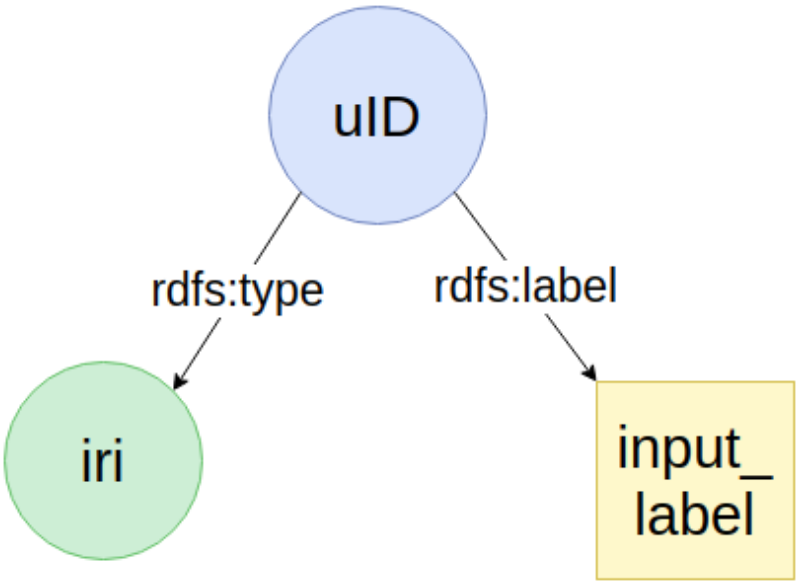

FIGURE 3.7: Schematic of History in Blazegraph

We use Listing 3.1 for fetching aggregated top two results by plugging the label to search in place of {ip_label}

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select ?iri (count(?iri) as ?count)
where
{
  ?s rdf:type ?iri;
     rdfs:label ?label
  FILTER regex(?label, "{ip_label}", "i")
```

```
}
group by ?iri
order by desc(?count)
limit 2
}
```

Listing 3.1: SPARQL to fetch aggregated results

3.2.4 String similarity algorithms

Database column labels are appended with prefix and post-fixes. To deepen the lexical search apart from the hard match we have included Levenshtein and Jaro-Winkler as string matching algorithms in the second tier of the of recommendation system. The functioning of string similarity algorithms helps trace the string patterns in a specific manner and recognise the regular variations in the nomenclature of column labels. There are many such algorithms which can determine specific patterns. To avoid overkill, we have limited the number to the following two string similarity algorithms:

Levenshtein algorithm

The edit distance (also known as Levenshtein distance) is the metric of similarity between two sequences of characters(strings). It is the number of single-character edits (deletions, insertions, or substitutions) needed to perform on one string to convert into another. Smaller the edit distance, mire similar are the input strings.(Levenshtein, 1966) Named after scientist Vladimir Levenshtein in 1996, this forms as a fundamental algorithm for text processing and search problems.

For calculating the distance between strings a and b with length `|a|` and `|b|` respectively we use following formula :

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \tag{3.1}$$

Here $1_{(a_i \neq b_j)}$ is the indicator function values to 0 when $a_i = b_j$ otherwise values to 1. $\mathrm{lev}_{a,b}(i,j)$ is edit distance between i characters of a and j characters of b. i and

j are index starting from 1. The first equation in the minimum bracket corresponds to deletion,the second to insertion and the third to match or mismatch. (*Levenshtein distance* 2020)

We can do three types of edits in this algorithm:

1. insertion
2. substitution
3. deletion

We have to use a minimum number of edits for calculating the distance by using the following algorithm:

Step 1: Initialization

1. Let n be the length of string s, and m be the length of t. For example if s is "cat" and t is "bat",m and n will be 3
2. Create a matrix containing 0..m rows and 0..n columns.
3. Initialize the first row to 0..n,
4. Initialize the first column to 0..m.
   The matrix for s as "cat" and t as "bat" would look in following form :

TABLE 3.2: Matrix Initialization Levenshtein algorithm

		b	**a**	**t**
	0	1	2	3
c	1			
a	2			
t	3			

Step 2: Processing

1. If character at s(i) equals character at t(j), the cost is 0.
2. If character at s(i) does not equal character at t(j), the cost is 1.
3. Set cell d(i,j) of the matrix equal to the minimum of:

(a) The cell immediately above plus 1 is d(i-1,j) + 1.This operation signifies deletion.

(b) The cell immediately to the left plus 1: d(i,j-1) + 1.This operation signifies insertion.

(c) The cell diagonally above on the left plus the cost: d(i-1,j-1) + cost.This operation signifies substitution.

Step 3: Result

Repeat step 2 till we get the value of d(n,m), which will be our Levenshtein distance. (Haldar and Mukhopadhyay, 2011)

In iteration 1, as first characters of s and t do not match each other the cost would be 1 as per Step2.2. Furthermore for cell value d(0,0) the minimum possible value will be diagonally above value on the left plus cost, which 0 + cost. Hence value on d(0,0) will be 1. Table 3.3 illustrates this cost and cell value calculation.

TABLE 3.3: First iteration of Levenshtein algorithm

		b	**a**	**t**
	0	1	2	3
c	1	1		
a	2			
t	3			

TABLE 3.4: Second iteration of Levenshtein algorithm

		b	**a**	**t**
	0	1	2	3
c	1	1	2	
a	2	2	1	
t	3	3	2	

TABLE 3.5: Third iteration of Levenshtein algorithm

		b	**a**	**t**
	0	1	2	3
c	1	1	2	3
a	2	2	1	2
t	3	3	2	1

Cell value at d(3,3) is the Levenshtein distance between "cat" and "bat". Here the only edit operation we did was the substitution of character "c" with "b". Using this as a metric, we have applied this algorithm for calculating and selecting the minimal most distance between the input column label and labels in the standardised ontology.

Jaro-Winkler algorithm

Jaro Similarity Jaro similarity is the metric of similarity among two strings based on the weighted sum of match characters from each string and the transposed characters (Jaro, 1989). Scaled from 0 to 1, higher the number, higher the similarity. Named of after scientist Matthew Jaro, the algorithm is used for applications which have the requirement of prefix matching. Jaro Similarity can be calculated by using following formula (*Jaro–Winkler distance* 2020) :

$$sim_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{3.2}$$

Where,s1 and s2 are string under comparison.

- $|s_1|$ is length of s1 and similarly $|s_2|$ is length of s2
- m is the number of "matching characters", while
- t be the number of "transpositions" or half the number of misplaced but matching characters ("Evaluating String Comparator Performance for Record Linkage").

The characters are considered matching if they are at distance of :

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1. \tag{3.3}$$

For example, In case when s1 = ”lenght” and s2 =”length” ,$|s_1|$ and $|s_2|$ will be 6. Value of m will also be 6 and t will be 1 as only 2 characters are miss placed. Jaro Similarity will be calculated as :

$$\frac{1}{3}\left(\frac{6}{6}+\frac{6}{6}+\frac{6-1}{6}\right) \tag{3.4}$$

which evaluates to 0.945.

Jaro-Winkler Similarity

Empirical studies suggested that typically the beginning of the string are more error-prone (Pollock and Zamora, 1984). To improve on this, William Winkler proposed a tweak in Jaro Similarity, which gave more emphasis on matching initial characters of strings(maximum four). Jaro Winkler Similarity can be calculated as :

$$sim_w = sim_j + \ell p(1 - sim_j), \tag{3.5}$$

Here sim_w is the similarity index, while sim_j is the Jaro Similarity of input strings. ℓ is the number matching prefix which can be of maximum value 4. p is constant scaling factor which is by default set to 0.1 and can be adjusted as per the number of common prefixes. (Winkler, 1990) While Jaro-Winkler distance d_w can be calculated as :

$$d_w = 1 - sim_w. \tag{3.6}$$

In example of ”lenght” and ”length”, we calculated sim_j as 0.945 in equation 3.4 Hence, Jaro-Winkler similarity sim_w will be:

$$0.945 + 4*0.1*(1-0.945)$$

This evaluates to 0.967. Hence we can see the rise of 0.2 in similarity. Also distance d_w will be 1-0.967, which is 0.033

Levenshtein VS Jaro-Winkler

Rules for string similarity may differ in every use case (Christen, 2006). The pattern for similarity criteria of the string is also different for every algorithm. Hence performance cannot be criteria for selection for the algorithm. In the case of example ”lenght” and

"length" the Levenshtein distance is 0.67 while Jaro Winkler distance is 0.96. Considering the prospects of patterns we had to recognise of Medical data domain, we have selected Levenshtein and Jaro-Winkler similarity to be our option for fundamental lexical analysis. For Levenshtein algorithm we are using `https://github.com/seatgeek/fuzzywuzzy` (*seatgeek/fuzzywuzzy* 2020) library, while for Jaro Winkler similarity we are relying on `https://pypi.org/project/pyjarowinkler/` (*pyjarowinkler*) library.

3.2.5 Natural Language Processing algorithms

Natural language processing, also abbreviated as NLP, is a discipline in artificial intelligence which deals with the interaction of data science and natural language humans use. Getting meaningful inference from unstructured data generated by conversations, manuscripts and declarations using traditional row-column structure is a complex task. Most of the NLP techniques aim to decipher this unstructured data and generate valuable cognitive outcomes. Recent advancements in data access and techniques and computational power are allowing researchers to explore the possibilities applications in various domains like media, social networks and health care.

For our use case, we need to analyse the relations between the input column labels and the labels in the standardised ontology. We require to comprehend the underlying meaning and the context of terminology labels. This will help use have analogies and the inferences, which could be scaled numerically. We have explored the Word Vector algorithms which address this kind of requirements.

3.2.6 Word Vectors

Machine learning models can only process numbers and no text. Word vectors is a strategic method of converting a word in number representation which retains its meaning. It is a multi-dimensional set of numbers which maps semantically similar words in proximity with each other. These numbers represent weights distributed across various dimensions. Indirectly these numbers represent the associativity of the word with that particular dimension or in other words dimensions are embedded with semantics (Mikolov et al., 2013b). In contrast to simple one-hot encoding Word vectors get the upper hand in computations due to its syntactic and semantic capabilities.

Suppose we have a vocabulary of skin, carcinoma, blood and leukemia the one-hot encoding vector would be :

TABLE 3.6: One-hot representation of vocabulary

	skin	carcinoma	blood	leukemia
skin	1	0	0	0
carcinoma	0	1	0	0
blood	0	0	1	0
leukemia	0	0	0	1

In table 3.6 we observe is not possible to establish any relationships between the words in vocabulary. In contrast, for forming word vectors, we consider various dimensions on which the word can be scaled. For example, let's consider we have dimensions like cancer, disease, benign, malignant, and WBC(white blood cells). Word vector representation for the vocabulary would look like:

TABLE 3.7: Word Vector representation of Vocabulary

	cancer	malignant	benign	tissue	WBC
skin	0.45	023	0.1	0.92	-0.3
carcinoma	0.87	0.72	0.65	0.84	0.1
blood	0.43	0.37	0.21	0.03	0.85
leukemia	0.93	0.79	0.69	0.19	0.81

The table 3.7 illustrates there is some abstract relationship between the word and the dimension based the value. We can make co-relations between the words by using these values of the dimension on which we want to scale/compare. This syntactic-semantic relationship can be used for various applications like analogy problems, answering questions, translations and information retrieval.

3.2.7 Word2Vec

Developed in 2013 by Tomas Mikolov, word2vec is a 2 layer neural net which transforms the word into numeric vectors. Word2Vec is not a deep neural net, but the numeric output vector forms input feed for the deep neural net. The algorithm is analogous to the autoencoder machine learning methodology. It does not learn from the input words but rather from the context words that are neighbour to it in the text corpus (Rong, 2016). Word2Vec can be implemented in two type of architectures:

1. Continuous bag of words (CBOW)
2. Skip-gram

CBoW

Consider the statement "Radiation is a physical agent, which destroys cancer cells by depositing high physical energy". With focus word as "destroys" and a sliding window of 4 words on each side for it, we would have :

"...tion is a physical agent, which destroys cancer cells by depositing hi..."

Continuous bag of words aims to determine the focus word destroy by using the context words in blue. The context words are the input layer in which each word is in a one-hot vector form. Hence, the length of these vectors is the number of words in the input vocabulary. Later we have a hidden layer which is of n dimensions. And late on the right, we have the output layer which with the dimension size of the vocabulary. Figure 3.8 illustrates the Continuous bag of words model.

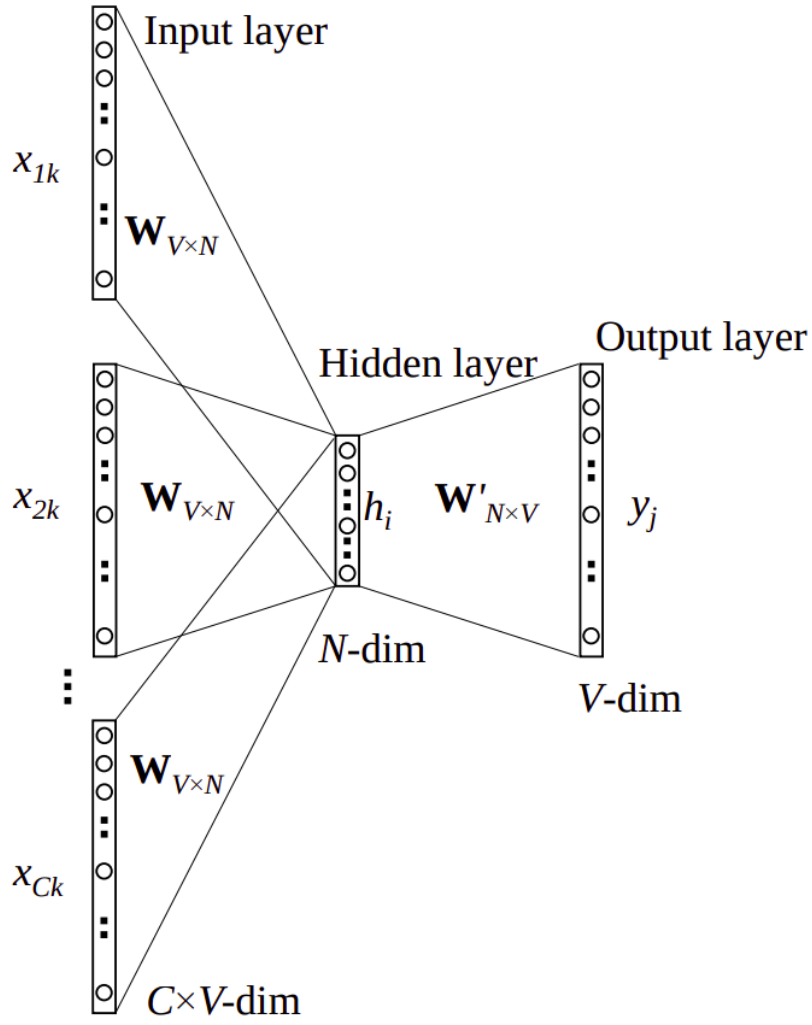

FIGURE 3.8: Continuous bag of words model
(Rong, 2016)

In Figure 3.8 (Rong, 2016) V is the size vector while N is number of neurons in hidden layer.W_{VN} is the weight matrix between input vectors and hidden layer.W'_{NV} is the weight matrix mapping the hidden layer and output layer. C is the number of words in context In the process of predicting the target word, we learn the vector representation of the target word.

As the input layer is one-hot vector, only a particular row is activated from W1. Hence the output of hidden layer is sum these selected rows, and divided by C for average.

$$\underset{\text{input}}{\begin{bmatrix}1 & 0 & 0\end{bmatrix}} \underset{\text{W1}}{\begin{bmatrix}a & b & c & d\\ e & f & g & h\\ i & j & k & l\end{bmatrix}} = \underset{\text{hidden layer}}{\begin{bmatrix}a & b & c & d\end{bmatrix}}$$

From Hidden layer to output layer second weight matrix W_2 is used scoring each word and later soft-max is used for posterior distribution

Skip-gram

The objective of training skip-gram is to get the context words as output from focus word as input. In other words, the skip-gram model is like CBOW model but inverted.

Similar to CBOW, activation function for hidden layer results in selecting a row from W1 weight matrix. At the output layer, we get multinomial distributions as per the size of the context window. The main objective of the training is to minimize the prediction error summed across context words at the output layer (Rong, 2016).

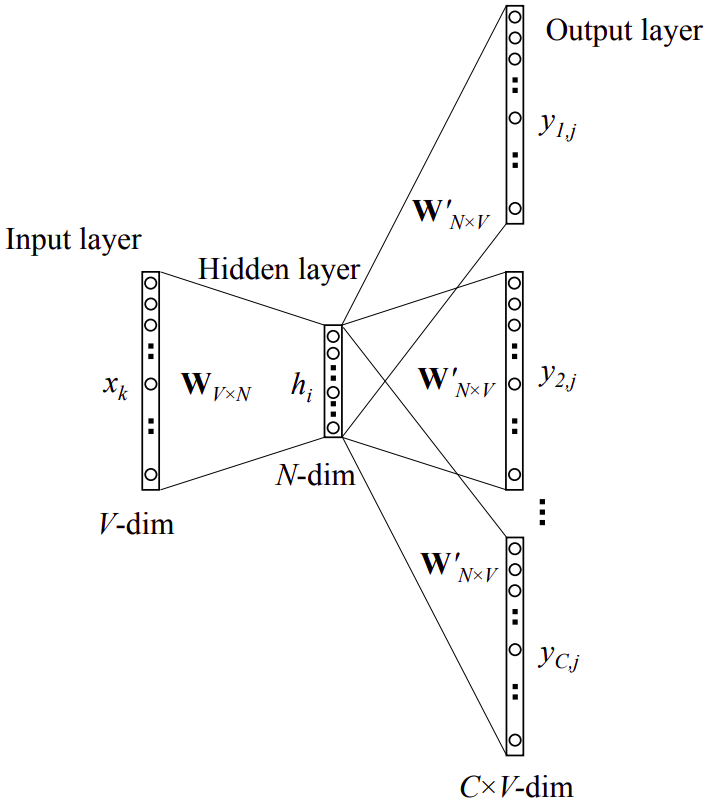

FIGURE 3.9: Skip Gram Model
(Rong, 2016)

Skip Gram vs.CBoW

Both have their own advantages and disadvantages. According to Mikolov, Skip Gram works well with small amount of data and is found to represent rare words well. On the other hand, CBOW is faster and has better representations for more frequent words (Mikolov et al., 2013b). In use-case of recommendation system, Skip gram model is a chosen as we need to find relevant context words (terminologies from ontology) based on input focus word (database column label).

3.2.8 GloVe

Unsupervised learning methods primarily use statistics of word occurrences for learning word representations. It is unclear what meaning these word vectors represent. Proposed by Pennington et al., 2014, Global vector model, also called GloVe, uses the insights of word representation to capture global corpus statistics directly. (Pennington, Socher, and Manning, 2014).

The core concept of GloVe is to deduce the semantic relationship between words from their co-occurrence matrix. These co-occurrence probabilities have the potential of encoding some meaning captured as vector difference. The training objective is learning word vectors such that dot product equals the logarithmic probability of words co-occurrence. It creates vectors which are capable of similarity tasks and word analogy. Global vectors are calculated using two main steps:

1. Construction of co-occurrence matrix using text corpus. For example, for statement "Radiation destroys tumorous cells", the matrix would be :

	Radiations	destroys	tumorous	cells
Radiation	0	1	0	0
destroys	1	0	1	1
tumorous	0	1	0	1
cells	0	1	1	0

TABLE 3.8: Co-occurrence matrix for statement "Radiation destroys tumorous cells". As observed, this matrix is a symmetric matrix

2. Factorization of co-occurrence matrix for obtaining vectors.

The glove algorithm aims to obtain a clear metric of semantic similarity of words. Instead of just considering raw co-occurrence probabilities of two words, the ratio of

probabilities with a third probe word is applied. Consider X as co-occurrence matrix which holds word-word count where X_{ij} is number time the ith term appeared next to the jth term. P_{ij} is the probability of i occurring next to j, which is calculated as X_{ij}/X_i.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

FIGURE 3.10: Ratio of Co-occurrence probabilities for GloVe (Pennington, Socher, and Manning, 2014)

In table 3.10 with a corpus of 6 billion tokens we consider $i = ice$ and $j = steam$ as target words and k as probe words(solid, gas, water and fashion). From the ratios $P_{k|ice}/P_{k|steam}$ we can infer that :

1. probability ratio is very high(>1) when the probe word is very similar to ice but very irrelevant to steam. (when k = solid)
2. probability ratio is meagre(<1) when the probe word is very similar to steam but very irrelevant to ice. (when k = solid)
3. probability is near to 1 when probe word unrelated to ice and steam(when k = water and fashion)

Using the ratio P_{ij}/P_{jk} in computation of the vector, we can achieve integration of statistics from the global corpus for learning word vectors.

3.2.9 GloVe vs. Word2Vec

The significant difference between Word2Vec and GloVe is that Glove is a count-based model, while word2vec is a predictive model. Using the Skip-Gram and CBOW, we can do arithmetics on the word vectors which were trained on words in context window (Mikolov et al., 2013a). This methodology efficiently preserves the local analogies and reference of the focus word but looses its intrinsic statistical properties.

Word2Vec is a predictive model, learns the vectors by loss of predicting the target words from given context words. In contrast, the count-based model GloVe learns by dimensionality reduction on the counts' co-occurrence matrix. The large count matrix is later factorized to obtain a reduced matrix in which row represents a vector of each word. (Pennington, Socher, and Manning, 2014)

In practice, it is easier to speed up the training process of Word2Vec by utilizing negative sampling and softmax functions. Depending on the requirement of semantic locality and type and the amount of training data, the optimal choice of algorithm is made.

3.3 SciSpacy framework

Even though with recent advancements natural language processing, many statistical models under-perform in new domains. It is a critical requirement in the biomedical domain for processing the clinical data, which is in raw format. Spacy is a python based tool which caters to practical needs of text processing needs in multiple languages. Based on the performance and robustness of Spacy, SciSpacy is explicitly built for processing biomedical data (Neumann et al., 2019).

We can use various pre-trained models in SciSpacy. Considering the efficiency required, we have opted for 'en_core_sci_lg' which is the largest available model. It is a set of 600,000-word vectors captured using Word2Vec algorithm. These word vectors are captured from medical text and literature available on Pubmed and PMC. This model is trained using the skip-gram model with context window size 5. For optimization, hierarchical soft-max training is used with sub-sampling threshold of the frequent word at 0.001. It created 200 vectors (*bio.nlplab.org*).

Along with this frame-work we have integrated these three algorithms in python. Furthermore, to increase the usability of this algorithms by non-it background user, we have implemented a interactive web-interface.

3.4 Methods of Evaluation

As a base for a standard set of taxonomy, we are using the Radiation Oncology Ontology (Traverso et al., 2018). As input database labels, we are using the database schema extracted from the SAGE Data warehouse, which is an investigation of rectal cancer treatment by Maastricht-Rome collaboration. Later, to establish the base annotation standards, we took the help of Doctors at Maastro Clinic. Furthermore, we have evaluated the outcomes of our algorithms against these reference standards. As a metric for evaluation of results, we have used Binary evaluation and Contextual similarity index.

3.4.1 Binary Evaluation

The results of the predictions were classified into the following outcomes of binary classification:

1. True positive (TP): Clinical label annotated with ontology concept when the concept is selected correctly and is present in the reference standard.

2. True negative (TN): Clinical label not annotated with ontology concept when the concept is absent in reference standard.

3. False Positive (FP): Clinical label annotated with ontology concept when the concept is selected correctly and is absent in reference standard.

4. False negative (FN): Clinical label not annotated with ontology concept when the concept is present in the reference standard.

Figure. 3.11 represents the relationships between the outcomes. The dotted circle represents annotations recommendations made by algorithm while the solid circle represents the recommendations which were made as a standard reference. In ideal situation annotation suggested must exist in standard reference of annotation, denoted by overlapping part of circles.

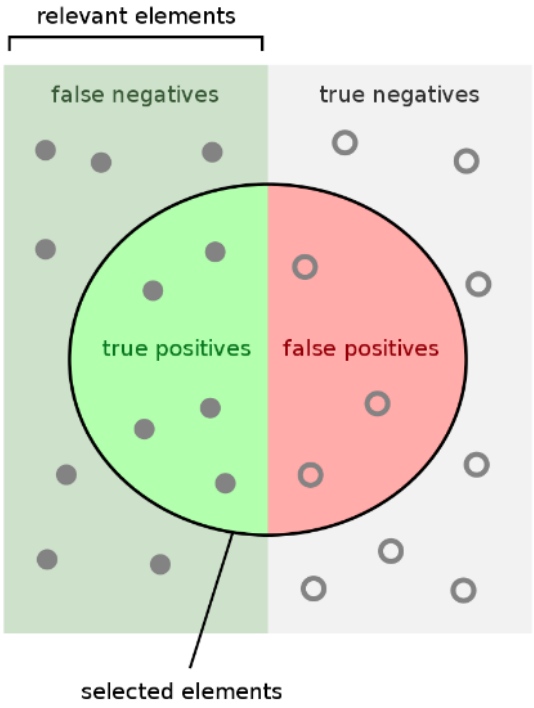

FIGURE 3.11: Classification of Outcomes
(*Sensitivity and specificity* 2020)

Combinations of binary classification outcome will be used to calculate following performance measures :

$$Precision = \frac{TP}{TP + FP} \quad (3.7) \qquad Recall = \frac{TP}{TP + FN} \quad (3.8)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.10)$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3.9)$$

Precision focuses on the accuracy of the ontology concepts with the annotations resulting from the annotation task. They help in determining the performance when False-positive recommendations are high. Recall focuses on the effectiveness of the annotation task to identify ontology concepts. It gives the estimate of how much True positives were predicted from the actual set of positives. F1 score is metric for accuracy and overall effectiveness of recommendations made against the standardised reference annotations. An ontology can have one or more relevant terms for given input label. Hence, it is difficult to evaluate specificity (ratio of actual negatives correctly identified)

as we can establish ground truth for every term in ontology being relevant with input label.

3.4.2 Contextual Similarity

Along with metric for checking the quality of recommendation, we are also evaluating the quality of string similarity. The algorithms, apart from making recommendations, also give us the percentage of the match. For NLP algorithms this will be a contextual match while a lexical match for string algorithms. Currently, post recommendation selection by the end-user, we are not storing by which particular algorithm made the recommendation. Hence we cannot scale the recommendations made from historical matches.

Chapter 4

Results and Performance evaluation

4.1 Introduction

Understanding of concepts is dependent on the perspective of the person establishing it. This perspective varies based on domain knowledge. Our framework aims to recommend the annotations on input labels extracted from the database schema. To assess our hybrid algorithm cluster usin the test use case, we need to evaluate the quality of annotations suggested by researchers with an extensive understanding of the medical domain and deep understanding of the concepts described in the ontology itself. Furthermore, using annotations as ground truth, we have measured the efficiency of our predictions.

The framework aim to target the user base, who mostly have domain knowledge but lack of software skills. Framework displays the recommendations made by algorithms, but the end-user chooses the appropriate option.

4.2 Performance of Automated Triple Conversion

As a part of initial testing, we ran triple conversion framework on institutional data warehouse containing diagnostic, treatment and questionnaire information of over 4000 rectal cancer patients treated with radiation therapy.

The framework was executed on a virtual machine running ubuntu 18.04 with 2 CPU and eight GB of ram. For graph data management, we used GraphDB by Ontotext, which was optimized for ”OWL-2RL” reasoning capabilities.

The tool can be utilized as a stand-alone java application or as a service by executing through docker. The framework performs reasonably for daily upload. Table 4.1 measures time required for various stages in the conversion process. This process was

scheduled to run daily. Uploading and internal processing in GraphDB are the most time-consuming activities in the workflow.

Table 4.1: Time measurements of framework

Step description	Mean time in seconds (daily run)	Remarks
Extract the ontology	0.6 (0.03)	2 tables with
Materializing triples	66.4 (1.14)	3.38 million triples
Upload ontology in RDF store	0.1 (0.02)	
Upload materialized triples	482,7 (3.46)	Dependent on number
Adding annotation reasoning rules	103.7 (1.03)	of triples

4.3 Performance of Historical Match

Annotations selected by the end-user is stored in history. Considering the that end-user will be domain expert, there is no metric defined for testing the semantic quality. A particular concept can be utilized across various triple schema and it can be defined by multiple standardized ontology. Hence we are providing statistics of a particular matched label against aggregated counts of possible IRI combinations picked from ontology. The statistics give the end-user a brief idea of which specific of Label and IRI is preferred by other users.

Table 4.2: Sample Historical data

IRI	**Label**
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16576	male
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16576	male
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16576	male
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C77777	male
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C77777	male
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16576	male
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16960	Patient
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16960	Patient
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C11111	male

Consider we have historical data from table 4.2 stored at http://sparql.cancerdata.org/. By using query 3.1 for label "male" we will get following statistics which are used to display on User Interface :

TABLE 4.3: Aggregated results for Historical data

IRI	count
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16576	4
http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C77777	2

4.4 String matching algorithms

Initially, best matching ontology label is selected for a given database label by running algorithm against every ontology label. Outcomes of annotations for string matching algorithm are evaluated using binary classification (Section 3.4.1) . This evaluation is done against the manual annotations done by domain experts. Figure 4.1 shows the labels under test, manual annotations by domain experts, and the outcomes of our string algorithms (Jaro-Winkler and Levenshtein).

In Figure 4.1, note that the "match%" columns for both the algorithms are Lexical match and not semantic match.

	Table	Column Label Input	Manual Annotations	predicted by Jaro-Winkler	match%	predicted by Levenshtein algorithm	match%
2	Outcome	Birth Month	Birth	Birth	89	Birth	90
3	Outcome	Gender	Gender	Gender	100	Gender	100
4	Outcome	Entry Date	Date	Entity	78	Date	90
5	Outcome	Question Type	NOT FOUND	Quantity	81	Censoring type	67
6	Outcome	Question	NOT FOUND	Version	78	Low Anterior Resection	68
7	Outcome	Answer	NOT FOUND	Anal Verge	75	Sex	60
8	Tumour_Treatment	Birth Month	Birth,Month	Birth	89	Birth	90
9	Tumour_Treatment	Date of Death	has date of death	Cause of Death	88	has date of death	95
10	Tumour_Treatment	Date of Diagnosis	has date of diagnosis	Date	85	has date of diagnosis	95
11	Tumour_Treatment	Gender	Gender	Gender	100	Gender	100
12	Tumour_Treatment	Death	Death	Death	100	Death	100
13	Tumour_Treatment	is alive last checked date	has alive last checked date	has alive last checked date	84	has alive last checked date	94
14	Tumour_Treatment	Diagnose	Diagnostic Procedure	Diagnostic Procedure	84	Diagnostic Procedure	79
15	Tumour_Treatment	clinical T stage	has clinical t stage	Clinical Target Volume	83	has clinical t stage	95
16	Tumour_Treatment	clinical N stage	has clinical n stage	has clinical n stage	81	has clinical n stage	95
17	Tumour_Treatment	clinical M stage	has clinical m stage	has clinical m stage	81	has clinical m stage	95
18	Tumour_Treatment	PBD	NOT FOUND	NOT FOUND	72	Body Weight	60
19	Tumour_Treatment	plan radiation target area	NOT FOUND	NOT FOUND	77	Radiation Therapy	86
20	Tumour_Treatment	Health plan	NOT FOUND	NOT FOUND	83	Health Care Activity	86
21	Tumour_Treatment	Target volume part	Target Volume	Target Volume	94	Target Volume	95
22	Tumour_Treatment	EPD - Prescribed fraction dose	Prescribed Radiation Dose	Prescribed Radiation Dose	74	Dose	90
23	Tumour_Treatment	EPD - Prescribed number of fractions	Number of Radiotherapy Fractions Per Day	Number of Radiotherapy Fractions Per Day	71	Fraction	90
24	Tumour_Treatment	EPD - Prescribed number of fractions per day	Number of Radiotherapy Fractions Per Day	Number of Radiotherapy Fractions Per Day	73	Fraction	90
25	Tumour_Treatment	EPD - Prescribed planosis	Prescribed Radiation Dose	Prescribed Radiation Dose	70	Prescribed Radiotherapy Total Treatment Dose	86
26	Tumour_Treatment	Fractions given for PBD	Fraction	Fraction	87	Fraction	90
27	Tumour_Treatment	Planned dose given for PBD	Planning Target Volume - Node	Planning Target Volume - Node	83	Dose	90

FIGURE 4.1: String Algorithms on SAGE database

4.4.1 Jaro-Winkler algorithm

Table 4.4 shows the binary performance evaluation of Jaro-Winkler algorithm. We can observe that even though the algorithm gives priority to initial four characters, the algorithm under performs for the terms like ”has clinical t stage”, ”has date of diagnosis” and ”has date of death”.

TABLE 4.4: Binary Classification of outcomes from Jaro-Winkler algorithm

Output Class	**TP**	**TN**	**FP**	**FN**	**Precision**	**Recall**	**F1**	**ACC**
Anal Verge	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Birth	2.00	27.00	0.00	0.00	1.00	1.00	1.00	1.00
Cause of Death	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Clinical Target Volume	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Date	0.00	27.00	1.00	1.00	0.00	0.00	0.00	0.93
Death	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Diagnostic Procedure	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Entity	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Fraction	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Gender	2.00	27.00	0.00	0.00	1.00	1.00	1.00	1.00
NOT FOUND	3.00	23.00	0.00	3.00	1.00	0.50	0.67	0.90
Number of Radiotherapy-Fractions Per Day	2.00	27.00	0.00	0.00	1.00	1.00	1.00	1.00
Outcome	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Planning Target Volume - Node	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Prescribed Radiation Dose	2.00	27.00	0.00	0.00	1.00	1.00	1.00	1.00
Quantity	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Target Volume	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Version	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
has alive last checked date	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has clinical m stage	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has clinical n stage	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has clinical t stage	0.00	28.00	0.00	1.00	None	0.00	0.00	0.97
has date of death	0.00	28.00	0.00	1.00	None	0.00	0.00	0.97
has date of diagnosis	0.00	28.00	0.00	1.00	None	0.00	0.00	0.97
Total	20	662	7	7	14	13.5	13.67	23.52
Average	0.83	27.58	0.29	0.29	0.67	0.75	0.57	0.98

4.4.2 Levenshtein algorithm

Table 4.5 shows the binary evaluation of Levenshtein algorithm. We are calculating the minimum number of changes to be made between the strings under comparison. This approach performs well for cases like ”has clinical t stage”, ”has clinical n stage” and ”has clinical m stage”. Levenshtein algorithm also performs good for singular labels like ”Gender”, ”Birth” and ”Date”.

TABLE 4.5: Binary Classification of outcomes from Levenshtein algorithm

Output Class	**TP**	**TN**	**FP**	**FN**	**Precision**	**Recall**	**F1**	**Accuracy**
Birth	2.00	27.00	0.00	0.00	1.00	1.00	1.00	1.00
Body Weight	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Censoring type	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Date	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Death	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Diagnostic Procedure	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Dose	0.00	27.00	2.00	0.00	0.00	None	0.00	0.93
Fraction	1.00	26.00	2.00	0.00	0.33	1.00	0.50	0.93
Gender	2.00	27.00	0.00	0.00	1.00	1.00	1.00	1.00
Health Care Activity	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Low Anterior Resection	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
NOT FOUND	0.00	23.00	0.00	6.00	None	0.00	0.00	0.79
Number of Radiotherapy-Fractions Per Day	0.00	27.00	0.00	2.00	None	0.00	0.00	0.93
Outcome	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Planning Target Volume - Node	0.00	28.00	0.00	1.00	None	0.00	0.00	0.97
Prescribed Radiation Dose	1.00	27.00	0.00	1.00	1.00	0.50	0.67	0.97
Radiation Therapy	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Sex	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97
Target Volume	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has alive last checked date	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has clinical m stage	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has clinical n stage	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has clinical t stage	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has date of death	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
has date of diagnosis	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00
Total	17	688	10	10	14.33	14.5	14.17	24.31
Average	0.68	27.52	0.40	0.40	0..65	0.81	0.57	0.97

4.5 Natural Language Processing -Word2Vec(skip gram)

The "match%" column in case of NLP denotes more contextual similarity than lexical similarity as we are using Skip gram model. Here in table 4.6, we observe that if concept of the input database label is defined ontology, appropriate label is selected as outcome.

TABLE 4.6: Binary Classification and Context match of outcomes from Skip-Gram algorithm

Output Class	TP	TN	FP	FN	Precision	Recall	F1	Accuracy	% match
Birth	2.00	27.00	0.00	0.00	1.00	1.00	1.00	1.00	82.59
Date	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	80.30
Death	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	100.00
Definitely Related to Intervention	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97	0.00
Diagnostic Procedure	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	46.69
Equivalent 2Gy Dose	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97	77.49
Fraction	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	62.95
Gender	2.00	27.00	0.00	0.00	1.00	1.00	1.00	1.00	100.00
Health Care Activity	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97	84.70
NOT FOUND	0.00	23.00	0.00	6.00	None	0.00	0.00	0.79	0.00
Number of Radiotherapy Fractions Per Day	1.00	27.00	0.00	1.00	1.00	0.50	0.67	0.97	83.19
Number of Radiotherapy Fractions Per Treatment	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97	66.29
Outcome	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	100.00
Person	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97	0.00
Planning Target Volume - Node	0.00	28.00	0.00	1.00	None	0.00	0.00	0.97	0.00
Prescribed Radiation Dose	1.00	27.00	0.00	1.00	1.00	0.50	0.67	0.97	69.70
Radiation Oncology Region of Interest	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97	74.87
TRG1 - Mandard	0.00	28.00	1.00	0.00	0.00	None	0.00	0.97	56.16
Target Volume	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	91.25
has alive last checked date	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	95.95
has clinical m stage	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	90.98
has clinical n stage	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	91.36
has clinical t stage	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	92.21
has date of death	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	91.56
has date of diagnosis	1.00	28.00	0.00	0.00	1.00	1.00	1.00	1.00	90.85
Total	18	745	11	9	16	15	15.33	26.31	None
Average	0.67	27.59	0.41	0.33	0.64	0.83	0.57	0.97	None

Overall, this study focused on the quality of ontology concept suggestions provided by string as well as NLP algorithm. Ultimately, as count of historical data increases per label, it becomes more reliable. In Table 4.7 we listed the average performance of all algorithms. In contrast, the recommendations made by the string and NLP algorithm will help to improve the performance of historical match algorithm. As recommendations in historical will be treated as ground truth, the performance values

of historical match in Table 4.7 are optimal. Historical match algorithm will also help the researchers to know the trend of concepts being used.

TABLE 4.7: Performance comparison of algorithms

Algorithm	Precision	Recall	F1
Historical Match	1	1	1
Jaro-Winkler	0.583	0.562	0.569
Levenshtein	0.573	0.58	0.566
NLP	0.615	0.576	0.589

Chapter 5

Software development

5.1 Introduction

An algorithm is normally not applicable in practice. It needs a software product to embed the algorithm, which hides the complexity and guides the user. To ease the end-user with access of the hybrid algorithm and have better utilization of outcomes, we have built a lightweight web interface. Using Flask framework, we have a minimalist web page which shows outcomes of prediction by each algorithm on the individual column level.

The code for the web-page and underlying algorithm can be found on https://github.com/nikhilg1312/Annotation-Recommendation-System. The technology stack and the architecture used for building the Web-interface in Figure 5.1.

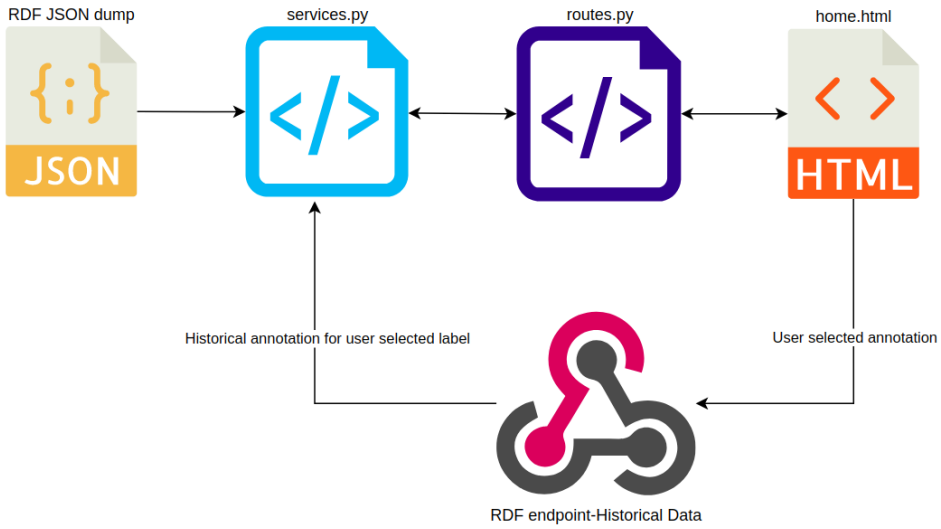

FIGURE 5.1: Architecture of User-Interface

Initially, we take the JSON dump of standardised ontology labels and database schema labels. Later we extract these dumps using ”services.py” and serve the required data elements as per user request in ”routes.py”. The routes.py script acts as a restful API. HTML pages are rendered as front-end. Open source RDF-endpoint acts as database for storing historical recommendations.

5.2 Home Page

For demonstration purpose, frame work uses cached list of database column labels on which the prediction of the annotation from standardised ontology is needed. This column list is derived from the schema ontology extracted by triplifier on SAGE database (Maastricht-Rome Collaboration) and can also be changed for other databases. The JSON response from private RDF endpoint of Maastro Hospital fetches all the required column labels. Fig 5.1 is the sample JSON dump of column labels. Appending integer id to the JSON response, we render individual column and use this id for by appending them to the URL for page traversals.

```
        "iri": {
          "type": "uri",
          "value": "http://localhost/rdf/ontology/Outcome.Geslacht"
        },
        "engLabel": {
          "xml:lang": "en",
          "type": "literal",
          "value": "Gender"
        },
        "label": {
          "type": "literal",
          "value": "Geslacht"
        }
```

Listing 5.1: Sample JSON data of ontology schema from SAGE DB

Also,the web page uses cached JSON from given ontology(in this case, Radiation Ontology Oncology(Traverso et al., 2018)) as standardised set of ontology. The JSON file can be changed as per requirement. The JSON dump of ontology includes concept label and associated meta-data. Figure 5.2 illustrates the home page. By clicking on any individual column, the researcher can traverse to individual column page for getting the recommendation of annotation for selected column label.

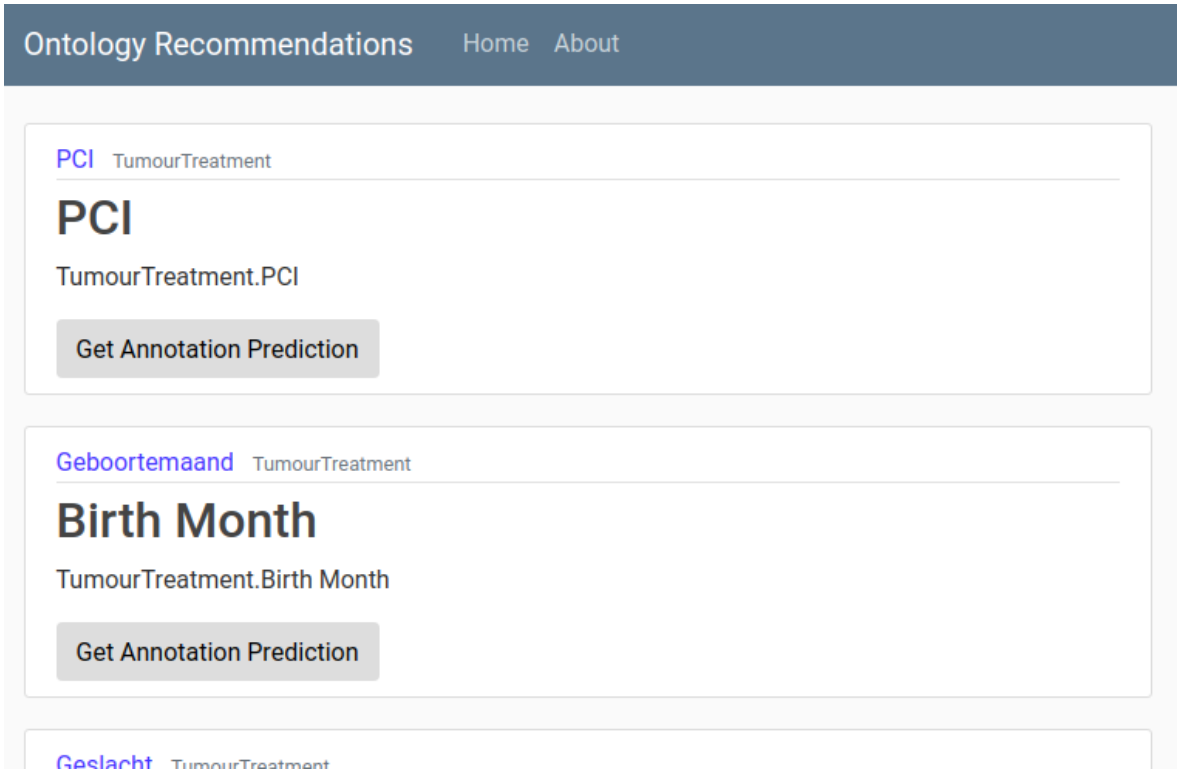

FIGURE 5.2: Homepage User-Interface

5.3 Column Page

Column page is rendered based on the user selection of a column on the homepage. Using the REST API, the meta-data for the column is sent from homepage to column page and used for making recommendations. Here we call the the prediction api of all the algorithms in our hybrid system Figure 3.6. In one api call, we get outcomes of all the three algorithms which are further rendered on column page. Accordingly, the user gets two options selecting appropriate annotation.

1. Manual self search in the ontology
2. Recommendations from Hybrid algorithm systems

By this approach, we give researcher liberty to select the most appropriate annotation apart from the recommendations made by the hybrid algorithm system. Furthermore, we are maintaining the history of these selected annotated recommendations.

Hence, the quality of historical recommendation will improve as the historical data increases. Figure 5.3 shows the recommendations made for column Gender. The framework needs to have unbiased results stored in history after selection of annotation. Hence, we do not mention the underlying algorithm for prediction made while rendering the results. Therefore, in Figure 5.3 we see that same annotation recommendation is made multiple times since both string and NLP algorithm made same prediction.

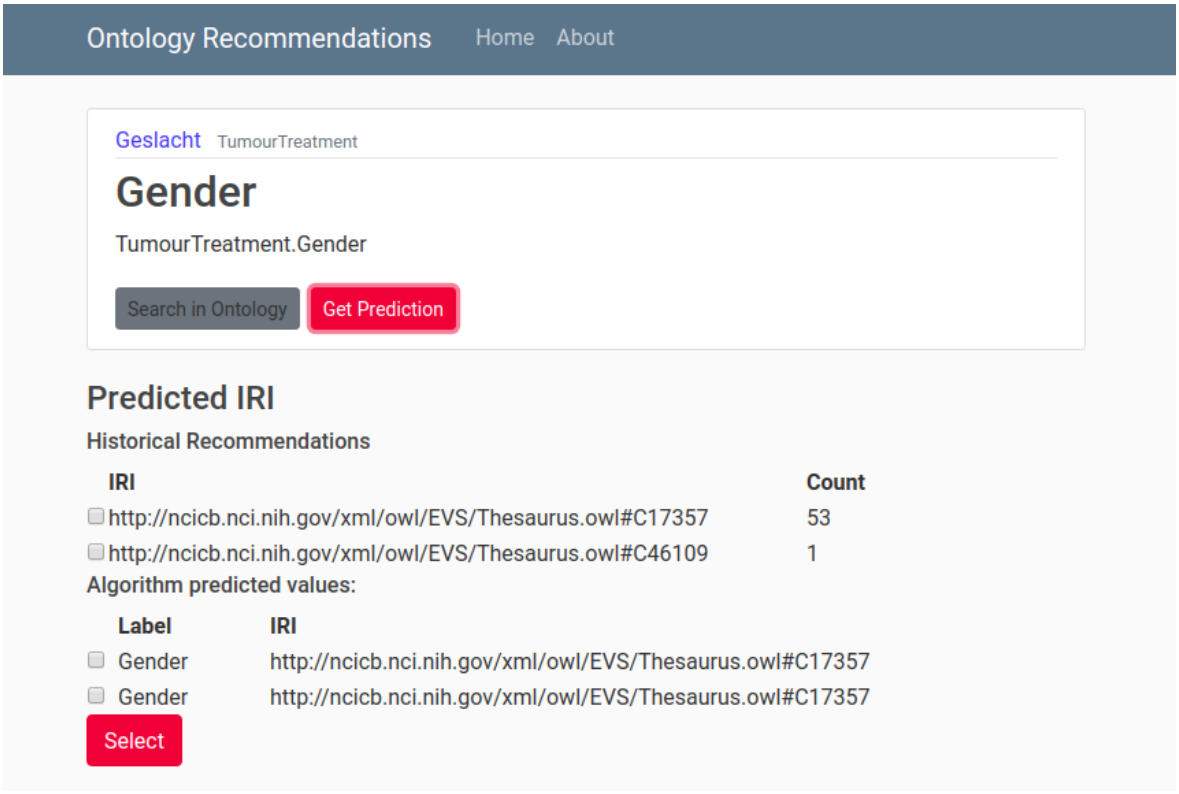

FIGURE 5.3: Annotation recommendations User-Interface

Fig 5.4 shows the User interface of manual search of concept directly in Standardised ontology. The user can select the appropriate annotation by using above options a get the query for inserting the annotation to RDF endpoint.

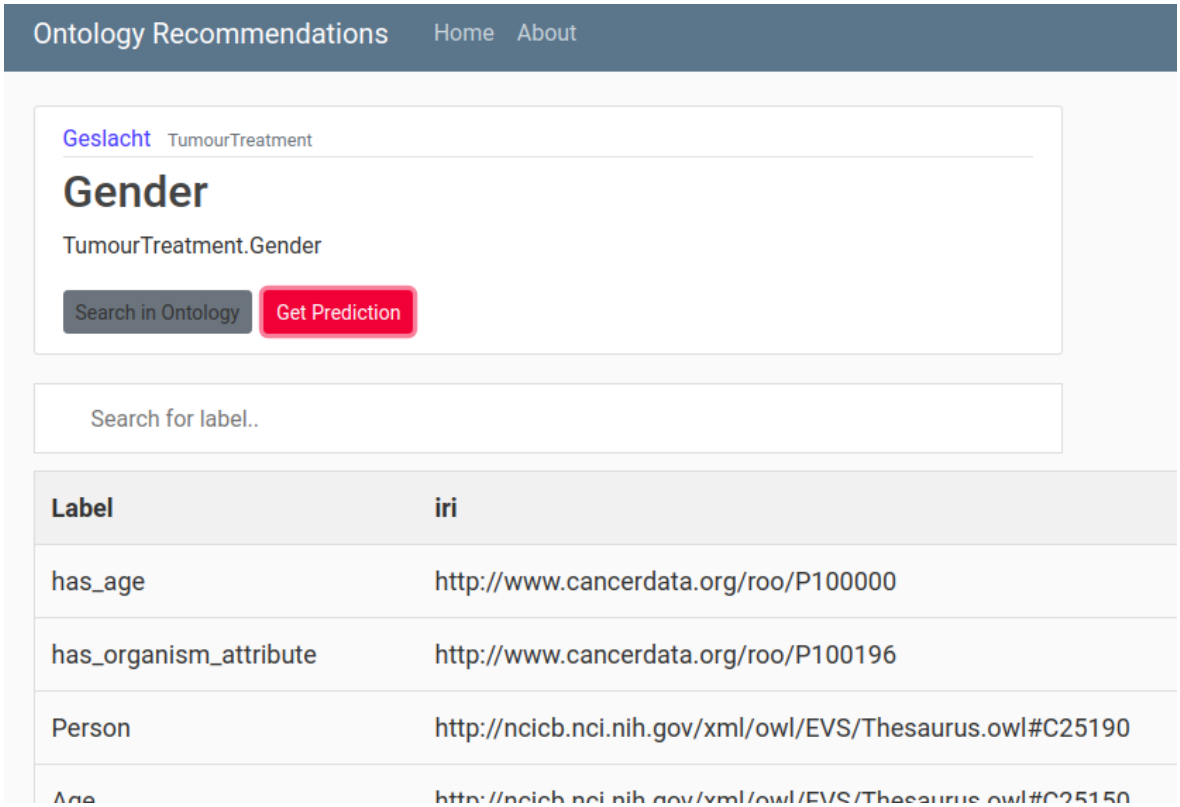

FIGURE 5.4: Manual search user-interface

Chapter 6

Discussion

By assisting the process of annotations for auto materialized RDF triples, we have demonstrated that it is possible to make data more FAIR. By substituting, the process of writing R2RML, we have isolated end-user from a complex workflow. Furthermore, by using the annotation recommendation web interface, we have shown that we can speed up the process of manual annotation using the schema ontology generated and assist the researcher with competences from different specializations.

Even though the R2RML pipeline solves many issues, it also has some shortcomings. The major bottleneck of the whole process is the tug of war between the researcher and the database administrator. In the initial setup of the RDF conversion pipeline, the researcher has to understand the data model of the data centre. As we are relying on SQL, a trivial change in the data model implies a change in the R2RML script. Therefore, maintaining this R2RML script is an overhead for the researcher as well as the database administrator.

In contrast, we have omitted this intermediate step of R2RML. Users of our framework do not need to understand this additional description language. The framework runs efficiently with minimal query time as it runs simple select all queries on the RDBMS tables.

The triple schema of the materialized data is robust to hold the inference rules, equivalent classes and literal values. Using the well-defined rich ontology and column labels with proper nomenclature, we notice the optimal performance of recommendations. Moreover, we take advantage of triple schema (Figure .3.2) and annotate these recommendations.

6.1 Answers to study research questions

Based on the evaluation of R2RML approach of the optimal way of converting relational data to RDF is to use the automated materialization tool. When used in combination with the recommendation system, this methodology becomes as good as a black box which does this conversion. The framework becomes user-friendly by hiding complex IT tasks for setup.

As the web-page is built on light weight Flask framework and automated triple conversion is less resource-intensive, we can have materialized triples readily. Having a virtual endpoint for triples at runtime or having an actual triple store will be dependent on use-case requirement and quantity of data.

The optimal methodology to have quality recommendations is to have it as a semi-automated methodology, and should include a hybrid selection of algorithms. String matching algorithms give us lexical matches while NLP gives us contextual matches. Furthermore, to optimize the performance time, we also give historical matches and improve the reusability of concepts.

Overcoming the hurdles IT competency, we empower researchers to have rich data as well as meta-data associated. It promotes their research work to be used for direct analysis as well as downstream research.

6.2 Strengths and Limitations

Strength and limitations of our study were totally dependent on the availability of concepts in standardised ontology and noise input column labels. For well-defined concepts like gender, date TNM staging the recommendation quality is optimal. In cases the standardised ontology does not hold the concept or when the input label has noise other than keyword, the quality of these recommendations drops. Also, the webpage has the fixed values of input column and standardised ontology labels. We can make this dynamic by further web development. For prediction, algorithm only compare the database column label and standardised ontology label. To increase the efficiency of NLP algorithm, along with database column label, the meta data associated should also be considered.

In case of historical repository of annotations, there is possibility that we might have saturation for a particular URI-label combination. This limits historical algorithm to be locked to URI-label combination.

Moreover, the combination of binary evaluation and contextual rating provide insight into the quality of annotation. The historical predictions made become much more relevant as the amount of historical predictions made for a label increases.

6.3 Results in relation to other similar works

Similar to our work we have Christen et al. algorithm which provide annotation recommendations for Medical forms defined in Unified Medical Language(UML) (Christen et al., 2015) . Similar to our workflow of algorithm, Christen et al.'s include a NLP model and semi-automatic annotation process. However, we think that the results cannot be compared as entities for annotation are database label in our case while a UML elements with different meta-data in their case.

6.4 Unanswered and New Questions

Currently, for conceptual matches, we do not differentiate between a class and a class property. We perform lexical or contextual matches between the input label and all the labels in the standardised ontology. We want to improve this prediction and suggest the user with the proper hierarchy of class and object.

The algorithm can only process English labels, both for input and output. Since the hospitals have international collaborations, the question of language barrier arises which can be tackled by implementing a language model built on biomedical data.

Chapter 7

Conclusion and Future work

Based on our research questions (section 1.5.2) the main findings of the frame-work are as follows:

1. It is possible to fully automate the process of conversion for tabular data into RDF triples by using DDLs of tables.

2. Annotation process of RDF triples cannot be fully automated as the perception of concepts defined in standardised ontology varies from researcher to researcher and may reduce the flexibility of choice.

3. Apart from computationally heavy algorithms, we need to reuse historical predictions. It will indirectly propagate the appropriate use of concepts from the standardised taxonomy by downstream users.

4. In case of the lexical match, irrespective of metrics like F1, precision and recall, recommendations for annotation needs manual intervention as string algorithms do the matching in a specific pattern (Jaro-Winkler gives more weight for matching first 4 characters).

7.1 Practical Relevance

Their is now a strong urgency in Medical research domain to make medical data FAIR. The concept of Findability, Accessibility, Interoperability and Reusability is relatively new, and there are no well-known standardized tools for this process. Most of the tools limit full capability usage as they need high level it proficiency, i.e. tools are "*Professorwares*" (Mons, 2018). Our study helps to bridge this gap by reducing the complexity of initial setups.

The terminology association can be done by the researcher, which enhances the FAIRification process, with limited efforts. Hence integrating international taxonomy into our framework will be the next logical step.

Smart data makes the algorithm smart. We emphasize the need for useful metadata for database labels as well as in standardized ontology. Researcher spends the majority of analysis time on unstructured datasets. Hence a well-annotated data is of great value.

7.2 Future Work

Most of the future work will be focusing on enhancing the framework's existing components. Before directly feeding the schema ontology file generated by triplifier to annotation recommendation system, We can have a user interface which reads this file and gather more meta-data associated with the schema and column contents. This will help to improve the efficiency of predictions by string and NLP algorithms. E.g. for a column with categorical value, the literals can also be easily annotated as they will be captured in the meta-data through this UI. Also, language tags can be associated with the labels, which can be a building block for the translation model.

A feedback mechanism can be built for the suggestions, to promote efficiency and quality. User could then rate the recommendations made, which will be stored in a history repository of annotations along with the label and opted IRI. This feedback can also be used as a metric for the next recommendations, which will help the future researcher. It could be more reliable and deciding factor as this a human-derived rating.

Furthermore, future studies should also focus more on conveying the perception of concepts defined in standardised ontologies. Sighting the advantages of FAIR data, most of the researcher are willing to move towards FAIRification of their results. Also, most of the research funding organisations demand FAIR data. The future study of the FAIR frameworks will help onboarding new researchers and give guidelines to existing ones.

7.3 Conclusion

The outcomes of our research work show that we have laid a more accessible track for the researchers to FAIRifiy their data. The suggestions given by hybrid recommendation algorithm have limitations, and hence we have given liberty to the researcher to

select their annotations manually. Improving this quality of recommendations will be a significant part of future works.

Bibliography

A Direct Mapping of Relational Data to RDF. URL: https://www.w3.org/TR/rdb-direct-mapping/ (visited on 09/13/2019).

Abernethy, Amy P. et al. (June 28, 2010). "Rapid-Learning System for Cancer Care". In: *Journal of Clinical Oncology* 28.27. Publisher: American Society of Clinical Oncology, pp. 4268–4274. ISSN: 0732-183X. DOI: 10.1200/JCO.2010.28.5478. URL: https://ascopubs.org/doi/full/10.1200/jco.2010.28.5478 (visited on 03/07/2020).

Ankolekar, Anshu et al. (Nov. 9, 2018). "The Benefits and Challenges of Using Patient Decision Aids to Support Shared Decision Making in Health Care". In: *JCO Clinical Cancer Informatics*. DOI: 10.1200/CCI.18.00013. URL: https://ascopubs.org/doi/pdf/10.1200/CCI.18.00013 (visited on 08/16/2019).

Baskar, Rajamanickam et al. (Feb. 27, 2012). "Cancer and Radiation Therapy: Current Advances and Future Directions". In: *International Journal of Medical Sciences* 9.3, pp. 193–199. ISSN: 1449-1907. DOI: 10.7150/ijms.3635. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3298009/ (visited on 02/18/2020).

Benson, Dennis A. et al. (Jan. 1, 2013). "GenBank". In: *Nucleic Acids Research* 41 (D1). Publisher: Oxford Academic, pp. D36–D42. ISSN: 0305-1048. DOI: 10.1093/nar/gks1195. URL: https://academic.oup.com/nar/article/41/D1/D36/1068219 (visited on 03/07/2020).

Berman, Helen, Kim Henrick, and Haruki Nakamura (Dec. 2003). "Announcing the worldwide Protein Data Bank". In: *Nature Structural & Molecular Biology* 10.12. Number: 12 Publisher: Nature Publishing Group, pp. 980–980. ISSN: 1545-9985. DOI: 10.1038/nsb1203-980. URL: https://www.nature.com/articles/nsb1203-980 (visited on 03/07/2020).

Berners-Lee, Tim. *The next web*. URL: https://www.ted.com/talks/tim_berners_lee_the_next_web (visited on 03/28/2020).

— (Nov. 7, 2000). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. 1 edition. San Francisco: Harper Business. 246 pp. ISBN: 978-0-06-251587-2.

bio.nlplab.org. URL: http://bio.nlplab.org/ (visited on 02/26/2020).

Bray, Freddie et al. (2018). "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries". In: *CA: A Cancer Journal for Clinicians* 68.6, pp. 394–424. ISSN: 1542-4863. DOI: 10.3322/caac.21492. URL: https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21492 (visited on 02/18/2020).

Choudhury, Ananya et al. (Nov. 1, 2019). "Distributed Analytics on Sensitive Medical Data: The Personal Health Train". In: *Data Intelligence*, pp. 96–107. DOI: 10.1162/dint_a_00032. URL: https://doi.org/10.1162/dint_a_00032 (visited on 11/08/2019).

Christen, Peter (Sept. 2006). *A Comparison of Personal Name Matching: Techniques and Practical Issues*. URL: http://users.cecs.anu.edu.au/~Peter.Christen/publications/tr-cs-06-02.pdf.

Christen, Victor et al. (2015). "Annotating Medical Forms Using UMLS". In: *Data Integration in the Life Sciences*. Ed. by Naveen Ashish and Jose-Luis Ambite. Vol. 9162. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 55–69. ISBN: 978-3-319-21842-7 978-3-319-21843-4. DOI: 10.1007/978-3-319-21843-4_5. URL: http://link.springer.com/10.1007/978-3-319-21843-4_5 (visited on 03/01/2020).

Cornet, Ronald and Nicolette de Keizer (Oct. 27, 2008). "Forty years of SNOMED: a literature review". In: *BMC Medical Informatics and Decision Making* 8 (Suppl 1), S2. ISSN: 1472-6947. DOI: 10.1186/1472-6947-8-S1-S2. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2582789/ (visited on 01/15/2020).

Crosas, Mercè (2011). "The Dataverse Network®: An Open-Source Application for Sharing, Discovering and Preserving Data". In: *D-Lib Magazine* 17.1. Publisher: Corporation for National Research Initiatives Section: D-Lib Magazine, p. 2. ISSN: 1082-9873. URL: https://dialnet.unirioja.es/servlet/articulo?codigo=3742561 (visited on 03/07/2020).

Deist, Timo M. et al. (June 1, 2017). "Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: euroCAT". In: *Clinical and Translational Radiation Oncology* 4. Publisher: Elsevier, pp. 24–31. ISSN: 2405-6308. DOI: 10.1016/j.ctro.2016.12.004. URL: https://www.ctro.science/article/S2405-6308(16)30027-1/abstract (visited on 03/31/2020).

Deist, Timo M. et al. (Mar. 1, 2020). "Distributed learning on 20 000+ lung cancer patients – The Personal Health Train". In: *Radiotherapy and Oncology* 144. Publisher: Elsevier, pp. 189–200. ISSN: 0167-8140, 1879-0887. DOI: 10.1016/j.radonc.2019.11.019. URL: https://www.thegreenjournal.com/article/S0167-8140(19)33489-9/abstract (visited on 03/31/2020).

Eriksson, Stefan and Gert Helgesson (Sept. 2005). "Potential harms, anonymization, and the right to withdraw consent to biobank research". In: *European Journal of Human Genetics* 13.9, pp. 1071–1076. ISSN: 1476-5438. DOI: 10.1038/sj.ejhg.5201458. URL: https://www.nature.com/articles/5201458 (visited on 09/13/2019).

FAIRification Process. GO FAIR. URL: https://www.go-fair.org/fair-principles/fairification-process/ (visited on 01/16/2020).

Haldar, Rishin and Debajyoti Mukhopadhyay (Jan. 6, 2011). "Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach". In: *arXiv:1101.1232 [cs, math]*. arXiv: 1101.1232. URL: http://arxiv.org/abs/1101.1232 (visited on 02/24/2020).

Hanahan, Douglas and Robert A Weinberg (Jan. 7, 2000). "The Hallmarks of Cancer". In: *Cell* 100.1, pp. 57–70. ISSN: 0092-8674. DOI: 10.1016/S0092-8674(00)81683-9. URL: http://www.sciencedirect.com/science/article/pii/S0092867400816839 (visited on 02/18/2020).

Igliński, Hubert and Maciej Babiak (Jan. 1, 2017). "Analysis of the Potential of Autonomous Vehicles in Reducing the Emissions of Greenhouse Gases in Road Transport". In: *Procedia Engineering*. 12th international scientific conference of young scientists on sustainable, modern and safe transport 192, pp. 353–358. ISSN: 1877-7058. DOI: 10.1016/j.proeng.2017.06.061. URL: http://www.sciencedirect.com/science/article/pii/S1877705817326073 (visited on 02/10/2020).

Jaro, Matthew A. (1989). "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida". In: *Journal of the American Statistical Association* 84.406, pp. 414–420. DOI: 10.1080/01621459.1989.10478785. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1989.10478785. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1989.10478785.

Jaro–Winkler distance (Jan. 29, 2020). In: *Wikipedia*. Page Version ID: 938177181. URL: https://en.wikipedia.org/w/index.php?title=Jaro%E2%80%93Winkler_distance&oldid=938177181 (visited on 02/24/2020).

Kalet, Ira J. and Mary M. Austin-Seymour (1997). "The Use of Medical Images in Planning and Delivery of Radiation Therapy". In: *Journal of the American Medical Informatics Association* 4.5, pp. 327–339. ISSN: 1067-5027. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC61250/ (visited on 02/18/2020).

Keizer, N. F. de, A. Abu-Hanna, and J. H. M. Zwetsloot-Schonk (2000). "Understanding Terminological Systems I: Terminology and Typology". In: *Methods of Information in Medicine* 39.1, pp. 16–21. ISSN: 0026-1270, 2511-705X. DOI: 10.1055/s-0038-1634257. URL: http://www.thieme-connect.de/DOI/DOI?10.1055/s-0038-1634257 (visited on 03/28/2020).

Kubben, Pieter (2019). "Data Sources". In: *Fundamentals of Clinical Data Science*. Ed. by Pieter Kubben, Michel Dumontier, and Andre Dekker. Cham: Springer International Publishing, pp. 3–9. ISBN: 978-3-319-99713-1. DOI: 10.1007/978-3-319-99713-1_1. URL: https://doi.org/10.1007/978-3-319-99713-1_1 (visited on 03/07/2020).

Lee, Tim Berners. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its inventor*.

Levenshtein, Vladimir Iosifovich (1966). "Binary codes capable of correcting deletions, insertions and reversals." In: *Soviet Physics Doklady* 10.8. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965, pp. 707–710.

Levenshtein distance (Feb. 16, 2020). In: *Wikipedia*. Page Version ID: 941053261. URL: https://en.wikipedia.org/w/index.php?title=Levenshtein_distance&oldid=941053261 (visited on 02/24/2020).

Maslow, Abraham Harold (Jan. 1966). *The Psychology of Science: A Reconnaissance*. First Edition edition. Place of publication not identified: HarperCollins. 168 pp. ISBN: 978-0-06-034145-9.

Mikolov, Tomas et al. (2013a). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf (visited on 02/17/2020).

Mikolov, Tomas et al. (Sept. 6, 2013b). "Efficient Estimation of Word Representations in Vector Space". In: *arXiv:1301.3781 [cs]*. arXiv: 1301.3781. URL: http://arxiv.org/abs/1301.3781 (visited on 02/25/2020).

Mons, Barend (Mar. 9, 2018). *Data Stewardship for Open Science: Implementing FAIR Principles*. CRC Press. 193 pp. ISBN: 978-1-315-35114-8.

Neumann, Mark et al. (2019). "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing". In: eprint: arXiv:1902.07669.

Ontology | metaphysics | Britannica. URL: https://www.britannica.com/topic/ontology-metaphysics (visited on 02/19/2020).

Open (FAIR) data. URL: https://www.nwo.nl/en/policies/open+science/data+management (visited on 02/14/2020).

Over Maastro. Maastro. URL: https://maastro.nl/over-maastro/ (visited on 01/15/2020).

OWL Web Ontology Language Guide. URL: https://www.w3.org/TR/owl-guide/ (visited on 02/19/2020).

OWL Web Ontology Language Reference. URL: https://www.w3.org/TR/owl-ref/#PropertyLogic (visited on 01/16/2020).

OWL Web Ontology Language Use Cases and Requirements. URL: https://www.w3.org/TR/2004/REC-webont-req-20040210/ (visited on 02/19/2020).

Peduzzi, P. et al. (Dec. 1996). "A simulation study of the number of events per variable in logistic regression analysis". In: *Journal of Clinical Epidemiology* 49.12, pp. 1373–1379. ISSN: 0895-4356. DOI: 10.1016/s0895-4356(96)00236-3.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: http://aclweb.org/anthology/D14-1162 (visited on 02/26/2020).

Pollock, Joseph J. and Antonio Zamora (Apr. 1, 1984). "Automatic spelling correction in scientific and scholarly text". In: *Communications of the ACM* 27.4, pp. 358–368. ISSN: 0001-0782. DOI: 10.1145/358027.358048. URL: https://doi.org/10.1145/358027.358048 (visited on 02/24/2020).

Radiation Oncology. Radiation Oncology. URL: https://ro-journal.biomedcentral.com/about (visited on 02/18/2020).

Ratte, Jean-Bernard. *pyjarowinkler: Find the Jaro Winkler Distance which indicates the similarity score between two Strings*. Version 1.8. URL: https://github.com/nap/jaro-winkler-distance (visited on 03/28/2020).

RDF Schema RDFS - Introduction to ontologies and semantic web - tutorial. URL: https://www.obitko.com/tutorials/ontologies-semantic-web/rdf-schema-rdfs.html (visited on 01/20/2020).

Regression modelling strategies for improved prognostic prediction - Harrell - 1984 - Statistics in Medicine - Wiley Online Library. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780030207 (visited on 09/12/2019).

Rong, Xin (June 5, 2016). "word2vec Parameter Learning Explained". In: *arXiv:1411.2738 [cs]*. version: 3. arXiv: 1411.2738. URL: http://arxiv.org/abs/1411.2738 (visited on 02/25/2020).

seatgeek/fuzzywuzzy (Mar. 28, 2020). original-date: 2011-07-08T19:32:34Z. URL: https://github.com/seatgeek/fuzzywuzzy (visited on 03/28/2020).

Sensitivity and specificity (Mar. 16, 2020). In: *Wikipedia*. Page Version ID: 945919083. URL: https://en.wikipedia.org/w/index.php?title=Sensitivity_and_specificity&oldid=945919083 (visited on 03/25/2020).

"The world's most valuable resource is no longer oil, but data". In: *The Economist* (). ISSN: 0013-0613. URL: https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data (visited on 02/10/2020).

Traverso, Alberto et al. (Oct. 2018). "The radiation oncology ontology (ROO): Publishing linked data in radiation oncology using semantic web and ontology techniques". In: *Medical Physics* 45.10, e854–e862. ISSN: 2473-4209. DOI: 10.1002/mp.12879.

Van Soest, Johan. *Annotation of existing databases using Semantic Web technologies: making data more FAIR*. URL: http://www.swat4ls.org/workshops/edinburgh2019/programme/accepted-papers/ (visited on 10/16/2019).

Wang, Dayong et al. (June 18, 2016). "Deep Learning for Identifying Metastatic Breast Cancer". In: *arXiv:1606.05718 [cs, q-bio]*. arXiv: 1606.05718. URL: http://arxiv.org/abs/1606.05718 (visited on 02/10/2020).

What Are Classes And Individuals? URL: http://www.linkeddatatools.com/help/classes (visited on 01/21/2020).

Wilkinson et al. (Mar. 15, 2016). "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific Data* 3, p. 160018. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18. URL: https://www.nature.com/articles/sdata201618 (visited on 11/29/2018).

Winkler, William E (1990). "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage." In: p. 9. URL: https://files.eric.ed.gov/fulltext/ED325505.pdf.

Yancey, William E. "Evaluating String Comparator Performance for Record Linkage". In: (), p. 42. URL: https://www.census.gov/srd/papers/pdf/rrs2005-05.pdf.