Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Abderrahmane Mira Bejaia
Faculty of exact sciences
Computer science department

Graduation thesis

To obtain an academic master's degree in Advanced Information Systems

Theme

Healthcare Big Data Warehouse integration

Presented by:

BOUAMRA Abdelbari.

Evaluated By:

Dr BATTAT Nadia: **President**
Dr CHIBANI Samia Wife SADOUKI : **Examiner**
Dr EL BOUHISSI Houda Wife BRAHAMI : **Superviser**

Promotion 2022/2023

In the name of Allah the Merciful

Thanks

I would like to express my deepest gratitude to my parents for their unconditional love and moral and financial support throughout my studies. Their sacrifices and their confidence in me have been an essential driving force in my academic success. I am grateful and honored to have them as parents, and I dedicate this work to them as well. Thank you so much for everything you have done for me. I would also like to express my deep gratitude to my supervisor, ***ELBOUHISSI Houda****, for her support and guidance throughout this project. Her sound advice, patience, and availability were essential to the success of this research.*

Dedications

I would like to thank Allah, the Compassionate and Merciful for giving me the strength and courage to complete this modest work. I dedicate this work to : My dearest parents, who guided me through the most difficult moments of this long journey. My mother was by my side and supported me throughout my life, and my father sacrificed himself to see me become what I am. May God the Almighty protect and keep them. My brothers: Abderrahim, Abdelmalek, Mehdi. My sisters: Raoua and Louiza. To my friends: Hichem, Abderrahim, Nassim, Youba, Abdou 03, Hakim.

Abdelbari

Résumé

De nos jours, les machines physiques connectées gèrent chaque jour une quantité très importante et variée de données appelées Big Data. Ces données proviennent de nombreuses sources hétérogènes et nous les utilisons à des fins diverses telles que la prise de décision, l'aide au traitement médical, le diagnostic médical, mais aussi l'accès rapide et pertinent aux données, etc. Cela a posé un défi majeur aux entreprises qui devraient faire face au problème du stockage, de l'analyse, du traitement et surtout de l'intégration des données. C'est pourquoi les entreprises ont besoin de nouveaux outils et techniques tels que l'utilisation d'ontologies pour l'intégration et l'interopérabilité des données afin de faire face aux difficultés d'intégration. Ces ontologies sont définies comme une spécification explicite et formelle d'une conceptualisation partagée entre les humains et les machines.

Notre mémoire de master passe en revue les approches les plus importantes en matière d'intégration de données et propose une nouvelle méthodologie qui intègre des sources de données multiples en utilisant des ontologies et l'apprentissage automatique afin de faciliter et d'améliorer la compréhension des données.

Mots clés : Intégration de données, Big Data, Interopérabilité, ontologies, Apprentissage automatique

Abstract

Nowadays, connected physical machines manage vast and diverse amounts of data, often referred to as Big Data. This data originates from numerous heterogeneous sources and serves various purposes, including decision-making, medical treatment support, diagnosis, and enabling fast and relevant data access, among others. This has presented a significant challenge for companies, as they grapple with issues related to data storage, analysis, processing, and, most notably, data integration.For this reason, companies need new tools and techniques, such as the use of ontologies for data integration and interoperability, to cope with integration difficulties. These ontologies are formally defined as explicit specifications of a shared conceptual understanding that can be interpreted by both humans and machines.

Our master thesis surveys the most important approaches to data integration and suggests a new methodology that integrates multiple data sources by using ontologies and machine learning, to facilitate and enhance data comprehension.

keywords: Data integration, Interoperability, ontologies, Big Data, machine learning

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abreviations list

AD	Alzheimer's disease
ANN	Artificial Neural Networks
API	Application Programming Interface
BD	Big Data
BGP	Basic Graph Patterns
CNN	Convolution Neural Networks
CSV	Comma-separated values
CT	Computed Tomography
COVID – 19	COrona VIrus Disease
DiiS	Data integration and indexing System
DQA	Data Quality Assessment
EHR	Electronic Health Record
EMR	Electronic Medical Record
ETL	Extract Transform Load
HDFS	Hadoop Distributed File System
HGSOC	High-Grade Serous Ovarian Carcinoma
HTML	Hypertext Markup Language
IA	Artificial Intelligence
ISAAC	International Society for Augmentative and Alternative Communication
IT	Information Technology
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
KNN	k-Nearest Neighbors
LIS	Laboratory Information System
ML	Machine learning
MIDAS	Medical Information Data Analysis System
MRI	Magnetic Resonance Imaging
MRS	Magnetic Resonance Spectroscopy
MSKCC	Memorial Sloan Kettering Cancer Center
NOSQL	Not only SQL
OWL	Web Ontology Language
PACS	Picture Archiving and Communication System
PDF	Portable Document Format
RDFS	Resource Description Framework
RDFS	Resource Description Framework Schema
RML	RDF Map Language
SPARQL	SPARQL Protocol And RDF Query Language
URL	Uniform Resource Locator
XML	Extensible Markup Language

Chapter 1

General Introduction

1.1 Introduction

The integration of massive data, also known as ”Big data integration”, is an essential process in the field of IT and data analysis. With the exponential growth in the volume, variety, and speed of data generated by businesses and users around the world, it is imperative to develop methods and technologies to manage and exploit this data effectively.

One of the major sectors affected by the integration of massive data is the healthcare domain. This integration plays a critical role in management, analysis, and informed decision-making in the medical sector. In the age of increasing digitization, medical data is produced on a massive scale and comes from a variety of sources such as electronic medical records, connected medical devices, laboratories, medical images, clinical research, and many others. Integrating this diverse and often voluminous data enables healthcare professionals to harness in-depth knowledge to improve patient care, medical research, and resource management.

Big data integration in healthcare aims to bring together, standardize and structure this disparate data from a variety of sources. This creates a complete and accurate picture of a patient's state of health, enabling more informed medical decisions to be made. This integration also facilitates communication between the various players in the healthcare sector, such as doctors, nurses, pharmacists, and hospital managers, by providing them with transparent access to relevant information. The integration of this data also facilitates the coordination of care, reduces medical errors, and improves the effectiveness of treatments.

In addition, the integration of massive data in the medical sector plays a crucial role in medical research. Scientists can aggregate and analyze large datasets to identify trends, patterns, and potential risk factors linked to certain diseases. This can accelerate the development of new therapies, medicines, and personalized treatment protocols.

However, data integration in healthcare also presents significant challenges such as Data velocity and security. Data security and confidentiality must be a priority in order to protect sensitive patient information from possible breaches. In addition, the standardization of data formats, the resolution of identity conflicts, and the management of updates are all complex issues that require particular attention to guarantee data quality and integrity.

1.2 Problematic

The integration of Big Data gives rise to a number of complex and interdependent issues that can impact the success and effectiveness of initiatives linked to its exploitation. A number of issues need to be taken into account such as the volume of data, the variety of data that comes from heterogeneous sources, interoperability so that systems and tools used to collect, store and analyze Big Data must be able to work together smoothly and efficiently, data analysis and exploitation by collecting and storing Big Data only makes sense if it is followed by analysis to obtain relevant information, and also the ability to extract meaningful knowledge from this data represents a major challenge.

These issues are closely linked and complex, requiring a global approach, from strategic planning to operational implementation, for effective management.

1.3 Objectives and contributions

The present thesis builds upon the proposal by **Elbouhissi et al.** [1], which focuses on the integration of Big Data. In the following points, we will summarize our contributions

The main contributions of the thesis are as follows:

- Conducting a comprehensive review of the major works related to the integration of Big Data within the scope of our study project.
- Introducing a novel approach to Big Data integration through the utilization of ontologies and Machine Learning techniques.
- Implementing a Machine Learning algorithm to calculate similarities, facilitating the creation of local ontologies.
- Constructing a global ontology by amalgamating local ontologies, enhancing data comprehension, interpretation, and consistent utilization.

1.4 Methodology

In particular, our work is based on the following steps:

- **Research and analysis:** We carried out an in-depth search of the state of the art of the various approaches proposed by researchers in the field of data integration, comparing the techniques used and the advantages of each approach.
- **Proposed solution:** propose a new effective approach that can deal with the problems associated with integrating Big Data.
- **Implementation and experimentation:** Implement the proposed solution as well as carry out experiments.

1.5 Thesis Organisation

The remainder of the thesis is organized as follows:

- **Chapter 2:** This chapter presents general concepts about the topic including the definition of Big Data, its characteristics, the sources of Big Data in the health sector, and the relation between Big Data and health. In addition, this chapter, presents the definition of ontology and its components. Finally, this chapter highlights the concept of Extract, Transform and Load (ETL).
- **Chapter 3:** This chapter reviews the main related works regarding Big Data integration.
- **Chapter 4:** This chapter presents the proposed approach in detailed, including the system architecture and the various stages involved.
- **Chapter 5:** This chapter details the technical aspects of our proposal, as well as the software and hardware environments used during this project, and finally the evaluation of the proposed solution.
- **Chapter 6:** Finally, this thesis is concluded by chapter 6 which provides a summary and assessment of the work carried out throughout the thesis, and also a set of perspectives for the continuation of this work.

Chapter 2

Fundamental concepts

2.1 Introduction

Big data integration in the healthcare sector is an evolving approach that uses high-level technologies to process, manage and analyze vast quantities of complex data generated within the industry. It involves aggregating and combining data from a variety of sources, such as electronic medical records, medical devices, patient-generated data, research studies, and administrative systems. By integrating these diverse data sets, healthcare organizations can gain valuable insights and models to improve patient care, personalize treatments, enhance clinical decision-making, and increase operational efficiency. However, as big data integration poses significant challenges around data privacy, security, and interoperability, implementation requires a robust infrastructure, data governance policies, and data sharing agreements to ensure a successful and ethical application in the healthcare ecosystem.

In this chapter, we will present some definitions and basic notions about data integration and its relation to the health domain. We will also provide a brief overview of ontologies, their elements, and machine learning since they are the focal points of our study.

2.2 Big data

2.2.1 Definition

The term ”big data” refers to datasets that are too large, complex, or fast-paced to be effectively perceived, acquired, managed, and processed using traditional IT and software/hardware tools within an acceptable timeframe [2].

According to Barton [3], big data refers to extremely large datasets that are generated from various sources such as environmental sensors, smart devices, electronic medical records, imaging, laboratory studies, and administrative data.

There are different sources of healthcare data such as (Figure 2.1:
- Medical Claims.
- Mobile data and wearables.
- Academic researchers.
- Hospital EHR.
- Opt-in genome registries.
- Government medical claims.
- Pharmacy claims.

Figure 2.1: Sources of big data
[4]

2.2.2 Characteristics

According to [5], Big Data is characterized by "5V" rules (Figure 2.2):

1. **Volume:** Able to process very large amounts of data from various sources.
2. **Variety:** Able to process data in different formats which can be collected from videos, images, text, etc. It can be structured or unstructured data.
3. **Velocity:** Able to process information in real-time and refers also to the speed of data transfers.

4. **Veracity:** Able to evaluate the reliability of data and accuracy of information.
5. **Value:** Able to focus on data of real value from large sets of data.

Figure 2.2: Characteristics of big data
[6]

2.2.3 Big Data in Healthcare

The healthcare system is a complex, multi-dimensional structure designed to prevent, diagnose, and treat health-related issues in humans. It comprises various components such as health professionals, health facilities, and financing institutions. Health professionals come from different sectors like dentistry, medicine, nursing, psychology, and more.

Healthcare services are provided at different levels based on the urgency of the situation, ranging from primary care to quaternary care involving rare diagnostic or surgical procedures. Health professionals handle different types of information, including patient medical history, medical and clinical data from imaging and laboratory exams, and other private medical data. This is why the healthcare sector requires a robust big data repository to effectively manage and store these diverse types of information [7].

2.3 Ontologies

Ontology (the ”science of being”) is a word, like metaphysics, that is used in many different senses. It is sometimes considered to be identical to metaphysics, but we prefer to use it in a more specific sense, as that part of metaphysics that specifies the most fundamental categories of existence, the elementary substances or structures out of which the world is made. Ontology will thus analyse the most general and abstract concepts or distinctions that underlie every more specific description of any phenomenon in the world, e.g. time, space, matter, process, cause and effect, system.

Recently, the term of ”(formal) ontology” has been up taken by researchers in Artificial Intelligence, who use it to designate the building blocks out of which models of the world are made.(see e.g. ”What is an ontology?”). An agent (e.g. an autonomous robot) using a particular model will only be able to perceive that part of the world that his ontology is able to represent. In a sense, only the things in his ontology can exist for that agent. In that way, an ontology becomes the basic level of a knowledge representation scheme. See for example my set of link types for a semantic network representation which is based on a set of ”ontological” distinctions: changing-invariant, and general-specific.

2.3.1 Definition

Gruber defines ontology as “a specification of a shared conceptualization of a domain” [8].

An ontology can be described as a structured representation of a particular domain, which includes elements such as objects, properties, and relationships. This formal abstraction is typically organized in a hierarchical manner [9].

2.3.2 Ontology Components

An ontology can be viewed as a 5-tuple, consisting of five components: Concepts (**C**), Relationships (**R**), Functions (**F**), Instances or Individuals (**I**), and Axioms (**A**) [10].

- **Concepts:** refers to the main formalized elements of the domain [9], which are described using specific properties that must be satisfied by them [11].

- **Relationships:** represent the links between the concepts, and are used to represent the structure of the ontology, whether it is taxonomic or not.

- **Functions:** are elements that calculate information from other elements within the ontology.

- **Instances:** refers to the main objects within the domain, as represented by the ontology structure.

- **Axioms:** are the rules, restrictions, and logic correspondences that must be satisfied in the relationship between the elements of the ontology [12].

2.3.3 OWL

OWL stands for Ontology Web Language, and it comprises a group of languages designed for the purpose of creating and exchanging ontologies. OWL builds upon the RDF and RDFS vocabulary, but offers significantly greater expressiveness in its ability to define classes and properties [13].

2.3.4 RDFS

RDFS, short for RDF Schema, constitutes a collection of vocabulary elements designed for modeling data in RDF. RDFs itself is a semantic extension of RDF. The role of RDFS is to define and describe resource categories, including classes and relationships between resources, as well as attribute categories, encompassing properties, and to systematically arrange them into hierarchical structures. These schemas are likewise exchanged and shared through the RDF format.

RDFS is used to define simple ontologies, i.e. lightweight ontologies [13].

2.3.5 SPARQL

SPARQL, short for SPARQL Protocol and RDF Query Language, serves as a structured and semantic query language for RDF knowledge bases. It functions as a means to access information within the Web of Data. Within a SPARQL query, there exists a collection of triple patterns known as basic graph patterns (BGPs). These patterns are employed to locate a portion or subgraph of data from the queried RDF data or RDF graph.

The outcomes of SPARQL queries can manifest as sets or RDF subgraphs. Each triple pattern in a BGP consists of RDF triples, where the subject, predicate, and object can either be unknown or represented by a variable denoted with a question mark, like ”?e,” with ”e” being the variable [13].

The SPARQL query structure is shown in the Figure 2.3 :

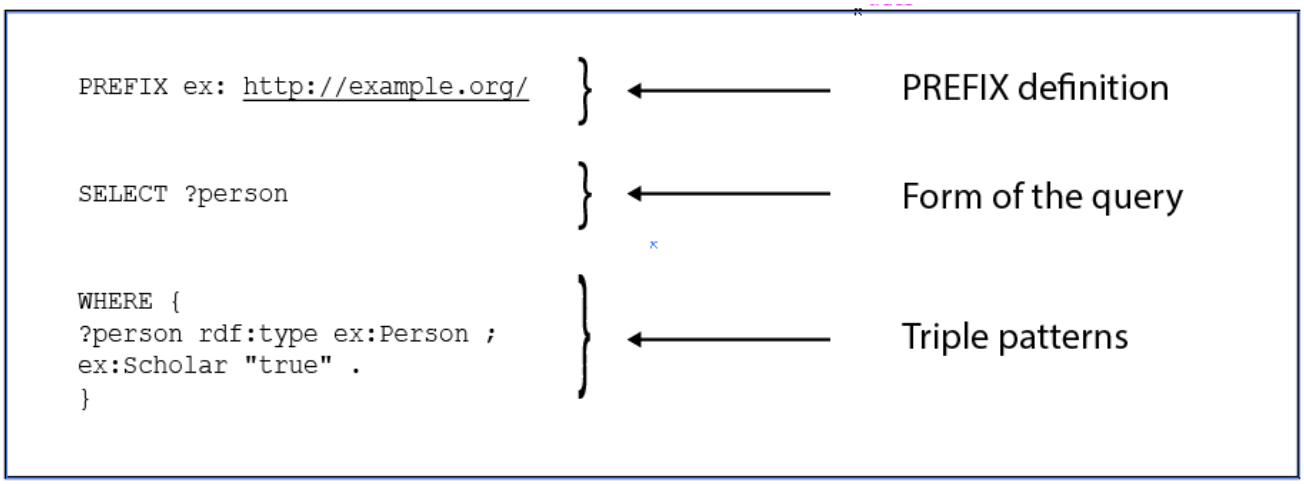

Figure 2.3: SPARQL query structure

2.4 Machine learning

Machine Learning is a discipline of artificial intelligence that is revolutionizing the way computers can learn and acquire skills through experience, just as a human being learns new skills by observing and interacting with the world around them. It is a powerful approach that enables computer systems to discover patterns and relationships in data and use them to make decisions, make predictions, or solve complex problems.

2.4.1 Definition

Arthur Samuel defined machine learning as “a field of study that gives computers the ability to learn without being explicitly programmed” [14].

2.4.2 Machine Learning types

Machine learning is divided into four categories [15] [16] (Figure 2.4):

- **Supervised learning:** Supervised learning is a type of machine learning where the goal is to learn a function that can map inputs to outputs based on a set of input-output pairs. This approach relies on labeled training data and a collection of examples to create a function. The purpose of supervised

learning is to accomplish certain goals from a specific set of inputs, making it a task-driven approach. The two most common types of supervised learning are classification, which splits the data, and regression, which adjusts the data.

- **Unsupervised learning:** it involves analyzing datasets without the use of labeled data or human intervention. This approach is based on a data-driven process and is commonly used for extracting generative features, identifying trends and structures, grouping results, and for exploratory purposes. The most common types of unsupervised learning are clustering and association.

- **Semi-supervised learning:** is a type of machine learning that lies between unsupervised and supervised learning approaches. In many real-world scenarios, labeled data may be scarce, while unlabelled data is abundant. Semi-supervised learning is useful in these situations, as it allows the model to use both labeled and unlabelled data to improve its performance. By exploiting both labeled and unlabelled data, semi-supervised learning can achieve better accuracy than unsupervised learning while requiring less labeled data than supervised learning.

- **Reinforcement learning:** it allows machines and software agents to evaluate automatically the optimal actions or improve their performance. This sort of learning is focused on reward or punishment, and its ultimate purpose is to use environmental activists' insights to take action to raise the reward or reduce the risk.

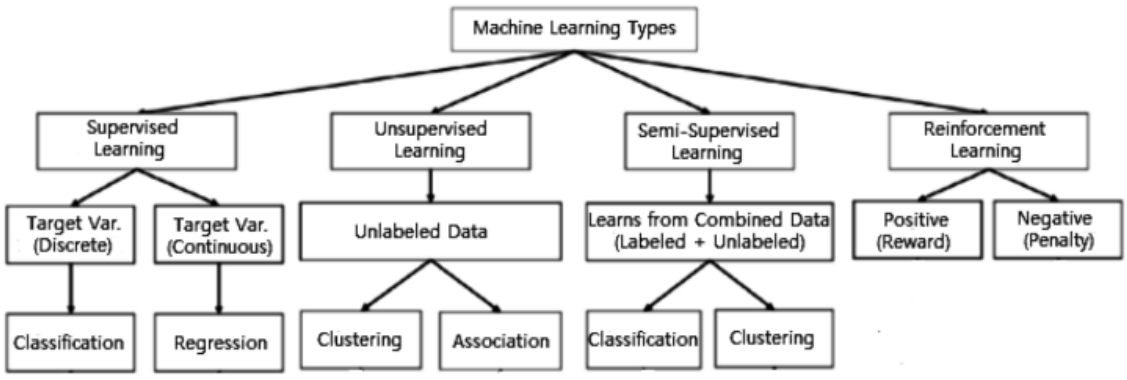

Figure 2.4: Types of machine learning
[16]

2.5 ETL

ETL means Extract, Transform, Load. It is the process of extracting data from various sources and transforming it and finally loading it onto the Data warehouse [17] (Figure 2.5).

- **Extract:** consists of extracting data from different sources these data may be in different formats like XML, CSV, files, or relational databases.
- **Transform:** During this process, the data that is extracted is often transformed into a format that is suitable for the user or client. As a result, the data is no longer in its original format and may require some cleaning, mapping, and transformations to be processed effectively. In some cases, additional customization steps or operations such as aggregation may also be performed to ensure that the data is in a format that is useful for the user's needs .
- **Load:** The load process is the final stage of the ETL (Extract, Transform, Load) process, where the transformed data is written into the destination or final database. This stage involves loading a large volume of data into the final database, and it is a critical step to ensure that the data is accurate, complete, and consistent with the desired format.

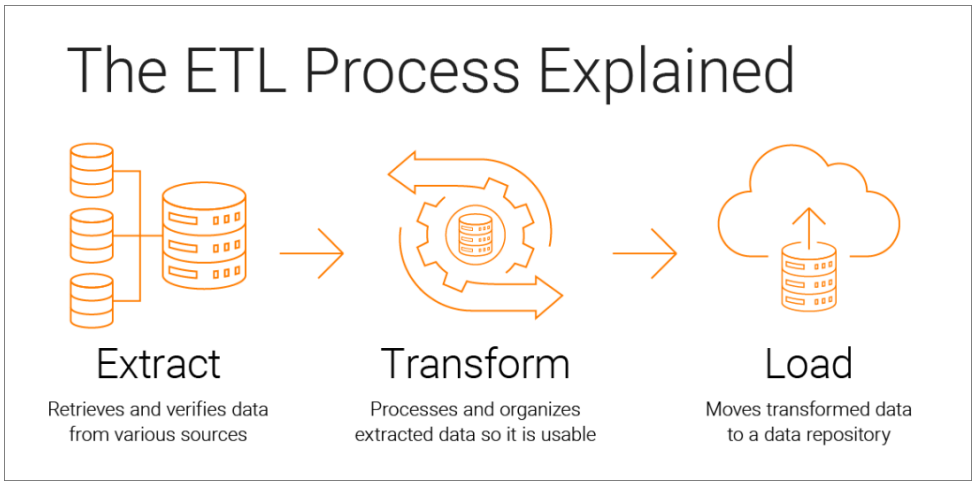

Figure 2.5: ETL process
[18]

2.6 Conclusion

The continuous growth in the volume of data in different sectors such as the health sector has led to the development of new techniques that can offer reliable, accurate, and accessible data at any time.

The notions we have discussed in this chapter allow us to understand the principle of data integration and therefore give us the possibility to propose solutions for the problems posed.

In the following chapter, we will study some works from the literature that have addressed the different approaches and techniques that concern big data integration.

Chapter 3

State of the art

3.1 Introduction

The incorporation of Big Data signifies a significant leap in how we capture, control, and exploit the immense quantities of data produced in the modern digital environment. As global connectivity grows, companies and institutions encounter the task of consolidating and synchronizing various data origins to unveil practical insights and formulate strategic choices.

The integration of big data encompasses a intricate procedure of seamlessly combining information from varied sources, encompassing conventional databases, cloud services, social media, sensors, and other platforms.

This chapter is dedicated to the presentation of some methods and techniques proposed in the literature that address the problem at hand.

We start in this section with a thorough literature search, including academic papers, and conference proceedings from reliable sources. Our search is guided by specific keywords related to data integration, health care, and ontologies, it describes the ensemble of data integration works that we have studied and which correspond to our work.

3.2 Related works

The proposed solutions can be classified into three classes (figure 3.1), the first being methods that use machine learning. The second is methods using ontologies. The third is works that use other techniques such as short-distance models, MapReduce, etc.

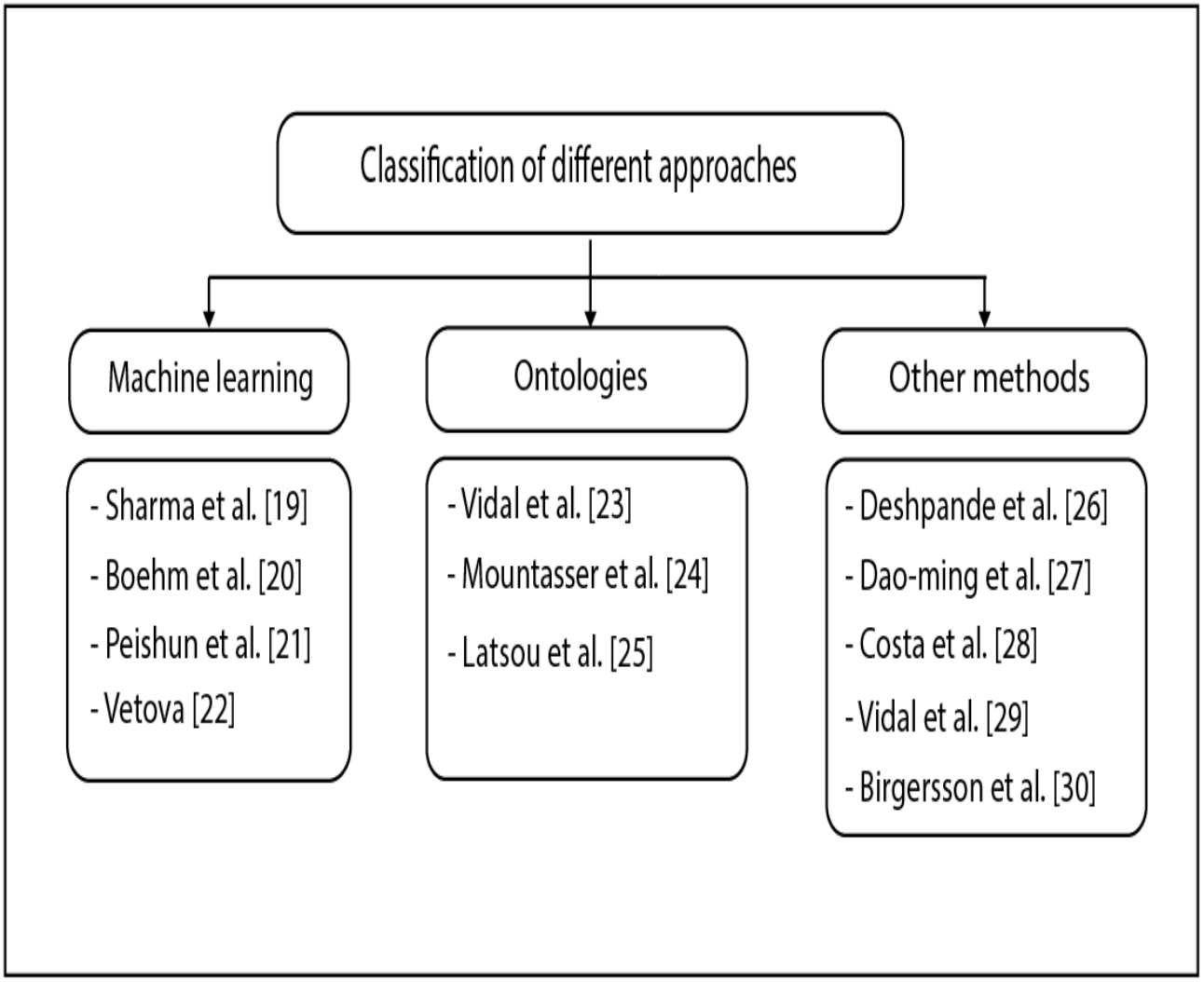

Figure 3.1: Classification of approaches

This section presents the main research works related to data integration.

3.2.1 Machine learning based approaches

Sharma et al. [19], proposed a Hadoop-based Big Data analytics framework for Alzheimer's disease (AD) classification and progression. This framework simplifies the integration of a huge amount of heterogeneous data, it uses organized data and several techniques for distributed storage and also algorithms based on machine learning, it consists of four components, the first is data normalization this component is responsible for organizing the heterogeneous data acquired from different modalities, such as using MedCon for MRI image conversion and a plugin for MRS data processing and exporting it into XML formats. The neuropsychological evaluations are downloaded in CSV format for further processing. The second component is data management, which includes tools for

user interaction using back-end and front-end (HUE, Apache Zookeeper, Oozie) systems, these systems help accessing, querying and synchronization of information. The third component is data storage using HDFS and Metadata to store a huge volume of data. The fourth and last component is data processing which contains checking quality of matrix, feature extraction and selection, decision and fusion, statistical analysis which validates the results of this framework.

Boehm et al.[20], assembled a multimodal data integration using machine learning to improve risk stratification of high-grade serous ovarian carcinoma (HGSOC). The study included 444 patients which were divided into two groups based on their risk of recurrence: low-risk and high-risk. The authors used machine learning to train models to predict which patients were in each risk group. The models were trained on data from the patients' histopathological slides, their CT (Computed Tomography) scans, and their clinical and genomic data. The authors found that these features contributed complementary prognostic information relative to one another and clinicogenomic features. The authors were able to enhance risk classification of HGSOC patients by combining histopathological, radiologic, and clinicogenomic machine-learning models which could lead to better treatment decisions and improved outcomes for patients.

Peishun et al. [21], argue that machine learning is a powerful tool for identifying patterns and relationships in complex data sets and that it can be used to improve our understanding of the gut microbiome and its role in human health. They begin by reviewing the different types of multi-omics data that can be collected from the gut microbiome. These include metagenomics, metatranscriptomics, metabolomics, and proteomics. Each of these data types provides a different perspective on the gut microbiome, and together they can be used to build a more comprehensive understanding of the microbiome's composition, function, and interactions with the host. After that, they discuss the challenges of analyzing multi-omics data which include the high dimensionality of the data, the presence of noise, and the lack of ground truth labels. Machine learning can be used to address these challenges by reducing the dimensionality of the data, filtering out noise, and identifying patterns that are not easily visible to the human eye. After that, they review the different machine-learning approaches that have been used to analyze gut microbiome data like random forest, gradient boosting, and decision tree, deep neural network. These approaches include supervised learning, unsupervised learning, and reinforcement learning. The authors argue that supervised learning is the most effective approach for tasks such as classification and prediction, while unsupervised learning is more effective for tasks such as clustering and feature selection. They conclude that machine

learning is a promising tool for integrating multi-omics data and identifying microbial biomarkers for disease.

Vetova [22], proposed a model of big data integration and processing system in the domain of biomedicine (COVID-19), this model consists of three sections. The first section is Information Organization, which enables the collection of biomedical data from clinical data sources, image formats, etc. Then store these data in file formats (.xls, .xlsx, .csv, .xlsm), and finally data integration process including cleansing, ETL, mapping, and transforming by using a tool called Talend open studio. After that, comes the second section which is information processing which consists of two phases, first is data and image processing which involves extracting meaningful information by performing operations and functions and manipulating the collected data. The second phase is the classification of the biomedical data by using KNN, K-means, and mean shift clustering to classify data, ANN (artificial neural networks), and CNN (convolution neural networks) to classify images. The last section is problem-solving by making decisions it uses various methods such as Machine Learning, decision support systems, optimization, etc.

3.2.2 Ontologies based approaches

Vidal et al. [23], proposed a knowledge-driven framework to integrate biomedical data. This framework generates a knowledge graph after extracting data from various sources such as clinical data and structured data, it contains four parts: knowledge extraction, knowledge graph creation, knowledge management and discovery, and data access control and privacy.
- Knowledge extraction consists of transforming unstructured data sets into structured datasets, it uses ontologies to standardize terms and describe the meaning of concepts.
- knowledge graph creation consists of receiving data sets to build a knowledge graph by transforming data into RDF triples, this knowledge graph is created by using a unified schema, entities that have a relationship with each other are integrated into the knowledge graph using ontological axioms, and entity linking methods.
- Knowledge management and discovery consist of exploiting the knowledge graph and determining new relationships between entities.

Mountasser et al. [24], proposed a semantic-based Big Data integration framework. This framework is composed of three modules. The first module is local ontology building, this module consists of assembling data from heterogeneous

data sources (unstructured, semi-structured, structured data) and converting it to an OWL ontology. The second module is a hybrid large-scale ontology matching to build a global ontology from local ontologies, this module consists of three layers, resources extraction layer that enables the analysis of local ontologies by using Jena framework after that this layer then takes and maintains these subsets in the HBase data store. The second layer is ontology clustering which consists of ranking entities, centroid extractions and creating clusters. The third layer is matching layer which involves identifying similar clusters, language-based matching, string-based matching, graph-based matching. Finally, the last module is probabilistic-logical-based governance, this module involves two-part, the first is constraint-based data quality management and Markov logic network paradigm.

Latsou et al. [25], have proposed a framework for DQA (Data quality assessment) using an ontology-based approach for capturing the knowledge obtained from the assessment, this framework includes three steps. The first step is user's requirements identification which has the objective of knowing the context of a dataset and how it will be utilised. This step consists of two phases, context identification form and data requirements identification form. After that the second step is data quality assessment which is divided on assessing the quality of data and visualizing the DQA insights. Finally, the third step is data quality ontology development which is composed of developing an ontology to capture DQA knowledge and populating the ontology with DQA insights. As a result, this framework helps users to understand and verify data usability.

3.2.3 Other methods-based approaches

Deshpande et al. [26], proposed a biomedical data integration framework (Diis) that depends on human intervention in some phases of data integration. The data shared by the users will be classified into multiple categories. This system is built through a layered architecture. The first is the operating system layer using Linux. The second layer is storing data which consists of two types: the in-house data and shared donor's data, this last is converted and mapped to correspond to the in-house data in the phase of the data adapter layer for supplying data to the next layer, this phase of storing is done by using Hadoop Distributed File System (HDFS). After supplying data, it will be charged into the database layer, the database is a combination of a relational and a NoSQL database system. The next layer is the search engine layer which contains all data retrieval algorithms and Application Programming Interfaces (API), there are many search engines

like Lucene, Solr, and Elasticsearch. This layer will order results in accordance with user queries (image, text, or image+text). After that, there is an authentication layer that verifies user authorizations and applies some specific rules. Finally, the subsequent layer to share information with users by providing an interface suitable to the domain.

Dao-ming et al. [27], proposed an architecture for clinical data integration using Hadoop platform. It consists of three parts: Hadoop platform, a rule-based message processing engine, and clinical information systems.
- The Hadoop platform consists of HDFS and MapReduce which may offer distribution storage, HBase cluster which is a non-relational database, and Zookeeper which may be utilized to organize and handle the distributed applications.
- Rule-based message-processing engine contains an interface layer to communicate with Hadoop platform and clinical information systems and the business process layer; this part is devised by using Apache Camel.
- Clinical information systems like EMR, PACS, and LIS. These systems produce data that is standardized by HL7. HL7 can transfer electronic information and describes the observations of a patient.

Costa et al. [28], designed and deployed a core data platform (MIDAS) for integrating a large volume of data for better understanding the disease and its impact and monitoring the different aspects of the evolution of the pandemic (COVID-19) across a diverse range of groups. It consists of different steps. The first step is data ingestion using a tool called GYDRA to prepare data by extracting data from external networks like public data, private data and ISAACUS metadata server. After that comes data storage using HDFS, Spark, Hive. The third step is data analysis by using MIDAS analytics application and showing the results on MIDAS dashboard.

Vidal et al. [29], have formulated a computational framework that has the capability to leverage information from annotations and semantic descriptions, with the ultimate goal of integrating equivalent entities into a knowledge graph by using unified schema and ontologies. The framework contains four components: Knowledge extraction, Semantic data integration, Exploration and traversal, and Knowledge discovery. In this approach, they concentrate on knowledge extraction and semantic data integration.
- Knowledge extraction: consists of extracting information from unstructured data sources and representing it in the form of predicates, objects, and subjects by using ontologies and vocabularies.
- Semantic data integration: After relevant entities are identified and labeled using ontologies, semantic data integration techniques are employed to determine

if two entities are equivalent based on their annotations. The entities that have been annotated are described with a unified schema and represented as triples through the execution of a set of mapping rules which are expressed by using RDF Map Language (RML). Subsequently, the mapping rules are also executed to represent the extracted entities and predicates as triples.
After that, in the third and the fourth component they used of federated query processing allows for exploration of the knowledge graph, while knowledge discovery aids in uncovering patterns within the graph.

Birgersson et al. [30], have developed a system that uses machine learning for data integration. The goal is to create a system that can connect two different systems by comparing their specifications, this system will then be able to correctly identify where a data value from one specification should be placed in the other specification. They developed 3 models (shortest distance model, maximum flow model, data value model) as the first step to automate data mapping which are tested dependently to determine the performance according to the score that returns, each one of these models takes 2 documents XML as input and confidence level as output. The results show that the distance model has the highest level of confidence and F-score=0.918.

We present in table 1 a classification of the approaches described in Section 2. Indeed, we describe the works according to several criteria to facilitate understanding: dataset, output, used technique, and advantages.

- Column 1: ”Approach” presents the names of the authors.
- Column 2: present the “Dataset” used.
- Column 3: “Output” show of approaches.
- Column 4: “Used technique” show the techniques used in the different stages of the approach.
- Column 5: “Advantage” shows the advantages of the techniques used.

Approach	Dataset	Output	Used technique	Advantage
Sharma et al.[19]	magnetic resonance imaging (MRI), MR spectroscopy (MRS), and neuropsychological test.	Hadoop-based Big Data analytics framework for Alzheimer's disease (AD).	-Data normalization. -Data management. -Data storage. -Data processing. -Accessing, and querying data. -storage using HDFS and Metadata. -Algorithms-based machine learning.	Accessible, facilitating making decisions, unifying various modalities in one platform.
Boehm et al. [20]	444 patient's datasets with HGSOC. 296 patients were treated at the Memorial Sloan Kettering Cancer Center (MSKCC) and 148 patients from The Cancer Genome Atlas Ovarian Cancer (TCGA-OV) data. - Clinicogenomic, histopathological, and radiologic features.	machine learning model.	fusing histopathological, radiologic, and clinicogenomic machine-learning models.	-lead to better treatment decisions. -improved outcomes for patients.
Peishun et al. [21]	-gut microbiome data.		- Machine learning. Deep neural network, random forest, gradient boosting, and decision tree.	- Identifying relationships in complex datasets. build a more comprehensive understanding of the microbiome's composition.
Vetova [22]	clinical data sources, image formats, visualization devices, and clinical datatypes.	Model of data integration and sorting of COVID-19 in .xls formats	- Information organization. - Storing data. - integration process (cleansing, ETL, mapping, transforming). -Information processing. -KNN, K-means, mean shift clustering, ANN, CNN, Machine learning.	-Improves the integration process performance. -Helps solve problems and decision-making.

Table 3.1: State of the art of related works (Machine Learning)

Approach	Dataset	Output	Used technique	Advantage
Vidal et al. [23]	Clinical and structured data.	knowledge-driven framework.	- knowledge extraction. - knowledge graph creation. - knowledge management and discovery. - data access control and privacy. - using ontologies.	Semantic reasoning and inference, efficient research collaboration.
Mountasser et al. [24]	Heterogenous data sources.	semantic-based Big Data integration framework.	- local ontology building. - hybrid large-scale ontology matching. - probabilistic-logical-based assessment. - Jena framework. -Markov logic network paradigm.	-Resolves data heterogeneity And interoperability issues. -Improves the integration process performance.
Latsou et al. [25]	User's requirement data.	Framework for data quality assessment (DQA)	- User'srequirements Identification. Context identification form. Data requirements identification form. -Data Quality Assessment: Assessing the quality of data. Visualising the DQA insights. -Data Quality Ontology Development Developing an ontology to capture DQA knowledge. Populating the ontology with DQA insights.	- Helps users to understand and verify data usability.
Deshpande et al.[26]	Electronic Health Records (EHRs).	Framework to integrate biomedical data	-HDFS. -API. -NoSQL database. -Multiple algorithms. -Solr, Lucene, Elasticsearch.	Accessible, improve data interoperability, reusable, facilitate the exchange of information, and cost reduction.

Table 3.2: State of the art of related work (Ontologies)

Approach	Dataset	Output	Used technique	Advantage
Dao-ming et al. [27]	Clinical data.	clinical data integration using Hadoop platform. word2-vec	-HDFS. -MapReduce. -HBase.	-Improve data sharing. -avoid the problem of isolated information.
Costa et al. [28]	Public data, private data, ISAACUS server.	MIDAS integration platform.	-data ingestion. -data analysis. -data storage and processing.	- insightful information.
Vidal et al. [29]	Structured and unstructured data sources.	Knowledge driven framework.	-Semantic data integration. -RML (RDF Map Language).	-can identify if two entities in the collection of datasets match or do not match.
Birgersson et al.[30]	-XML files.	Data integration system.	-shortest distance model. -maximum flow model. -data value model.	-accuracy of 0.918 % - could be used to save time in the integration process.

Table 3.3: State of the art of related work (Other methods)

3.3 Analysis and Comparison

Big Data are collections of data sets so large and complex to process using classical database management tools. Their main characteristics are volume, variety and velocity. Although these characteristics accentuate heterogeneity problems, users are always looking for a unified view of the data. Consequently, Big Data integration is a new research area that faces new challenges due to the aforementioned characteristics. Ontologies are widely used in data integration since they represent knowledge as a formal description of a domain of interest. With the advent of Big Data, their implementation faces new challenges due to the volume, variety and velocity dimensions of these data.

The objective of the chapter was to analyze data integration approaches. Various approaches, including semantic data integration, data integration frameworks, and machine learning, were used.

The existing works attempt to solve data integration problems from various perspectives. Some of these works [19], [20], [21], [22] have utilized Machine Learning algorithms, they used algorithms like KNN, mean shift clustering, K-means, random forest, decision tree, and CNN. This method has multiple advantages like Improves the performance of the integration process, unifying various

modalities, identifying relationships in complex datasets, leading to making better decisions and building a more comprehensive understanding of data. However, using these algorithms needs parameters that are fixed randomly.

Other works [23] [24] [25] have used ontologies to integrate Big Data that comes from heterogeneous data sources by using different techniques such as local ontology building, and hybrid large-scale ontology matching. Using ontologies in the process of integrating Big Data helps users to understand and verify data usability, resolving data heterogeneity and interoperability issues, efficient research collaboration, and improving the integration process performance.

Current research efforts on modular ontologies composition are focusing on either overlap detection between modules or hierarchy organization between concepts or modules. However, we notice the following limitations:

- they fail to set up complex correspondences between the compared modules. Authors of the studied approaches compare concepts to concepts and ignore concept/attribute and attribute/concept comparisons,
- they do not consider the difference between granularity levels of the composed modules, which may cause problems in information exchange due to the different levels of specification of an entity in the real world.

In addition, [26] [27] [28] [29] [30] have used other several techniques in the process of integration such as the shortest distance model, data value model, Map-Reduce, RML, data ingestion, and analysis. These methods help in making insightful information, improving data interoperability, and facilitating the exchange of information.

Managing Big Data about a specific domain remains an important issue worldwide since users usually look for an integrated view of the available data. Ontologies were widely used as a solution for data integration. The basic issue in Big data integration is to automatically build the ontology model and to bring out the hidden semantics which are not directly available from the data sources. The resulting ontology serves to represent knowledge integrated from Big Data sources and to provide a shared model for data sources.

This work concentrated on ontology building for Big Data integration, our approach consists of building local ontologies from several data sources, then building a global ontology through a fusion stage using machine learning, and finally querying them using a query module. Using these techniques, our aim is to obtain better performance in the context of data integration.

The main innovation of the proposed approach lies in its ability to be applied to all domains and its dynamicity. Indeed, it is easy to integrate a new Big Data

source in the integration process, since data sources are treated independently from each other.

3.4 Conclusion

In this chapter, we have presented a survey of a the main works related to data integration, which is an open area of research given its importance in the technological world, we have summarized each work in a small paragraph indicating the important points. After that, we summarized all the works in a table indicating their main techniques used and advantages. Each work has its own characteristics, noting that there may be common features.

In the next chapter, we will work to present our contribution. We will explain in detail the different steps and techniques used.

Chapter 4

Contributions

4.1 Introduction

Big data integration focuses on dealing with various data differences by offering solutions for multiple interoperability challenges, including representation, structuring, and correspondence conflicts. These discrepancies arise due to the usage of diverse data models and representation techniques across different data sources. Additionally, similar syntax in data can lead to different interpretations, further complicating the process of distinguishing between them. Consequently, effective data integration strategies are necessary to tackle these interoperability issues while considering the complex nature of the data.

researchers propose various tools and methods to handle data integration. However, the majority of these works focused on the data structure without semantic.

The aim of our study is to propose an approach to integrate healthcare big data from various data sources using ontologies and machine learning algorithms. Ontologies provide a semantic layer to understand data in a a specific context. While machine learning algorithms aim to manage data to make prediction and recommendations.

The proposed methodology is inspired from the proposal of el bouhissi et al. [1] and a continuation.

In this chapter, we present in detailed our approach for integrating Big Data in the healthcare sector.

4.2 Proposed approach

In this section, we describe in detail our approach which consists of building local ontologies from data collected from several resources , then building a global ontology which is a fusion of the local ontologies and finally querying the global ontology using SPARQL.

The proposal described in figure 4.1 involves five main steps. The first step **“Data collection”**, which includes collecting data from various data sources. The second step called **“Data processing”** which includes correcting errors and inaccuracies. The third step is **” Local ontology engineering”**, which consists of using the K-means algorithm to calculate similarities for building local ontologies. The fourth step named **“Global ontology building”** involves using a fusion algorithm to create a global ontology, and finally, the fifth step is **“Query interpretation and execution”** which consists of reformulating queries in the form of SPARQL code and executing these queries.

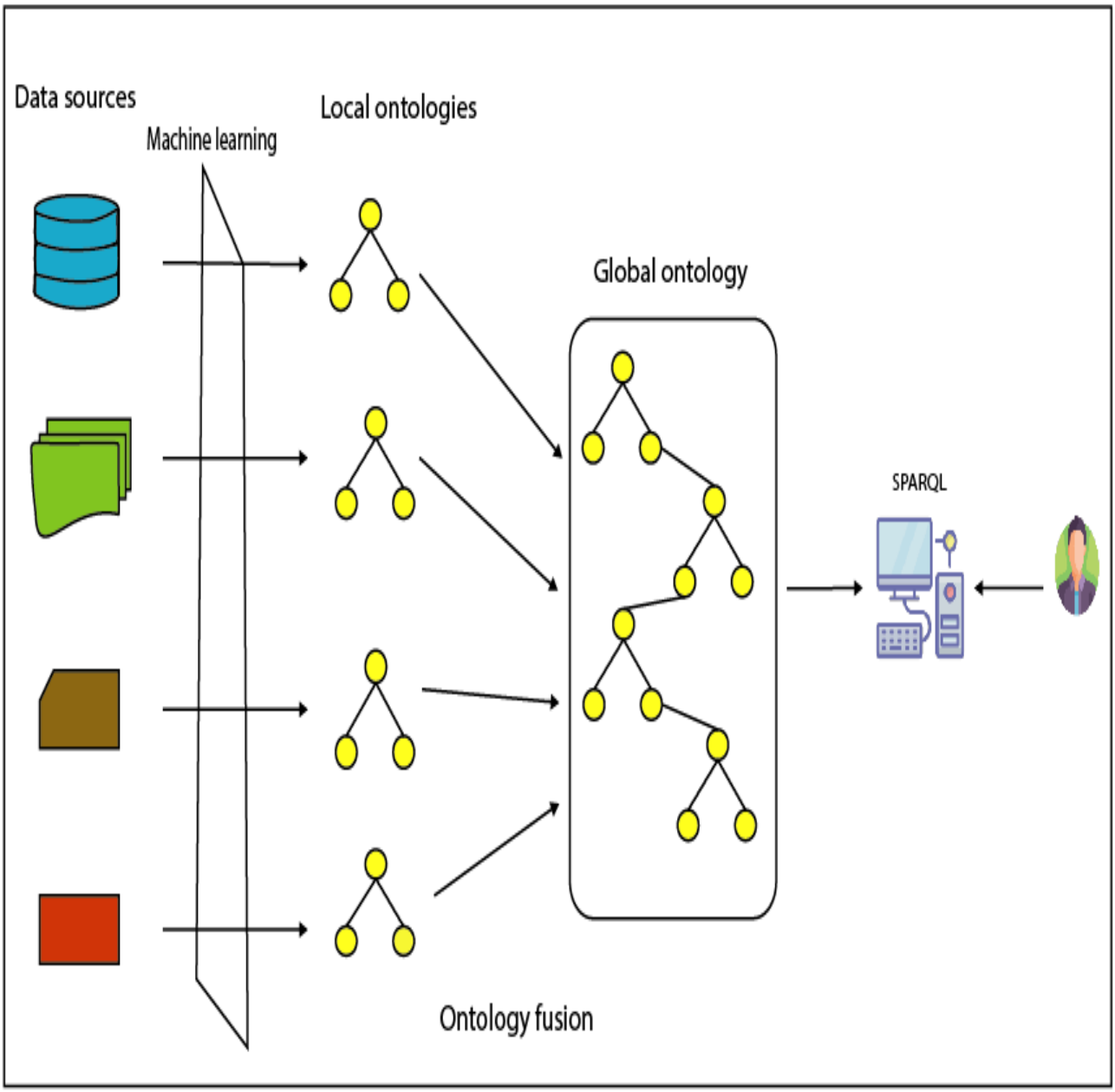

Figure 4.1: System architecture

4.2.1 Data collection

Data collection refers to the process of acquiring and evaluating information and datasets about specific variables within an established system, facilitating the ability to answer relevant questions and evaluate results. These input sources include several data types: structured, semi-structured, or unstructured. It serves as a fundamental basis for data integration, data analysis and data-driven decision-making.

There are various data collection methods, including:

- **Surveys:** These are forms used to gather information from a group of individuals and can be conducted in person, or any means of communication.
- **Interviews:** An exchange between two parties, one asking questions and the other providing answers. Interviews can be structured, semi-structured, or unstructured.
- **Observation:** The act of observing and recording the behaviour of individuals, whether in a natural environment or in a controlled laboratory.
- **Experiments:** Controlled investigations where one or more variables are manipulated in order to assess their impact on another variable.
- **Document analysis**: The process of examining various types of documents in detail, such as paper documents, digital files or audio recordings, in order to extract information.

The choice of data collection method depends on the specific research requirement, the resources available, and other factors. The elements to be taken into account when making this choice are as follows:

- **The objectives of the study:** What information do we want to obtain from the data?
- **The type of data required:** What type of information needs to be collected?
- **The target population:** Who will the data be collected from?
- **Resources available:** How much time is available?
- **Ethical considerations:** Are there any ethical aspects to take into account?

Once a data collection method has been selected, it is essential to plan and execute the process. This includes:

- **Developing a data collection tool:** A survey, interview guide or observation protocol, for example.
- **Carrying out pilot tests:** This stage enables any problems with the tool to be identified and any necessary improvements to be made.
- **Collecting data:** This may involve carrying out surveys, interviews or observations.

This step consists of gathering and merging patient data from diverse sources and healthcare settings into a cohesive, interoperable platform. The main objective is to enhance patient care and strengthen communication among healthcare professionals.

4.2.2 Data processing

Data processing is the process of collection and manipulation of digital data to produce meaningful information. It involves finding and correcting errors and inconsistencies, replacing null values, normalising, eliminating duplicates and inaccuracies in data sets to improve their accuracy and reliability. This is a form of information processing, i.e. the modification of information in a way that is detectable by an observer.

The data processing cycle comprises the following stages:

After data collection, the cleaning process consists of eliminating errors and inconsistencies from the data. This is an important step in ensuring the accuracy and reliability of the data. Once the data has been cleaned, it needs to be transformed and converted into a format that is easy to process. This may involve converting the data into a different format, such as a spreadsheet or database. After this, the data needs to be analyzed, the process of extracting meaning from the data. This may involve using statistical techniques to identify patterns and trends in the data. Finally, the results of the data analysis need to be communicated to others. This may involve creating tables, graphs or reports.

4.2.3 Local ontology engineering

The local ontology construction module is charged with converting input data sources into local ontologies. It provides a semantic data model that homogenizes the data being integrated by exploiting several ontology learning mechanisms to bring together data from heterogeneous sources, in spite of its content and nature, and transform it into a common representation.

The Figure 4.2 ,presents the schema of building local ontologies:

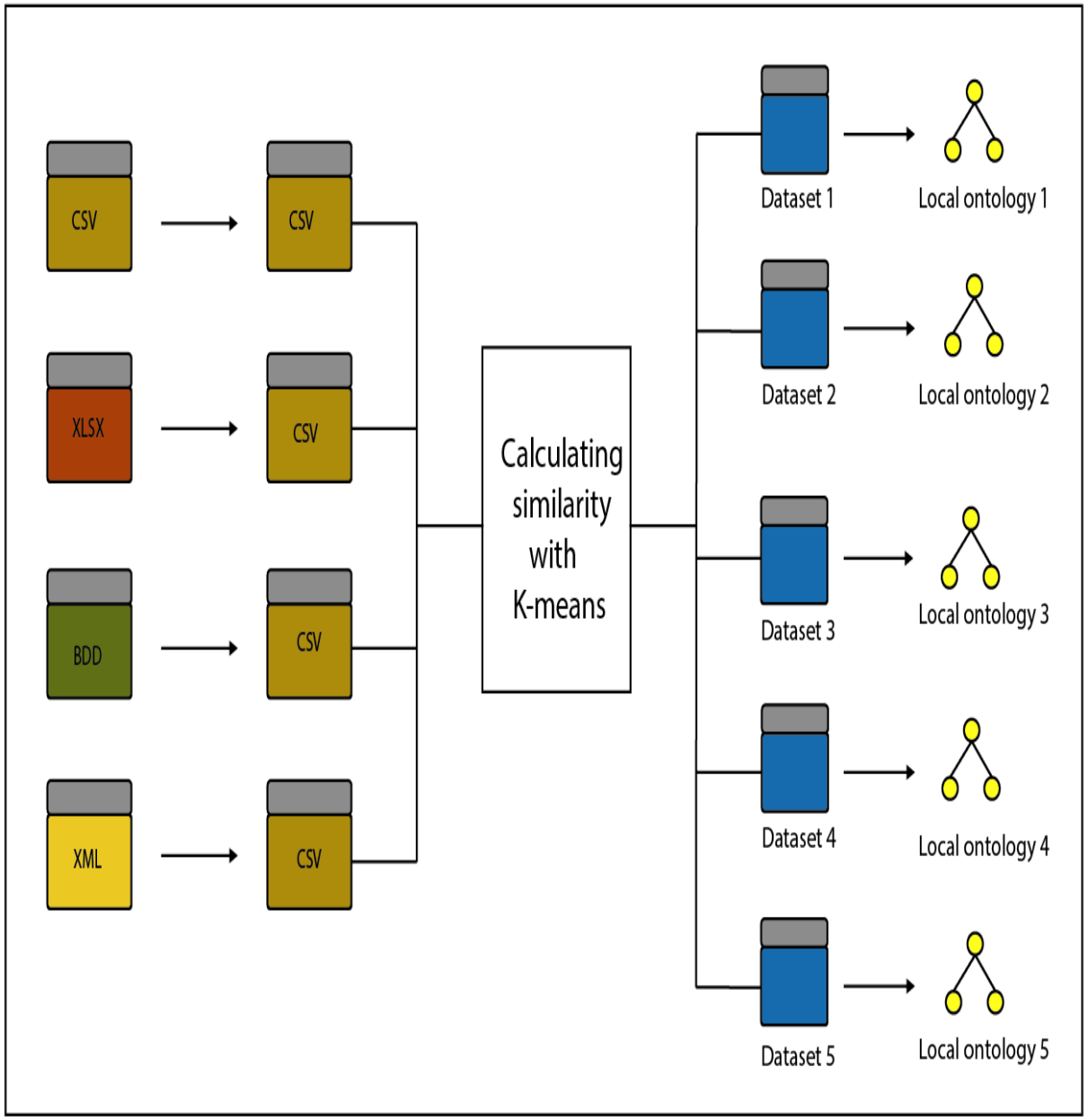

Figure 4.2: Local ontologies building

In our approach, we use datasets in different formats such as CSV, XLSX, BDD, XML, etc. After collecting these datasets, we convert them to CSV format in order to facilitate the process, then we proceed to manage the data by eliminating the inaccuracies, then we apply the K-means algorithm to calculate the similarities between the datasets, after completing this process the K-means algorithm categorizes the datasets according to their similarities and at the end build the local ontologies associated with each dataset.

The creation of an ontology from a CSV data file is done by using the Protégé tool[31]. Users have the option to either directly upload the CSV file or provide its accessible Uniform Resource Locator (URL). The present procedure exclu-

sively accepts CSV data flows as input, but it's adaptable to include additional modes like Hypertext Markup Language (HTML), Keyhole Markup Language (KML), and JavaScript Object Notation (JSON).

For parsing CSV files, the Apache Commons CSV library is utilized. It determine CSV file structure including headers, line endings, and escape characters. This aids in processing text-based fields within CSV files. The CSV parser employs various functions to read and interpret rows, cell values, and reference values within the CSV data.

The conversion of CSV data to OWL primarily necessitates appending metadata annotations that elucidate the data interpretation methodology. Designed for non-experts, this tool expedites the creation of clear and informative records of information resources, enabling effective resource searches in a connected environment. The Dublin Core Resource Description Framework Architecture (RDFS) vocabulary is employed to define shared metadata. RDFS is chosen due to its widespread recognition as a user-friendly mechanism for constructing precise and comprehensive records for data sources. Simultaneously, it supports resource searches in interconnected environments.

The user-entered CSV file name becomes the ontology's title, while the CSV file's column headers are allocated as data properties. The values stored within these data properties are treated as ontology entities. Each row under a column header is considered an individual entity, with each record assigned a unique identifier. Throughout this transformation, corresponding axioms are established between data properties and individuals, elucidating class meanings and relationships. Following axiom addition, the ontology resides in local memory, and the resulting ontology is visualized using Protege tools [32].

Before creating ontologies, we need to use a machine learning algorithm for calculating similarities. Calculating similarities between datasets involves various techniques and methodologies depending on the nature of the data and the specific goals of the analysis. Here's a general process that can be followed:

1. **Data understanding** Before calculating similarities, it's crucial to have a deep understanding of the datasets working with. This includes understanding the data types, structures, and any inherent patterns or characteristics.

2. **Data processing** Clean and process the datasets to ensure consistency and eliminate noise. This may involve handling missing values, standardizing units, normalizing data, and addressing outliers.

3. **Similarity Selection and Calculation** Choose an appropriate similarity algorithm based on the nature of the data. Different algorithms are used for

different types of data, such as numerical, categorical, text, and image data. In our approach, we use the K-means algorithm because we are in a type of unsupervised learning.

4. **Calculate Similarity:** Apply the chosen similarity algorithm to calculate the similarity between data points or instances in your datasets.

5. **Normalization** Depending on the algorithm used, it may be necessary to normalize the similarity scores to ensure that they fall within a specific range, generally between 0 and 1. Normalization makes the similarity more interpretable and comparable.

6. **Similarities Aggregation** If similarities are calculated between multiple data points within each dataset, it is necessary to aggregate these individual similarities to get an overall measure of similarity for the entire dataset.

7. **Results classification** The categories of similarities obtained will be classified according to the K chosen the first time.

The efficiency and accuracy of the matching will depend on the quality of the data, the relevance of the features extracted, and the performance of the machine learning model used.

K-Means algorithm:

K-Means is an unsupervised learning algorithm for clustering data based on similarity. It is one of the simplest and most efficient clustering algorithms, and is also relatively fast to run, making it suitable for large datasets.

The procedure follows a simple and easy method for classifying a given set of data into a number of groups. The main idea is to choose the desired number of clusters, which is usually determined by the application. Each cluster has centroids, which are the central points. They are randomly initialized in the data space. Each piece of data is then allocated to the cluster whose centroid is closest to it. These centroids are updated according to the data assigned to them [33].

The Figure 4.3 , presents an exemple of K-Means clustering algorithm:

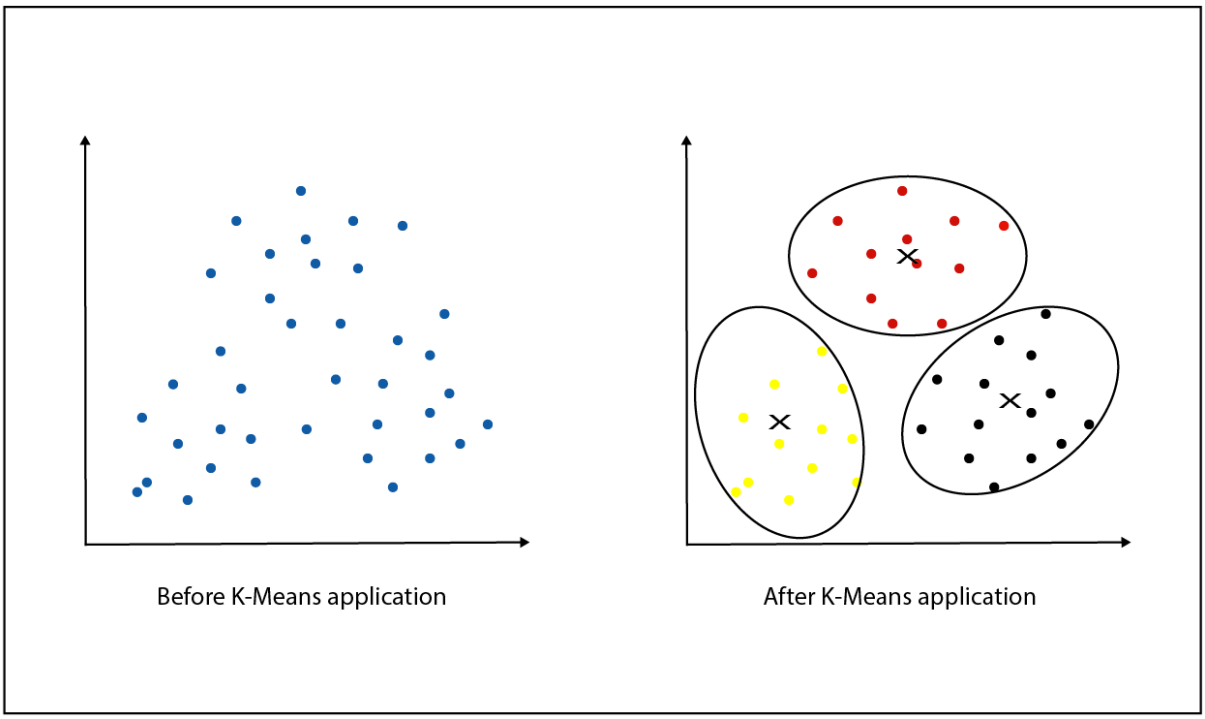

Figure 4.3: K-Means exemple

Here's an illustration of the steps involved in using K-Means:

- **Initialization:** Initialise the algorithm with the number of clusters (K), the dataset (X) and the maximum number of iterations (max-iterations). In addition, K initial cluster centroids are randomly selected.
- **Iteration loop:** The algorithm runs for a maximum of max-iterations or until convergence is reached.
- **a. Assignment step:** For each data point in the dataset X, find the nearest of the K centroids. Assign this data point to the group represented by the nearest centroid. This step constitutes the initial grouping.
- **b. Update step :** For each cluster k (from 1 to K), calculate the average of all data points assigned to cluster k. This average becomes the new centroid for cluster k.
- **Iteration increment:** Increment the iteration counter by one after completing a full cycle of assignment and updating for all data points and centroids.
- **End of algorithm:** The algorithm ends when the maximum number of iterations is reached or the centroids no longer change significantly between iterations (convergence).

Algorithm 1 K-means Algorithm

Require: K (number of clusters), X (data points), $max\_iterations$
1: Initialize K cluster centroids randomly
2: $iterations \leftarrow 0$
3: **while** iterations $<$ max_iterations **and** changes in centroids **do**
4: Assign each data point to the nearest centroid
5: **for** $k = 1$ **to** K **do**
6: Calculate the mean of data points assigned to cluster k
7: Update centroid k to the calculated mean
8: **end for**
9: $iterations \leftarrow iterations + 1$
10: **end while**
11: **return** Final cluster assignments, cluster centroids

The algorithm 1 describes the K-means clustering algorithm, which is a popular machine learning technique used to divide a dataset into K distinct non-overlapping groups. Here is an explanation of the steps in the algorithm:

4.2.4 Global ontology engineering

Merging local ontologies to build a global ontology is a process that combines the knowledge represented in several ontologies to create a single ontology that covers a broader domain. This process is often necessary in situations where heterogeneous information systems need to be interconnected or where data from different sources needs to be merged, as in our case.

Merging local ontologies involves resolving differences between ontologies, such as definitions and relationships between concepts, and involves updating or modifying the structure of the local ontologies to accommodate the newly mapped concepts and relationships.

By using a common ontology, it becomes easier to communicate and share data between different applications and platforms, it enables data to be integrated by creating semantic correspondences between different data sources, and it helps to enrich data, which facilitates categorization, classification and enrichment of data, which is useful for analysis and visualization.

The figure 4.4 describes the global ontology process, which begins with the identification of concepts, then the calculation of similarities carried out in the previous step, and finally the merging of local ontologies, which produces a global ontology.

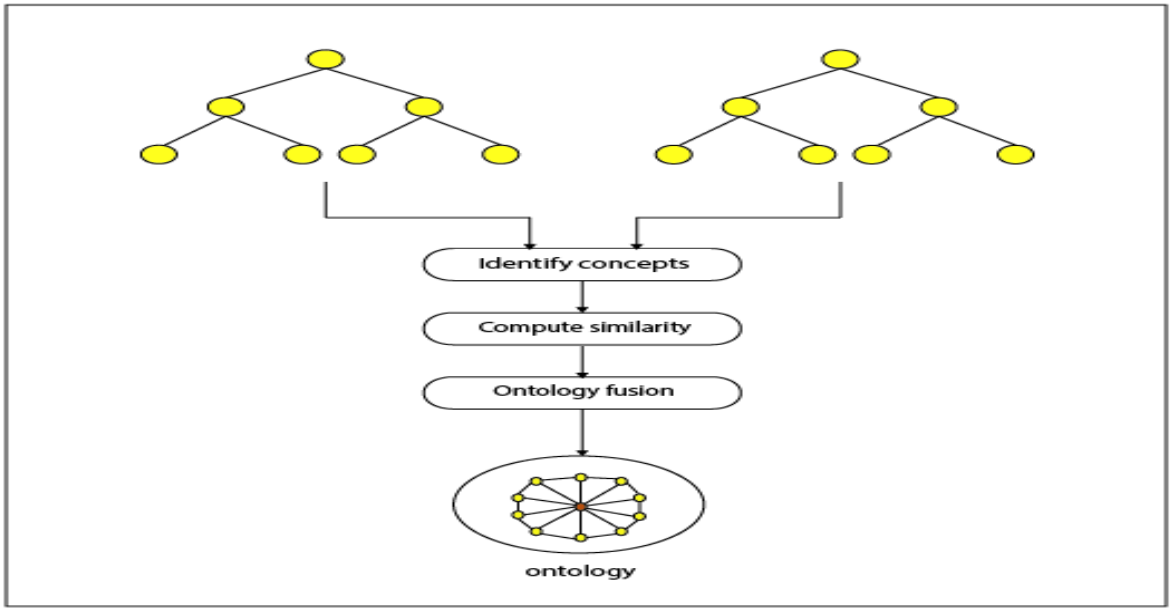

Figure 4.4: Global ontology

4.2.5 Query interpretation and execution

Query interpretation is a valuable tool that can be used to improve the usability of information retrieval systems. Query answering is important to big data integration because it provides a process by which users and applications can interact with data sources using ontologies, it provides users an easier access to the information that they need, by converting natural language queries into formal queries.

The query interpretation and execution are responsible for retrieving the results from the RDF repository. This may involve rewriting the query to use more efficient operators or distributing the query over RDF repositories.

The fundamental component used to create SPARQL query structures is called a Basic Graph Pattern (BGP). A BGP consists of a collection of triple patterns, which are RDF triples capable of including query variables in the subject, predicate, and object positions. During the query's execution, it identifies the connections between the variables and the values.

The SPARQL engine evaluates the query and returns the required responses once it has been created. The query is constructed directly using the language constructors at this level, which requires some prior knowledge. we construct a query based on the user's requirements using the Simple Protocol And RDF Query Language (SPARQL). Query response is critical in the period of BD integration because it offers a framework for users and applications to communi-

cate with data sources via ontologies. To run queries with SPARQL, we need to convert the OWL ontology into RDF format.

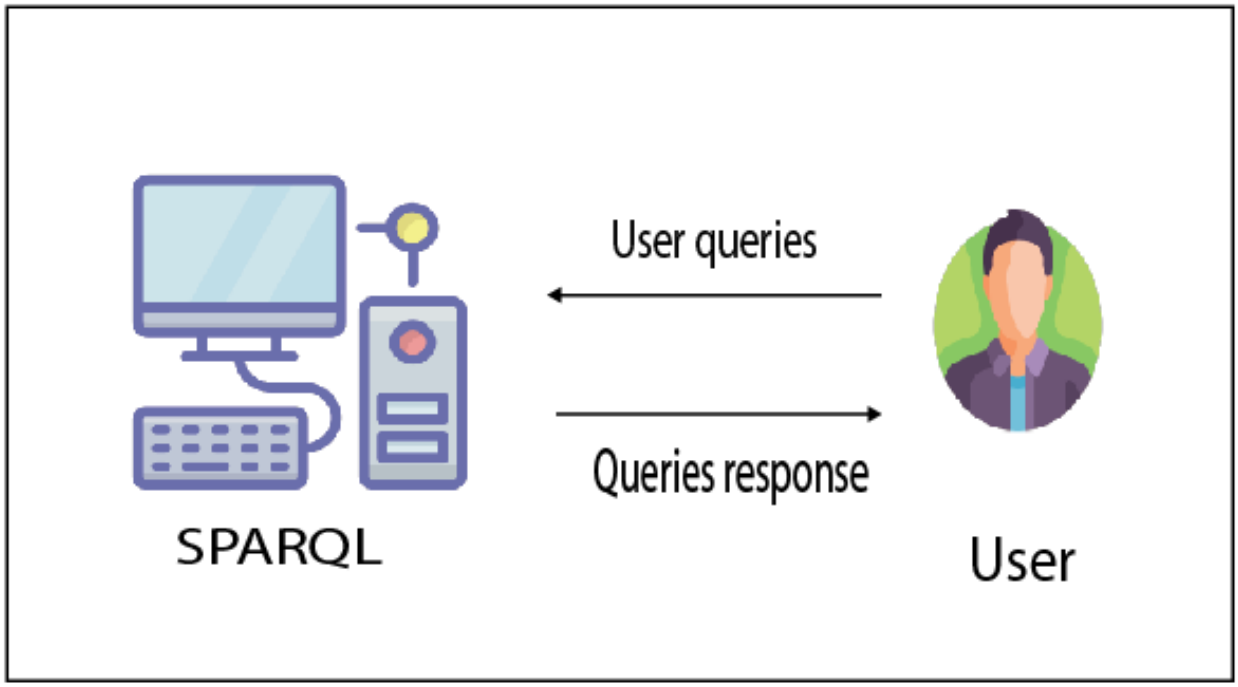

Figure 4.5: Query interpretation and execution

4.3 Conclusion

In this chapter, we have described the main steps of the proposed approach for healthcare big data warehouse integration. Our system involves various steps namely data collection, data processing, local ontology engineering, global ontology engineering, query interpretation, and query execution. We have defined what the matching ontology algorithm is, and we have defined how the fusion algorithm works.

In the next chapter, we will implement and evaluate our approach to big data integration in the healthcare sector. We also present the used tools and the development environment.

Chapter 5

Experimentation

5.1 Introduction

When it comes to integrating massive and diverse volumes of data, theory alone is not enough. Experimentation and implementation are essential to transform the promises of Big Data into tangible reality.

In the context of our research, we have chosen to use unsupervised learning to calculate similarities and merge local ontologies. The clustering process will be carried out on datasets in order to group them into different categories to build local ontologies, as well as build a global ontology from these ontologies.

In this chapter, we will describe the datasets used to carry out our experiment, as well as the hardware and software tools used for the development of our solution, and finally the implementation and evaluation of our solution.

5.2 Development environment

5.2.1 Hardware environment

We used a Laptop computer with the following configuration :

- Processor: Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz
- Installed Memory (RAM):8.00 GB
- Operating System Type: 64-bit, x64 processor
- Windows 10 Family

5.2.2 Software environment

- **Jupyter Notebook** Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equa-

tions, visualizations, and narrative text. It is widely used for data analysis, scientific computing, machine learning, and data visualization tasks. Jupyter supports a wide range of programming languages, including Python, R, Scala, etc. [34].

- **Python** Python is a high-level, interpreted, object-oriented programming language. It is easy to learn and use, making it popular for a wide variety of programming tasks, including data science, scripting, and web programming. The Python interpreter, along with its comprehensive standard library, is accessible in both source and binary forms at no cost on major platforms, and it can be freely shared [35].

- **NumPy** NumPy is a numerical calculation library in Python. It provides data structures for representing multidimensional arrays and matrices, as well as mathematical functions for working with these arrays. NumPy is widely used in data science, machine learning, and image processing [36].

- **Matplotlib** Matplotlib is a data visualization library in Python. It provides tools for creating 2D and 3D graphs from data, as well as functions for customizing the appearance of graphs. Matplotlib is widely used in data science, finance, and other fields for data visualization. [37]

- **Pandas** Pandas is a data manipulation library in Python. It provides data structures to represent arrays of data, called Data Frames, as well as functions to manipulate, clean, and analyze them. Pandas are used for processing data in data science and other fields [38].

- **Scikit-learn:** Scikit-learn is a popular open-source machine-learning software library for Python. It offers a wide range of supervised and unsupervised machine learning algorithms, as well as tools for data preparation and transformation, model selection, performance evaluation and much more. Scikit-learn is widely used to develop predictive models, and perform classification, regression, clustering, and dimensionality reduction tasks. It is valued for its ease of use, and extensive documentation [39].

- **Protégé** Protégé is an extensible open-source environment for creating ontologies. It was created at Stanford University and is very popular in the field of the Semantic Web and computer science research. Protégé is developed in Java. It is free and its source code is published under an open license. Protégé can read and save ontologies in several ontology formats: RDF, RDFS, OWL, etc. It is recognized for its ability to work on large ontologies [31].

5.3 Datasets Description

A dataset is a structured collection of data, usually organized into rows (instances) and columns (attributes) in a tabular format.

Datasets are used in a variety of fields, such as scientific research, data analysis, machine learning, statistics, and more. They serve as a basis for analysis, experimentation, and modeling. They can be in a variety of formats, such as CSV, XLSX, etc.

For the purpose of our experimentation, we collected 3 datasets from Kaggle [40].

These datasets collect information from over 100,000 medical appointments in cities in Brazil and are focused on the question of whether or not patients show up for their appointments. A number of characteristics about the patient are included in each row. It was created by JoniHoppen in 2016.

The first dataset includes 10001 rows and 13 columns and is organized as shown in Figure 5.1 and involves the following columns:

	ID	Sexe	Day	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	5758417	F	2016-08-01T08:17:04Z	2016-08-01T00:00:00Z	20	ILHA DAS CAIEIRAS	0	0	0	0	0	0	No
1	5523159	F	2016-03-29T16:09:39Z	2016-05-03T00:00:00Z	37	RESISTÊNCIA	0	0	0	0	0	1	No
2	5693080	F	2016-05-12T17:33:56Z	2016-05-20T00:00:00Z	38	MARIA ORTIZ	0	0	0	0	0	0	Yes
3	5654129	F	2016-05-03T13:54:51Z	2016-06-03T00:00:00Z	24	SANTO ANDRÉ	0	0	0	0	0	1	Yes
4	5641070	F	2016-04-29T12:16:28Z	2016-04-29T00:00:00Z	41	MARIA ORTIZ	0	0	0	0	0	0	No

Figure 5.1: Description of dataset 1

- **ID:** Identification of a patient.
- **Sexe:** Male or Female .
- **Day:** The day someone called or registered the appointment, this is before the appointment of course.
- **AppointmentDay:** The day of the actual appointment, when they have to visit the doctor.
- **Age:** How old is the patient.

- **Neighbourhood:** Where the appointment takes place.
- **Scholarship:** True of False.
- **Hipertension:** True of False.
- **Diabetes:** True of False.
- **Alcoholism:** True of False.
- **Handcap:** True of False.
- **SMS-received:** 1 or more messages sent to the patient.
- **No-show:** True of False.

The second dataset comprises 5411 rows and 13 columns.This dataset is organized as shown in Figure 5.2 with the following columns:

- **ID-personne':** Identification of a patient.
- **Gender:** Male or Female.
- **ScheduledDay:** The day someone called or registered the appointment, this is before the appointment of course.
- **Date:** The day of the actual appointment, when they have to visit the doctor.
- **Age:** How old is the patient.
- **Neighbourhood:** Where the appointment takes place.
- **Scholar:** True of False.
- **Hipertension:** True of False.
- **Diabetes:** True of False.
- **Alcoholism:** True of False.
- **Handcap:** True of False.
- **SMS-received:** 1 or more messages sent to the patient.
- **No-show:** True of False.

	ID_personne	Gender	ScheduledDay	Date	Age	Neighbourhood	Scholar	Hipertension	Diabetes	Alcoholism	Handcap	SMS_received	No-show
0	6853555	F	2016-05-03T12:37:27Z	2016-05-03T00:00:00Z	18	JESUS DE NAZARETH	0	0	0	0	0	0	No
1	5597470	F	2016-04-18T16:39:48Z	2016-05-19T00:00:00Z	0	MARIA ORTIZ	0	0	0	0	0	0	No
2	6853877	F	2016-05-03T13:28:54Z	2016-05-30T00:00:00Z	35	JESUS DE NAZARETH	0	0	0	0	0	1	No
3	5630683	M	2016-04-27T16:01:26Z	2016-05-16T00:00:00Z	67	SANTOS DUMONT	0	1	0	0	0	0	No
4	5635542	M	2016-04-28T13:43:04Z	2016-05-06T00:00:00Z	1	SANTOS DUMONT	1	0	0	0	0	1	No

Figure 5.2: Description of dataset 2

The third data set comprises 13433 rows and 12 columns. This dataset is organized as shown in Figure 5.3 with the following columns:

- **Personne:** Identification of a patient.
- **Gender:** Male or Female .
- **ScheduledDay:** The day someone called or registered the appointment, this is before the appointment of course.
- **AppointmentDay:** The day of the actual appointment, when they have to visit the doctor.
- **Age:** How old is the patient.
- **Neighbourhood:** Where the appointment takes place.
- **Scholarship:** True of False.
- **Hipertension:** True of False.
- **Diabetes:** True of False.
- **Alcoholism:** True of False.
- **Handcap:** True of False.
- **No-show:** True of False.

	Personne	Gender	ScheduledDay	AppointmentDay	Age	Neighbourhood	Scholarship	Hipertension	Diabetes	Alcoholism	Handcap	No-show
0	5659242	F	2016-05-04T13:26:45Z	2016-05-04T00:00:00Z	40	CONSOLAÇÃO	1	0	0	0	0	No
1	5598382	F	2016-04-19T07:31:36Z	2016-05-10T00:00:00Z	50	SANTA TEREZA	0	1	0	0	0	No
2	5690145	F	2016-05-12T09:34:57Z	2016-05-12T00:00:00Z	30	RESISTÊNCIA	0	0	0	0	0	No
3	5635911	F	2016-04-28T14:16:52Z	2016-05-17T00:00:00Z	16	TABUAZEIRO	1	0	0	0	0	Yes
4	5661500	F	2016-05-05T07:32:21Z	2016-05-19T00:00:00Z	29	ROMÃO	0	0	0	0	0	No

Figure 5.3: Description of dataset 3

5.4 Implementation

5.4.1 Data collection

The first step is to import the dataset from files in various formats. In our case, we import CSV files (Figure 5.4). The Figure 5.4 shows how to import a dataset from a CSV file. The same process is performed to all datasets.

```
import pandas as pd
```

```
data = pd.read_csv(r'C:\Users\abdel\Desktop\KaggleV2-May-2016.csv')
```

Figure 5.4: Import datasets

5.4.2 Data processing

In this step we remove columns that are not necessary to apply our algorithm, and also transform columns that have a character type into a numeric type.

For the first dataset, as shown in Figure 5.5, we remove the **"AppointmentDay"** and **"No-show"** columns.

We transform the **"Address"** column from a character into a numerical value, and each of the cities will be assigned a number, we do the same for the **"Sexe"** column, **'F'** becomes 0, and **'M'** becomes 1.

```
data = pd.read_csv('dataset_1.csv')
display(data.head())

# Deleting date => unnecessary
del data['AppointmentDay']

# Delete, no need for K-means
del data['No-show']

# Use of a LabelEncoder to convert Addresse into numerical values
address_encoder = LabelEncoder()
data['Adresse'] = address_encoder.fit_transform(data['Neighbourhood'])
del data['Neighbourhood']

# Transformation of Sexe values into numerical values
gender_encoder = LabelEncoder()
data['Sexe'] = gender_encoder.fit_transform(data['Sexe'])

display(data.head())

data1 = data.copy()
```

Figure 5.5: Dataset 1 preparation

For the second dataset, as shown in Figure 5.6, we remove the **"ScheduledDay"** and **"Date"** and **"No-show"** columns.

We transform the **'City'** column from a character into a numerical value, and each of the cities will be assigned a number, we do the same for the **'Gender'** column, **'F'** becomes 0, and **'M'** becomes 1.

```
data = pd.read_csv('dataset_2.csv')
display(data.head())

# Deleting dates => unnecessary
del data['ScheduledDay']
del data['Date']

# Delete, no need for K-means
del data['No-show']

# Use of a LabelEncoder to convert City into numerical values
address_encoder = LabelEncoder()
data['City'] = address_encoder.fit_transform(data['Neighbourhood'])
del data['Neighbourhood']

# Transformation of Gender values into numerical values
gender_encoder = LabelEncoder()
data['Gender'] = gender_encoder.fit_transform(data['Gender'])

display(data.head())

data2 = data.copy()
```

Figure 5.6: Dataset 2 preparation

For the third dataset, as shown in Figure 5.7, we eliminate the **”ScheduledDay”** **”AppointmentDay”** and **”No-show”** columns.

We transform the **”City”** column from a character into a numerical value, and each of the cities will be assigned a number, we do the same for the **’Gender’** column, **”F” becomes 0, and ”M”** becomes 1.

```
data = pd.read_csv('dataset_3.csv')
display(data.head())

# Deleting dates => unnecessary
del data['ScheduledDay']
del data['AppointmentDay']

# Delete, no need for K-means
del data['No-show']

# Use of a LabelEncoder to convert City into numerical values
address_encoder = LabelEncoder()
data['City'] = address_encoder.fit_transform(data['Neighbourhood'])
del data['Neighbourhood']

# Transformation of Gender values into numerical values
gender_encoder = LabelEncoder()
data['Gender'] = gender_encoder.fit_transform(data['Gender'])

display(data.head())

data3 = data.copy()
```

Figure 5.7: Dataset 3 preparation

Next, we prepare the datasets to group them together into one dataset. Before doing this, the datasets must have the same column names and the same number. This process is performed as shown in the Figure 5.8.

```
# Front View
print('Before Preaparation to calculate similarity:')
print('data1 :',data1.shape,'columns:',len(data1.columns))
print(data1.columns)
print('data2 :',data2.shape,'columns:',len(data2.columns))
print(data2.columns)
print('data3 :',data3.shape,'columns:',len(data3.columns))
print(data3.columns)


# Deletion of 'Day' and 'SMS_received' from data1 and data2
del data1['Day']
del data1['SMS_received']
del data2['SMS_received']

# Renaming columns
data1 = data1.rename(columns={'Scholarship': 'Scholar'})
data3 = data3.rename(columns={'Scholarship': 'Scholar'})

data1 = data1.rename(columns={'Adresse': 'City'})
data1 = data1.rename(columns={'Sexe': 'Gender'})
data1 = data1.rename(columns={'AppointmentID': 'ID'})
data2 = data2.rename(columns={'ID_personne': 'ID'})
data3 = data3.rename(columns={'Personne': 'ID'})

# View after
print()
print('After Preaparation to calculate similarity:')
print('data1 :',data1.shape,'columns:',len(data1.columns))
print(data1.columns)
print('data2 :',data2.shape,'columns:',len(data2.columns))
print(data2.columns)
print('data3 :',data3.shape,'columns:',len(data3.columns))
print(data3.columns)
```

Figure 5.8: Preparing datasets

The Figure 5.9 shows the number and names of columns before and after eliminating the columns we don't need.

```
Before Preaparation to calculate similarity:
data1 : (10001, 11) columns: 11
Index(['ID', 'Sexe', 'Day', 'Age', 'Scholarship', 'Hipertension', 'Diabetes',
       'Alcoholism', 'Handcap', 'SMS_received', 'Adresse'],
      dtype='object')
data2 : (5411, 10) columns: 10
Index(['ID_personne', 'Gender', 'Age', 'Scholar', 'Hipertension', 'Diabetes',
       'Alcoholism', 'Handcap', 'SMS_received', 'City'],
      dtype='object')
data3 : (13433, 9) columns: 9
Index(['Personne', 'Gender', 'Age', 'Scholarship', 'Hipertension', 'Diabetes',
       'Alcoholism', 'Handcap', 'City'],
      dtype='object')

After Preaparation to calculate similarity:
data1 : (10001, 9) columns: 9
Index(['ID', 'Gender', 'Age', 'Scholar', 'Hipertension', 'Diabetes',
       'Alcoholism', 'Handcap', 'City'],
      dtype='object')
data2 : (5411, 9) columns: 9
Index(['ID', 'Gender', 'Age', 'Scholar', 'Hipertension', 'Diabetes',
       'Alcoholism', 'Handcap', 'City'],
      dtype='object')
data3 : (13433, 9) columns: 9
Index(['ID', 'Gender', 'Age', 'Scholar', 'Hipertension', 'Diabetes',
       'Alcoholism', 'Handcap', 'City'],
      dtype='object')
```

Figure 5.9: Datasets prepared

Once we have obtained the same number and names of columns, we need to concatenate the datasets into a single one, to make it easier to apply the K-Means algorithm later.

```
# Merging DataFrames using pd.concat()
data_final = pd.concat([data1, data2, data3], ignore_index=True)

# Display the first lines of the combined DataFrame
display(data_final)
```

Figure 5.10: Merging datasets

5.4.3 Local ontologies building

Once we have prepared the datasets and cleaned them up, we will apply the K-Means algorithm.

We performed a data normalization, we chose **K= 4** to do the clustering, and at the end, we created a **'Category'** column to find out where each dataset row

belongs.

```
# Application of KMeans

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Normalization
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data_final)

num_clusters = 4  # Replace with the desired number of clusters
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(scaled_data)

cluster_assignments = kmeans.labels_
cluster_centers = kmeans.cluster_centers_
```

```
data_final['Category'] = cluster_assignments
data_final
```

Figure 5.11: Application of K-Means

As shown in the Figure 5.12, we have filtered the data according to the category of each of them. This step is used to create the local ontologies associated with each category.

```
# Filter data according to the value in the "category" column
data_1 = data_final[data_final['Category'] == 0]
data_2 = data_final[data_final['Category'] == 1]
data_3 = data_final[data_final['Category'] == 2]
data_4 = data_final[data_final['Category'] == 3]
```

Figure 5.12: Filtering data

After data filtering, we obtained 4 datasets as shown in the Figure5.13, these datasets will be converted into local ontologies.

(1886, 10)

	ID	Gender	Age	Scholar	Hipertension	Diabetes	Alcoholism	Handcap	City	Category
11	5646843	0	68	0	1	1	0	0	71	0
61	5387908	1	79	0	1	1	0	0	38	0
87	5728246	1	60	0	1	1	0	0	15	0
91	5754675	0	88	0	0	1	0	0	18	0
99	5772305	0	88	0	0	1	0	0	9	0

(17096, 10)

	ID	Gender	Age	Scholar	Hipertension	Diabetes	Alcoholism	Handcap	City	Category
0	5756417	0	20	0	0	0	0	0	28	1
1	5523159	0	37	0	0	0	0	0	56	1
2	5693080	0	38	0	0	0	0	0	41	1
3	5654129	0	24	0	0	0	0	0	65	1
4	5641070	0	41	0	0	0	0	0	41	1

(9009, 10)

	ID	Gender	Age	Scholar	Hipertension	Diabetes	Alcoholism	Handcap	City	Category
6	5697447	1	13	1	0	0	0	0	0	2
8	5723075	1	8	0	0	0	0	1	32	2
10	5660217	1	9	1	0	0	0	0	73	2
14	5735949	1	48	0	0	0	0	0	35	2
16	5683656	1	10	1	0	0	0	0	48	2

(854, 10)

	ID	Gender	Age	Scholar	Hipertension	Diabetes	Alcoholism	Handcap	City	Category
52	5669777	0	54	0	0	0	1	0	7	3
78	5735234	0	47	0	0	0	1	0	63	3
86	5670093	0	53	0	1	0	1	0	63	3
98	5587737	1	66	0	1	0	1	0	54	3
100	5435669	1	61	0	0	0	1	0	49	3

Datasets sauvgardés.

Figure 5.13: Datasets obtained

Now we must convert these 4 datasets that we have obtained into local ontologies by using the Protege tool. The Figure 5.14 shows one of the local ontologies that we have built. These local ontologies are built each one separately according to their category which is calculated by the K-Means algorithm.

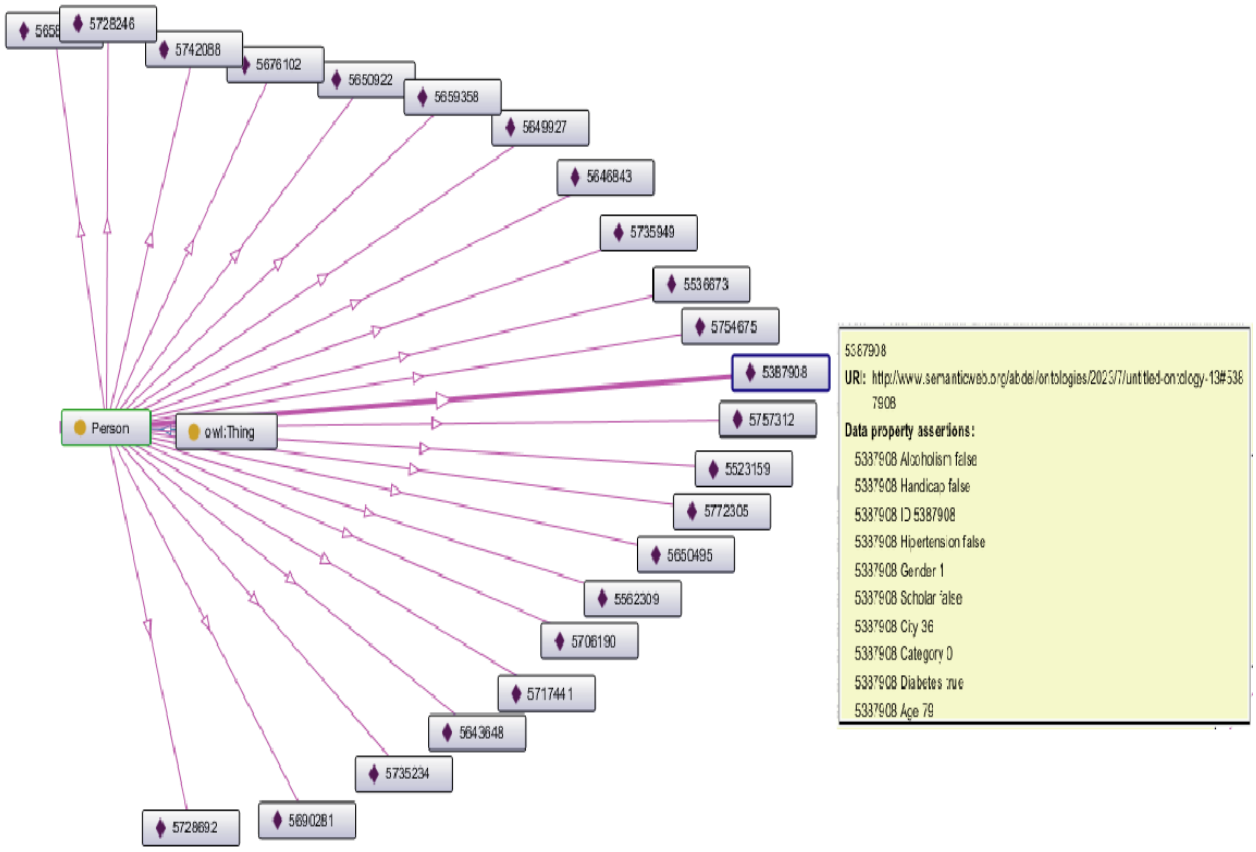

Figure 5.14: Local ontology

The local ontologies elements are:

- **Class:** which is ”Person”.
- **Data property:** ”ID”, ”Gender”, ”Age”, ”Scholar”, ”Hipertension”, ”Diabetes”, ”Alcoholism”, ”Handcap”, ”City”, ”Category”.
- **Individuals:** like ”5387908”.

  Once, we introduce a new individual, we have to introduce the data property assertions.

5.4.4 Global ontology building

After building the local ontologies, we merge them and group them together to build a global one by using the Protege tool. This global ontology would be our Data Warehouse, containing all the data: classes, data properties, individuals representing each person and their information in the datasets used in the beginning.

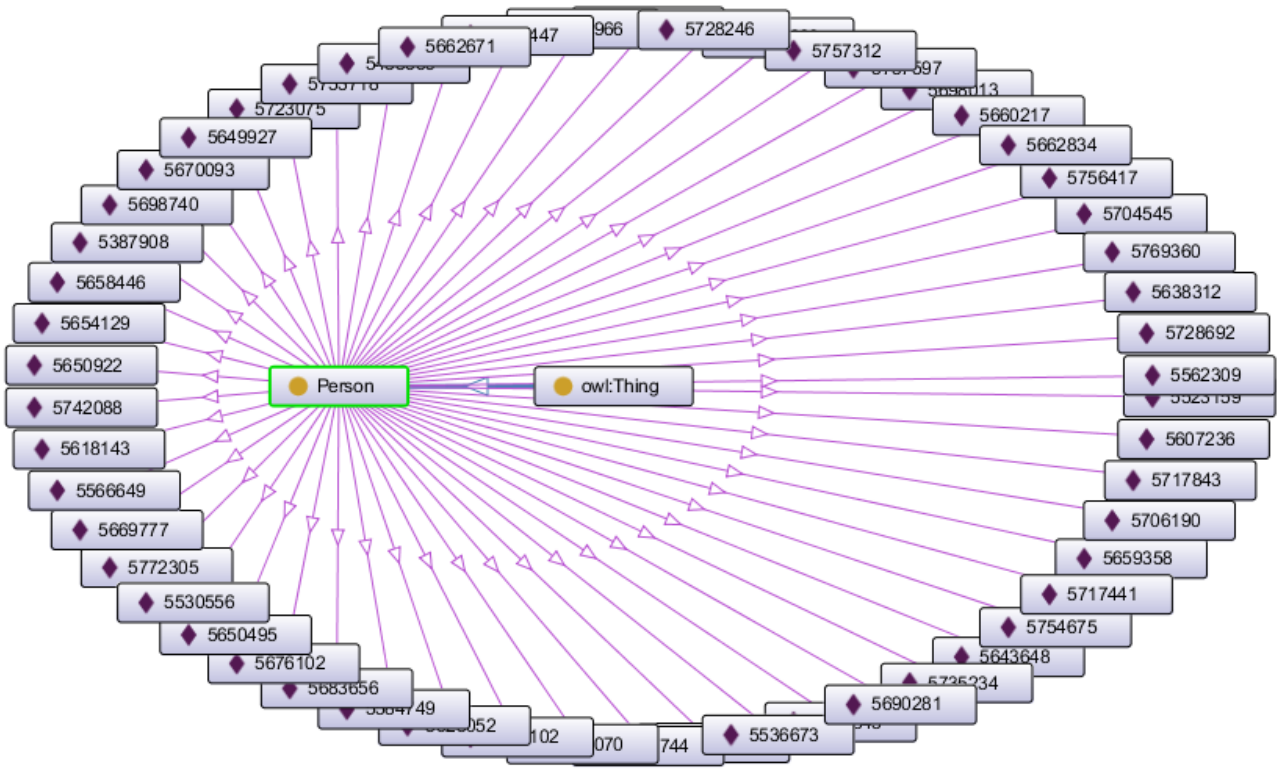

Figure 5.15: Global ontology

5.4.5 Query intepretation and execution

SPARQL queries will be inserted as illustrated below:

1. To get individuals of the class Person who are scholars and live in a specific city:

```
PREFIX ex:
<http://www.semanticweb.org/abdel/2023/7/Global_ontology#>
SELECT ?person ?id
WHERE {
?person rdf:type ex:Person ;
ex:ID ?id ;
ex:Scholar true ;
ex:City "CityName" .
}
```

Figure 5.16: Query 1

2. To find all people with the gender ’male’:

```
PREFIX ex:
<http://www.semanticweb.org/abdel/2023/7/Global_ontology#>
SELECT ?person
WHERE {
?person rdf:type ex:Person ;
ex:Gender "0" .
}
```

Figure 5.17: Query 2

3. To find all people with the age greater than 60:

```
PREFIX ex:
<http://www.semanticweb.org/abdel/2023/7/Global_ontology#>
SELECT ?person
WHERE {
?person rdf:type ex:Person ;
ex:Age ?age .
FILTER(?age > 60).
}
```

Figure 5.18: Query 3

4. To find all people who are scholars:

```
PREFIX ex:
<http://www.semanticweb.org/abdel/2023/7/Global_ontology#>
SELECT ?person
WHERE {
?person rdf:type ex:Person ;
ex:Scholar "true" .
}
```

Figure 5.19: Query 4

5. To get all individuals of the class Person with their IDs and ages:

```
PREFIX ex:
<http://www.semanticweb.org/abdel/ontologies/2023/7/untitled
SELECT ?person ?id ?age ?gender
WHERE {
?person rdf:type ex:Person ;
ex:ID ?id ;
ex:Age ?age .
}
```

Figure 5.20: Query 5

5.5 Conclusion

In this chapter, we have outlined the main steps in our proposed approach. We have presented the datasets we used for the experimentation step, as well as the application of the K-Means algorithm and the SPARQL queries. These steps are all illustrated by screenshots.

The implementation employs different technologies and focuses on ontologies to provide a semantic layer to the data and Machine learning for ontology matching

In the next chapter, we conclude our thesis with a general conclusion covering the steps followed in each chapter, and we give a provide outlook on our future objectives.

Chapter 6

General conclusion

6.1 Introduction

Big Data integration is a dynamic and complex process that requires a combination of technological solutions, data management practices, and strategic planning. Organizations that can effectively integrate and leverage their diverse data sources stand to gain a deeper understanding of their business landscape, leading to improved performance, innovation, and growth. It's important to stay updated on the latest developments in the field to ensure successful integration and utilization of Big Data.

6.2 Problematic

The integration of Big Data raises a number of problems that can have an impact on the effectiveness of the methods and techniques used to exploit it. It faces a number of challenges, such as the variety of data coming from different sources, the very large volume of data, and interoperability, so the systems and tools used to integrate Big Data must be able to work together effectively. Collecting and storing Big Data are important steps, but they are not enough. The data must then be analyzed to extract relevant information. This analysis can help to discover trends, models, or correlations that would not be visible to the human eye.

6.3 Methodology

In this thesis, we proposed a new approach to integrating Big Data in the healthcare sector using ontologies and Machine Learning. In the second chapter, we have presented general concepts about Big Data, ontologies, Machine Learning, ETL, and the relationship between Big Data and Healthcare. In the third chapter, we studied a number of works that deal with this subject, comparing them

in order to gain a clearer understanding of the subject. In the fourth chapter, we proposed a new approach based on the use of ontologies and machine learning to integrate Big Data. In Chapter 5, we created a simulation environment to test our method. We discuss the key elements of the implementation of our approach, as well as the programming language and the datasets used.

6.4 Perspectives

For future prospects, we aim to continue improving our approach. We plan to improve the creation of ontologies from datasets by using well-known frameworks such as Jena to make ontology creation very efficient. As well as this, we plan to create an interface for interpreting and executing SPARQL queries, this interface is created using the front end and back end as well as creating a database for querying the queries. In addition, we plan to generalize our approach to other domains.

6.5 Limits

In our brief, we encountered several problems such as the lack of resources, the need to put security measures in place as the processing of big data can raise confidentiality and security concerns, so it is necessary to guarantee that the data is protected against unauthorized access. We also need the Hadoop cluster, which is specifically designed to store and analyze large quantities of unstructured data.

The lack of datasets that have almost similar data. Voluminous data is often heterogeneous, which can make it difficult to integrate, as it has to be cleaned, normalized, and transformed before it can be combined.

References

[1] H. E. Bouhissi, A. Patel, and N. C. Debnath, "Toward data integration in the era of big data: Role of ontologies," in Semantic Web Technologies, pp. 359–380, CRC Press, 2022.

[2] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," Mobile networks and applications, vol. 19, pp. 171–209, 2014.

[3] A. J. Barton, "Big data," J. Nurs. Educ., vol. 55, pp. 123–124, Mar. 2016.

[4] H. Dhayne, R. Haque, R. Kilany, and Y. Taher, "In search of big medical data integration solutions-a comprehensive survey," IEEE Access, vol. 7, pp. 91265–91290, 2019.

[5] G. Bello-Orgaz, J. J. Jung, and D. Camacho, "Social big data: Recent achievements and new challenges," Information Fusion, vol. 28, pp. 45–59, 2016.

[6] S. Gutta, "The 5 V's of Big Data — medium.com." `https://medium.com/analytics-vidhya`. [Accessed 03-05-2023].

[7] S. Dash, S. K. Shakyawar, M. Sharma, and S. Kaushik, "Big data in healthcare: management, analysis and future prospects," Journal of big data, vol. 6, no. 1, pp. 1–25, 2019.

[8] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?," International journal of human-computer studies, vol. 43, no. 5-6, pp. 907–928, 1995.

[9] H. Hlomani, Multidimensional Data-driven Ontology Evaluation. PhD thesis, University of Guelph, 2014.

[10] I. Artificial, "Ontologías y web semántica 2005,"

[11] G. E. Barchini and M. M. Álvarez, "Dimensiones e indicadores de la calidad de una ontología," Avances en sistemas e informatica, vol. 7, no. 1, pp. 29–38, 2010.

[12] M. Chmielewski, M. Paciorkowska, and M. Kiedrowicz, "A semantic similarity evaluation method and a tool utilised in security applications based on ontology structure and lexicon analysis," in 2017 Fourth International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), pp. 224–233, IEEE, 2017.

[13] T. H. Nguyen, Mining the semantic Web for OWL axioms. PhD thesis, Université Côte d'Azur, 2021.

[14] A. L. Samuel, "Some studies in machine learning using the game of checkers," IBM Journal of Research and Development, vol. 3, no. 3, pp. 210–229, 1959.

[15] M. Mohammed, M. B. Khan, and E. B. M. Bashier, Machine learning: algorithms and applications. Crc Press, 2016.

[16] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," SN computer science, vol. 2, no. 3, p. 160, 2021.

[17] J. Sreemathy, S. Nisha, G. P. RM, et al., "Data integration in etl using talend," in 2020 6th international conference on advanced computing and communication systems (ICACCS), pp. 1444–1448, IEEE, 2020.

[18] "informatica.com." `https://www.informatica.com/nl/resources/articles`. [Accessed 06-08-2023].

[19] A. Sharma, D. Shukla, T. Goel, and P. K. Mandal, "Bharat: an integrated big data analytic model for early diagnostic biomarker of alzheimer's disease," Frontiers in neurology, vol. 10, p. 9, 2019.

[20] K. M. Boehm, E. A. Aherne, L. Ellenson, I. Nikolovski, M. Alghamdi, I. Vázquez-García, D. Zamarin, K. Long Roche, Y. Liu, D. Patel, et al., "Multimodal data integration using machine learning improves risk stratification of high-grade serous ovarian cancer," Nature cancer, vol. 3, no. 6, pp. 723–733, 2022.

[21] P. Li, H. Luo, B. Ji, and J. Nielsen, "Machine learning for data integration in human gut microbiome," Microbial Cell Factories, vol. 21, 11 2022.

[22] S. Vetova, "Big data integration and processing model," WSEAS Transactions on Computers, vol. 20, pp. 82–87, 2021.

[23] M.-E. Vidal, K. M. Endris, S. Jozashoori, F. Karim, and G. Palma, "Semantic data integration of big biomedical data for supporting personalised

medicine," Current Trends in Semantic Web Technologies: Theory and Practice, pp. 25–56, 2019.

[24] I. Mountasser, B. Ouhbi, F. Hdioud, and B. Frikh, "Semantic-based big data integration framework using scalable distributed ontology matching strategy," Distributed and Parallel Databases, vol. 39, pp. 891–937, 2021.

[25] C. Latsou, G. I. Minguell, A. N. Sonmez, O. I. Irurre, M. M. Palmisano, S. Landon-Valdez, J. A. Erkoyuncu, P. Addepalli, J. Sibson, O. Silvey, et al., "Developing an ontological framework for effective data quality assessment and knowledge modelling,"

[26] P. Deshpande, A. Rasin, J. Furst, D. Raicu, and S. Antani, "Diis: A biomedical data access framework for aiding data driven research supporting fair principles," Data, vol. 4, no. 2, p. 54, 2019.

[27] D.-M. Lyu, Y. Tian, Y. Wang, D.-Y. Tong, W.-W. Yin, and J.-S. Li, "Design and implementation of clinical data integration and management system based on hadoop platform," in 2015 7th International Conference on Information Technology in Medicine and Education (ITME), pp. 76–79, IEEE, 2015.

[28] J. Pita Costa, M. Grobelnik, F. Fuart, L. Stopar, G. Epelde, S. Fischaber, P. Poliwoda, D. Rankin, J. Wallace, M. Black, R. Bond, M. Mulvenna, D. Weston, P. Carlin, R. Bilbao, G. Nikolic, X. Shi, B. De Moor, M. Pikkarainen, J. Pääkkönen, A. Staines, R. Connolly, and P. Davis, "Meaningful big data integration for a global covid-19 strategy," IEEE Computational Intelligence Magazine, vol. 15, pp. 51–61, Nov. 2020. Funding Information: Supported by the GYDRA Tool Funding Information: This project was funded by the European Union research fund 'Big Data Supporting Public Health Policies,' under GA No. 727721. Publisher Copyright: © 2005-2012 IEEE. Copyright: Copyright 2020 Elsevier B.V., All rights reserved.

[29] M.-E. Vidal, S. Jozashoori, and A. Sakor, "Semantic data integration techniques for transforming big biomedical data into actionable knowledge," in 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), pp. 563–566, IEEE, 2019.

[30] M. Birgersson, G. Hansson, and U. Franke, "Data integration using machine learning," in 2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW), pp. 1–10, IEEE, 2016.

[31] "Protégé." https://protege.stanford.edu/. [Accessed 23-08-2023].

[32] P. Kaur, P. Nand, S. Naseer, A. A. Gardezi, F. Alassery, H. Hamam, O. Cheikhrouhou, and M. Shafiq, "Ontology-based semantic search framework for disparate datasets.," Intelligent Automation & Soft Computing, vol. 32, no. 3, 2022.

[33] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," Electronics, vol. 9, no. 8, p. 1295, 2020.

[34] "Project jupyter | home." https://jupyter.org/. [Accessed 23-08-2023].

[35] Python.org, "What is python? executive summary." https://www.python.org/doc/essays/blurb/. [Accessed 23-08-2023].

[36] "Numpy." https://numpy.org/. [Accessed 23-08-2023].

[37] "Python plotting — matplotlib 3.4.3 documentation." https://matplotlib.org. [Accessed 08-08-2023].

[38] "pandas documentation — pandas 2.0.2 documentation." https://pandas.pydata.org/docs/. [Accessed 23-08-2023].

[39] "scikit-learn: machine learning in Python x2014; scikit-learn 1.3.0 documentation — scikit-learn.org." https://scikit-learn.org/stable/. [Accessed 28-08-2023].

[40] "Kaggle: Your Machine Learning and Data Science Community — kaggle.com." https://www.kaggle.com/. [Accessed 08-09-2023].