Università degli Studi di Salerno

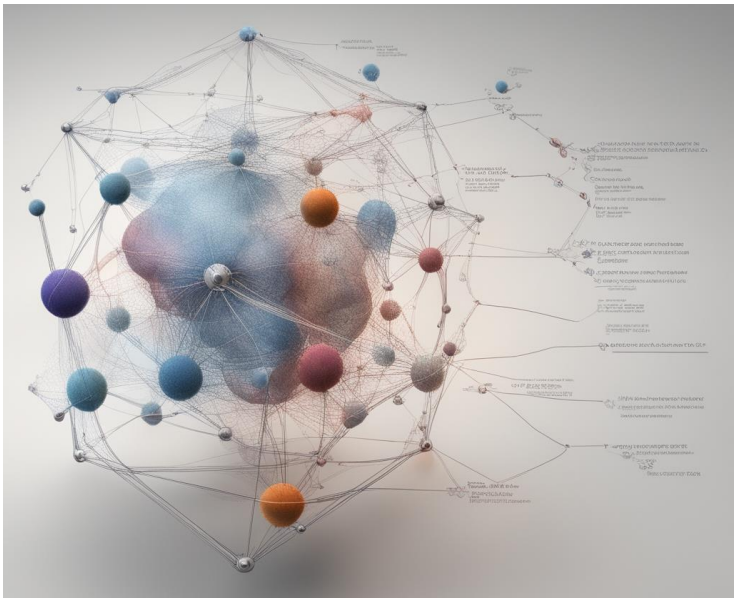
.DIEM

Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata

Tesi di Dottorato/Ph.D. Thesis

Knowledge Base Construction Methods for Neuro-Symbolic IA Exploitation

Diego Rincon-Yanez



Supervisor: **Prof. Sabrina Senatore**

Ph.D. Program Director: **Prof. Pasquale Chiacchio**

Dottorato di Ricerca in Ingegneria dell'Informazione
Ciclo 35 – anno 2023

Università degli Studi di Salerno

.DIEM

Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata

Dottorato di Ricerca in Ingegneria dell'Informazione
Ciclo 35

TESI DI DOTTORATO / PH.D. THESIS

Knowledge Base Construction Methods for Neuro-Symbolic IA Exploitation

DIEGO RINCON-YANEZ

SUPERVISOR: **PROF. SABRINA SENATORE**

PHD PROGRAM DIRECTOR: **PROF. PASQUALE CHIACCHIO**

Anno 2023

Acknowledgements

The Ph.D. journey and this thesis have been a life-changing experience for many reasons, but especially for remembering who I am, where I come from, and what drives me daily to accomplish life goals, especially in times like when this thesis has taken place.

Thanks to all the people who enabled me in one way or another to accomplish this.

(Version 3.0 - 11th October 2023)

Abstract

Knowledge representation has long been considered a key task in the information sciences. Nowadays, with AI's broad scope in all aspects of daily life, understanding how to represent, display, and analyze human knowledge has become increasingly vital. This thesis relies on computational systems' inherent ability to deliver comprehensive and accurate responses using a combination of inductive learning, commonly known as Deep Learning, and logic learning; the topic is Neuro-Symbolic AI (NeSyAI). To fully use the NeSyAI system-like characteristics, Knowledge Bases must be built, which are a collection of statements ordered to form a Graph type network structure, also known as a Knowledge Graph (KGs) in the semantic community. Knowledge Graphs can offer background information on one or more domains, adding extra context to the task at hand. This thesis offers numerous methods for integrating contextual knowledge into Neuro-Symbolic AI systems by producing semantically enhanced knowledge bases for later exploitation.

Initially, the Internet of Things (IoT) real was explored, where sensor network characteristics are mapped into a traditional OWL-type ontology using and building a knowledge base of the sensor, events and observations with the aim of predicting events and places. For this, a state-of-the-art ontology was created for IoT event recognition purposes, followed by the implementation of a convolutional Neuron Network, which is able to take the KG and discover new wave information. These two tasks combined in a single flow produce and validate a new architectural pattern for

Neuro-Symbolic AI. To validate the proposed approach, two use cases were implemented: a volcano discovering seismic events and a cleanroom monitoring environment for discovering pollution events.

In the second part, knowledge graph embeddings (KGE) were exploited to push the boundaries of the knowledge base construction task in several alleys, such as knowledge validation, question answering, and link prediction from the Neuro-symbolic AI perspective. These tasks were used along with well-known metrics to test the construction approaches from prosaic text leveraged by the external knowledge of one of the biggest Open KG available, such as DBpedia and transformers-type models, arriving at a compelling results compared to the current SOTA, allowing to test also new and improved NeSy AI architecture patterns. On the other hand, the need to explore synthetic data generation was raised considering the need for continuous experimentation and automation to test the experimentation scenarios. Finally, an initial explainability approach for KGE was conceptualized and applied to the commodities tradeflow prediction use case.

List of Publications

Journal

- Falanga, M., De Lauro, E., Petrosino, S., Rincon-Yanez, D., & Senatore, S. (2022). Semantically Enhanced IoT-Oriented Seismic Event Detection: An Application to Colima and Vesuvius Volcanoes. IEEE Internet of Things Journal, 9(12), 9789 – 9803. 10.1109/JIOT.2022.3148786

- Rincon-Yanez, D., De Lauro, E., Petrosino, S., Senatore, S., & Falanga, M. (2022). Identifying the Fingerprint of a Volcano in the Background Seismic Noise from Machine Learning-Based Approach. Applied Sciences, 12(14), 6835. 10.3390/app12146835.

- Di Paolo, G., Rincon-Yanez, D., & Senatore, S. (2023). A Quick Prototype for Assessing OpenIE Knowledge Graph-Based Question-Answering Systems. Information, 14(3), 186.

- Rincon-Yanez, D., Ounoughi, C., Sellami, B., Kalvet, T., Tiits, M., Senatore, S., & Yahia, S. ben. (2023). Accurate prediction of international trade flows: Leveraging knowledge graphs and their embeddings. Journal of King Saud University - Computer and Information Sciences, 35(10), 101789.

Conference

- Rincon-Yanez, D., Lauro, E. De, Falanga, M., Senatore, S., & Petrosino, S. (2020). Towards a semantic model for IoT-based seismic event detection and classification. In 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020 (pp. 189–196). IEEE. 10.1109/SSCI47803.2020.9308329.

- Rincon-Yanez, D., Crispoldi, F., Onorati, D., Ulpiani, P., Fenza, G., & Senatore, S. (2021). Enabling a Semantic Sensor Knowledge Approach for Quality Control Support in Cleanrooms. In International Semantic Web Conference (ISWC) (Vol. 2980).

- Rincon-Yanez, D., & Senatore, S. (2022). FAIR Knowledge Graph construction from text, an approach applied to fictional novels. Proceedings of the 1st International Workshop on Knowledge Graph Generation From Text and the 1st International Workshop on Modular Knowledge Co-Located with 19th Extended Semantic Conference (ESWC 2022), 94–108.

- Rincon-Yanez, D., Mouakher, A., & Senatore, S. (2023). Enhancing downstream tasks in Knowledge Graphs Embeddings: A Complement Graph-based Approach Applied to Bilateral Trade. Procedia Computer Science, 225, 3692–3700.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Knowledge representation (KR) is focused on designing and improving symbolic notations to express and capture real-world facts, [4, 5] that better enable automatized decision-making tasks. Over the past ten years, Knowledge Graphs (KG) have received a lot of interest since they effectively organize and represent knowledge, allowing further use in different baselines and applied solutions. Knowledge graph models have been widely used to arrange and describe data in various sectors [6], including medicine [7], health care [8], biology [9] and finance [10]. KG are playing an increasingly critical role in many applications [6], such as drug discovery [7], user recommendation [11], dialogue systems [12], and question answering [13, 14].

Despite the wide range of research paths that the term knowledge graphs (KG) intervenes and even several years of development in the research community, there is no consensual definition of a knowledge graph. Some authors, for example, treat the KG as any populated ontological Knowledge Base (KB) indiscriminately, while others transform the results of traditional Knowledge Engineering into a KG. Generally, the KG has become an umbrella term where the Knowledge Bases, representations, populated ontologies, ontological schemas, or any other type of graph-oriented data connected base single or multi-structured data conveys; observing and experiencing this premise, this thesis aims to wrap (but not limit) the Knowledge Graph concept from an applied computing viewpoint. First, Chapter 2 draws a topic-framed applied taxonomy. The intention is to provide context on recent trends,

technologies, tasks and frameworks used in the KG community; then, using the automated knowledge base/graph construction as a main research challenge, a path is taken from the traditional semantic web and knowledge engineering methods, explored in Chapter 3 to describe more recent approaches applying vector representations the knowledge graph/base construction, this is consigned in Chapter 4.

However, to provide more contextualized research on the works developed in this thesis, it is essential to introduce the concepts of NeuroSymbolic Systems [15], some initial approaches into a Design Patterns for Hybrid Learning [16, 17] and the more recently introduced, Semantic Web and Machine Learning Systems (SWeMLS) [1]. The overlap of this research provides an initial stage for the results presented in this thesis.

The Semantic Web Technologies, since the release of the semantic cake in 2006 [18], detailed in Section 2.6.1, open the chance to integrate information from multiple domains, languages and fields of human knowledge through a standardized framework, and supported by a set of concrete specifications, such as XML, RDF, OWL, SHACL, etc. Later, the systems able to leverage semantic knowledge were called NeuroSymbolic Systems, thanks to the inherent capacity to exploit the data relations defined through semantic specifications and logic-based representations [15]. Let us notice that NeuroSymbolic refers to the representation and logic and the integration of methods in standard AI, such as stochastic gradient descent (SDG), backpropagation, etc. This type of integration is also known as Hybrid Learning [19]. The impact of this research path relies on producing robust and adaptable systems able to exploit the hybrid approach, reaching human-explainable answers by combining logic and associative techniques.

Moreover, from a high-level perspective in the Knowledge Representation field, the meta-cognition in AI has been studied very recently in [20]. The study defines what the authors have called fast and slow thinkers, referring to *fast* as procedures or systems able to learn from past experiences, which is also called intuitive, and *slow*, to traditional solvers that employ reasoning and well-defined rules, in other words, rational. The task of swapping from slow to fast and vice versa positively affects the learning capabilities of basic methods and reveals the profound importance of the thinker's coordination while denoting the inherent characteristic of NeuroSymbolic systems to be modular.

Normally, these systems are based on modular atomic tasks performed on a few KBs, normally assembled to perform complex tasks favoring comprehensibility and maintenance.

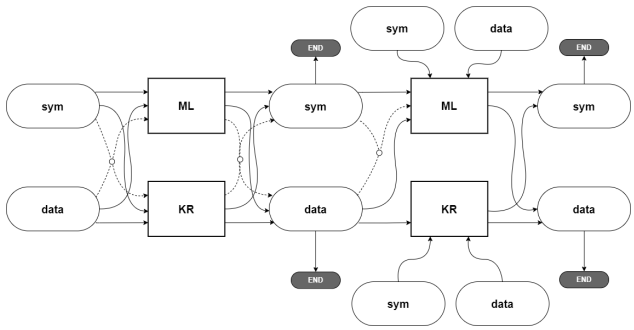


Figure 1.1: Meta processing flow, adapted from [1], representing the over 41 SWeMLS patterns for hybrid learning presented

The appropriate usage of these fast and slow thinkers raises an additional endeavor for researchers to build proper Neuro-symbolic approaches. However, in the early work of [16], 15 initial patterns were defined for hybrid-oriented learning systems; of these 15 patterns, 11 are oriented to NeuroSymbolic systems; later, in [1], more than 20 different patterns were introduced. Figure 1.1 provides the general pattern of the approximately 41 existing Design Patterns extended in [1]. The figure indicates two different types of shape: rounded, representing that the input can be symbolic *(sym)*, and referring to semantic data or non-symbolic, i.e., raw data *(data)*, and square, indicating a Knowledge Representation domain process *(KR)*, such as a logic-based rule, or Machine Learning *(ML)*, such as Artificial Neural Networks (ANN).

The inductive learning task, i.e., ML tasks and the rational approach, i.e., KR, traditionally have been seen as independent tasks that strive to resolve separately. Recently, a parallel was defined on these two learning tasks [1], describing the synergy between them and integrating them into a broader field called NeuroSymbolicAI. This emerging field holds the research and development of AI systems, combining these two tasks. With the recent definition of NeuroSymbolicAI, An additional problem arises, given the strong advancement of ML and KR individually to being used unified as complementary approaches.

The unification task problem leads to the need to demarcate the

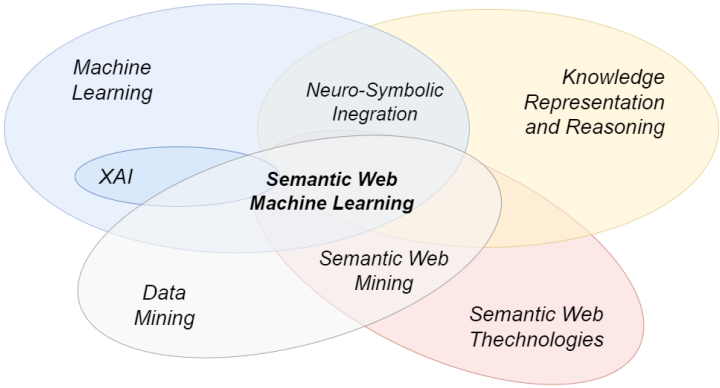


Figure 1.2: SWeMLS landscape, taken from [1]

Semantic Web resources and Machine Learning Systems (SWeMLS) components; this newly indicated research area is defined as combining SWT with an inductive model (ML) to solve a specific task. Figure 1.2 shows the intersection of the involved domains. The research area is marked as the intersection between Knowledge Representation and Reasoning, leveraging Semantic Web Technologies with the more well-known Machine Learning models; this intersection denotes neuro-symbolic integration or Hybrid Learning. Moreover, Data Mining, Explainable AI (XAI), and Hybrid Learning coveys into the Semantic Web Machine Learning Systems (SWeML).

Exploring the early semantic extensibility from Semantic Web Technologies (SWT) to NeuroSymbolic Systems, to later have the symbiosis within AI methods and reach NeuroSymbolic IA; prove that these approaches can follow taxonomical integration rules, a.k.a, design patterns as bonding mechanisms to design test and deploy SWeMLS. Finally, the KG represents the primary element able to converge all the concepts explored so far, becoming a strong articulation ingredient for delivering knowledge across all the technologies, approaches and computing paradigms.

From the previously denoted research scope, the main contribution of this thesis is twofold, listed as follows.

- The traditional use of Knowledge Representation approaches followed by a rigorous Knowledge Engineering process to design

and test an interoperable Ontology, focusing on later integrating with Deep Neural Networks to solve an Event Recognition task. The produced SWeMLS has been tested using sensor networks to gather data implementing Internet of Things (IoT) concepts. The proposed work was tested in two diametrically opposite domains, like Vulcanology, by sensing signals from actual mountains to Smart Manufacturing by monitoring environmental metrics in Cleanrooms. This last solution is working at the production level.

- The use of Knowledge Graph Embedding models as a baseline framework for solving, validating, and enhancing NeuroSymbolic integration approaches with different triple-oriented tasks such as Domain-Specific Knowledge Graphs Construction, Synthetic Triple Generation, and a novel method of knowledge graph construction using gravitational models with explainable characteristics that allow enhancing downstream tasks in traditional machine learning algorithms.

The research lines previously described frame the contribution presented in this thesis; each line provides a comprehensive study of the topics and intersections illustrated in Figure 1.2. The following structure is proposed to thoroughly present the results archived in this thesis.

The chapter 2 explores the topic of Knowledge Graph (KG) through a taxonomic structure, which provides an all-around scope of KG from the perspective of applied computing. The chapter starts with a preliminary section on the fundamental notion of KGs. Then a formal definition of the KG creates a defined concept baseline, starting from the characterization of the atomic statement. The formal definition presented open space to the knowledge tasks describing the operations range, articulated within the KB life-cycle: being an applied-oriented taxonomy, the specifications for supporting the design gain importance in the concept definition as well as the models to interoperate with the knowledge graph. Therefore, a look at semantic web technologies and knowledge graph embeddings is given. Finally, a view of the most prominent OpenKG and how articulate the research ecosystem give space to numerous applications and real-world use cases where knowledge graphs are used in different fields.

The contribution sections start with chapter 3 with a comprehensive state-of-the-art revision on semantic sensor network approaches from

the semantic stack, such as standardized ontologies, like SSN and SOSA; similar approaches exploring some deep learning methods, such as multilayer perceptrons (MLP), and other more conventional like rule-based approximations. Further in the chapter, the designed ontology, namely VEO (Volcano Event Ontology), is debriefed from the formal perspective by describing the properties of SOSA classes and extensions and other more generic ontologies such as TIME and GEO ontologies. The deep learning method is introduced by presenting the feed-forward architecture to classify the ontological annotated data. Finally, the two applied scenarios are presented. The first is from the Seismic domain: it is based on an integrated system capable of performing event recognition leveraging the semantically annotated data in controlled experiments using specific seismological events. An additional experimentation set aims to detect the background noise of specific mountains. The second scenario presents a reduced version of the same VEO ontology oriented to the case of environmental control for a manufacturing use case in one of the most prominent companies in Italy.

The second contribution chapter, Chapter 4 explores the vector representation focusing on the knowledge graph embeddings (KGE), which attracted the community to the link-prediction task, besides the knowledge representation capabilities of these low-dimensional representation methods. The chapter also aims to validate methods of constructing unsupervised knowledge graphs by exploiting vector representations (KGE models) from the perspective of zero-shot learning (ZSL), where ZSL refers to an approximation of AI in which no data samples or data scenarios are provided to an AI method to learn a given task inductively. In consequence and with the goal of augmenting and improving the downstream tasks in KGE models, a method of constructing synthetic triples is provided, tested with benchmark experimental scenarios that demonstrate the effectiveness of the method in augmenting training metrics. In the last section, the gravitational model is integrated into the knowledge graph construction process with the intention of creating a vector representation (KGE) to elicit latent relationships to leverage as feature inputs for other ML methods. The bilateral trade domain was explored to test this procedure, and well-known methods such as Decision Trees and Logistic Regression were tested. Still, in more recent approaches, such as Graph Neural Networks, the results of each experiment proved to positively impact the vector representation as an

input feature for forward models in a given pipeline.

In synthesis, this thesis provides a comprehensive panorama on the state of the knowledge Graph Construction task from two different NeuroSymbolic AI or SWeMLs (Semantic Web Machine Learning Systems) perspectives, exploring two different and contrasting approaches, one starting from the more logical approach by exploring Knowledge Engineering techniques and the other from the inductive perspective exploring different scenarios in the Vector Representations or Embeddings. In both explored efforts, valuable contributions were made by publishing in top-tier venues and impacting respective journals.

Chapter 2

Knowledge Graphs - A Taxonomy Study

To gather, organize and represent information, that is intelligible for humans, and can later be employed by machines, is an arduous undertaking. The appropriate and discriminated use of information in natural language needs a model that describes and organizes facts in an enhanced way allowing their applications to be accurate and pertinent, also helping to discern from the data what is relevant and what is not. With the massification of the internet arriving at almost 60%[1], the vast amount of data disseminated on the web and the multi-source environment nowadays available in different domains need solutions to organize practical knowledge in various domains and contexts[21].

From a more technical viewpoint, the heterogeneity of the source formats and technologies gives a high level of complexity and needs integration to guarantee compatibility. This is why Semantic Web Technologies (SWTs) has been a standard and agnostic approach for information enhancement and representation around knowledge description and contextualization [22]. Moreover, SWTs become essential for data understanding providing domain-invariant structures to represent the conceptualization of different fields, describing data instances, values, and the relations among them.

Big Internet companies like Google, Wikipedia, and Facebook have found a simple but powerful unified tool in the Knowledge Graph (KG)

[1]World Pop. Internet Access https://data.worldbank.org/indicator/IT.NET.USER.ZS

field to describe their multi-structured and multi-dimensional knowledge base, collecting data from its users and transforming it into vast KBs.

Conversely, a KG can be described as an unstructured semantic-directed data network; this network is built to discover the intrinsic relations between concepts of one or more correlated domains. Even though the KG concept is not recent [23], many approaches and definitions have emerged in the last years, arriving at new and exciting advancements in many directions.

This explosion of research lines leads to a community debate about a more accurate topic characterization and indiscriminate term usage. Given the previous discussion and contributing to a more rounded topic usage, a characterization of the topic was made with a holistic computational perspective to exploit it and provide computational, conceptual and practical support for further research.

2.1 Introduction

Knowledge representation (KR) is focused on the design and improvement of symbolic notations for the expression and the capture of real-world facts [4, 5] for later automatized decision-making. In this way, knowledge representation (KR) is crucial to provide a straightforward way to describe information inside a finite group of facts from a specific domain to create a knowledge base (KB). One way of KR is Knowledge Graphs (KG). An early literature definition of the Knowledge Graph topic was defined in [24], in which a KG is presented by the combination of *multiple data sources*, a *knowledge base (KB)* and a *reasoning system*. Combined, these three elements can provide a dynamic for performing knowledge operations in different scenarios inside of the knowledge base and the outside world.

A complete KG definition is not trivial. A KG is often defined as a set of semantic statements grouped by a graph-like schema [25, 26], or in other definitions it is a Knowledge Base [27, 28] (KB), the constant interchange between Knowledge Base (KB) and Knowledge Graph (KG) concepts is remarked also by the authors [24, 29, 30]. this draws the need to debrief the KG concept more profoundly.

One of the research community's widely accepted definitions was made early in the Editorial of the Journal of Web Semantics in 2016

[31, 32].

> "Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities."

Starting from this definition is possible to create a taxonomy to frame the preliminary needed concepts and the complementary techniques, methods and technologies. is important to notice that there is no intention to draw a boundary that limits the line of study from either a theoretical or practical perspective, but rather provide a more structured view and explore each leave in the taxonomy that could lead to further research.

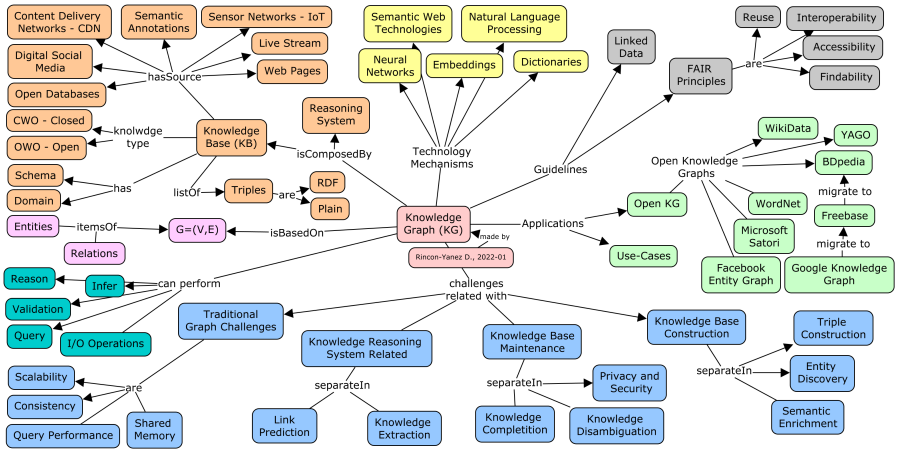


Figure 2.1: Proposed Knowledge Graph Taxonomy

The proposed taxonomy, shown in Fig. 2.1, presents the topic from its foundations, starting from the formal perspective represented by the colour Purple and Orange for the properties of the Knowledge Graph. The colour blue represents the challenges for the construction, maintenance, and other different task related to the knowledge graph life cycle; the grey branch presents guidelines developed by the research community and finally, the green branch present the applications derived from the KG especially making an emphasis on the Open Knowledge Graphs.

The sections of this chapter are divided into the branches of the proposed taxonomy. However, before starting with the KG definition, it is relevant to introduce some preliminary concepts. On the one hand, the *FAIR Principles* described in Section 2.2.2 hold the foundational cornerstones to the openness and massiveness of the KG in the general scope. Section 2.2.1 references the *Open Data Cloud* community that gave maintenance to the most effective integration of ontological and open knowledge bases in the world. These two elements serve as leverage for the KG research community and determine the foundation upon which the KG research community is built.

The main characteristics of the KG need to be defined: a description in Section 2.3 uses a formal approach to conceptualize the components; this formal definition will work as a pillar for the developed work in Chapter 4. After the knowledge base is formally defined is necessary to manipulate the information inside this network-based data model; therefore, the *Knowledge Tasks* are presented using the Knowledge Base life cycle to separate each of the tasks and their purpose in the different time moments of KB life span; this cycle begins with the description of the issues involved in creating and maintaining a KG, the actions produced; those tasks are described in Section 2.4. Finally, the *Technology Mechanism* like, Semantic Web Technologies and Embeddings, provide tools to implement standard methods providing a computing perspective used to operate the KB data; these are described in section 2.6.

In addition to the components and tasks involved in the KG, the *Open Knowledge Graph* is introduced in Section 2.5, these are large disclosed KBs available to be used openly across the web. these OpenKG represent the actions and active contributions not only from the academic but also from the industry to provide a seamlessly open semantic platform available to the general public. To finalize the chapter, a review on *Applications*, specifically industry-oriented use cases, are presented in Section 2.7, reflect not only the potential of the topic but also the active research and industrial community working around the topic.

2.2 Preliminaries

Even though the Semantic Web Technologies (SWT), explained in Section 2.6.1 provides a standardized framework to define knowledge representation across domains and technologies, there is the fundamental need to share and reuse knowledge requires the definition of a group of even higher-level best practices to help standardize [29]; these practices provide quality features in the knowledge structure and potentially a point of integration with other KBs. This section introduces two concepts, Linked Open Data and the FAIR principles; these two visions are responsible for enabling the actual networked data ecosystem.

2.2.1 Linked Open Data

The history of the Linked Open Data (LOD) movement [33] begins with the well-known Linked Data principles [18], proposed by Tim Berners-Lee at the general technical level; these principles provided simple best practices for building KB on the Web. Later, they were widely adopted by w3c as a vision for interconnecting data in a federated scheme in which a resource identified by a Unified Resource Identifier (URI) is shared and consumed across the Web, while retaining control of the source of the data. From the perspective of the Semantic Web, these principles are important design rules for describing data about the real world and other resources in a machine-understandable way.

The proposed LOD was a foundation schema for Open Government[2]. This effort spawned the Open LOD Cloud Community [3], providing an umbrella platform to integrate all the known Open KGs into a massive collection of semantically enhanced data with over 190 billion triples [4].

2.2.2 FAIR Principles

The acronym FAIR [34] is widely used, especially when it is related to scientific data. It represents a machine-reusable approach for publishing data; the principles refer to the data descriptors, the accessibility of

[2]https://www.w3.org/egov/wiki/Linked_Open_Data
[3]Open LOD Cloud - https://lod-cloud.net/
[4]https://dbpedia.org/resources/knowledge-graphs/

the general collection, and the protocols implemented to publish or build a dataset. The letters FAIR refer to the Findability, Accessibility, Interoperability, and Reuse concepts.

- **Findability:** is related to the metadata explanation and how it is represented and linked by a global identifier, e.g., URI, IRI, DOI, SKU. This covers the content data or the metadata.
- **Accessibility:** draws the descriptor persistence, even if the actual data is no longer available, and the implementation for open or protected access collections through standardized communication protocols, e.g., HTTP, TCP, FTP, normally defined by Internet Engineering Task Force (IETF). Data should be openly available and accessible, with clear policies and procedures for access.
- **Interoperability:** explores the possibility of reusing metadata by sharing the same language conventions and references to other datasets or data points; this can be ensured by the use of standard vocabularies of high-level ontologies, e.g. SKOS (Simple Knowledge Organization System).
- **Reuse:** Data should be well documented and licensed for reuse, with clear guidelines for proper licensing of use, source trust, and community quality standards for data sharing, e.g. CC-BY-SA, MIT, etc.

In the past years, the use of the FAIR principles has increased, not only for scientific data or community-oriented data management inside the SWT and KGs; but also in other contexts, such as IA communities or developers. Encouraged by this, communities such as Open Linked Data Community (OpenLOD) and Semantic Web, in general, aim to join efforts in defining a FAIR Knowledge Graph topology [35].

2.3 Formal Definitions on KG

This section will cover the **Knowledge Base** and **Reasoning System** concepts. The knowledge facts are normally expressed as triples, and each one is defined by a combination of unique *Subject, Predicate* and *Object*. The **Subject** represents an entity: it can be any object, concept, or idea in the real world. The **Predicate** represents the action performed by the Subject over an **Object**. The **Predicate**, on some occasions,

can represent a Subject property, and an Entity Property can be defined as a primitive characteristic of that entity. The triple description is given as $triple = (Subject, Predicate, Object)$ or $T = (S, P, O)$, where an example of a predicate as an Entity can be expressed, e.g., $T = (BarackObama, birthPlace, Hawaii)$ and an example of Object as a Primitive, e.g., $T = (Barack_Obama, birthYear, 1961)$

2.3.1 Knowledge Base

According to [4], a Knowledge Base is characterized by facts, assertions, and rules grouped in some collection $Kb = \{St_1, St_2, St_3, \ldots, St_n\}$. Normally a graph, from its most basic form, is composed **Vertex (V)** and **Edges (E)**. This collection of facts can be seen take as a collection of triples as well that, arranged in a collection, can constitute a graph.

Therefore, the network models can be divided by the nature of the relations having a more generalized and homogeneous[29] form such as defined in Definition. 1; a homogeneous graph is a graph where all nodes and edges have the same type, which means there is only one type of entity and one type of relationship in the graph. For example, a graph where all nodes represent people and all edges represent friendship relationships between them is a homogeneous graph.

Definition 1 *A homogeneous KB is given by defining the edges* (E) *as a unique type of directed relationship between the vertices* (V)*. where the* $Kb = \{St_1, St_2, St_3, \ldots, St_n\}$*, and given a single triple* $T = (S, P, O)$ *in* $G = (V, E)$*, given the previous context the* $S \cap O \equiv (V)$ *and* $P \equiv (E)$

Given the complex nature of the *Vertex (V)* and the implications of representing the real-world interaction between the entities, it is necessary to extend the homogeneous graph features by inserting additional subsets of *Relations (R)* to the *Edges (E)*. Those relations describe a more organic connection between the nodes; according to some authors, this type of graph is heterogeneous [36, 29] in the Definition 2 is detailed.

Definition 2 *The Heterogeneous Graph is represented in the form* $G = (V, E, R)$*. This type of graph can represent an unlimited amount of relations among the Vertex.* $T = (S, P, O)$ *in* $KB = (V, E, R)$ *where each edge* $p \in E$ *is the unique relation type between the vertices* $V \equiv$

$(s \cup o)$; and relation $r_i \in R : r_i = (v_i, p_h, v_j)$ with $i, j = 1, \ldots, |V|$, $h = 1, \ldots, |E|$ describes a triple, composed of an edge connecting the vertices.

To use a Knowledge Base (KB) as an interchangeable information source is necessary to adopt the W3C Resource Description Framework (RDF) standard[5] to represent each knowledge assertion of the KB. This provides a standard way to add semantics to entities and relationships among them, making it machine-understandable, i.e., the machine can process using query systems or traditional reasoners or any AI model to make tasks about the triple-based assertions.

In the RDF data model, facts are expressed as subject-predicate-object (SPO) triples in a graph-based representation. Each statement T is expressed as $T = (S, P, O)$. To summarize, the traditional heterogeneous graph definition and the RDF triple definition combined can be expressed in the Definition. 2.

Semantic Web Technologies provide a way to incorporate knowledge modeling and automatic deduction in the Web. From a semantic viewpoint, the data that compose a KB are concepts defined in predefined schemes [37]. These schemes are normally OWL-based conceptualizations of the real world in the form of entities, relations, and semantic rules, grouped in ontology form. The ontology is a formal representation of the conceptualization of a reference domain. It is done by implementing the W3C Ontology Web Language [6] (OWL) that uses RDF, adding major expressive power to describe more structured statements.

Although the most common datatype inside a KB is natural language text, this will not represent a limitation of the supported data type. It is possible to represent data from any *Data Source* type, leveraging web-oriented tasks, such as web scraping [38], social media analysis [39], sensor networks mapping [40] or even traditional relational data analytics.

Finally, the statements collected inside a KB can be categorized into Close or Open World Assumption (CWA / OWA) [41, 36]. This assumption represents the knowledge representation approach for a ground truth into the KB, granting the Close assumption as false to any other states outside of the KB. Open Assumption is the possibility

[5]W3C RDF Official Website - https://www.w3.org/RDF/

[6]https://www.w3.org/OWL/

to find the truth in a statement outside or not yet represented into the KB.

In conclusion, the KB can be defined as a collection of semantic statements that, using a directed graph model, represent actions and/or characteristics of a group of entities in one or more knowledge domains.

2.3.2 Reasoning System

In light of the definition of KG earlier, it is necessary to incorporate some mechanism that allows KG to perform operations on the statements or within the KB with external systems, in [24] is called a Reasoning System; this reasoner intent to be like a traditional reasoner, capable of perform logic assumptions based on predefined rules based on the defined schema. However, in the KG context, the Reasoner or a KG Reasoning System (KRS) combines one or more computing methodologies or techniques to read, write or compute data inside the KB statements. These methodologies could include using one or more Artificial Intelligence techniques like Machine learning or Logic Engines to operate inside the graph schema. e.g., Knowledge Graph Embeddings (KGE) as shown in Definition 3.

Definition 3 *Let* $KG = (KRS, KB)$ *or* $KG = (KRS, (V, E, R))$ *where:* $KRS = \{r_1, r_2 \dots r_n\}$ *where:* $n \geq 1$ *e.g.,* $R_i : KGE(KB) \equiv TransE(\{St_1, St_2 \dots St_n\})$*, in which TransE is determined by their loss function* $\mathcal{L}, TransE_{St_i} = -\|e_s + e_p - e_o\|_n$ *where* $n = St_n$

In a nutshell, the Knowledge Graph (KG) is composed of a Knowledge Base (KB) and Knowledge IA techniques operating on top of the KB, a KG Reasoning System (KRS), meaning $KG = (KRS, KB)$.

Leveraging the power of the KG in the given broader form, a wide range of domains and applications are covered, starting from the ground graph knowledge analytics [42], and relation validation methods [43], moving through the semantic field such as ontology reasoning about graph models [44, 45], using multiple RDF data sources [46], and a more traditional fuzzy approach in data development[47], to reach more innovative usage methods such as using natural language techniques to make queries over the KB [48], among many.

2.4 Tasks in Knowledge Graphs

The KB life cycle is shown in Fig. 2.2 and consists of three main phases: *Knowledge Creation* when the initial knowledge is discovered by providing the initial statements and foundation to the Knowledge Base, *Knowledge Curation* where the internal triples need to be annotated, and validated to comply with the semantic purposes of the KB, and finally, *Knowledge Deployment* where the knowledge is in a mature state and can be used together with a KRS to manage it.

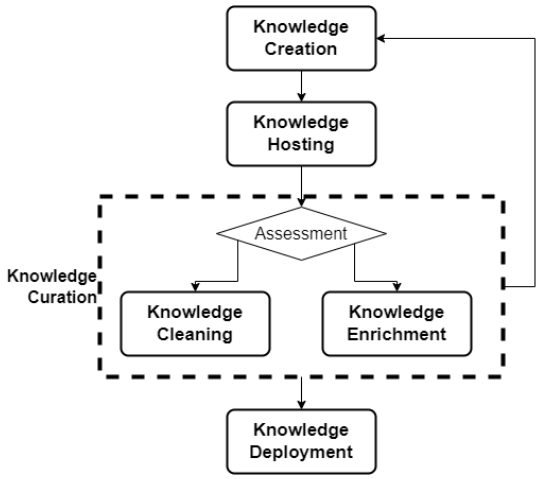


Figure 2.2: Knowledge Base Life Cycle, adapted from [2]

Due to the complexity of the KG topic, a wide spectrum of challenges has been spawned, according to [49, 28, 50, 42, 51]. So these challenges can be split into three groups that can match the KB Life Cycle, even though some overlaps can be among the groups; these three can be roughly categorized into (1) KB Construction process, (2) KB Maintenance, (3) Related to the AI techniques to operate the KB.

The knowledge tasks, normally, are atomic operations performed over a KB; in the following list, some tasks are associated with the states in the KB life cycle. These are mainly oriented to the stage where the knowledge base is, but they are not limited to that. Additionally is important to notice that if well some tasks are representative of the semantic domain, thanks to recent research, the task covers methods

from the traditional IA spectre [1], extending the KB to what they call Semantic Web Machine Learning Systems.

- **Knowledge Base Construction:** Entity Recognition (i.e., Node Recognition), Entity Discovery, Semantic Enrichment, Knowledge Disambiguation, Triple Construction.
- **Knowledge Base Maintenance:** Knowledge Assessment (i.e., Knowledge Completition), Privacy and Security, Coverage
- **Knowledge Reasoning System:** Knowledge Extraction, Link Prediction, Knowledge Fusion (i.e., Integration), Knowledge Representation, Ontology Inferencing, Fuzzy Logic, Deterministic Models, etc.

2.4.1 Knowledge Base Construction

The knowledge base construction (KBC) can be achieved by assembling the first two stages of the KB life cycle through the process to populate a KB defined in [52] and then annotating the knowledge with semantic information [53]. The annotation process can be either automated or human-supervised. Specifically, the last one can be (1) curated by a small group of experts or (2) collaboratively done by a large group of volunteers. Supervised annotation process methods must meet one or more schemas to get a proper ontology-based population, while the automated method can uphold the schema-free and schema-based approaches [2].

The KBC has evolved over time in response to the need for Knowledge Representation [54] for data ingestion as well as human consumption data. The evolution has increased the complexity of data dimensions, the number of sources, and the diverse data typology[55]; for these new additional challenges, semantic tools [56] can boost the gathered data, providing cross-context and enabling inference from the native data relations or external data sources, such as status reports or supply chain data.

Text analysis techniques are the most straightforward way to address the KBC tasks. In the past, the text-based KBC process was composed of a group of basic text analysis methods [57]. First, **Rules and Queries** are applied to the logical framework to check facts from external sources, and then **linguistic Patterns** define the text recognition techniques, e.g., regular expressions or ontology population, finally

Learning and Reasoning, exploits all the techniques oriented to some ML/IA methods. This procedure normally is in some way generalized in most cases, but given the nature of current data found on the Web and the complexity of newer requirements, the previously mentioned procedures have increased complexity.

A starting point for KBC is Commonsense Knowledge (CSK). CKS refers to the facts that humans consider as general knowledge, e.g.,*Human are mammals*; classify and get inside into a more specific CSK arriving into the discovery of new statements can be performed using advance semantics, such as the case of ASCENT[58], which propose a three-step architecture: (1) Retrieval, (2) Extraction, (3) Consolidation for fact construction, using an open information extraction and assertion models enabling the possibility to extract multiple key-value statements pairs from the previous assertions.

In current times, and giving the impetus to share and democratize knowledge through the Web [55], philosophies like Linked Open Data and FAIR principles, described in Sections 2.2.1 and 2.2.2 respectively, have played an essential role in the KG development. These proposals not only increased the number of sources like the Open KG (see Section 2.5) but also catapulted the KBC process by providing enormous leverage to cross-reference and improve the accuracy of the generated KB.

Using the OpenKG to compare detected terms based on a previously existing ontology is an ordinary use case for these platforms, yet inferring on the ontology is called ontology reasoning. OWL-based reasoning and a Linked Hypernyms Dataset (LHD)[59] Framework propose a process that extracts lexico-syntactic patterns over the text sentences, and in combination with a Support Vector Machines (SVM) algorithm, produces types, then, this is string-matched with DBpedia and integrated using a Statistical Type Inference (STI) algorithm to create a resulting RDF KB. Furthermore, using the unique reference inside a KB that RDF schemas provide, the Open Information Extraction (OpenIE) provides small information units that simplify the future query of the data in external databases, e.g., DBpedia [60].

Open KG, like DBpedia [61], are typically extensive, handling extracted data from SPARQL queries or any other request, consuming considerable processing time and resources. DBpedia On Demand [62] provides a lightweight version of the known KB using semi-structured

fields of the DBpedia articles on the user request, real-time delivering updated information by analyzing infoboxes (ontology structured knowledge) using pre-trained mappings to interpret the incoming data.

2.4.2 Knowledge Base Maintenance

Once an initial KB has been deployed, i.e., Knowledge Completion can be accomplished, leveraging on the open-world assumption (OWA) using masking models [41]. In particular, this masking model uses text features with the structure $T = (S, P, ?)$ to learn entity and relation embeddings by high dimensional semantic representations based on TransE [63]. This approach also can be reused as a **Link Prediction** technique, e.g., DistMult, ComplEX, among others, implementing inference on the form $T = (S, ?, O)$.

The KGE provides a broad solution to many of the general knowledge management challenges introduced at this section's beginning; many are in the KB deployment phase. TransE, being one of the most studied KGE models, explores a range of different solutions, such as the case of knowledge validation. On the other hand, many of the explored KBC methods start by creating truth statements to reflect real-world reality. TorusE is a TransE-based embedding method, this method combined with the use of KG Lie Groups (KGLG), is used[64] to validate an inferred KB; the method was tested using the de-facto standards datasets for evaluating the multi-dimensional distance between the calculated embeddings spaces.

Pre-trained models such as GPT-2/3 or BERT, which have been a significant advancement in the Artificial Intelligence field, can also be used to enhance and produce a highly dense KB, such as the case of MAMA [65], that exploits these models along to Wikidata [66] to create additional KB.

2.5 Open Knowledge Graphs

The so-called Web of Data [33] gives rise to several Open Knowledge Graphs (OpenKBs). The OpenKB are RDF triple stores that describe real-world knowledge. They are published to be queried by users or systems by a SPARQL interface responding to one or many ontologies

Table 2.1: Number of OpenKG Statements

Open KG Name	Number	Year
DBpedia Community	1.89 billion	2020
WikiData	1.1 billion	2020
YAGO	2 billion	2020
WordNet	207+ thousand	2011
GoogleKG	Over 100 billion (claimed)	2020
OpenLOD Cloud	Over 200 billion (estimated)	2020

depending on the purpose of the database, reaching millions of statements as shown in Table 2.1[7]. OpenKG can be considered a general domain of domain-specific [29, 67]. The most important OpenKG is integrated into the OpenLOD Cloud, thanks to the Open Linked Data effort described in Section 2.2.1.

The general domain or open domain KG provides multi-lingual facts integrating ontologies over a wide number of knowledge disciplines, serving purposes such as collecting general knowledge facts like $St = (Paris, capitalOf, France)$ becoming nodes with high centrality in the OpenLOD Cloud. These KG were built from initiatives like Google Knowledge Graph [51], evolving to what is now the most valuable KG on the Web, DBpedia[61]. Similar cases like the initial WikiNet [68] and later WikiData [66] have the foundation in the factual knowledge extraction from the Wikipedia project. Moreover, cases like YAGO [69], which combines the non-human-readable entities from WikiData, with the standard *schema.org* ontology to provide a person and machine-understandable facts is also considered the simplified version of Wikidata.

Among the over 1500 domain-specific OpenKGs existing in OpenLOD Cloud, there are exceptional cases like WordNet[8]. This network provides detailed semantic information about the words arranged in objects called synsets. It is typically used as a grammar dictionary, helping to improve all kinds of analysis or studies about, e.g., speech analysis or any other textual use case. On the other hand, Semantic

[7]Statement stats - https://www.dbpedia.org/resources/knowledge-graphs/
[8]WordNet Official site - https://wordnet.princeton.edu

Scholar [70] was designed as a search engine to provide a one-sentence overview of research papers; nowadays, it also provides API access to explore research knowledge across domains. Other similar approaches are the case of DBLP [70] for the computer science domain.

2.6 Technologies

The Knowledge Graph definition is delivered in Section 2.1, and subsequently, formalized in Section 2.3. The need to integrate various data sources for the KB creation is noted in these sections, with the scope of consistently and coherently combining them into an integrated and stable representation.

In order to provide a standard platform to represent, transfer, share and reuse knowledge in the world wide web[18]; The semantic stack provides a set of specifications built on top of concepts and abstractions[33]. On the other hand, more contemporary approaches and technologies, like Machine learning, have given the embeddings the possibility to map the statements described using the SWT into vector representations[71] and leveraging knowledge by exploiting specific loss functions $\mathcal{L}$, according with the KB characteristics, some example of KGE loss functions are shown in Table 2.3.

The combination of these two technologies holds great potential for advancing the state-of-the-art in relevant areas such as natural language processing, recommendation systems, and intelligent search.

2.6.1 Semantic Web Technologies

The Semantic web Stack, or ”cake”, is a layered architecture, presented in [33], which describes a set of high-level concepts, following the former early vision for the web from Berners-Lee. An enhanced version of the well-known architecture is taken from [3] and presented in Figure 2.3; the isometric view allows us to divide the concepts from the actual specifications that it supports. Moreover, a parallel with the Linked-Data concept also is provided[18], giving a complete of the stack and the current implications in his adoptions.

The semantic cake present in the bottom layer, the current standard web platform as we know it, is supported by large specification low-level

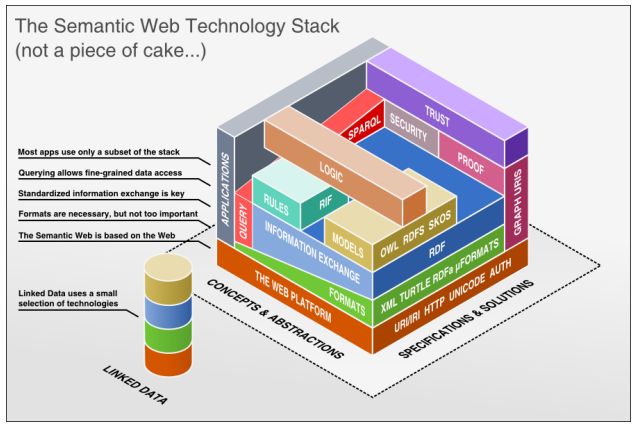


Figure 2.3: Semantic web Stack from an isometric perspective, parallelizing the concepts of linked data, the concept at high-level and the specifications supporting the framework, taken from [3]

protocols such as HTTP, Unicode, Auth, FTP, URI, and many others. Once a transport layer is set, the following step needs a blueprint to frame the represented knowledge; in this layer, it can be found file schemas like XML or knowledge description languages N-triples to define the statements in triple statements, e.g., *¡Bob¿ ¡is a¿ ¡Person¿*, define the statement of an entity *Bob* is connected with the object *Person*, by the relationship *is a* in an informal way, or Turtle, as a DL where the statements can be combined enhancing on top a more digestible user view. In the third layer, a form of information exchange is given, presenting RDF, this framework, for example, grants vocabularies for describing the relations in the KB, like, e.g., *rdf:type*, *rdf:Property*, among others.

The final two layers present RDFS as a more structured and comprehensive version of RDF, and OWL as a technique to define custom schemes; this enables the building of domain-specific ontologies. By granting the knowledge specialization by the ontologies, is possible to create rules using, e.g., First-Order Logic, that can interconnect, complete, or even in some cases, infer new knowledge using semantic reasoners on top it this ontology.

In Algorithm 1, a complete example of 1,2,3,4 layers is given; by integrating two well-known Open KBs such as WikiData and DBpedia.

Algorithm 1: Example of a Wikidata entity definition using Turtle language with five relations, using DBpedia.

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>

wd:Q12418
    dcterms:title "Mona Lisa" ;
    rdfs:label "Mona_Lisa" ;
    rdf:type dbo:Person ;
    dbp:museum dbr:Louvre ;
    rdfs:label "Mona_Lisa" ;
    dcterms:creator dbr:Leonardo_da_Vinci .
```

Layer 1 holds the communication between the endpoints by enabling the URI, and URL definition, i.e., Client App, Dbpedia and WikiData; Turttle enhances the data readability by providing a simple grouping-based notation. RDF, provides a clear and standard definition of types and namespaces definition for the complete URI, and finally, RDFS and OWL grant an extra characterization of an entity by allowing to define domain-specific properties, e.g., *dbp:museum* or standard properties, e.g., *rdfs:label*.

2.6.2 Knowledge Graph Embedding

Knowledge graph embeddings (KGEs) are supervised learning models that learn vector representations of labeled, directed multigraph nodes and edges. The KGE model provides a mechanism to perform operations with a KB, such as fact-checking, question-answering, link prediction and entity linking. The most used KGE models are explored in Table 2.3 along with some of the main features and non-functional properties.

They learn dimensional representations of entities and relations to predict missing facts. Roughly speaking, given an incomplete knowledge base, one of the possible tasks is to predict unknown links. KGE

models achieve this through a scoring function ϕ that assigns a score $s = \phi(h, r, t) \in \mathbb{R}$, which indicates whether a triple is true, intending to be able to score all missing triples correctly. The score function $\phi(h, r, t)$ measures the salience of a candidate triple (h, r, t). The goal of optimization is usually to assign a higher score to the true triples (h, r, t) than to corrupted false triples (h', r, t) or (h, r, t'). Let us remark that a triple one between the head and the tail can be corrupted (denoted by superscript).

Table 2.2: Main Characteristics of most used Knowledge Graphs Embeddings

Model	Main Publication	Type	Complexity
TransE	Bordes2013[72]	Translational	Low
DistMult	Yang2015 [73]	Bilinear	Medium
ComplEx	Trouillon2016[74]	Bilinear	High
HolE	Nickel2016[75]	Non-Bilinear	High
ConvE	Dettmers2018[76]	Convolutional	High

The KGE is commonly used for link prediction; the task focuses on the missing part of a triple against a specific KB that was trained. In the dimensional embedding space, these KGE models are denoted by various score functions [77] that quantify the distance between two entities through the relation type, as shown in Figure 2.4. The KGE models are trained using these score functions, some score/loss functions are denoted in Table 2.3, so those entities connected by relations are close to one another, and entities without connections are far away.

In more formal terms, the model learns vector embeddings of the entities and relationships for each training set S of triples (h, r, t) composed of two entities $h, tinE$ (the set of entities) and a relationship $rinL$. The embeddings take values in $\mathbb{R}^N$, where N is the space dimensionality.

For instance, TransE considers the translation of the vector representations, i.e., the head entity embedding should be close to the tail entity embedding, plus relation embedding when the head entity is similar to the tail entity (when (h, r, t) holds). Otherwise, the head entity should be far away from the tail entity. According to the energy-based framework proposed in [78], the energy of a triple is equal to

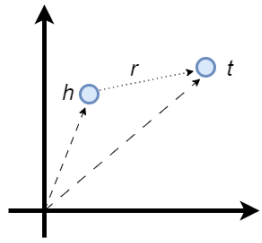


Figure 2.4: Embedding space representation of a single triple (h, r, t).

Table 2.3: Loss function details of the most used Knowledge Graphs Embeddings and the family type

Model	Scoring Function	Family
TransE	$-\|e_s + r_p - e_o\|_n$	Geometric
DistMult	$\{r_p, e_s, e_o\}$	Matrix Factorization
ComplEx	$Re(\sum_{k=1}^{K} W_l, e_h, \overline{e_t})$	Matrix Factorization
HolE	$\frac{2*Re(\sum_{k=1}^{K} W_l, e_h, \overline{e_t})}{n}$	Matrix Factorization
ConvE	$\sigma(vec(g([\overline{e_s}; \overline{r_p}] * \Omega))W)e_o$	Deep Learning

$d(h + l, t)$ for some dissimilarity measure d, which can be either the L1 or the L2-norm.

To learn such embeddings, the model minimizes a margin-based ranking criterion over the training set described in the following equation:

$$\sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'_{h,r,t}} [\gamma + d(h + r, t) - d(h' + r, t')]_+ \qquad (2.1)$$

Where:

- $[x]_+$: denotes the positive part of x;
- $\gamma > 0$: is a margin hyper-parameter;
- $S'_{h,r,t} = \{(h', r, t') \mid h' \in E\} \bigcup \{(h, r, t') \mid t' \in E\}$

The set of corrupted triplets $S'_{h,r,t}$ is composed of training triplets with either the head or tail replaced by a random entity (but not

both simultaneously). The loss function (2) favors lower energy values for training triples than for corrupted triples and is, thus, a natural implementation of the intended criterion.

KGE provides a comprehensive framework to perform operations [42] over a KB, usually provided in the KB deployment phase. for example, TransE [72], being one of the most studied KGE models, given by the low dimensionality in the results and the model explainable capabilities, $fTransE = -\|e_s + r_p - e_o\|_n$, computes the similarity between the $e_{subject}$ translated by the embedding of the predicate e_p and the object e_o.

2.6.3 NeuroSymbolic AI Architecture Patterns

When the term neural-symbolic computation was introduced [15], the goal was to propose a standard unification to describe logic and network models; unfortunately, with the rise of modern deep neural networks both in literature and in real-world applications, there is no standardization in how the inductive and rational models connect between one and another. For this reason, in [16], a notation to report the findings in the hybrid-learning approaches was proposed, and from these initial studies, 15 patterns were depicted, of which 11 denote a neuro-symbolic interchange in the learning procedure.

This notation was structured by the abstraction of rational tasks, defined by *KR (Knowledge Representation)* and inductive tasks, defined by "*ML (Machine Learning)*"; these tasks normally perform a transformation on their respective I/O, such as *sym* stands for the symbolic or semantic data and *data* for the plain or regular data. These four components (*KR, ML, sym, data*) create a baseline for the NeuroSymbolic AI patterns later exploited by [1] where from 11 patterns, the count was extended over 40, as shown in Figure 1.1.

The usage of standardized patterns provides a roadmap for implementing NeuroSymbolic AI systems integrating annotated and raw data providing a set of advantages for both types of interactions and delivering more complete results. In the case of the transition from inductive (*ML*) to rational (*KR*), the module allows the learning, improvement and application of different rule sets previously not discovered; on the other hand, from the rational to the inductive algorithm, the first module contributes to the enrichment of the sample data delivering more

data points for the model to discover more information and enriching the model improving downstream task.

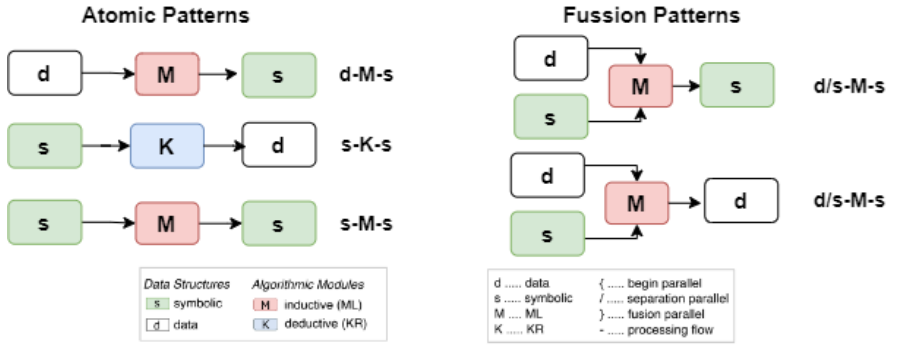


Figure 2.5: Graphical and textual representation of NeuroSymbolic AI patterns using the Boxology notation, adapted from [1]

These patterns can be categorized from simple algorithmic modules like the *Atomic* and *Fusion* to a more complex chain of modules such as *I, Y and T* types. Each one of these categories is denoted by the ingestion of data from one task to the other, providing a system overview for specific purposes. In order to represent in a seamless way the description of the patterns, the authors of these last extensions have decided to design a notation to describe the interaction between the main components *KR, ML, sym, data*, the details of the text-based notation is described in Figure 2.5, showing an example the Atomic and Fusion patterns.

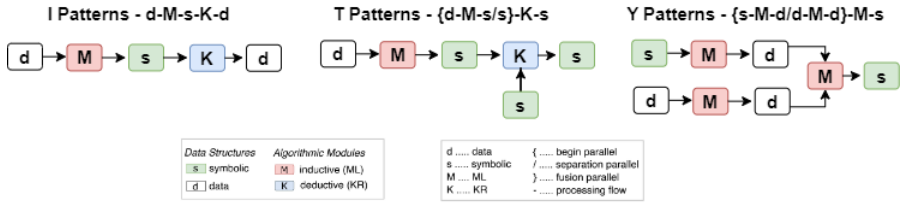


Figure 2.6: Graphical and textual representation of complex NeuroSymbolic AI patterns using the Boxology notation, adapted from [1]

- **Atomic Patterns:** These patterns are denoted by simple tasks such as *KR* or *ML* where a single input is given to one of the previous modules e.g. d-M-s, s-K-d, s-M-s, etc.
- **Fusion-type patterns:** Determined as the more complex single module systems where two types of input are fed to the task e.g. d/s-M-d, s/d-M-s, etc.
- **I-type patterns:** Normally, these patterns are the concatenation of different *Atomic Patterns* in pipeline manner e.g. d-M-s-K-d. where the patterns d-M-s and s-K-d are arranged, one after the another
- **T-type patterns:** Normally is formed by the combination of one *I-type Patterns* and a *Fusion Patterns* .e.g. {d-M-s/s}-K-s in this particular case the I-Pattern d-M-s-K-s and the Fusion Pattern s/s-K-s.
- **Y-type patterns:** Relies in the combination of one (or more) *Atomic Patterns* with *Fussion Pattern* e.g. {s-M-d/d-M-d}-M-s is formed by the combination of s-M-d and d-M-d Atomic Patterns and d/d-M-s Fussion pattern.

2.7 Applications

Given the current trend of data sharing explosion, brought by Industry 4.0 and Open Goverment [79], Knowledge-based applications are in constant growth. More in-depth, the KG application ecosystem can be dismantled [80] into several use cases giving various scenarios based on the possible use of the knowledge and its characteristics. These use cases are Query Answering oriented to chat and bot systems, Recommendation Systems, Semantic Searching by providing SPARQL endpoints over a fixed KB schema, Traditional Knowledge Sharing, Dashboard, and visualization support, or any other domain-specific applied to any particular industry, like Health, Government, Manufacturing, etc. However, all the use cases can not work with the support of vast KBs with quality standards, open and available, across generalized and specific fields for the broader utilization of diverse systems.

Some use cases were selected within the broad spectrum of KG applications, providing valuable experiences around the most significant problems in the base construction, e.g., Triple Construction, Knowledge

Management, Ontology Matching, and population, among others.

BabelNet [81] is a huge KB with a comprehensive lexicographic and encyclopedic coverage of terms, enriched by a semantic graph (ontology) that links concepts and named entities (synsets) in a vast network of semantic relations. The KB was built by combining WordNet and the multilingual versions to integrate them with WikiData statements through an integrated interface. Similarly, Babelpic [82] creates a KB of semantic annotated high-quality images extracted from popular repositories. Additionally, Babelfy [83], acting as a KGR, provides a disambiguation service using the BabelNet KB.

Considering a case study from the journalism domain, the ontology modeling and population approach described in [84] creates a KB starting from an existing ontology and, with the combination of NLP and ML techniques, provides an internal organization service for facts collaboration in a real Norwegian newsroom.
In the smart city domain, graph schemas have been combined to integrate sensor networks with official public data as an integrated KB to create a context-aware system [85]. Besides the statements extraction from external data sources, e.g., sensors, social media, existing databases, and knowledge, there is the need to validate an existing KB, the OpenKG (Section 2.5), such as DBpedia become a valuable tool like the case of HAPE[86], additionally using big data technologies to ensure the storage and redundancy aspects of the data management.

Fake news detection brings the associated challenge of Social Media and Content Delivery Networks (CDN) integration[87], considering the social source of the fake news: an approach to categorize the news into a subgraph, for a question-answer domain is presented [88]. Some further methods explore the semantic analysis of the statements and rules of the tourism industry [89], leading to the question-answer scope.

Knowledge or Statement Integration constitutes a part of the KBC, allowing external knowledge into an existing KB. For example, besides the enrichment challenge, adding semantic knowledge to the already existing statement is shown in [90] from the Humanities field: a link prediction method using KGE was implemented for the historical record linkage process.

While the KBC is taking place, it is necessary to supervise the KG life cycle by moderating the inconsistent and mismatched statements, a product of the initial discovery process. Therefore, approaches like

[91] enable user-friendly applications for human intervention in the reviewing process of mismatched statements.

In the bioinformatics domain, creating subgraphs that give the genomic information of proteins can lead to discovering new relations that interact with larger KBs [92], annotating protein data and the innate multi-relation of these, ending in the development of new pharmaceutical solutions.

Chapter 3

Knowledge Graphs Construction with Traditional Semantic

For over a decade, Semantic Web Technologies (SWT) have provided a reliable data representation and integration framework. Semantics represents an interoperability backbone to get homogeneous data representation across single or multiple domains, query integrated data, and solve complex problems, leveraging data taxonomies to predict information. This chapter focuses on the knowledge graph study from the perspective of the traditional Semantic Web. It focuses on exploiting neuro-symbolic AI system patterns to perform schema-based knowledge-base construction approaches, specifically on event recognition tasks.

The introduction details an initial scenario and the research gap that, at the same time, is justified by framing and motivating the research path across the thesis contributions in this chapter. In the next sections, a SOTA revision was performed, and then the used and implemented baseline methods were described. From Section 3.5 to Section 3.7, the contribution results are presented. The publications are listed in Section 3.1.1, and Table 3.1 gives a bibliographic detail of journals and conferences.

3.1 Introduction

Nowadays, SWT not only continues to provide a KR framework but has become a crucial method to give meaning to data across different domains; this enables cognitive-based tasks by integrating machine learning and automated reasoning with semantic annotation. This new approach, which uses SWT in actual systems architecture and data analysis processes, is called Neuro-Symbolic integration [15].

A neuro-symbolic system consists of an ensemble representation of a particular domain, e.g., Plants and Medicinal Herbs; this can be at the same time combined with another representation network, e.g., the state-based of a specialized agent, that is able to interact and gaining new features with the support of the previously represented characteristics [15].

Recent trends in AI, like neural networks, have required the adoption of new ways in which the Neuro-Symbolic integration needs to be performed; as stated in [1] in recent years, the NeuroSymbolic integration with existing AI downstream tasks has become a powerful combination in many stages, from knowledge representation, data augmentation for enhancing downstream task, to domain interoperability, etc. In Fig. 3.1, the dotted lines show the contribution scopes of this thesis chapter of the ecosystem intersection in the NeuroSymbolic IA or SWeMLS[1].

Normally, NeuroSymbolic AI systems are composed of atomic tasks; the tasks allow for performing single-role jobs and executing hybrid learning strategies. Design patterns have emerged from the hybrid learning perspective; they describe data interactions such as learning from symbolic sources and explainable systems [16]; in this initial study, 15 design patterns were described for hybrid learning tasks and complemented with 33 additional ones described in [1].

The NeuroSymbolic approach proposed in this chapter is motivated by patterns No. 5 and 6, defined in[16], where pattern No 5. describe learning systems from symbolic sources, i.e., ontologies; and pattern No.6, built upon pattern No. 5, introduces the concept of extending machine learning (ML) procedures with traditional reasoning.

This chapter presents a NeuroSymbolic-AI modification of pattern 6 [16, 17], shown in Fig. 3.2 and adopted in the proposed framework. The adapted pattern is represented as *(d-s-K-M-K-d)*, according to [1] and is shown in the bottom part in Fig. 3.2; it is composed of a

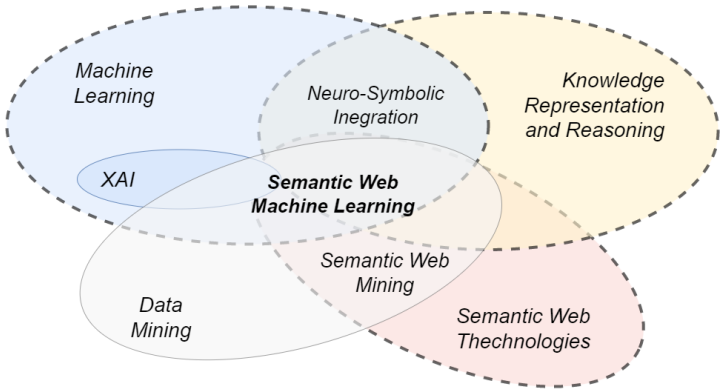


Figure 3.1: SWeMLS landscape, taken from [1], and adapted to focus on the contribution scope of the chapter of Knowledge Graph Construction with Traditional Semantic

data annotation architecture (d-s-K) for model enrichment and then, using the inductively calculated data, a reasoning approach based on a pre-defined schema (M-K-d). The AI task, named *M*, in the figure, is covered by a traditional neural network able to classify events from a specific knowledge base and append the results to the existing KB. The NeuroSymbolic layer is divided into two parts. Initially, the signal data is captured (1) and annotated by an expert-defined ontology, creating an initial and contextual knowledge base of the sensor network. Then a real-time annotation schema captures actual data from the sensor to the KB to serve the annotated event data as input to the previously trained IA task; once the AI task returns the initial classification results to the KB, (2) the KB takes the event detection and runs a logic query to output the event detection.

An IoT ecosystem consists of web-enabled smart devices and sensors that interact and operate under a common format to collect, send, and act on data acquired from the environment. The synergy between IoT and Semantic Web Technologies promises to overcome the barrier of standardized heterogeneity and provide seamless solutions in remote sensing applications by taking benefits in knowledge representation, expression, and discovery.

Semantic modeling of the seismic domain can describe a wide range

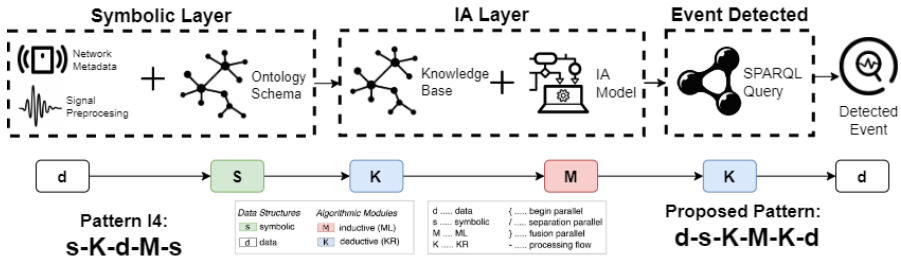


Figure 3.2: NeuroSymbolic IA fusion pattern approach for event detection

of sensing data from heterogeneous sources; in addition to the traditional seismic network sensors, low-cost wearable sensors are available with the advent of the IoT era. In light of these ongoing perspectives, this chapter presents an IoT-oriented framework to support signal acquisition. The proposed framework annotates, processes, and classifies/recognizes key events, resulting in ontology-based storage of the collected data to retrieve aggregated, time-based information to support humans in analysis and decision-making scenarios.

A thorough state-of-the-art review has been conducted in Section 3.2, which focuses on semantic integration and interoperability in IoT environments, integration of SWT with current IoT technology, through deep learning models for event recognition. The SOTA review reveals a lack of studies on sensor networks enhanced with NeuroSymbolic AI capabilities to improve event recognition in the IoT domain.

Standard ontologies for sensing scenarios like SSN and SOSA were examined; this study is described in Section 3.4.1; by early 2020, a new revision of SOSA was released, where a seismic sensing description highlight[1] is provided. The sensing modelling defined by the W3C provides a technical baseline for building a SOSA extension where the focus is the Event Recognition paradigm.

The massive nature of incoming data points from the sensor network, enhanced on top by the neuro-symbolic approach, required to use of a deep learning experimentation framework to search, fine-tune, train, and validate models based on the generated semantically annotated data. The experimentation framework and the validation metrics employed

[1]SOSA Example - http://www.w3.org/TR/vocab-ssn/#seismograph

are described in Section 3.4.2.

To model the domain of interest, leveraging on SOSA/SSN Ontology, a formal representation of involved data, captured by the sensors transforming it into a related event, was given in Section 3.3.

Once the initial baselines of Ontologies for Sensor Networks and NeuroSymbolicAI are defined and the formal definition of NSAI design patterns is in place, the main contribution of this chapter can be described. A comprehensive Knowledge Engineering procedure was developed to design the proposed ontology, detailed in Section 3.5, including peer-review evaluation, domain-expert consultation, and two rounds of revisions by one of the SOSA coauthors. The proposed ontology, VEO, provides a homogeneous data representation of different domains (seismic, sensing, geographical, etc.) by integrating heterogeneous data (IoT, seismologic, SAR, GPS, telemetric data, etc.) and ensuring syntactic and semantic interoperability. Behind the homogeneous sensing data representation, the knowledge base represents a common access point for enhanced data retrieval. The semantic layer allows for leveraging quality event data to exploit traditional AI approaches.

The ontology-based KB is processed by two Deep Learning methods, namely, MultiLayer Perceptron and Convolutional Neural Networks, described in Section 3.6, that are trained for event detection.

The NeuroSymbolic IA approach, described in 3.7, is designed for two domains. The first is the seismic domain. An initial version of the ontology and the two ANN-based validation procedures were described and used on a small dataset for a simple event recognition scenario. This preliminary approach was published in [93]; this research was invited to make a more thoughtful contribution to the IEEE IoT Journal[94], where a more curated and consolidated version of the ontology was designed and more extensive experiments were added in different scenarios and a more complex dataset. Section 3.7.1 describes the experimentation scenarios and results. Finally, a further event recognition scenario using the NSIA approach on a controlled dataset aims to detect the noise background through a subset of experiments merely focused on source detection, specifically volcano mountain. The results shown in [95] evidence high accuracy and satisfactory F1-score values. The complete research also is described in Section 3.7.2.

The second approach was carried out in a more practical scen-

ario. Cleanrooms are controlled environments where certain production goods are manufactured. In this case, the NSIA approach was applied to measure and control three environmental variables in three connected rooms, involving legacy-type sensors and offline sensors to detect humidity, temperature, and particle counting thresholds that comply with international standards. A middleware for recollection and annotation was built from scratch to integrate various sensors. Finally, the developed approach was deployed in platform form on Leonardo SpA in the headquarters of Rome. The details of the ontology and the annotation procedures are described in Section 3.7.3; additionally, a contribution was presented in the Industry Track of the ISWC 2021 [96]. Unfortunately, the training and validation results were not possible to publish, given the company data privacy policy.

3.1.1 Contribution

The main objective of this chapter is to explore the Knowledge base construction methods from the traditional semantic web technologies perspective leveraging architectural patterns for Neurosymbolic IA integration, performing integration with known Deep Learning methods to enhance the event recognition task.

To offer a comprehensive overview of the chapter's contributions in a more refined manner, the subsequent list presents a synthesized compilation of all the noteworthy accomplishments within this chapter.

- A new **design pattern** is proposed extending functionality for NeuroSymbolic AI Systems, used for ingestion and reasoning purposes in integration tasks.
- An expert-validated **event detection ontology** schema, extending SOSA/SSN, that can be exploited on sensor networks.
- A **convolutional network approach** to classify semantically annotated sensor events. Additionally, a similar architecture achieves the detection of seismic signals from volcano background noise signals.
- A **NeuroSymbolic AI framework** that can annotate data using VEO ontology and efficiently detect events occurring on a sensor network, demonstrated in at least two different domains.
- An additional case study in an industry scenario using legacy technologies.

3.1.2 Publications List

The publications derived from this chapter work are listed as follows. The journal/conference details are described in Table 3.1.

1. Rincon-Yanez, D., Lauro, E. De, Falanga, M., Senatore, S., & Petrosino, S. (2020). Towards a semantic model for IoT-based seismic event detection and classification. In 2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020 (pp. 189–196). IEEE. 10.1109/SSCI47803.2020.9308329.
2. Rincon-Yanez, D., Crispoldi, F., Onorati, D., Ulpiani, P., Fenza, G., & Senatore, S. (2021). Enabling a Semantic Sensor Knowledge Approach for Quality Control Support in Cleanrooms. In International Semantic Web Conference (ISWC) (Vol. 2980).
3. Falanga, M., De Lauro, E., Petrosino, S., Rincon-Yanez, D., & Senatore, S. (2022). Semantically Enhanced IoT-Oriented Seismic Event Detection: An Application to Colima and Vesuvius Volcanoes. IEEE Internet of Things Journal, 9(12), 9789 – 9803. 10.1109/JIOT.2022.3148786
4. Rincon-Yanez, D., De Lauro, E., Petrosino, S., Senatore, S., & Falanga, M. (2022). Identifying the Fingerprint of a Volcano in the Background Seismic Noise from Machine Learning-Based Approach. Applied Sciences, 12(14), 6835. 10.3390/app12146835.

Table 3.1: Impacted (C)onferences and (J)ournals, ordered by publication Date. Cite Score (CS), Impact Factor (IF). Conference)

Pub.	Name	Type	Rank	CS	IF
1	IEEE SSCI	B	B	–	–
2	ISWC	C	A+	–	–
3	IEEE IoTJ	J	Q1	14.9	10.2
4	Applied Sciences	J	Q2	3.7	2.8

3.2 Related Work

Semantic Web technologies enable data interoperability and integration by transforming raw sensor data into high-level machine-understandable

knowledge; they also provide reasoning capabilities that allow the domain of interest to be enriched with additional inferred information [97]. Semantics provide a basic infrastructure for data representation, ensuring shared agreement on the types of features, sources and data. Alongside these technologies aimed at structuring and managing natural events and ecological activities, the role of ML is crucial in defining accurate models for data analysis and classification activities.

The spread of IoT technologies and the role of the Semantic Web in modeling sensor networks have highlighted new opportunities in the collection and integration of sensing data from different tools and heterogeneous sources. Especially in Earth Observations, sensor networks are low-cost solutions for monitoring land activities and mountain surveillance [98]. The role of Semantic Web technologies in the sensor network domain provides many benefits to the data stream processing: for instance, the use of JSON-LD (JavaScript Object Notation for Linked Data, a method of encoding linked data based on JSON format) represents an efficient way to semantically annotate data and, at the same time, collect an ontology-based description of the domain of interest to speed up the data filtering solutions.

The remainder of the section provides an overview of the main works involving the research domains, such as Semantic Technologies and Machine Learning with reference to the Sensing Technologies domain. Their synergy provides interesting solutions to face natural phenomena, specifically in the geological and seismological fields.

3.2.1 Ontologies and sensing technologies from the seismological viewpoint

With the IoT technological explosion and the need to extract data easily to interpret and process them, semantic technologies represent an important framework, a step forward for collecting and processing semantically annotated data.

The IoT technologies suffer from some drawbacks [99]: heterogeneity of physical objects, both regarding physical features and data and/or operations that they provide and/or accomplish; incomplete or missing metadata to describe things and their features. Moreover, sensing technologies are heterogeneous, coming from different sources generating data with incompatible formats.

These problems find a possible solution in adding a semantic layer for annotating sensing devices: ontologies have provided a seamless framework for formally describing the domain of interest.

Semantic modeling produces an explicit description of the meaning of data in a structured way by merging domain knowledge and context-relevant information with raw measured data. In the IoT domain, ontology modeling provides an explicit description of the things and devices and their features in a structured way by combining domain knowledge and contextual information with raw measured data aimed at syntactic and semantic interoperability [100]. Several research works define high-level descriptions and semantic extensions of *Things* by means of ontology-based annotation. Just for example, GoAAL [101] exploits SSN-based ontologies to support goal-oriented tasks in the Ambient Assisted Living (AAL) environment, specifically in the non-critical services for comfort and home management for elderly persons.

In [102] a smart home ontology is proposed as a way to collect housing data and information about the appliances through the use of a Non-IP mechanism with monitoring purposes. The ontology-based development of IoT solutions is crucial to obtain universal solutions which ensure adequate interoperability, especially in complex and risky environments.

The ontology modeling has been used indeed for describing natural phenomena or hazards [103, 104, 105]. In the seismic domain, volcano ontologies [106, 107, 108] are presented from different perspectives: for example, in [107] the eruptive styles and several dynamic volcanic processes that occur during eruptions are described from a physics viewpoint, whereas in [108], ontology modeling focuses on representing conceptual relationships between data and phenomena associated with volcanic domains, particularly linking plate tectonic features with volcanic systems.

The proposed ontology VEO focuses mainly on event identification by collecting sensing data available in the domain of interest. Like in [99], VEO uses SSN/SOSA ontologies to describe IoT sensors and sensing devices embedded in stations. The idea is to integrate into our knowledge base seismic data collected from traditional sensor-based station networks but also IoT-based sensor devices (for example wearable ones) to get a comprehensive and integrated view of what is happening from the seismic viewpoint.

To support our proposal, recent literature proposes IoT-based systems based on a synergy of technologies such as microelectromechanical systems, wireless communication, and increased computing power to collect seismic data [109]; some advances in mobile accelerometer technology reveal how smartphones [110] are becoming a valid alternative to fixed seismometers as the traditional sensing devices for the earthquake early warning [111].

Other alternatives to traditional networks include a microelectromechanical system, accelerometers installed in buildings, USB attached to personal computers, or other low-cost sensory equipment [112, 113]. Ontology-based technologies are used in the literature as well to model the seismic domain too, building information systems on seismology [114], also focusing on the earthquake emergency evaluation and response domain [103, 115].

3.2.2 Machine Learning in Volcano Seismology

In volcano seismology, several techniques of ML have been adopted such as those based on Artificial Neural Networks (ANN) and Multi-Layer Perceptrons (MLP) [116, 93], Cluster Analysis (CA) [117], Self-Organizing Maps (SOM) [118], Support Vector Machines (SVM) [119], Convolutional Neural Networks (CNN) [120], Recurrent Neural Networks (RNN) [121], Hidden Markov Models (HMM) [122].

More explicitly, with respect to Italian volcanic areas, the seismic signals were classified by using the classic ANN/MLP approaches, applied to Vesuvius [123], Stromboli [124], and Etna [125] mountains. Indeed, Scarpetta et al. (2005) [123] introduced an automatic classification of local seismic signals and Volcano-Tectonic earthquakes at Vesuvius based on MLP trained with a single hidden layer by using the quasi-Newton algorithm and the scaled conjugate gradient descent. Falsaperla et al. (1996) [124], by adopting the MLP approach combined with the Back Error Propagation algorithm, concentrated the attention on the classification of the explosion quakes at the Stromboli volcano, identifying four different classes of shocks. Furthermore, Langer et al. (2009) [125] investigated the seismic tremor at Etna volcano by comparing the results achieved by applying both supervised (SVM/MLP) and unsupervised (CA/SOM) techniques. CA algorithms are unsupervised networks very useful for looking for similar shapes, independently of

the sizes of the seismic events such as Long-Period signals that are generally clustered into families (see the case of Soufrière Hills Volcano, Montserrat [126]). In Ruano et al., 2014 [127], the SVM method was used to detect seismic events by a distributed sensor network in the framework of earthquake prediction in Portugal.

On the other hand, RNN and CNN (or ConvNets), which are a type of regularized MLP, are well-established methods in processing signals; specifically 1D CNN [128]. Similar approaches were presented in [121] to classify different types of seismic events of the Colima volcano (Mexico). HMM classifiers have been adopted to classify the seismic events [122] in Colima and the Popocatepetl volcano (also located in Mexico). Recently, Deep Neural Network (DNN) with various architectures has been applied for automatic seismic event classification and detection. In this framework, it is noteworthy to cite the studies of Titos et al., 2018 [129] conducted at Colima to classify volcano-seismic events.

3.3 Events and Waves a formal definition

The goal of the proposed ontology; is to capture the event description in combination with the environment network semantic data, of the environmental network, and the sensor types, etc. Given this general objective is necessary to focus on how the raw data coming from the station can be arranged and processed. For this purpose, it is necessary to describe the wave with the natural characteristics embedding some of the features from the environment as well as the proper sensor characteristics. In this section, the main ontology classes defined in VEO, namely, *veo:DataPoint*, *veo:DataPointArray*, and *veo:DataPoint* described from a formal perspective.

3.3.1 veo:DataPoint

Building from the smallest point, the minimum source of data relies on any observation (x) captured by a sensor (S) at any time (t). The *veo:DataPoint* represents a measurement from a seismic trace captured by a sensor over time. More formally:

Definition 4 *Considered a sensor device s, a* ***DataPoint*** DP_t^s *at time t, can be described as follows:*

$$DP_t^s = \{x_1(A,t); x_2(A,t); x_3(A,t)\}, \tag{3.1}$$

Where each component $x_i(A,t)$ with $i = 1, \ldots, 3$, describes the amplitude (or, in other words, the displacement speed of ground) of the measured wave A at the time t when the sample is taken by the sensor device s.

Notice that, depending on the sensor s, DP_t^s can contain one or three components of a seismic wave (along V, NS, and EW directions of motion), which can be one or three dimensions, respectively.

3.3.2 veo:DataPointArray

The collection of *veo:DataPoint* compounds an array, as described as follows.

Definition 5 *A* ***DataPointArray*** $DPA_{\Delta t}^s$ *is a sequence of* DataPoints (DP_t^s)*, gathered in a time interval $\Delta t = (t_j - t_i)$ by a given sensor s.*

$$DPA_{\Delta t}^s = [DP_{t_i}^s, \ldots DP_{t_j}^s]. \tag{3.2}$$

*The time interval is undefined when considering the whole sensor lifespan, i.e., from t_i where $i = 0$ (when the sensor is turned on) to $t_j =$ where $j =$*tMAX *(the sensor device s until it is powered-off).*

3.3.3 veo:DataPointMatrix

However, once defined a sequence of the data points it is possible to detect a sub-sequence of data points describing a seismic event or a generic event E.

Definition 6 *Let us consider a sensing network S composed of a set of sensor devices $S = \{s_1, s_2, \ldots, s_n\}$; an* ***Event*** $E_{\Delta t}^S$ *acquired by any sensor $s \in S$ in the time interval Δt is described as follows:*

$$E_{\Delta t}^S = (DPA_{\Delta t}^{s_1}, DPA_{\Delta t}^{s_2}, \ldots, DPA_{\Delta t}^{s_n}) \tag{3.3}$$

The collection $(DPA_{\Delta t}^{s_1}, DPA_{\Delta t}^{s_2}, \ldots, DPA_{\Delta t}^{s_n})$ represents ***DataPointMatrix****, i.e., a collection of DataPointArrays gathered by each sensor*

of the seismic network. In particular, the DataPointMatrix can be classified as an event $E^S_{\Delta t}$ among those can be: BLs, VTs, and LPs.

In other words, each sensor from the sensor network collects one or more DataPoints in a specific time interval. The collection of DataPointsArray describes an event.

3.4 Baseline Methods

3.4.1 SSN & SOSA Ontologies

SSN [130] is a well-known ontology model for describing sensors and their observations, the observed properties, domain interest features, methods and deployments used for sensing [131]. It is defined by foundational Dolce UltraLight (DUL) ontology, whose model platforms are physical objects [132]. At the same time, SSN has been used to extend into an additional, and self-contained ontology called SOSA (Sensor, Observation, Sample, and Actuator) designed as a core vocabulary for supporting domain applications and reuse-by-extension in the sensing platform environment [133]. The main reasons for using SNN/SOSA rely on the high level of flexibility and the level of standardization of that ontology. The role of SOSA/SSN concepts is crucial in the Knowledge Representation process. Some proposed classes extend SOSA ontology introducing additional characteristics. The SOSA/SSN concepts (or classes) framed in our ontology model are listed as follows:

- **sosa:Platform**: It is the container that hosts the other entities. In our ontology model, it is specialized as a seismic network.
- **sosa:Sensor**: It is a generic class that can represent a device, an agent (including humans), or software (simulation) for a sensing task. In our ontology model, it is a seismic domain-specific device or station for acquiring features measured by the network.
- **sosa:Observation**: It is the class describing measurement values captured by the sensor about a property related to a Feature of Interest.
- **sosa:ObservableProperty**: Represents an observable property or quality of the investigated phenomena.

- **sosa:FeatureOfInterest**: This class describes the feature whose property is estimated in the course of an observation.
- **sosa:phenomenonTime**: This class is the duration time associated with the observation reading. Depending on the Feature of Interest, it might be a time interval (time:Interval) or an instant *time:Instant*, defined in the OWL Time Ontology [134].
- **sosa:ObservationCollection**: this class has been introduced in an in-progress extension of the SSN ontology [135], that enables representing homogeneous collections of observations.

3.4.2 Neural Network Preliminaries

Besides the adoption of well-consolidated methodologies based on the seismological analysis of the waveforms [136, 137, 138], the characterization of typical events occurring on volcanoes in these last years has been also performed by using the machine learning (ML) techniques, and artificial neural networks (ANN), such as multi-layer perceptrons (MLP) [123, 93], convolutional neural networks (CNN) [120].

Experimentation Framework: Figure 3.3 shows a standard ML training framework with annotation and normalization tasks performed to test our proposed approach. The raw data, i.e., the event signals from the sources, are preprocessed, splitting it into separate time windows lasting 24 s (20 s devoted to the event signal and 4 s due to the overlapping windows). Due to the different types of acquisitors gathered from the stations, the raw signals are resampled at 100 Hz to ensure data normalization across the entire dataset. The samples are separated into two parts: one for model training and validation, and the other (test data) part called *unlabeled data* is the unseen signals that were be classified by the model once trained.

The *labeled data* is validated by cross-validation (stratified k-fold) to guarantee a reliable and robust classification model. Cross-validation, indeed, allows for improving the flexibility and performance of the proposed model. Normally, the method splits the dataset into k subsamples, in which one sample is used for testing, and the remaining $k-1$ dataset is used for training purposes. The whole process is repeated k times by varying the training samples and testing samples. After the k iterations, the best instance of the model (i.e., *validated model*) is

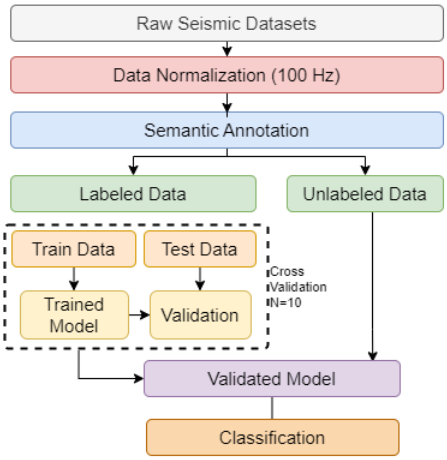


Figure 3.3: Workflow for training and validating a ML model fed by semantically annotated signal data.

selected and used for the classification of *unlabeled data*.

Evaluation Metrics The performance evaluation for NN methods is calculated considering *precision (P)* Eq. 3.4a, *recall (R)* Eq. 3.4b, *F1 score (F1)* Eq. 3.5a, and *accuracy (ACC)* Eq. 3.5b. These measures are based on the evaluation of the following parameters:

- ***tp* (true positive)** is the number of samples correctly assigned to a class.
- ***tn* (true negative)** is the number of samples correctly not assigned to a given class.
- ***fp* (false positive)** is the number of samples wrongly assigned to a class.
- ***fn* (false negative)** is the number of samples wrongly not assigned to a given class.

$$P = \frac{tp}{tp + fp} \qquad (3.4a) \qquad R = \frac{tp}{tp + fn} \qquad (3.4b)$$

More precisely, these metrics, reported in Expressions 3.4 and 3.5; can be understood as precision describing how precise/accurate the model is out of those predicted positives; the recall calculates how many of the actual correctly classified instances are effectively labeled by

the model as true positives. The F1 score is the weighted average of precision and recall, while accuracy is the simplest metric, providing a ratio of correctly classified samples to the total sample dataset.

$$F1 = 2 * \frac{P * R}{P + R} \quad (3.5a) \qquad ACC = \frac{tp + tn}{tp + tn + fp + n} \quad (3.5b)$$

3.5 A Event-Oriented Ontology

The VEO ontology is composed of a total of 20 classes, 6 of which are inherited by SOSA ontology, 17 properties, and 10 types of relations between classes, as shown in Figure 3.4. Our ontology is designed to describe the seismic domain, starting from the raw data acquired by an individual sensor or, more in general, by a sensor network located in a volcanic environment, until to recognize the specific waveform that describes a seismic event. Moreover, the ontology models the outcome of a possible alerting situation, supporting, for example, the authorities to make decisions in the risk scenarios. It is important to mention that this ontology can also be applied to sampling other domains where a sensor network takes low-periodic samples of the environment [96].

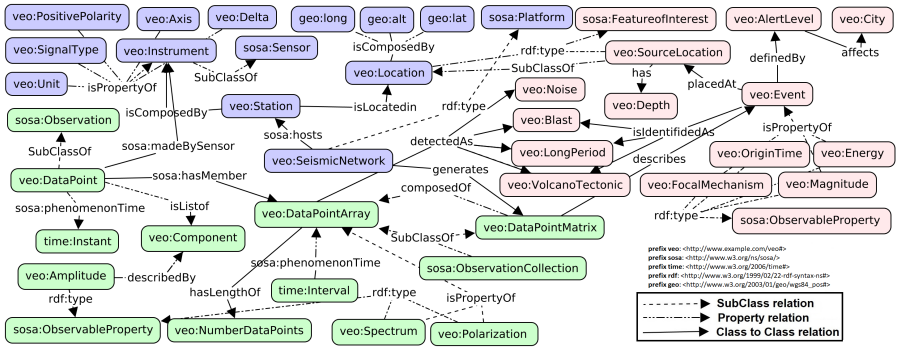


Figure 3.4: Core classes of Volcano Event Ontology divided into Knowledge Models: *Context* in purple, *Collection* in green, and *Knowledge* in pink.

According to the color-based ontology class partitioning, the concepts in VEO ontology are divided into *Context*, *Collection*, and *Know-*

ledge models describing the seismic domain from different perspectives and abstraction levels.

3.5.1 Context Model

It describes the sensing infrastructure related to the domain of interest. The concepts, colored in violet, describe the physical layer of the sensor networks, the sensors, and complementary meta-data associated with the raw data.

- **veo:SeismicNetwork**: it extends the generic *SOSA:Platform* class, a specialized subclass for the seismic domain. It is an ensemble of seismic stations (e.g., TRZ, CPV) located at a certain distance from each other, according to a particular geometry. In a broader geophysical context, *Veo:SeismicNetwork* can be easily included in a general *Geophysical Network* in which all the geophysical/geochemical instruments are taken into account (e.g., tiltmeters, GPS, gravimeters, etc.).
- **veo:Station**: it is a station composed of one or more sensors and equipped with a data logger (for local acquisition, stand-alone instrument) or with transmission apparatus (to send real-time signals to the acquisition center). Geographical coordinates are well-defined for each site where the station is installed and represent a *sosa:FeatureOfInterest*. One station measures only one physical parameter, composed of at least one *veo:Instrument*.
- **veo:Instrument**: it is a specialized class, a subclass of *sosa:Sensor*, extended with additional properties, i.e., *veo:Unit*, represents the measuring unit used by the instrument. *veo:Delta* representing the time sampling in which a sample is taken; *veo:SignalType* describing the signal type (analog or digital, depending on the sensor type and axis), represented by one or three components.

3.5.2 Collection Model

The classes colored in green in the VEO model are designed to support the process of collecting raw data from stations over time and represents the formal aspects defined in Section 3.3. The concepts involved in this model are listed as follows:

- **veo:DataPoint**: is a subclass of sosa:Observation; as stated in the Exp. 3.1, it represents an individual measurement of the wave signal gathered by a sensor device (*veo:Instrument*), measured by the *veo:Station*, at a given time instant *owl:Time*. A datapoint can be composed of one to three *veo:Component*s (a relation labeled *veo:asListOf* exists between *veo:DataPointArray* and *veo:Component* in the ontology model, see Figure 3.4).
- **veo:Component**: represents the component x_i (see Eq. 3.1, Section 3.3) describing the ground motion along one of the directions V, NS, EW., depending on the device type installed on that station.
- **veo:DataPointArray**: this class extends *sosa:ObservationCollection* describing the sequence of *veo:DataPoint*s collected in a given time interval *time:Interval*, by a station sensor, according to Exp. 3.2.
- **veo:DataPointMatrix**: this class also extends *sosa:ObservationCollection*, but represents a collection of *veo:DataPointArray*s that are gathered by each sensor of the *veo:SeismicNetwork*. It can describe a *veo:Event* located in the estimated hypocenter, corresponding to the class name *veo:SourceLocation* (see Exp. 3.3 for details).

3.5.3 Knowledge Model

The concepts colored in pink provide a higher-level description of the collected and processed data, i.e., they allow the seismic event identification and classification, accomplished by a supporting ML-based Classification Layer. The main concepts are given as follows:

- **veo:Location**: this class describes a geographical location by the geographical coordinates *geo:longitude*, *geo:latitude*, and *geo:altitude*. The prefix *geo* means that the properties are imported by the WGS84 Geo Positioning Vocabulary[2]. The class *veo:Location* is related to *veo:Station* by the property *veo:isLocatedIn*, stating the geographical coordinates where the station is located; at the same time, it is a superclass of the specialized class *veo:SourceLocation* representing the hypocenter of the seismic event (i.e., the point

[2]Available: http://www.w3.org/2003/01/geo/wgs84_pos

from which the rupture originates). A suitable data triangulation calculates the source location through the station network.

- **veo:Event**: it is the class describing the source event detected. The instances of this class are indeed the earth ground motions detected by the sensor network and processed to identify the *veo:SourceLocation*, at *veo:OriginTime* (representing the starting time when the rupture occurs at the hypocenter). In our context and experimentation, the classes *veo:Blast*, *veo:VolcanoTectonic*, *veo:LongPeriod* are the possible events that the ML-based classification Layer identifies. The *veo:Event* class has several properties (sosa:ObservableProperty), e.g., *veo:Magnitude*, *veo:FocalMechanism*, *veo:OriginTime*, and *veo:Energy* which can be appropriately associated with a particular event.
- **veo:Blast**: the class describes the blasts (such as the quarry blasts that occurred in Nola or Terzigno caves or the sea explosions due to illegal fishing).
- **veo:VolcanoTectonic**: it is the class describing VT earthquakes.
- **veo:LongPeriod**: it is a class describing LP events.
- **veo:Noise**: is the class describing the background seismic signal.
- **veo:City**: is a class describing the city involved in a seismic event. It is related to the **veo: AlertLevel** class, which describes the impact (i.e., consequences) of the alert level the event could have on the cities.
- **veo:AlertLevel**: a class that provides the alerting level, evaluated after the event detection and the estimation of its severity.

The relations mapped in the proposed ontology provide a comprehensive wave and sensor-network description; besides capturing nature entity relations, they are designed to capture the waveform hierarchy up to the event composition.

The main relations designed in VEO Ontology are reported as follow.

- **veo:isComposedBy:** it is a 1-to-1 relation between the *veo:Station* and the *veo:Instrument*, representing the technical device and the role that is portrayed in the sensor network.
- **veo:isListOf:** denotes the inner decomposition for several events for a higher level container; more explicitly, the *veo:DataPoint* is composed of a list of axis *veo:Component* (V, NS, EW).
- **veo:isLocatedIn:** this relation provides double functionality, express the *veo:Station* geo-referenced location and serve as a

base for the *veo:SourceLocation* of a specific event after the wave detection.

- **veo:hasLengthOf:** describes a numerical value as the number of *veo:DataPoints* contained in a specific *veo:DataPointArray*, this can be offered from the single "wave sample" description through the history of a specific *veo:Station* or the description of a *veo:Event*.
- **veo:generates:** provides a direct connection between a specific sensor network and all the generated *veo:DataPointArray* in that particular *veo: SeismicNetwork*.
- **veo:detectedAs:** deliver the connection between the Collection and the Knowledge model, where a single Wave-form with a finite length can be identified as an existing event.
- **veo:describes:** provide the connecting functionality for the detected event sampling and referencing each of the *veo:DataPointArray* detected to describe that particular *veo:Event*
- **veo:isIdentfiedAs:** This relation shows the identification of one particular *veo:Event* to one of the possible detected types e.g.: *veo:Blast*, *veo:VolcanoTectonic*, etc.
- **veo:definedBy:** describe the *veo:AlertLevel*, according to the *veo:Event* computed properties.
- **veo:affects:** connects the emergency level defined by the *veo:Alertlevel* with the direct impact on a specific *veo:City*, located in the surroundings of the *veo:SensorNetwork*.

3.6 Neural approaches fed by semantic event data

Symbolic knowledge representation by exploiting semantic technologies is related to deep learning models. Neural networks (NN) have become a standard de-facto solution for classification problems in hundreds of domains. A comparative study was carried out by implementing two DL classification methods: on the one hand, the traditional Multilayer Perceptron (MLP) algorithm combined with Linear Predictive Coding (LPC) specially designed for seismic wave feature extraction, and on the other hand, a convolutional neural networks (CNN) approach.

The data is fed to the AI model following the designed experiment-

ation framework, described in Section 3.4.2, where the training data is associated with a label that identifies the event (identified by the ontology reference, the URI of veo:Event). The model is used to predict the correct signal identification; once the classification is performed, the results (recognized events) are added to the current Knowledge Base (KB), following the specific *OWL* constraints, which, later on, can be queried to retrieve the contextualize semantic information on the occurred events, alerting situations or just to get statistics in the future.

Multilayer Perceptron (MLP): The first model is a traditional MLP algorithm. MLP is a layered feedforward neural network with a back-propagation algorithm [139], described in terms of the stochastic gradient descent algorithm, and it minimizes the categorical cross-entropy loss function. The activation function used for the hidden layer is ReLU (rectified linear unit), while the softmax function is used in the output layer. Other training parameter configurations of MLP are reported as follows.

- Number of Layers: 42
- Number of samples by iteration: 50
- Training Rate: 0.001
- Gradient influence direction: 0.9
- Epochs: 640

Let us notice that the input data after the annotation and normalization process described in Section 3.4.2, is additionally preprocessed with linear predictive coding (LPC) [123] to extract spectral features and, in addition, a discretized waveform parameterization of signals composed of 24 s windows is also considered as further input features. The total number of features is 80: 56 spectral features from the LPC (using 8 sub-windows of 3 s and setting the order equal to 6) and 24 waveform features from the discrete waveform parametrization.

Convolutional Neural Networks (CNN): The second model is based on CNN [140], becoming a de facto standard for many applications in computer vision and machine learning [141, 142]. The method has been widely used in characterizing signals in many research fields, especially those oriented to biomedical applications[143]. In our approach, a one-dimensional convolutional neural, 1D CNN, has been

employed; it has become a popular choice among researchers for the significant complexity reduction against the 2D CNN, reducing the computing times, and being used in real-time and low-cost applications. In particular, 1D CNN methods are used for processing time-series data or, particularly, for patient-specific ECG signals in the biomedical domain [144]. The architecture of the proposed network is shown in Figure 3.5.

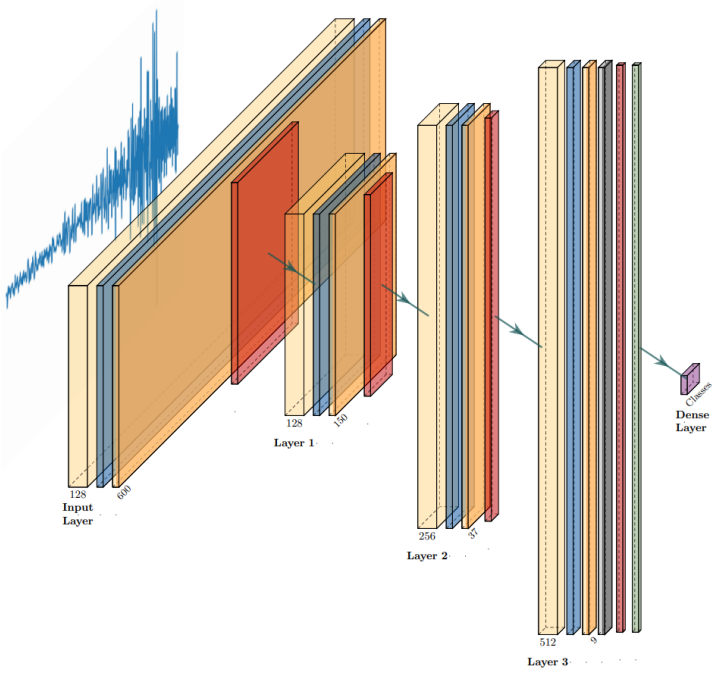


Figure 3.5: CNN architecture design with the colored layers. Yellow: convolution 1D; blue: batch normalization; orange: activation (ReLu); red: MaxPooling1D; gray: dropout (50%); light green: Lambda (mean); and purple: dense (softmax).

The CNN is composed of four-layer blocks; each block includes a convolution one-dimension layer, the batch normalization operation, the ReLU activation function, and, finally, a MaxPooling operation. As shown in Figure 3.5, the initial convolutional layer has 128 filters used for our time-series data of size 2400, a kernel size of 80, and a stride of 4. The Xavier (also known as Glorot) uniform initializer) was used

to randomly initialize the weights of the layer. This block structure appears three times but with 128, 128, and 256 filters, respectively, in the layers called the input layer, layer 1, and layer 2 in Figure 3.5. Layer 1 and layer 2 have instead a kernel size of 3 and stride 1. Finally, layer 3 has a similar design but uses 512 filters, with kernel size 3 and stride 1. This last block also provides a dropout layer with a 50% probability to avoid overfitting along with a Lambda "mean" layer. Finally, a dense layer with softmax activation provides a three-neurons output layer. The proposed CNN model architecture traced approximately 558,597 parameters on the samples trained to perform the classification.

The parameter configuration of 1D CNN is chosen as follows.

- **Layers:** 4
  - **Filters:** 128, 128, 256, 512 (respectively).
  - **Kernel initializer:** Glorot uniform
  - **Activation function:** Rectification linear unit (ReLU);
- **Epochs:** 200
- **Dropouts:** 50%
- **Last layer activation function:** softmax.
- **Loss function:** categorical cross-entropy.
- **Optimizer:** Adam.
- **Callbacks:** ReduceLROnPlateau.
  - **Monitor:** 'val_loss'
  - **Patience:** 5
  - **Redution Factor:** 0.1
  - **Learning Rate:** 0.0001

3.7 Implemented Scenarios

To test our approach, a framework was designed to process data, annotate it, and use a trained model. Figure 3.6 shows the system logical view of our designed framework along with its main modules and the relative tasks in the data flow of its pipeline. Initially, sensing devices in the sensor network acquire the data, whose stream is collected and processed by the *Data Acquisition* Layer, transforming the collected data into a digest form for processing. The system is designed to be easily extended with additional IoT-based devices, such as environmental sensors, even wearable ones. The *Preprocessing* step indeed is in charge

of the seismic data analysis by time sampling and then storing the seismic waves collected.

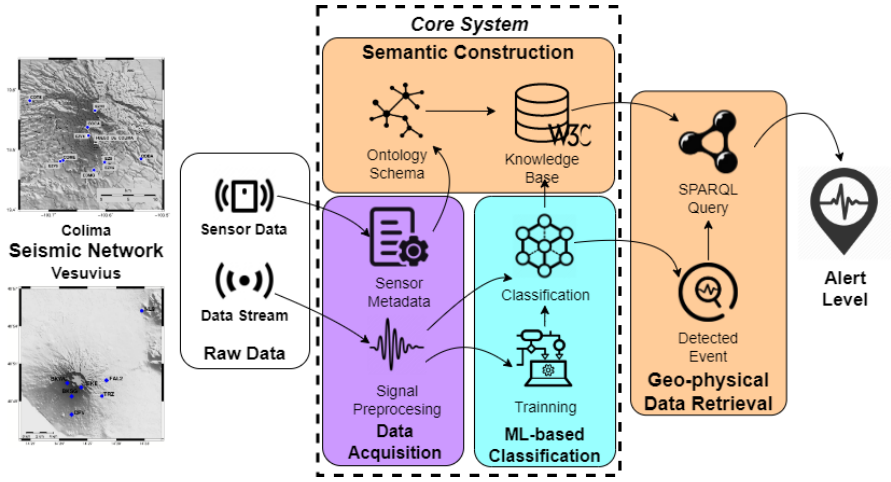


Figure 3.6: Implemented Framework Overview.

The *Semantic Construction* Layer translates the processed data in RDF (Resource Description Framework) triples, according to the defined ontology schema to populate it and infers assertional data to feed and enrich the knowledge base. In the meanwhile, preprocessed data from *Data Acquisition* Layer are also given as input to the *ML-based Classification* Layer that is in charge of the seismic event detection and classification. Initially, this module works on pre-collected seismic data to train the ML model. Also, the knowledge base is populated by adding event-related triples. Then, incoming real-time data from station sensor streams were processed in a digested way and given as input to the neural network-based model to predict if the acquired seismic waveform is a potentially seismic event. The last layer, namely *Geo-physical Data Retrieval* Layer allows querying the knowledge base by the query language SPARQL (a recursive acronym for SPARQL Protocol and RDF Query Language) that retrieves and manipulates data stored in RDF format.

In a nutshell, our system provides a seamless infrastructure to collect data from the seismic domain, analyze them to get processable traces and identify relevant events. The semantics plays a crucial role as an interoperability backbone to get homogeneous knowledge

representation, storage and integrating on-top classification, detection, and recognition in a whole framework that is able to combine Knowledge Representation capabilities with the prediction capability of traditional ANN. The final goal is to improve the ability to uniquely and profitably exploit all revealed features and gain an integrated view by interrogating knowledge at any time.

3.7.1 Classifying Seismic Events

An IoT ecosystem consists of smart devices and sensors that interact and operate under a common format to collect, send, and act on data acquired from the environment. The high-level IoT applications aim at collecting data over time by the spatially distributed sensors and detecting and recognizing events in the real world to interpret the phenomenological data. In a geophysical context, in cases of natural disasters such as severe earthquakes or strong volcanic explosions, an accurate forecast of the occurrence and magnitude is crucial for hazard assessment. For this purpose, to promptly identify anomalies, observatories worldwide manage different sensors collecting continuous seismic signals.

The challenge around data integration, exchange and interpretation, and IoT device interoperability, find an effective solution in the use of Semantic Web Technologies (SWT) for modelling and integrating data from heterogeneous and distributed sensor networks [145].

To this purpose, a comparative analysis between the Multi-Layer Perceptron algorithm and Convolutional Neural Networks has been carried out, aimed at assessing the classifier's performance in recognizing the seismic events leveraging semantically annotated data.

Experimental Results

The knowledge base was built on the VEO ontology, defined for the seismic domain, given in Section 3.5. Collected seismic data are relative to two very dangerous volcanoes, i.e., Vesuvius (Naples, Italy) and Colima (Mexico). VEO extends the well-known ontologies, namely Semantic Sensor Network (SSN) and Sensor, Observation, Sample, and Actuator (SOSA), designed to support the modeling of seismic sensors and stations.

Table 3.2 shows the considered experiments, using different dataset configurations and the relative event classes involved for the classification. Experiments #1, #2 and #3 propose a binary classification, grouping the seismic samples in different ways: by types and also by sources. The goal is to assess if the classifier is able to discriminate the seismic wave from the background noise (the other class is composed of all noise data).

Table 3.2: Signal Classification - Experiment Design: BL: Blast, VT: VT Earthquake, LP: Long-Period, LPV: Long Period from Vesuvius, LPC: Long Period from Colima, NO: Noise.

		Event Class				
Exp.	# Classes	BL	VT	LPV	LPC	NO
1	2	–	–	–	5530	5530
2	2	–	–	726 + 5530		6256
3	2	832 + 928 + 0 + 900				2765
4	4	416	464	–	450	450
5	4	832	928	–	900	900
6	5	832	928	726	1000	1000
7	3	–	–	726	1000	1000
8	3	–	928 + 726		2765	2765

More specifically, **Experiment #1** considers a dataset composed of 5530 LP traces from Colima and 5530 traces of noise data. Since the size of Colima LPs traces is not big as for noise, a data augmentation process has been applied by inverting the polarity of the Colima LP traces. The experiment results show an accurate classification with respect to the two designed classes for both classifiers. The results are promising with high values for Precision, Recall, and F1-score. Figs. 3.7(a-d) show also training trend, validation, and loss, across the fixed epoch number. Notice that LP recognition is very relevant for volcano hazards because the occurrence of LPs is very often associated with a renewal of volcanic activity.

Experiment #2 considers the same two classes, but the dataset is larger: now the LP class includes the LPs from both volcanoes. As already said, merging these two classes makes sense because the LPs are

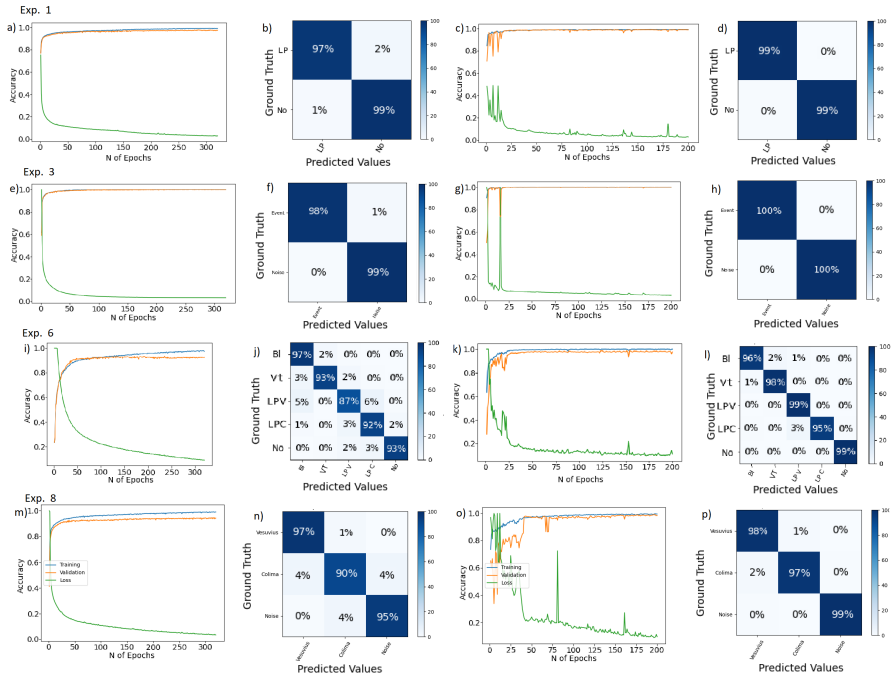


Figure 3.7: Charts representing the results of the training, test and loss iterations and confusion matrices using the MLP and CNN classification, respectively, for the experiments #1,#3,#6,#8.

similar in waveform, duration, and spectral content. The class of noise samples increases accordingly. Performance results are comparable with the previous experiment, showing that the models are able to discriminate the LPs from noise.

Experiment #3, reported also in Figs. 3.7(e-h), proposes a binary classification targeted at the discriminating *event* from *no event*. The models were trained considering two classes: one composed of all the seismic events: BLs, VTs, and LPs from both Vesuvius and Colima (*event* class), whereas the other class includes the noise (*no event* class). The two classes have the same sample size. The performance shown in Table 3.3 emphasizes that both the network models can discriminate input data correctly.

In **Experiment #4**, the classification aims at detecting the different seismic events. The dataset is now composed of four classes: BLs, VTs,

LPs from Colima, and noise data. To ensure balanced classes, each class is composed of about 400 samples. No data augmentation is applied in this experiment. The results are good even if the values for *P*, *R*, and *F1* are slightly lower compared to the previous experiments. It is due to the increase in the number of classes. This experiment is designed to get balanced classes, so the number of samples for Colima's LP and noise was restricted to the same for the samples for blasts and VTs.

Experiment #5 is similar to the previous one, but the sizes of the event classes are increased. Precisely, samples for the blasts and VTs are obtained by data augmentation. Sizes for BL and VT classes are 832 and 928, respectively, i.e., they are duplicated. Sample sets with similar sizes are selected for building LPC (LP from Colima) and Noise (NO) classes guaranteeing a homogeneous class balancing.

Performance results for both experiments are satisfying with slight improvements with the increased size of datasets.

Experiment #6, reported in Figs. 3.7(i-l), includes all the individual classes: BL, VT, and LP from Vesuvius have duplicated size, whereas the size of LP from Colima and noise (NO) is restricted to consider just 1000 traces for class balancing purposes. Performance results for MLP in this experiment are slightly lower than the other experiments, probably due to the increased number of classes.

This case represents a strong improvement in the detection process of volcano observatories because it allows the prompt distinction between those events having a volcanic origin from those not volcanic (such as BLs).

Experiment #7 aims at assessing the accuracy of the two models in discriminating LP from Colima, LP from Vesuvius, and noise. By balancing the class size (augmenting LP samples from Vesuvius), CNN presents more accurate data classification. Although the similarity of LP waveforms, the model recognizes even slight differences between them, distinguishing LPs acquired at Vesuvius from those at Colima. Let us remind that LPs are a marker of the renewal of volcanic activity.

Finally, the last experiment (**Experiment # 8**) aims at recognizing the "signature" of the volcano, taking into account the combined acquisition of all VTs (from Vesuvius) and LPs (from both Vesuvius and Colima). Therefore, the classes of this experiment are Vesuvius, Colima, and Noise. The results reported in Figs. 3.7(m-p) show the trends for training, validation and loss along with the confusion matrix

for both models. Also in this experiment, the performance of the two models is very good, especially for the CNN gets 0.98 for P, R, and F1-score.

This last experiment could be a step toward a very global database where seismic signals from volcanoes worldwide are recognizable. This work was expanded in [146].

Table 3.3: Signal Classification - Experiment Result: Precision (P), Recall (R) and F1-score (F1).

		MLP			**CNN**		
Exp.	Classes	P	R	F1	P	R	F1
1	2	0.98	0.98	0.98	1.00	0.98	0.98
2	2	0.98	0.98	0.98	0.99	0.99	0.99
3	2	0.99	0.98	0.99	1.00	1.00	1.00
4	4	0.94	0.94	0.94	0.93	0.93	0.93
5	4	0.97	0.98	0.97	0.98	0.98	0.98
6	5	0.93	0.93	0.93	0.97	0.98	0.98
7	3	0.94	0.94	0.94	0.98	0.99	0.98
8	3	0.94	0.95	0.94	0.98	0.98	0.98

Table 3.3 reports the eight experiments, providing training and validation tendencies along with the confusion matrices in Figure 3.7. The performance for MLP and CNN are given in terms of the macro-average for the precision (P) (Eq. 3.4a), recall (R) (Eq. 3.4b), F1-score (F1) (Eq. 3.5a).

The described experiments show the effectiveness of our models in classifying seismic signals, proposing different classifications according to the event classes taken into account.

Discussion

This work introduces a semantically-enhanced framework for seismic event detection, starting from data acquired by two seismic sensor networks. An ad-hoc defined ontology describes the seismic domain, modeling seismic signals for seismic event recognition. It has been

built on the well-known SSN/SOSA ontology, modeled to describe sensor-based systems.

Besides ontology modeling, the framework accomplishes the classification/detection of seismic events to feed the knowledge base of new triple-based assertions. Several experiments were carried out on seismic signals recorded at two different volcanoes: Mt. Vesuvius (Naples, Italy) and Colima volcano (Mexico), achieved through comparative analysis of two ML-based methods, MLP and CNN models.

The obtained results highlight the effectiveness of the proposed framework in terms of flexibility and high accuracy. Successful detection and classification are crucial for evidencing any alteration in the volcano dynamics; our framework is designed to support human experts, providing complete access to the knowledge base for a prompt alert in volcanic crises.

The synergy between Artificial Intelligence techniques and Semantic Web technologies provides a solid and formal infrastructure to annotate discovered wave patterns by the semantic integration of raw data of different types coming from heterogeneous sources. Besides the seismic networks with highly sensitive and costly seismometers, wearable [147], low-cost IoT sensors have also been developed for earthquake detection.

Our approach provides a forefront framework designed to be easily extended in a larger sensing domain, including traditional station network approaches and IoT-based applications and devices, to build a comprehensive knowledge base to support users in seismic alerting identification.

3.7.2 Identifying a mountain fingerprint

A further experience in the seismic domain was carried out, starting from the semantic annotated VEO-based model. The presented framework was adapted to process the background seismic signal acquired in different volcanic areas. To the best of our knowledge, *this is the first attempt in the seismological field in which the proposed approach successfully identifies the source that generates it.* The ML-based classification of background seismic signal, as a hallmark for recognition of the generating volcanic source, is thus the main contribution of the present research.

Experimental Results

Specifically, we studied four volcanoes: Campi Flegrei, Ischia Island, Vesuvius (Italy), and Colima (Mexico), each characterized by peculiar patterns of seismic activity. The main aim was to show that the MLP and CNN methods are able to classify the background seismic signal, which contains different contributions (microseism, cultural noise, and volcanic/hydrothermal tremor). The classified signal can be considered a kind of fingerprint, which allows the identification of a volcano, among others. The proposed approach, when applied to continuous seismic acquisitions, can promptly reveal anomalies or variations of the background motion, thus providing information about changes in the sourcing process much earlier than the occurrence of potentially dangerous events such as volcanic eruptions.

Table 3.4: Experiment settings with the class sizes associated with each background noise source of Campi Flegrei, Vesuvio, Ischia Island and Colima

Exp	Campi Flegrei	Vesuvio	Ischia	Colima	Events
E1	18,622	18,837	18,837	18,837	–
E2	18,622	18,837	18,837	18,837	8014

Starting from the results presented in [94], two experiments have been introduced aimed at the detection and classification of background noise signals. The first experiment was carried out on all the background noise signals. The ML methods were trained to distinguish among the noise signals at the different sources used. As reported in Table 3.4, four classes of signals have been considered; the dataset results are balanced since each class is approximately composed of the same number of signals. The second experiment has the same classes reported in the first experiment, with one more composed of transients, i.e., VTs, LPs, quarry blasts, and undersea explosions[93]. This class represents a set of no-background signals used to validate the ML method's capability to learn and classify noise signals.

Our goal with both experiments was to discriminate the volcano fingerprint (in the stationary phase) across the background noise and also in the presence of higher energy transient events. More specifically,

Table 3.5: Models Evaluation Results: Train Accuracy (Tr.A) and Loss (Tr.L), Test Accuracy (Te.A), Average Accuracy (A), Precision (P), Recall (R), and F1-Score (F1) for the two presented experiments

MLP							
Ex.	**M.**	**Tr.A**	**Loss**	**Te.A**	**P**	**R**	**F1**
E1	AVG	95.424	0.179	94.622	0.948	0.946	0.946
	STD	0.675	0.018	0.696	0.006	0.008	0.008
E2	AVG	93.978	0.234	93.123	0.929	0.923	0.925
	STD	0.720	0.017	0.611	0.006	0.008	0.005
CNN							
Ex.	**M.**	**Tr.A**	**Loss**	**Te.A**	**P**	**R**	**F1**
E1	AVG	99.588	0.019	99.417	0.990	0.990	0.990
	STD	0.399	0.009	0.399	0.000	0.000	0.000
E2	AVG	95.648	0.118	95.229	0.970	0.951	0.954
	STD	3.774	0.089	3.748	0.017	0.037	0.034

the precision (Eq. 3.4a) describes how accurate our model is out of those predicted positives; the recall (Eq. 3.4b) calculates how many of the actual correctly classified our model captures by labeling them as true positives. The F1 score (Eq. 3.5a) is the weighted average of precision and recall, while accuracy (Eq. 3.5b) is the ratio of correctly classified samples to the total sample dataset.

The experiment results can be seen in Table 3.5 for the two models employed, MLP and CNN. According to the workflow sketched in Figure 3.3, the models have been validated by using a cross-validation technique. Precisely, for the second experiment proposing unbalanced class samples (the *Event* class is smaller than the others), a stratified k-fold algorithm from the Python Scikit-learn library was carried out to enforce the class distribution in each split of the data to match the distribution in the complete training dataset.

Table 3.5, also shows the mean and the standard deviation of accuracy results for the training data and loss, as well as for testing

data. The mean and the standard deviation are also provided for P, R, and F1. As reported, the performance of the results of the two ML models are quite satisfactory: with the MLP model, *Experiment 1* provides better performance than *Experiment 2*, which involves not only the background signals but also additional transient signals. The latter is different from the background signals of the predefined classes, used in some sense to *dirty* the dataset to evaluate the ability of the model in discriminating extraneous transient events from the background signals to be classified.

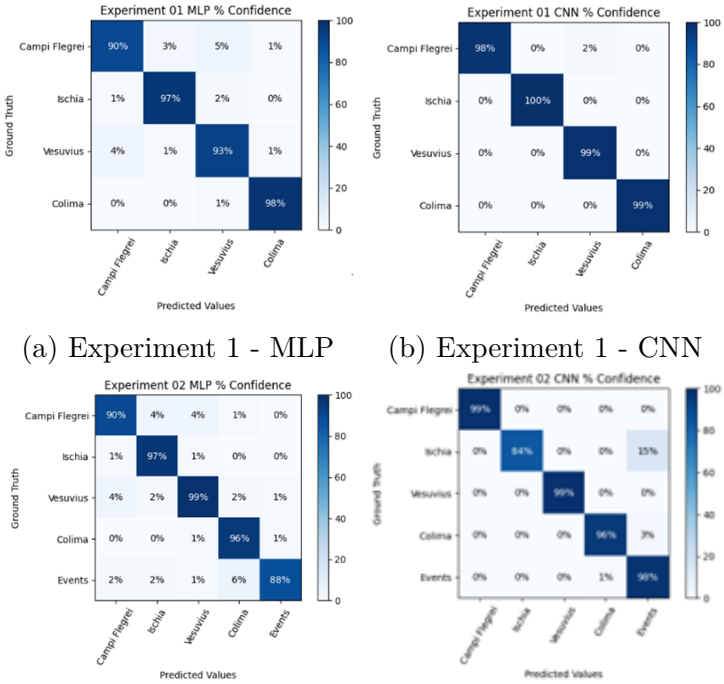
Similar behaviour appears for the CNN model, whose performance in the case of *Experiment 1* is very promising since the values of both *accuracy* and *F1* are close to 100%. This result emphasizes how CNN is able to correctly discriminate between balanced classes of noise signals. In the case of *Experiment 2*, instead, a certain unbalance in the class distribution reflects in the results returned by the model, with a significant standard deviation value for *accuracy*. The *F1 score*, although assuming a similar value to accuracy, is more stable (i.e., the standard deviation is low), according to its characteristic to work better with unbalanced classes.

The confusion matrices computed by considering the predictions for test data in each fold by cross-validation are also reported in Figure 3.8, according to the model used and for each experiment. The confusion matrices confirm the observations reported above, highlighting the satisfying performance of the models in both experiments.

Discussion

In the framework of forecasting volcanic eruptions, much attention is generally devoted to the analysis of seismic transients for prompt detection and cataloging of waveforms. The reason is the strict link between the dynamic state of a volcano and the seismic signals such as LPs or VTs. On the other hand, the seismic acquisition is a complex superposition of several signals emitted by different sources, which can be internal or external to a volcano, that can share common properties, especially the frequency content, making the discrimination even more difficult.

Here, we demonstrate that it is possible to recognize the fingerprint of a specific volcano by basically analyzing the background seismic noise, taking advantage of continuous acquisition in a stationary volcanic phase.



(a) Experiment 1 - MLP (b) Experiment 1 - CNN

(c) Experiment 2 - MLP (d) Experiment 2 - CNN

Figure 3.8: Normalized confusion matrices for the MLP and CNN models for the two experiments.

This puts on a more quantitative basis the now consolidated idea that persistent tremors, both volcanic and hydrothermal, and some volcanic transients (i.e., LPs or explosion quakes) are generated by the exact physical mechanism, eventually in different regimes involving different levels of seismic energy.

Investigations were conducted by using ML techniques for the Neapolitan volcanoes and the Colima volcano. These analyses are advantageous in weak ground motions as the background seismic noise is plenty of information often neglected by traditional signal processing techniques. The ML-based models provide very reliable solutions when all the traces of all the volcanoes are mixed together, allowing the identification of pieces of background tremor. However, more than this, our method ascribes each recognized tremor to the relative volcano with its characteristics. In addition, introducing new classes of labeled

noise waveforms in the ML techniques framework enriches the training phases so that the classifier can better discriminate between low-energy volcanic signals from spurious ones, just reducing the probability of detecting false positive events [148]. The accurate identification and classification of continuous seismic noise represent progress toward the characterization of background signals, which can also be implemented in real-time for monitoring purposes.

The promising performance shown by our ML approach in background noise analysis applied to volcanoes with different activities (e.g., hydrothermal, volcanic, anthropogenic, etc.), in quiescent or in unrest phases, and with even different geological features (calderas, stratovolcano), makes us confident that the method can be extended to a large variety of volcanoes for extracting their fingerprints and significantly contributing to the study of the seismic/volcanic sources.

Separate classification of background noise and transients opens the way to hybrid approaches that use specific classifiers integrated into decision models to employ both signals, even constructed from a dataset provided by a single seismic station. This would be successful in those areas that are not accessible to locate seismic networks or seismic arrays. Finally, our network should recognize any anomalies in the background noise, thus indirectly evidencing changes in the dynamics of a volcano.

3.7.3 Detecting in Alerts levels on Cleanrooms

With the explosion of big data technologies (BD), the possibility of integrating those tools into daily company operations is more affordable and straightforward. On the other hand, Knowledge-based approaches such as knowledge graphs and semantic approaches, although those have not been so popular in the industry in the past years, nowadays, thanks to the high availability of heterogeneous data inside of the company context, they are being used more to enhance or enrich data and processes and make more informed decisions about the business. In light of this scenario, a platform called SEMT (Sensor Environment Monitoring Tool) combines a Big Data recollection from a legacy/manual sensor environment. It performs a knowledge enhancement process to support the semiconductor development and production operations inside a cleanroom.

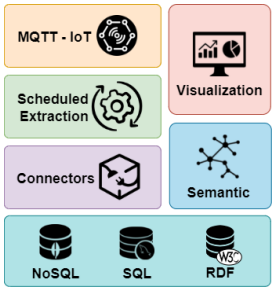
SEMT platform was deployed in the company Leonardo SpA, in

Rome headquarters, for monitoring a manufacturing environment, i.e., cleanroom. Cleanrooms are controlled R+D closed manufacturing environments that deliver controlled levels of particles, temperature, and other climate-related characteristics. The goal was to collect and analyze sensor environmental data over a production line, to support the cleanroom quality control process and the production of semiconductors in this facility.

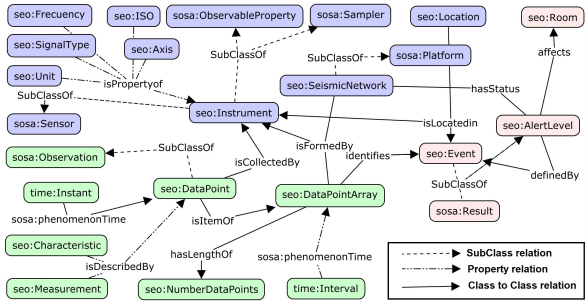
For these heterogeneous domains, semantic tools can boost the gathered data [56], providing context and enabling inference starting from the native data relations or external data sources, such as status reports or supply chain data. This was followed by pattern discovery, querying the system for contextual answers related to the status of a product using, for example, SPARQL queries [79]. The motivation for this industrial use case was to test the production-level use of a semantic knowledge base pattern to create the foundation for a context-aware environmental sensor system within limited and security-constrained environments.

Implementation Results

The system was designed to extract environmental data from heterogeneous sensors scattered in different laboratory segments. The general approach in this architecture was to create a data microservices-based supply chain; those services are divided into layers grouped by functionalities at a higher level, as shown in Fig. 3.9a.



(a) High-Level Architecture view



(b) Knowledge-Base Schema

All the components use HTTP REST messages, allowing transparent and standard message interchange between the services in general.

Storage layer provides persistence to the data, descriptors for sensors, collected data, and semantic data, The **Connectors layer** interacts directly with the device and its data; a connector for each device type should exist. The **Scheduled Extraction layer** provides the ability to coordinate the extraction from the sensor network, **Visualization layer** acts as a web app for displaying the dashboards, data reports, and device management interfaces. Finally, **MQTT IoT layer** can forward the recollected data via MQTT to the Leonardo 4.0 platform.

Some of the most compelling challenges at the design phase of this system were: (a) the inherent need to maintain historical data in an arbitrated way; for later trace and product quality control and (b) usage of temporal data stored, both immediately and distant. A simple adaptation of a contextual sensor network ontology [93] was used to attend to these requirements. The knowledge schema enhances the sensor data, mainly focusing on the collection model see Fig. 3.9b (Green Section); for this knowledge representation, two entities were exploited: (1) *DataPoint* and (2) *DataPointArray*, those are mainly focused on the live data collection.

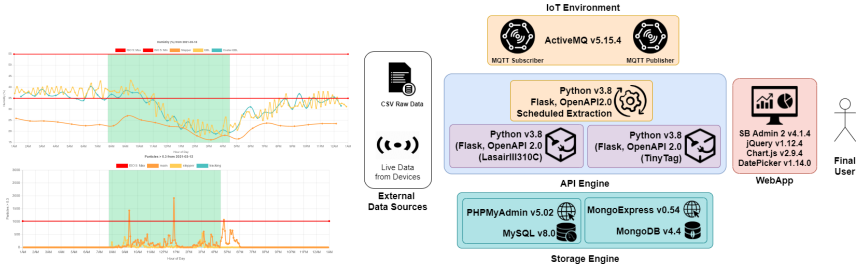


Figure 3.10: Generated Environmental Reports in Humidity and Particle Counter (Left) and Detailed Technology Blueprint(Right). This environment was deployed in production.

A long-term purpose is to consider a later knowledge-discovery process tto combine this data with product data, personnel access and activity logs, and other relevant and context-aware data. Initially, in this same vein, the semantic layer is also used to query temporally stored data to run reports and predictive analyses on the various internal cleanroom locations, with the goal of evaluating and assisting a risk management procedure that could compromise product profitability.

Discussion

This work's main intention was to develop the closest to an industry-grade (see Fig. 3.10 solution that brings value into an existing production line implementing cutting-edge approaches and delivering a product complying with the standards of industry 4.0.

As controlled and restricted environments, cleanrooms require fast and efficient technologies to deliver real-time, contextual, and aware data of the room's reality, enabling further monitoring, tracing, controlling, and predicting product characteristics. With the development of this platform, it was possible to demonstrate that a fast semantic-data ecosystem can be set up in restrictive and productive spaces and contribute added value to the process.

Chapter 4

Vector-Based Representation in Knowledge Graph Construction

Knowledge graphs (KGs) are a way to efficiently collect, organize, and manage knowledge from huge amounts of data, increase the quality of information services, and provide smarter services to consumers. These aspects are based on overlapping layers of knowledge that enhance the ability to extract, infer or reason about a given KB, making it one of the most important features of KGs [149]. Finding inconsistencies and drawing new conclusions from existing data are the objectives of knowledge extraction over knowledge graphs. In addition, new relationships between entities may be derived through knowledge inference, which can subsequently be used to enhance knowledge graphs and enable sophisticated applications. Similarly to logical frameworks, entity vector representations, also called embeddings, provide another learning strategy while maintaining the same functionality, known as link prediction. This chapter will discuss the contributions related to Knowledge Graph Embeddings (KGEs).

4.1 Introduction

With the boom of Machine Learning and especially Deep Learning techniques such as LSTM, Encoder-Decoder approaches, and Transformers in the past half of a decade, the need to integrate inductive Learning techniques into the semantic ecosystem has become more prevalent in upcoming research scenarios.

The embeddings are numerical vector generalisations computed with the intention of representing feature values of data. In the Knowledge Graph domain, the embeddings represent multi (low or high) dimensional abstractions for the predicates across all KG for a given entity. Knowledge Graph Embeddings (KGE) exploits Deep Learning (DL) concepts such as back-propagation and loss functions to learn step by step from the KG data. Recently, KGE has become the default technique for link prediction tasks in the semantic web community.

This chapter mainly focuses on the intersection of Machine Learning, Explainability, Data Mining and Semantic Web technologies from the landscape described in [1]; the figure 4.1 shows the intersection of these topics between the dotted lines.

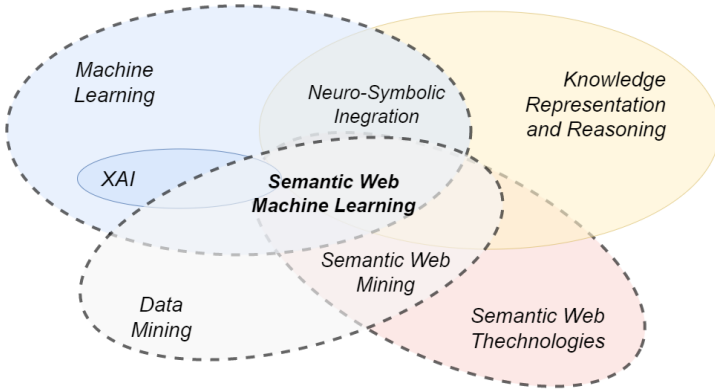


Figure 4.1: SWeMLS landscape, taken from [1], and adapted to focus on the contribution scope of the chapter Vector Representations Approaches in Knowledge Graph Construction

KGEs are representation learning techniques that aim to encode entities and relationships in a knowledge graph as vectors. Typically,

embeddings are implemented for multi-link prediction, which means that computed entity embeddings are able to predict links for all predicates in a given heterogeneous graph. This prediction is assessed by means of a ranking score known as the Rank (MR), described in Section 4.3.2. However, recent approaches push forward the KGE entity ranking feature for Knowledge representation (KR) tasks [150], boosting the functionalities and the spectrum of the KGE, enabling them to be implemented in a broad new set of ways.

This chapter presents contributions related to vector representation in the Knowledge Graph construction task. Section 4.2 presents an overview of KGE-oriented approaches. Section 4.3 presents a complete baseline of methods and additional supporting mechanisms to complement the Knowledge Base Construction; in Subsection 4.3.4, a complete and detailed list of the standard datasets used to validate the proposed methods in the contribution sections.

The first contribution is described in Section 4.4, focusing on the validation of KG using embeddings as vector abstractions from the knowledge representation perspective. One pathway was oriented to the validation of a KG built by text using the zero-shot learning (ZSL) approach: the idea is to create the KG from prosaic text without previous knowledge and validate the constructed knowledge graph by using KGE and synthetic false and true data, obtaining values $\simeq 0$ for the false data and $\simeq 1$ for the true data. To complete this task, an RDF-based annotation task of KG generated from the prosaic text was designed, developed and published using a public repository[1]. The resulting KB is evaluated by three different types of embedding, with results confirming the initial validation hypothesis and supporting the original premise.

The previous validation of KG exploiting embeddings highlights the synthetic triple generation's crucial role in providing a consistent methodology to augment the data in a KG, directly impacting the performance of downstream tasks in the AI models derived from that KB. In Natural Language Understanding (NLU) domain, machine learning models are sometimes validated using synthetic triples using expert systems, yielding controlled negative or positive outputs on the validation experimentation settings. In Section 4.5, a synthetic triple-generation algorithm for enhancing downstream tasks in KGEs is

[1]GitHub Repository - https://github.com/d1egoprog/Text2KG

described. It is based on the graph complement: the algorithm identifies missing relationships between entities by exploiting the complement graph and generates high-quality synthetic triples that improve the accuracy of predictions; the generated triples were benchmarked using several knowledge graphs embedding methods and scoring metrics for these types of models, demonstrating the effectiveness of the proposed approach in enhancing the overall performance of knowledge graph embedding models. In particular, the performance of all four selected models improved when the original knowledge base was supplemented with synthetic triples.

The final contribution described in Section 4.6 conveys an approach that leverages Knowledge Graph Embedding for modelling the KG as a gravitational system and applies to an international trade domain. The main goal for the gravity KBC was the later integration of traditional machine learning methods with the resulting entity vector representations, such as decision trees, linear regression, and graph neural networks. The research findings demonstrate the potential for improving prediction accuracy in traditional ML methods and provide insights into embedding explainability in knowledge representation purposes.

4.1.1 Contribution

This chapter is devoted to two objectives. The first objective is to advance the ranking capabilities of Knowledge Graph Embedding Methods, enabling the validation of Knowledge Base Construction methods and synthetic triple generation methods. The second objective aims to enhance traditional Machine Learning methods by leveraging Knowledge Graph embeddings as vector representations.

To offer a comprehensive overview of the chapter's contributions in a more refined manner, the following list presents a synthesized compilation of all the noteworthy accomplishments within this chapter.

- A experimentation framework for RDF-based Knowledge Graph construction evaluation from plain text sources
- Release of a system tool used for the RDF Knowledge Graph construction from plain prosaic text.
- MoviesQA (wikimovies) dataset extension for support link prediction task and publication in a public repository.
- Synthethic triple algorithm for candidate triples generation.

- A gravity-embeddings generation methodology for downstream task ingestion.

4.1.2 Publications List

The publications derived from this chapter work are listed as follows. The journal/conference details are described in Table 4.1.

1. Rincon-Yanez, D., & Senatore, S. (2022). FAIR Knowledge Graph construction from text, an approach applied to fictional novels. Proceedings of the 1st International Workshop on Knowledge Graph Generation From Text and the 1st International Workshop on Modular Knowledge Co-Located with 19th Extended Semantic Conference (ESWC 2022), 94–108.
2. Di Paolo, G., Rincon-Yanez, D., & Senatore, S. (2023). A Quick Prototype for Assessing OpenIE Knowledge Graph-Based Question-Answering Systems. Information, 14(3), 186. 10.3390/info14030186
3. Rincon-Yanez, D., Mouakher, A., & Senatore, S. (2023). Enhancing downstream tasks in Knowledge Graphs Embeddings: A Complement Graph-based Approach Applied to Bilateral Trade. Procedia Computer Science, 225, 3692–3700.
4. Rincon-Yanez, D., Ounoughi, C., Sellami, B., Kalvet, T., Tiits, M., Senatore, S., & Yahia, S. ben. (2023). Accurate prediction of international trade flows: Leveraging knowledge graphs and their embeddings. Journal of King Saud University - Computer and Information Sciences, 35(10), 101789.

Table 4.1: Impacted (C)onferences and (J)ournals, ordered by publication Date. Cite Score (CS), Impact Factor (IF). Conference)

Pub.	Name	Type	Rank	CS	IF
1	ESWC	C	A1	–	–
2	Information	J	Q2	4.2	2.38
3	KES	C	B	–	–
4	JKSUCIS	J	Q1	11.9	9.0

4.2 Related Work

In recent years, *Knowledge Graph Embedding* (KGE) techniques have attracted a lot of interest since they can learn the representations (i.e., embeddings) of entities and relations in low-dimensional vector spaces [6], and these embeddings may be used as features to enable link prediction, entity categorization, and question answering.

A Question-answering (QA) system typically consists of a natural language question (the input), a knowledge base, the engine (agent), and an answer (the output). A knowledge base is typically structured as a knowledge graph [151], which offers a consistent way to express heterogeneous entities and concepts in the form of triples *(head entity, relation, tail entity)* (denoted as (h, r, t)). An increasing amount of attention is being directed to QA systems with the widespread of KGs in academia and industry. With the introduction of large-scale open-domain KGs, such as Freebase [152], Wikidata [153] and DBpedia [61], it can be said that KGs are the sources from which the QA system draws for answers.

Noteworthy is the Open Information Extraction (OpenIE) systems built by exploiting the OpenIE paradigm [154]; this paradigm allows the extraction of open relations and arguments in the form: *subject; relation; object* from sentences, with no domain restriction and without requiring any previous training data. This technique enables a semantic representation via simple predictive statements [155], useful for extracting relations from simple statements, like questions (e.g., who directed the Star Wars movie?); once this knowledge has been properly collected, it can be used for a wide variety of downstream tasks, such as precisely Question Answering. Thanks to the OpenIE paradigm, a number of advantages are achieved: automatic annotation replaces time-consuming and error-prone manual annotation; domain-independent design is possible due to the rich knowledge representation, a scalable solution for extracting facts and relationships from unstructured text is offered, and it is a powerful tool for automatic knowledge management and retrieval.

4.2.1 Knowledge Graph Construction

Natural Language Processing (NLP) tasks play an important role in KG construction. Recent research finds relations from sentences by

exploiting a graph neural network for sentence representations and aligning them to the KG [156]. Tasks such as Named Entity Recognition (NER) [157] and Relation Extraction (RE) [158], [159] aim at extracting structural information at sentence and document level [160], exploiting also a description-based representation of entities [161] as well as ontologies [162] starting from unstructured texts. Identifying and extracting accurate and comprehensive information from unstructured data could be a challenging task for knowledge construction. The OpenIE paradigm allows for the automated extraction of structured knowledge without the need for pre-defined schema, ontologies, evolving knowledge bases, or manual annotation [13].

Also, prompt-tuning for few-shot classification tasks has gained some attention in enhancing knowledge extraction [163]. However, most of the recent literature on KGs focuses on particular facets of KGs, consisting of their construction [164] and embedding [29].

Despite the fact that KGs contain a significant number of entities and relations, link prediction is a critical issue for knowledge graphs because they are typically incomplete. Indeed, large-scale KGs such as NELL [165], Freebase [152], WordNet [166], and YAGO [167] have emerged as important sources of auxiliary information for many AI-related tasks like question answering, report extraction, and recommendation systems [29], [13].

The learning of low-dimensional representations of entities and relations for missing link prediction has recently been the subject of extensive research on topics like completeness, partiality, and newly-added information [168]. At the same time, since generated knowledge graphs can contain millions of entities and relationships, embedding them all in a low-dimensional space might be computationally demanding, making scalability difficult [169].

In terms of completeness, some academic trends have centered on link prediction, which seeks to detect missing facts in KGs. Knowledge graphs can be sparse, many entity pairs lack any known relationships. Due to this sparsity, it may be challenging for embedding models to properly represent and then predict the relationships between entities [170].

Existing approaches for link prediction are the Knowledge Graph Embedding models (KGE) [171], [161]. To retain the basic structure of the KGs, KGE models learn the embedded representation of both

the relations and the entities, preserving the inherent structure of the KGs. Predicting missing links with knowledge graph embedding (KGE) methods has been extensively investigated in recent years. The general methodology is to define a score function for the triple [72], [73], [74]. Geometric properties are used in several KG embedding techniques. Some improvements have been gained by utilizing either more complicated spaces (e.g., moving from Euclidean to complex or hyperbolic space) [72], or more advanced procedures (e.g., from translations to isometries, or to learning graph neural networks) [74]. Other alternative strategies have progressed in both directions [172].

Nevertheless, knowledge graph embedding models are typically trained on a specific knowledge domain [7], [6] and may not generalize well to other knowledge graphs or new entities and relationships that are not present in the training data [6]. Moreover, KGE models are not able to handle complex queries with multiple entities and relations. In [173], the method called LEGO creates a latent execution-guided reasoning framework that infers a question's latent structure and reasons in the latent space for multi-hop logical inference.

4.3 Baseline Methods and Datasets

4.3.1 NLP for OpenIE (Information Extraction)

NLP-based parsing tasks extract plain triple-based sentences. The *Triple Construction* component initially carries out the *Basic POS annotation* on the input raw text, consisting of traditional preliminary text analysis, namely, the tokenization, stop-word removal, and tagging routines to identify and annotate the basic part of speech entities.

Once the *Basic POS annotation* is accomplished, parallel activities, namely *Information Extraction* and *Chunking Rules* and *Tagging Pattern Execution* are carried out, as shown in Figure 4.2.

Information Extraction:

This block is composed of three basic tasks necessary to get further processing of the tagged text yield by the *Basic POS Annotation*. The

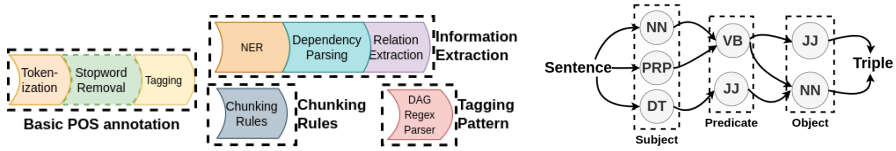


Figure 4.2: Basic plain Triple Construction: from raw text to the POS annotation, chunking rules to grammar dependency rules. The dashed lines represent a process that is not executed and applies to the Tagging Pattern Execution step. (left). Directed Acyclic Graph (DAG) describing the Tagging Pattern task (right)

text is indeed given to the Named Entity Recognizer (*NER*) which identifies named entities according to some specific entity types: person, organization, location, event. Then a *Dependency Parsing* task allow locating named entities into *subject* or *object*, and finally, the *Relation Extraction* task is in charge of the triple composition, starting by the detected entities. Let us remark that the named entities has a crucial role in the triple identification and generation.

Tagging Pattern:

This task accomplishes ad-hoc text processing, considering the nature of the text from a linguistic perspective. Linguistic patterns were defined specifically for processing the prosaic narrative and for this reason, for example, some stop words are important and not discarded. Those patterns can be implemented to discriminate additional triple-based statement to feed the KB; in particular, a finite-state machine is proposed, implementing three evaluation steps, one for each triple component $T = (S, P, O)$. Figure 4.2 sketches a few of the implemented state transitions used in the development of this methodology.

Chunking Rules:

This task performs syntactic parsing of the text in which specific subsequences of words forming key phrases are identified. A regular expression was defined on the word tag sequence to select and extract basic triples to enrich the KB, such as the following tag string $TP : \{< DT >$

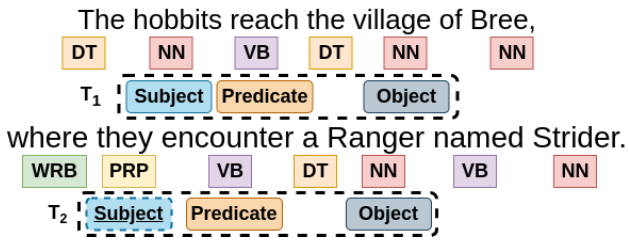


Figure 4.3: Chunking Rule Evaluation Example.

$? < JJ* >< NN >? < VB* >< DT >? < JJ* >< NN >\}$. This regular expression pattern says that a triple pattern *TP* chunk should be formed by the subject chunker which finds an optional determiner (DT) followed by any number of adjectives (JJ) and then a noun (NN); then the predicate-chunker composed of the verb (in its tenses) an finally the object-chunker that is similar to the subject chunker, except that it could be a pronoun besides the noun. Subject and object chunkers are typical noun phrases. Some rule example application is shown in Figure 4.3.

As a final result of the *Triple Construction* component, all the generated data are integrated into a set of plain triples, $T_{Plain} \equiv T_{IE} \cup T_{Pattern} \cup T_{Chunking}$, where the T_{IE} represents resulting triples of Information Extraction task, the $T_{Pattern}$, the result of Tagging Pattern and $T_{Chunking}$ is the Chunking Rules result, once redundant and duplicated are discarded. Finally, the resulting triple set T_{Plain} is stored in a plain format file.

4.3.2 Knowledge Graph Embeddings

KGE embedding models can be categorized based on the type of scoring function they use (1) Geometric, (2) Matrix (Tensors) Factorization or (3) Deep Learning [77]. However, to evaluate the selected KBC methods, a model of each family was chosen, with the aim of obtaining a wider evaluation of the characteristics of each extracted KB. In this experience, we would evaluate the link prediction. To leverage the KGE robustness and validate the extracted KBs, a cross-knowledge graph validation approach was implemented, specifically *Entity Replacement* [174] approach on a pair-wise of the extracted KBs can be defined more

formally as follows:

Definition 7 *Given two knowledge bases, KB_m and KB_n, and let e_S and e_O be two entities that are in the two KBs, a triple $T_m \in KB_m$, $T_m = (e_S, r_P, e_O)$ and a relation $r_P^n \in KB_n$, the Entity parts (e_S, e_O) can be placed in the form (e_S, r_P^n, e_O) to add an additional statement to KB_m or placing $r_P^m \in KB_m$ as (e_S, r_P^m, e_O) to add the new triple to KB_n.*

Evaluation Metrics The metrics considered are the Mean Rank (MR), the Mean Reciprocal Rank (MRR) and the Hits@N.

The Mean Rank, defined in Exp.4.1, computes the query average ranking position among samples.

$$\mathbf{MR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} q \tag{4.1}$$

MRR defined in Exp. 4.2 is a statistical measure for evaluating any process that produces a list of possible responses to a sample of queries (in this case, triples), ordered by the probability of correctness.

$$\mathbf{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{q_i} \in [0, 1] \tag{4.2}$$

More formally, the mean reciprocal rank is the average of the reciprocal ranks of results for a sample of triples Q; where $rank_{qi}$ refers to the rank position of the first relevant document for the i^{th} query q. It represents the ability of the model to produce a correct answer.

The *Hits@N* is defined in Exp. 4.3 and represents the percentage rate of original triples ranked at the top N in prediction.

$$\mathbf{HITS@N} = \frac{|q \in Q : q < n|}{|Q|} : rank_i < N| \in [0, 1] \tag{4.3}$$

HITS@N with $N = 1$ is the conventional accuracy; the model prediction (the one with the highest probability) must be exactly the expected answer. It captures the fraction of triples that appear in the first N triples of the sorted rank list, or in other words; it calculates the percentage of examples for which the predicted label matches the

specific target label. Different values of Hits have been calculated, in particular: HITS@1, HITS@3, HITS@10. This measure represents the ability of the model to produce a correct answer in the top 1, 3, or 10 answers produced, respectively.

4.3.3 Transformers for OpenIE

Transformers are neural networks modes designed with self-attention mechanisms to capture knowledge in sequential data; this strategy is commonly used for NLP and NLU tasks. REBEL [158] translate natural language sentences into triples, with an end-to-end approach that takes unstructured textual input and generates structured output compliant with a given vocabulary. REBEL is a BART-based transformer, fine-tuned on common-sense relation extraction datasets such as the news-based: NYT-Multi[2] and CONLL04 [175], document level: DocRED [176], and pharmaceutical ADE [177].

REBEL is a seq2seq model built on BART [178] that performs end-to-end relation extraction for several relation types. The goal is to translate raw input sentences into a set of triples. This model tackles Relation Extraction and Classification as a generation task, just like a [44] "translation". It is based on the Teacher Forcing method widely used in RNN [179] for translation tasks. It leverages text pairs in two languages by conditioning the decoded text on the input. At training time, the encoder receives the text in one language, and the decoder gets the text in the other language, outputting the prediction for the next token at each position. A similar mechanism is used in REBEL: a raw input sentence containing entities is translated with implicit relations between them into a set of triplets that explicitly refer to those relations. The triplets must therefore be expressed as a series of tokens for the model to decode. The model uses a reversible linearization with special tokens that enable the model to output the relations in the text in the form of triplets while minimizing the number of tokens that need to be decoded.

REBEL takes raw text as input and outputs linearized triples. If our input sentence is Q_1 to x and the result of linearization of the relations in Q_1 is T_1 to y, REBEL's task is to autoregressively generate y given x, as shown in [158]:

[2]https://ir-datasets.com/nyt.html

$$pBART(y|x) = \prod_{i=1}^{len(y)} pBART(y_i|y < i, x) \tag{4.4}$$

4.3.4 Datasets

MoviesQA (Wikimovies)

The MovieQA (WikiMovies) dataset [180] is a question-answering pair dataset built by Wikipedia that includes the raw source text and the corresponding KB framed in the movies domain. The dataset comprises three forms of knowledge representation: i) raw Wikipedia *documents* describing movies; ii) a classical graph-based *KB* consisting of entities and relations created from the Open Movie Database (OMDB) and MovieLens (disclosed in separate files); and iii) information extracted (*IE*) by processing the Wikipedia pages to build a KB. The dataset matches the query with the three knowledge types mentioned.

The experiments were conducted on a dataset with a graph-based KB representation. This dataset is made up of $T =< h, r, t >$ triples that correspond to the structure (film, relation, object). The KB dataset holds information on 18 thousand movies using OMDb and MovieLens metadata with entries for each movie and nine different relation types: director, writer, actor, release year, language, genre, tags, IMDb rating and IMDb votes, with 10k related actors, 6k directors and 43k entities in total, in which 75k+ are distinct entities, on ten structured relations, reaching over 186 thousand triples.

The CEPII gravity database

The CEPII gravity database is a dataset containing information on bilateral trade flows between countries worldwide. The dataset is based on the gravity model of international trade, which states that the trade volume between two countries is proportional to their economic size and inversely proportional to the distance between them [181]. More than 220 countries and territories are represented in this database, with details on trade flows, GDP, population, and other pertinent variables available from 1948 to the present. The dataset is available on the

CEPII website[3].

4.4 A Validation Framework for RDF OpenIE KBs

In light of the FAIR principles for interoperability and reuse of data, Open KBs are standardized, curated and reliable data sources for ingesting quality data and improving knowledge exploitation tasks. In the Triple Construction task, leveraging the integration of these Open KBs can help with tasks such as cross-validation of statements and fact-checking by relying on external sources like other Open KBs.

The goal is to define an initial framework for validating a built KB starting from textual resources (documents, web pages, tweets, posts, reports, etc.)

This work is divided into two parts. First, a methodological approach is presented to build domain-specific knowledge graphs compliant with FAIR principles from an automated perspective and without using knowledge-based semantic models. Then, a validation framework based on Knowledge Graph Embeddings (KGE) that measures the accurateness of a given RDF-based KB by relying on the ranking-based measures delivered by the Mean Reciprocal Rank (MRR) and the Hits@N scores.

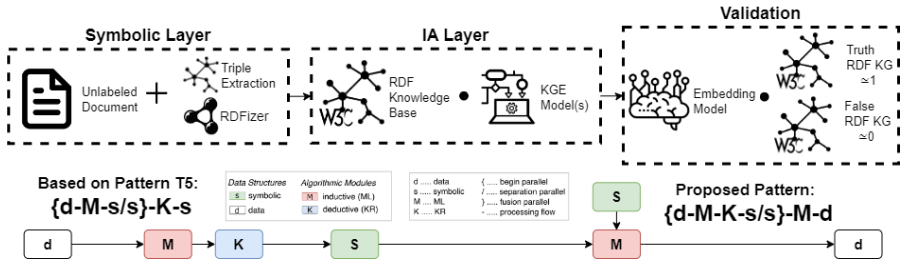


Figure 4.4: Proposed high-level unattended Knowledge Base Construction methodology.

From a high-level perspective, Fig. 4.4 shows the pipeline of our framework. First, a plain text document is given as input for plain triple

[3] http://www.cepii.fr/CEPII/en/bdd_modele/

extraction; the plain KB processed by an RDFzier component is transformed into RDF-based triples. Then querying DBPedia, additional resource triples are extracted and entity disambiguation is resolved: these tasks compound the Symbolic Layer. Once the RDF KB is stable, training is done by using KGE models; this module is named the AI Layer. The trained model will be used by the validation framework to evaluate the KB against Synthetic True and False triples from the same domain.

From the NeuroSymbolic AI perspective, the pipeline design is based on the T5 design pattern ({d-M-s/s}-K-s), which facilitates train plain data, produce a semantic answer (d-M-s) and use the produced semantic answer to perform a logic task. The T5 architecture pattern is enhanced by introducing semantically annotated data to the inductive learning approach, producing a semantic output (d-M-K-s) and reuse the semantic output to perform an additional inductive task ({s/s}-M-d).

In a nutshell, The KBC method leverages traditional Natural Language Processing (NLP) techniques to generate simple triples as a starting point and enhance them using external OpenKG, i.e., DBpedia, to annotate, normalize, and consolidate an RDF-based KG. As a case study, novels in literature were explored to validate the proposed approach by exploiting the features of the narrative construction and relationships between characters, places, and events in the story. This methodology builds a domain-specific and FAIR-compliant knowledge graph, focusing on the integration and completion of assertions and, in this case, starting from a state of non-knowledge and a domain-specific source of information.

4.4.1 Knowledge Base Construction from Text

The process starts from the *Unlabeled Document*, representing the domain of interest, composed of textual resources, generally unstructured, in natural language, available from the Web, even though proprietary textual resources could also be used. To construct the initial plain triples, some NLP techniques were used, such as Chunking Rules and Extraction Patterns creating Directed Acyclic Graphs (DAGs); these techniques are described in Section 4.3.1; the main focus was on using online dictionaries [54] and NLP and NLU techniques to produce an initial KB of plain triples, taking advantage of the text structure to

identify the relevant entities in the text and the actions performed by those entities.

Algorithm 2: SPARQL Query example to URI and Label extraction

```
SELECT * WHERE {{
  {{  dbr:{word} rdf:type dbo:
      FictionalCharacter .
    dbr:{word} dbp:name ?label .
  }} UNION {{ {{ dbr:{word} dbo:
      wikiPageRedirects ?URI }}
    UNION {{ dbr:{word} dbo:
      wikiPageDisambiguates ?URI }} .
    ?URI rdf:type dbo:FictionalCharacter .
    ?URI rdfs:label ?label
  }}
  FILTER (lang(?label) = 'en')
}}
```

The *Triple Enrichment* step focuses on the search of external databases to disambiguate and normalize the data. To this aim, a set of SPARQL queries was executed systematically against the DBpedia endpoint [182] for one of the existing statements of the plain KB. One of the executed queries can be seen in Alg 2; in this SPARQL sentence it can be seen the disambiguation process performed against DBpedia, performing at the same time the extraction of the URI, the normalized name and the ¡*rdfs:label*¿ for that particular entity in the plain KB in the English language.

The final result for one of the founded entities in the disambiguation and cross-reference process can be seen in Figure 4.5, where from the discovered entity "*Bree*", the URI i.e. "*dbr:Bree_(Middle-earth)*", the normalized name in English i.e., "*Bree (Tolkien) (en)*" and the rdfs:label resource.

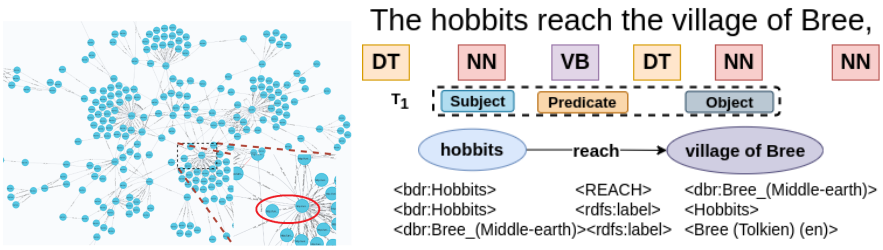


Figure 4.5: Result of the RDFizing process with an example triple.

4.4.2 Validation Framework for RDF KG

The embeddings training phase, mostly done by the AI layer in Fig 4.4 consists of the use of KG embeddings (KGE) models such as DisMult, ComplEX, and TransE to score and validate the results [183]. Noteworthy is Ampligraph [184], a Python library which provides a comprehensive evaluation framework for KGE models.

The proposed validation framework takes the principle of the embedding evaluation where the Mean Rank (MR) of an unseen triple T_u confronted with a given model, trained over a specific knowledge base KB should be higher if the triple is near the dimensional space of the entities in the KB; this means that if the unseen triple belongs to the knowledge base, the ranking should be close or equal to 1. In practical terms, $MR(Embeding_{\text{KB}}(T_u)) \simeq 1$, the unseen triple belongs to that knowledge base, therefore for that particular model is *True* concerning the trained KB.

Considering a controlled scenario, where for a domain $D1$, two unseen triples are given to the trained model, a UT_{True} and UT_{False} and evaluation results are close or equal to 1 and 0, respectively, is possible to state that the KB used for train that model belongs to the domain D1, otherwise do not belong the domain.

4.1: Knowledge Base Validation Approach

$$\begin{aligned} \forall r \in KGE \subseteq SynKB_{True} &\simeq 1 \\ \forall r \in KGE \subseteq SynKB_{False} &\simeq 0 \end{aligned} \tag{4.5}$$

Therefore, the evaluation of all the relations in a given KB is $\simeq 1$ if is confronted with a TrueKB and $\simeq 0$ for the FalseKB as shown in the Expression 4.5, we can state the Domain-Specific Knowledge Base Construction is correct.

Experimentation

For the testing of the KBC validation, the three used sub-tasks to generate triples, namely, *Information Extraction*, *Chunking Rules* and *Tagging Pattern Execution* described in 4.3.1, where taken, as T_{IE}, $T_{Chunking}$, $T_{Pattern}$, respectively for the experimentation phase.

The triples generated by using the Information Extraction (T_{IE}) are approximately 75% of the discovered triples. The remaining 25% can be split between the triples (about 10% of the total KB generated) coming from the Chunking Rules ($T_{Chunking}$) and those (around 15%) from Tagging Pattern components ($T_{Pattern}$). Overlapping statements represent 1% of the triples discovered.

Table 4.2: KBC evaluation results, exploring TransE, DistMult, ComplEX. src is the reference data; Val. is the data to validate the Src. All the KB are the final T_{RES} for each process

Model	Src.	Val.	MRR	H@1	H@3
TransE	T_{KB}	ST_{Pos}	0.95	0.92	0.99
	T_{KB}	ST_{Neg}	0.01	0.01	0.01
DistMult	T_{KB}	ST_{Pos}	0.92	0.85	0.99
	T_{KB}	ST_{Neg}	0.01	0.01	0.01
ComplEx	T_{KB}	ST_{Pos}	0.99	0.98	0.99
	T_{KB}	ST_{Neg}	0.01	0.01	0.01
HolE	T_{KB}	ST_{Pos}	0.93	0.86	0.99
	T_{KB}	ST_{Neg}	0.01	0.01	0.01
ConvE	T_{KB}	ST_{Pos}	0.96	0.92	0.99
	T_{KB}	ST_{Neg}	0.01	0.01	0.01

Table 4.2 shows the evaluation process results by pair-wise comparison of the statements generated by the KB generated by each subtask. The first column represents the set of reference triples used to validate

the set of triples in the second column. Then, the relative values of the metrics for each KGE model were analyzed for the five KGE models. In the table, it can be show the overall high performance of the evaluated embedding models, with some emphasis on the *ComplEx* and *ConvE* models. They deliver the best results, in particular the *ComplEx* model gets 99% on the MRR and more than 98% on the HITS@1 and HITS@3, asserting the effectiveness of higher dimensional and complex models to knowledge base validation tasks. On the other hand, the disclosed result reflects similar performance in term of MRR, HIT@1 and HIT@3 between the Geometrical models (*TransE*) and Convolutional ones (*ConvE*), as well as the High dimentional model (*DistMult*) and the Holographic one (*HolE*).

The percentage of the top one and three ranked triples (H@1, H@3) confirms that using triples from the Information Extraction task (T_{IE}) for both reference and validation often produces good triple rankings. To consider the validation of the external knowledge, the last rows of Table 4.2 shows the result of the comparison with the synthetic triples ST_{Pos} and ST_{Neg}.

The results suggest that the validation of the triples produces high values (about 0.95) of MRR for the positive ST_{Pos} triples and very close to 0 for the negative ST_{Pos} triples. The synthetic statements accurately reflect the knowledge discovered by the methodology and the correspondence implementation, and the positive statements are potential candidates to feed into the existing knowledge base.

Discussion

This work provides a general methodological approach to automatically build a knowledge base on a preferred domain given in textual resources. It allows the creation of different types of KBs. It can be leveraged to construct ad-hoc knowledge bases tailored to a specific domain. Question-answering systems can act as validation systems when verifying the veracity and reliability (degree of truth) of a given statement is required, for example, for discovering fake news or just for fact-checking. The approach accomplishes an unsupervised KG construction from specific domain, given a natural language text; in particular a case study designed on prosaic text was performed with promising results.

4.5 Synthetic triples for enhancing downstream tasks in KGE models

Synthetic triple generation is the artificial process of automatically generating new triples based on an existing KB. It aims to enrich the KG with further information that is not explicitly stated but can be computed leveraging algorithms or patterns to generate new knowledge from the existing information [185].

A popular approach within the synthetic triple generation is the construction of a negative knowledge base; it is an important task in many fields, where identifying false or negative information can be just as valuable as identifying positive information. Negative knowledge can be used in various applications, such as medical diagnosis, fraud detection, and reasoning in expert systems.

In Natural Language Understanding (NLU) disciplines, trained machine learning models are validated using expert systems, yielding the desired negative output. On the other hand, negative triple generation involves the generation of false triples or facts that are not present in the original knowledge graph[146].

To create a valid knowledge representation (KR) of a real-world scenario, it can be necessary to complement the existing data with other relations that enable the current models to comprehend and integrate the complex links in the real world. To address this challenge, many researchers have focused on the synthetic triple generation at a large scale [50], highlighting in most cases the negative sampling [174], the last ones representing false information over existing KB. Usually, synthetic triples are generated systematically to enhance the results of downstream tasks, like training or validating IA models or obtaining controlled answers.

This research introduces a synthetic triple-generation algorithm based on the complement graph. The proposed algorithm identifies the missing relations between entities, exploiting the graph's nonexistent relationships, the complement. The generated triples are validated against the original knowledge base using several knowledge graphs embedding methods and computing metrics such as MRR and HITS@N. The evaluation results demonstrate that the proposed algorithm generates high-quality synthetic triples that improve the model accuracy,

directly influencing the predictions.

Our approach provides a practical and effective solution for addressing the challenge of predicting international trade flows [186]. The results demonstrate the effectiveness of triple generation based on the complement graph in enhancing the performance of knowledge graph-based models.

This section is structured as follows: Section 4.5.1 introduces the proposed synthetic triple-generation algorithm. Then, Section 16 describes an experimentation scenario to validate the proposed method. Finally, Section 16 highlights the discussions and future work.

4.5.1 The Synthetic triple generation

This section describes a method of synthetic triple generation. Initially, a formal definition of KB and KG concepts is provided; this will serve as a precise starting point for understanding how they are used and related to key concepts to generate the synthetic triples. Then, the method is presented as an algorithm, depicted in Alg. 3. Finally, the experimentation scenario is based on KGE to confront the training impact on learning scenarios against the initial KB.

The Complement Graph

A complement of a graph G is a simple graph G$'$ having all the vertices of G and an edge between two vertices v_i and v_j if and only if there exists no edge between v_i and v_j in the original graph G. More formally,

Definition 8 *Let $G = (V, E, R)$ a graph and let K be the set of all possible triples (v_i, p_h, v_j), with $i, j = 1, \ldots, |V|$, $h = 1, \ldots, \frac{1}{2}|V| * (|V| - 1)$, by considering all the edges connecting two different vertexes in V (with $v_i \neq v_j$). Then $G' = (V, E', R')$ is a complement of the graph G where $E \cap E' = \emptyset$, and $R' = K - R$.*

Considering rule-based methods expressed in recent research [187, 188], this approach leverages the KG complement, considering all the triples not mapped into the KB as possible candidates. The method is expressed in its initial form as an algorithm and shown in Alg. 3. The algorithm receives the existing KB as input and produces the sets of

Algorithm 3: Algorithm for KG for Synthetic triple generation

Data: $G = (V, E, R)$
Initialize:
  $R' = \emptyset$
  $E' = \emptyset$
Input : $KB = \{St_1, St_2, St_3, \ldots, St_n\}$
Output : $G = (V, E', R')$
foreach $r \in R$ **do**
  $p' = synth(p)$
  add p' to (E')
  foreach $v_i \in V$ **do**
    foreach $v_j \in V$ **do**
      if $(v_i \neq v_j)$ ***AND*** (v_i, p, v_j) ***NOT*** *in* R **then**
        $r' = (v_i, p', v_j)$
        add r' to (R')
      end if
    end foreach
  end foreach
end foreach
return (R', E')

unique edges in a vector mode E' and the relations as a list of generated statements R'.

The procedure iterates over the graph entities V and queries the graph for those who do not have the relations; when a queried relationship is not found, a new candidate triple is created. The triple generation strategy aims to create missing links on the KB by mapping them as potential candidates for improving the posterior training procedures. Furthermore, the semantic component of the relation in the triple can be extracted from a given vocabulary or negate the queried relation, as shown in [189].

Finally, it is important to note that the complement graph generated by this procedure does not intend to create a final mapping on the graph as a knowledge competition task but rather an approach to leverage the

semantics in the triple relations to build a bridge between the initial KB and the training model to learn the missing important relations in the KB.

Recently, the KGE has been used in more complex frameworks like fact-checking [190], question-answering [191], and others, providing competitive results by providing similarity or distance computations leveraged in the low-dimensional vectors. However, the main reason for the embedding usage in the triple validation is the natural feature for discovering semantic relationships between the entities and concepts, making it an evaluation framework suitable for the synthetic triple generation. Several comparative experiments, leveraging KGE models, were conducted, starting from the original KB and adding synthetic triples: 1) as part of the test set and 2) as part of the train/test/validation set. Table 2.3 shows the KGE models explored in this experimentation and their respective scoring functions. Table 2.2 provides the KGE main features and additional non-functional model properties for some of the most popular KGE models.

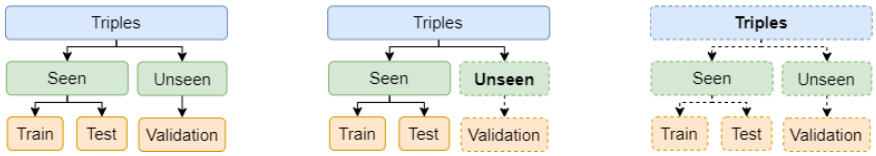
The KGE model selection criteria is based on the embedding model family, regardless of the accuracy provided by the SOTA benchmarks. The use of a specific model for each category was inspired by a recurring concern in the literature that different embeddings may be appropriate for different types of relationships [192], and given the diversity and complexity of KG, different types of modeling are needed to represent relationships in a KG [193].

To measure the impact of the generated synthetic triples in the KB training procedure, the metrics MRR and HITS@N are used. Different values of HITS have been calculated, in particular: HITS@1, HITS@10, and HITS@100. As stated, this measure represents the ability of the model to produce a correct answer in the top 1, 10, or 100 answers, respectively.

Experiment Results

The experimentation designed for the synthetic generation method uses the CEPII Gravity dataset to validate the triple generation method. The experiment achieves training, testing, and validation results in different scenarios. A traditional machine learning evaluation framework was used, by splitting the data for training, testing, and validation in

60/20/20, respectively, among the total triples to proceed with a cross-validation strategy.



(a) Baseline statements without synthetic triples - KB

(b) Synthetic triples used as the unseen set - KB+Synth1

(c) Baseline triples plus the synthetic ones - KB+Synth2

Figure 4.6: (a) Baseline triples without synthetic statements and Triple split and experimentation test cases: (b) the input triples are used to train and test the model, getting a baseline; then a similar experiment is accomplished by leveraging the synthetic triples (*Unseen*) for the validation; (c) the triples consist of the initial KB plus the synthetic triples, mixed together for training, testing and validating the model.

Figure 4.6 presents our experimental scenario. Initially, the input triples are trained and tested (as shown in Figure 4.6a to establish a baseline for a given model. The baseline approach is called *KB*. Then, a similar experiment was carried out but added the synthetic triples (*Synth1*) in the model as *Unseen* triples for the validation task to assess how the computed scores are affected by the established baseline. This scenario is called *KB+Synth1* in Figure 4.6b. The last scenario, instead, considers the input triples composed of all the initial triples from KB plus the synthetic triples (called *KB+Synth2*) mixed. Then the typical training, testing and validation approach is carried out, as shown in Figure 4.6c, with the same split ratio.

Regarding the model parameters, the setup for all scenarios was $k = 150$, $eta = 10$, optimizer *Adam* with a learning rate of 1^{-3} and a loss function *pairwise*. Let us notice that the model parameters were not tuned during testing. The purpose was to examine the impact of the synthetic KB in the training operation. In addition, a GitHub repository was prepared in which some additional resources were posted[4].

[4]GitHub repository - `https://github.com/d1egoprog/SyntheticTrip`

Table 4.3: Evaluation results using the KGE confronting the two synthetic sampling scenarios versus the initial knowledge base training score.

MRR			
Model	**KB**	**KB+Synt1**	**KB+Synt2**
TransE	0.1023	0.1086	**0.3789**
DistMult	0.195	0.2243	**0.8320**
ComplEx	0.5699	**0.6928**	0.625
HolE	0.6132	**0.6802**	0.5934

Table 4.3 shows the training results without using synthetic triples; it is possible to observe the MRR score difference between the "simple" embedding type, such as the Translational one (TransE), and more complex models, in terms of dimensionality, computing complexity and score function.

The initial version of the synthetic triple generation method has a positive impact on the training results, with MRR and $HITS$ increasing in most cases for each N. As Table 4.4 shows, we can see that the evaluation metrics, MRR and H@N, are higher for the knowledge base (KB) integrated with the synthetic triples generated by both Synt1 and Synt2 than the training score of the initial knowledge base.

These results indicate that the performance of all four embedding models improved when the original knowledge base was supplemented with synthetic triples generated by Synt1 and Synt2. In particular, the TransE and DistMult models had the highest MRR score with Synt2, while the ComplEx and HolE models were shown to have improved MRR and H@N scores with both synthetic sampling methods. Increasing synthetic data seems to be a good technique to improve the performance of knowledge graph embedding models.

Discussion This section was devoted to presenting a synthetic triple-generation method, leveraging the missing relations on the graph as a complement. The added triples show improvement in all the computed scores by all of the selected KGE models.

---

les

Table 4.4: Evaluation results using the KGE confronting the two synthetic sampling scenarios versus the initial knowledge base training score.

Model	KB	KB+Synt1	KB+Synt2
H@1			
TransE	0.0437	0.0430	**0.1294**
DistMult	0.0688	0.0967	**0.7693**
ComplEx	0.4835	**0.6196**	0.4841
HolE	0.5458	**0.6254**	0.4314
H@10			
TransE	0.2013	0.2407	**0.7764**
DistMult	0.5193	0.5870	**0.9466**
ComplEx	0.7371	0.8367	**0.8660**
HolE	0.7378	0.7822	**0.8629**
H@100			
TransE	0.9348	0.9534	**0.9957**
DistMult	0.9871	0.9814	**0.9998**
ComplEx	0.9642	0.9814	**0.9988**
HolE	0.9678	0.9606	**0.9978**

In this study, we investigated the effectiveness of generating synthetic triples to improve the performance of knowledge graph embedding models. Our results show that incorporating synthetic triples generated by the proposed method and in the scenarios tested, both Synt1 and Synt2, significantly improves evaluation metrics, particularly MRR and HITS@N, for all four embedding models evaluated. Possible future work will be aimed at reducing the complexity of the algorithm and evaluating it with state-of-the-art benchmarks to compare the impact on learning downstream tasks.

As a multi-dimensional problem, bilateral trade allows for exploring baseline solutions such as those presented in this research using real data. A future extension is to take this initial synthetic generation method and improve it by elevating the triple quality by capturing the different

semantic relations and exploring more profound strategies, implementing additional techniques such as incorporating external knowledge sources, vocabularies, or more straightforward approaches such as reinforcement learning or others.

4.6 Gravity Knowledge Representation for enhancing prediction tasks

Knowledge Base Construction (KBC) leverages available data to create representative features with the sole intention of arriving at a stable KR model able to capture real-world characteristics. The quality of a KB or a KG, in our context, depends on the < subject, predicate, object > triples, and it is strictly affected by the relation connecting between entities. On the other hand, the gravity model is a mathematical model widely used in international economics to predict bilateral trade flows between two countries. It is used to measure the commercial trade synergies between two countries using specific economic indicators such as GDP, GDP per capita, etc. There are some affinities between the gravity model and a KG: a relation between two entities can be seen as a flow between two locations. Starting from this observation, this approach aims at modeling the KG construction, exploiting the gravity model to represent trade flows (gravitation interactions among entities) by triples.

Several recent studies have attempted to apply machine learning techniques to predict international trade flows. However, most studies are done on a limited number of products and countries. The availability of the bilateral trade dataset and modern machine learning tools provides a unique opportunity to understand and predict international trade flows to inform decision-making by governments, policymakers, and businesses.

The KG approach is particularly relevant to studying international trade, a significant cornerstone of economic and social development in the globalized economy [194, 195]. The world of international trade is complex and interconnected, with multiple entities (commodities, companies, and countries) interacting in multiple ways [196]. This

method helps to understand those complex interactions in a structured and intuitive way.

This section introduces an approach for addressing International Trade modelling by leveraging knowledge graphs' natural characteristics of exploiting contextual and relational information. The novelty of this work is not only the construction of the KG for link prediction using embeddings but, by leveraging the gravity model for knowledge base construction, the embeddings thus generated for that KB can be used to feed traditional methods such as decision trees or neural graph networks, improving performance and prediction results.

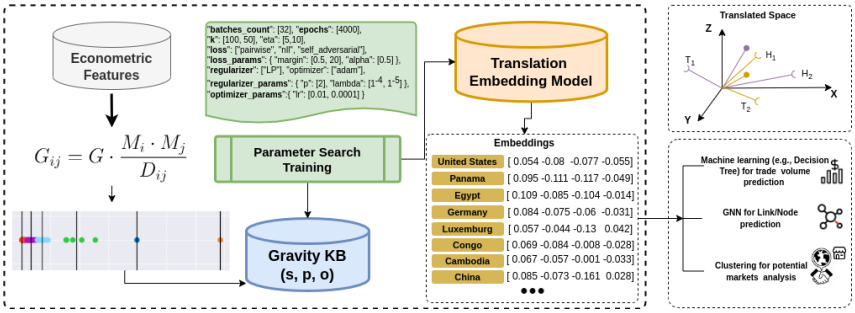


Figure 4.7: Gravity-based Knowledge Base Construction method presents the data flow at a High-level. The resulting embeddings can be exported to feed additional IA methods such as Decision trees and GNN

A high-level overview of the Gravity-based KBC model is shown in Fig. 4.7. The method exploits certain behaviours of gravitational interaction G_{ij} between two entities (S and O) in a KB.

Then, a classification of the value ranges of gravitational forces are carried out (applying a straightforward clustering algorithm). The resulting partitioning allows us to label the predicate p in our triples $< s, p, o >$. So computed triples compound the initial Knowledge Graph (KG), according to the definition provided in Section 2.3, or Gravity Knowledge Base (*Gravity KB*), as shown in Fig. 4.7.

Once the KG is built, the *Embeddings Computation* component accomplishes a parameter search optimization task that allows the *Resulting Embeddings* component to generate, by translational space-based methods, a low-dimensional vector entity representation; these vectors

are projected in the final translated space for an explainability purpose (*Explainable Representation*). Each pipeline component/task is detailed in the following subsections. The resulting embedding is used to feed the traditional machine-learning models (*Knowledge Exploitation* component) to leverage the intrinsic information revealed by the embedding and enhance the prediction capability of the models.

In the remaining of this section, a gravity-enhanced KBC model for vector representation exploitation is described. Section 4.6.1 presents the gravity-inspired knowledge base construction method. Then, a comparative analysis of the influence of embedding methods as an input for other IA methods, such as GNN and Decision Trees, is provided (Section 4.6.2). Also, a projection of the embeddings in a multi-dimensional space provides a preliminary tool for explainability and the interpretation of the knowledge model (Section 4.6.3).

4.6.1 Gravity-oriented Knowledge Graph Construction

By integrating the gravity model into the knowledge base construction process, the method captures the essential factors influencing trade relationships and allows for predicting trade patterns.

The model takes as input relevant national economic Key Performance Indicators (KPI) extracted from the CEPII dataset, described in Section 4.3.4. The *Knowledge Graph Construction* component is in charge of processing these features to evaluate the gravity score (using Eq. 4.6) among entity pairs s and o and generate a triple-based representation $< s, p, o >$, according to the evaluated gravity score p.

Gravity Model adaptation

The gravity model is a mathematical model widely used in international trade analysis to predict and explain bilateral trade flows between two countries. It is used in various fields, including economics, transportation planning, and international relations, to explain and predict the flow of goods or information between different locations. Formally, it describes the attraction force of G_{ij} between two entities in space, e_i and $e_j \in E = \{e_1, e_2, e_3, \ldots, e_n\}$ the set of entities:

$$G_{ij} = G \cdot \frac{M_i \cdot M_j}{D_{ij}} \tag{4.6}$$

where M_i and M_j are the masses of the entities e_i and e_j respectively, and D_{ij} is the distance between them, and G a gravitational constant. This model provides a robust empirical relationship with bilateral trade flows and is widely used in many trade datasets.

The CEPII Gravity Dataset [197], for instance, is a comprehensive resource extensively employed in gravity model analyses, a principal method for understanding and examining bilateral trade patterns. This dataset encapsulates various variables, facilitating a deeper understanding of international trade dynamics.

Considering the notions about *triple* and *KG* provided in Section 2.3, our model maps the triple $T =< s, p, o >$ establishing a direct relationship between two entities $s \rightarrow e_i$ and $o \rightarrow e_j$, obtaining the following final triple expression $T =< e_i, G_{ij}, e_j >$, where G_{ij} is the attraction force, as given in Eq. 4.6.

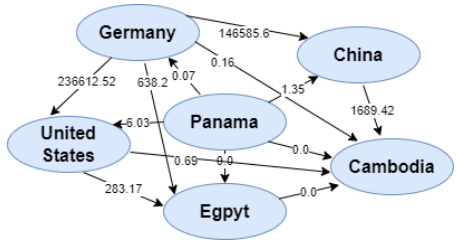


Figure 4.8: Graph example with countries as entities, their respective gravity values computed from trade flows and distance as relations

Figure 4.8 presents a trade flow-based graph with countries as nodes and gravity scores as edges between node pairs.

As mentioned, the edge or relationship between nodes in the graph represents a gravity score, that is, a numerical value. An automatic clustering-based process allows score partitioning by simply mapping value ranges into categories or classes to avoid manually classifying these gravity forces among countries.

Clustering gravitational forces among entities

The numerical values of the gravity score could be very tricky to model in the graph-based representation, where the relation between subject and object is a label and describes a kind of correlation among entities. Defining as many classes (or categories) as the score values could not be effective for the KGE process. To deal with this issue and avoid bias in the categorization, a clustering method is applied to the gravity scores, yielding a partitioning whose clusters can be mapped to some categories.

To this purpose, in the range of clustering families [198], five clustering methods were considered were carried out. In Table 4.5, the selected algorithm and the families to which they belong are mentioned.

Table 4.5: Implemented Clustering methods with family type and suited designed dataset constraints.

Method	**Family**	**Suited Data**
K-means	Centroid-Based	Large even datasets
Dendogram	Hierarchical	Small and Medium datasets
DBSCAN	Density-based	Large with arbitrary shapes
Gaussian Mix	Model-based	Normally Distributed Data
Mean Shift	Kernel-based	No particular distribution

The algorithms cover various requirements regarding data distribution, complexity, and dataset size. Being the simplest to understand and implement, K-means commonly assumes spherical clusters of equal size; this method is designed for evenly distributed large datasets. On the other hand, Hierarchical clustering, such as dendrogram, does not have any assumptions on shapes and sizes and, therefore, performs well on unevenly distributed clusters; besides, it is computationally expensive, making it optimal to work with small to medium-sized datasets. DBSCAN is a density-based technique that finds clusters of any shape and size but requires tuning for parameters set, in contrast with Dendogram, which performs well with large datasets in arbitrary cluster shapes and sizes. Finally, Mean-Shift follows a more unattended approach with no cluster assumption regarding shape and size, automatically deciding the number of clusters; it is computationally expensive and performs well

with small to medium-sized datasets with non-linear cluster shapes.

4.6.2 Leveraging Gravity Embedding as input Features

Knowledge Graph embeddings (KGE) are typically used for downstream tasks like link prediction and entity recognition. However, recent research on NeuroSymbolic IA trends[15, 1] shows the embeddings as an additional step into the KR procedure and not a final task.

Energy-based models are founded on the hypothesis of benefiting from a given function $E(x)$ by assigning low energy values to some inputs and high energy values to others [199]. Similarly, the translational space embedding model, also known as TransE [72], is created with the perception that the head entity h is close to the tail entity t embeddings through the relationship l. TransE exploits the hierarchical relationship concept, meaning the distance between two connected entities through a relationship is small since they share similar attributes. The loss function for the TransE model [72] is described as follows:

$$\mathcal{L} = \sum_{(h,l,t)\in S} \sum_{(h',l',t')\in S'_{(h,l,t)}} [\gamma + d(h+l,t) - d(h'+l,t')]_+ \tag{4.7}$$

$$S'_{(h,l,t)} = \{(h',l,t)|h \in \mathcal{E}\} \cup \{(h,l,t')|t' \in \mathcal{E}\}. \tag{4.8}$$

Where S is the set of positive triples in the training set, S' is the negative triples, defined in Eq. 4.8 shadowing *head* or *tail* and $d(h+l,t)$ is the distance measure between the entities.

TransE provides a simple but powerful framework for capturing semantic associations between entities and relationships in a KG, in contrast to more advanced models such as DistMult [73] and ComplEx [74], which use high-dimensional spaces; HolE [75] which uses circular correlation in tensor spaces; RotatE [200] employs complex-valued rotation matrices, or the convolutional approach of ConvE [76].

TransE provides a simple but powerful framework for capturing semantic associations between entities and relationships in a KG, in contrast to more advanced models such as DistMult [73] and ComplEx [74], which use high-dimensional spaces; HolE [75] which uses circular

correlation in tensor spaces; RotatE [200] employs complex-valued rotation matrices, or the convolutional approach of ConvE [76].

During the experimental setup of the machine learning models, we performed a comparative analysis to investigate the impact of utilizing the embedding components of both origin and destination countries, which gravity-based KBC created. The models were trained on the UN Comtrade and BACII datasets, consisting of various features such as year, month, GDP, harmonic distance, and commodity code, to forecast the trade volume between the two countries.

Decision Trees

The candidate model for our first experience was the Decision Tree (DT) model. The model creates a tree structure that captures the relationships between the input features and the target variable. Thanks to the generation of the tree structure, it is easily understood, allowing users to gain insights into the decision-making process.

Fig. 4.9 shows an example decision tree model for predicting the trade volume between the United States and Luxembourg.

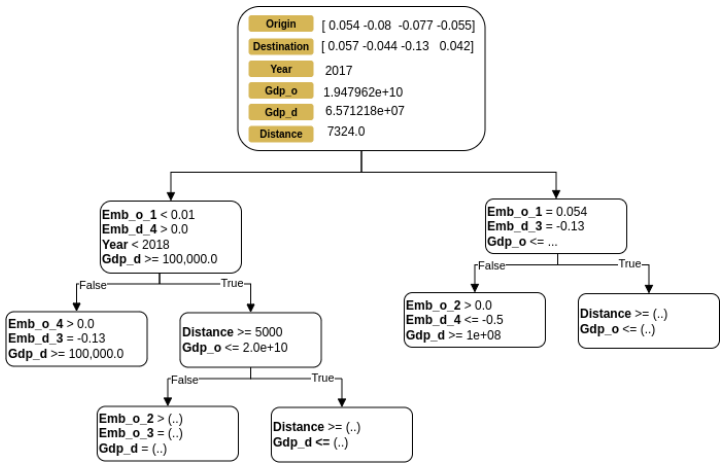


Figure 4.9: Decision Tree model using the KBC embedding results for trade volume prediction between USA and LUX example.

The model uses the embedding components of origin and destination countries created by gravity-based KBC and several other features,

including the year, GDP, distance, and commodity code, to predict the trade volume. The decision tree is structured with a series of nodes that split the data based on the values of these features. At each node, the model chooses the feature that provides the most information gain for the prediction task. The tree's leaf nodes contain the predicted trade volume for each combination of feature values. This decision tree model can predict the trade volume between the United States and Luxembourg for any combination of feature values by traversing the tree from the root to a leaf node and using the corresponding predicted value.

Table 4.6: Experimental setup parameters.

Parameter	**Value**
Dataset size	367, 315, 312
Reporter codes	140
Partner codes	247
Commodity codes	5,204
Train/Test	80%/20%
Decision tree	Max depth: 50
Trade flow	Exports

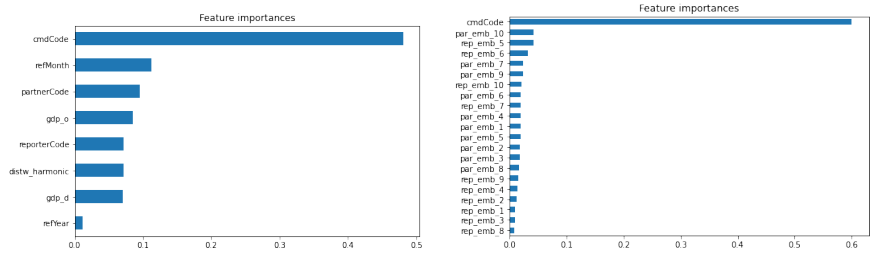
Table 4.7: Decision Tree performance evaluation with Basic Features (BF) and Embedding Features (EF).

Metric	**MAE**	**MAPE**	**MPE**	**R-square**
BF	0.6745	633,799.3726	-633,784.5952	0.5276
Log BF	0.3661	**18,981.6164**	**-18,951.7242**	0.5143
EF	0.3554	580,450.3465	-580,438.4712	0.5564
Log EF	**0.1229**	657,425.2525	-546,900.2780	**0.6826**

Table 4.7 presents the performance evaluation of the decision tree model with different feature variations based on various metrics. The first variation experiment used basic features, including trade value, country codes, year, month, commodity code, harmonic distance, and

GDP. A logarithm function was applied to the log variation's trade value, harmonic distance, and GDP. Both countries' embedding vectors were used alongside the commodity code for training to predict the trade value of embedding features and embedding features with the log. The evaluation used four metrics: MAE, MAPE, MPE, and R-square. The results indicate that the embedding features of the log model outperformed all other models in terms of MAE and R-square, which suggests that this model has the smallest prediction errors and the highest goodness of fit. On the other hand, the basic features model had the highest MAPE and MPE values, indicating that this model's predictions deviated most significantly from the actual values.

Overall, the evaluation suggests that the embedding features with the log model are the best-performing model, indicating that Gravity-based KBC embedding features and log transformation can significantly improve the model's performance in predicting the target variable. Therefore, it can be concluded that incorporating embedding features and log transformations can enhance different machine learning models' accuracy in predicting trade value.



(a) Basic features (b) Embeddings features

Figure 4.10: Decision Tree feature importance.

Figure 4.10 shows the feature importance results of the two decision tree models with log transformation trained with basic features and embedding features, respectively. The DT basic features model shows that the commodity code is the most critical feature with an importance value of 0.48, followed by the partner code with 0.10, and the reference month with 0.11. However, other features, such as the reporter code, harmonic distance, and GDP, have relatively low importance values compared to the commodity code. In contrast, the DT with embeddings

model indicates that the commodity code is still the most crucial feature, with an importance value of 0.60. Embedding features, however, have a higher importance value than basic features, ranging from 0.01 to 0.04 for the ten embedding features. This suggests that the embedding features are crucial to improving the model's performance. Additionally, the embedding features for the partner and reporter codes have similar importance values, with the top features having similar importance values. These feature importance results suggest that incorporating embedded features improves the model's performance in predicting trade value. The embedding features capture more detailed information about the partner and reporter countries, leading to better prediction accuracy. However, the commodity code remains the most critical feature in predicting the trade value, regardless of the type of features used.

Graph Neural Networks

Figure 4.11 represents the results obtained when gauging the MSE loss function for the GNN-based link prediction tasks while training the original gravity dataset and another training using our embedding features. The MSE values reported in this figure indicate that the GNN regression model based on embedding features has quickly decreased to 0; therefore, it has significantly outperformed the GNN model based on the basic features.

On the other hand, Figure 4.12a represents the confusion matrix of the GNN model with basic features that provide interesting insights. It achieved a true positive rate (TP) of 83.3%, meaning it correctly identified a significant portion of true links. However, the false positive rate (FP) was 16.7%, indicating a tendency to identify non-existent links as positive mistakenly. Moreover, the false negative rate (FN) was 22.2%, implying that the model failed to recognize some true links. On the positive side, the true negative rate (TN) reached 77.8%, denoting a reasonably high accuracy in identifying non-links.

In contrast, Figure 4.12b, which represents the confusion matrix of the GNN model with embedding features, demonstrated substantial improvements. It achieved a TP rate of 97.4%, indicating its ability to predict true links accurately. The FP rate dropped to 2.6%, reflecting a reduced tendency to identify non-existent links incorrectly. However, the FN rate remained at 13.7%, suggesting that some true links were

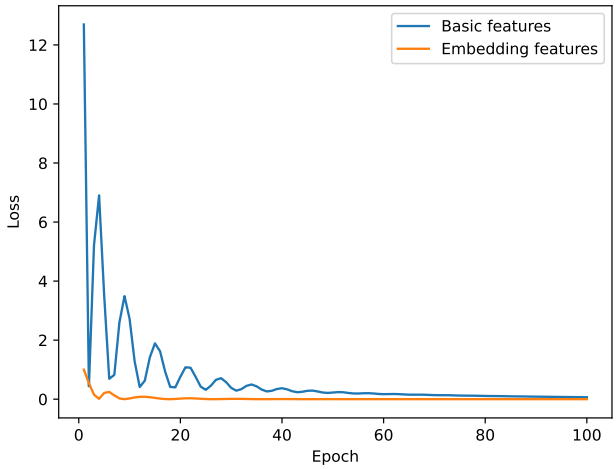


Figure 4.11: MSE Loss function for GNN-based Link Prediction

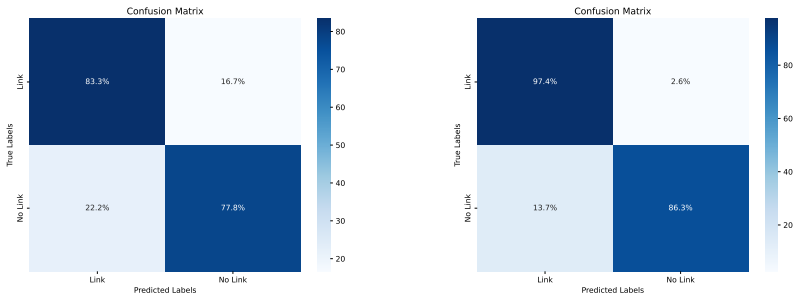
still missed. Nevertheless, the TN rate rose to 86.3%, highlighting the model's improved accuracy in identifying non-links. It is also shown in table 4.8, which represents the accuracy measured for two models, indicates that the GNN model based on our embedding features has a much higher accuracy value, with a mean of almost 0.9185 against the other model.

Comparing the two models, it is clear that the GNN model based on embedding features outperformed the one based on basic features. The embedding-based model's higher TP and TN rates signify improved accuracy in predicting both positive and negative links. Although the FN rate is still relatively higher than the TP rate, it is significantly lower than that of the basic features model.

Table 4.8: Accuracy for link prediction task

	Accuracy
Basic features	0.8055
Embedding features	0.9185

Overall, the fact that the GNN regression model, based on the em-



(a) with basic features (b) with embedding features

Figure 4.12: Confusion Matrix for GNN-based Link Prediction.

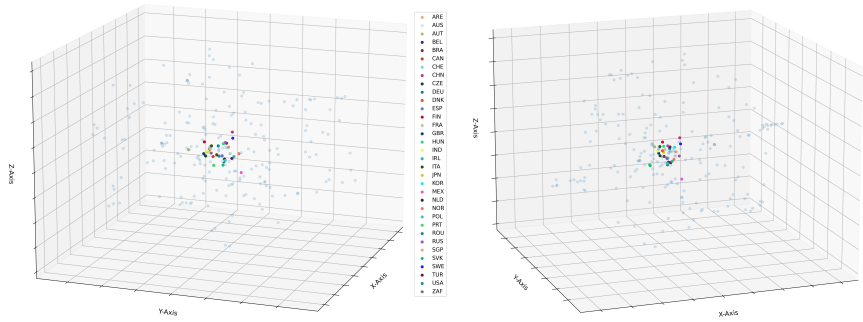
bedding features, outperforms the original dataset regarding predictive accuracy suggests that a model is a promising approach for the given task.

4.6.3 Towards an Explainability approach for KGE

In large-scale and real-world decisions, understanding how an IA model makes decisions is a mandatory requirement; this is why explainability approaches are becoming more popular every day. For this scenario, the space representations displayed in Fig. 4.13 provide a comprehensive view of the previous gravity scores and the relations between the nations.

Two three-dimensional plots of the same embeddings are shown from two different perspectives to grasp the depth perception and get the different ”distances” between entities.

In Figure 4.13, the international trade closeness of Germany (DEU) for 2019 is portrayed. The primary trade partners include, for example, the United States (USA), France (FRA), China (CHN), the Netherlands (NLD), the United Kingdom (GBR), Italy (ITA), Poland (POL), and Austria (AUT). This illustrative figure also provides a portrayal of the interconnectedness amongst these principal trade partners as well as with other economies globally. This three-dimensional representation facilitates an in-depth comprehension of the global trade ecosystem from Germany’s perspective. It also unveils the primary clusters of



(a) Left View (b) Rigth View

Figure 4.13: Isometric view of translational embeddings model projection in a 3D space

countries with which Germany does most of its trade transactions and how those countries are connected to other countries around the globe through trade.

This way, our graphical representation visually displays clusters or groups of countries participating in substantial trade relationships. The visualization provides a platform to comprehend the formation of global economic blocs based on trade interactions. It allows for straightforward and accessible recognition of key players and hubs within the trade network, thereby enabling the identification of potent regional economies and influential trading nations.

Moreover, a comparison of these maps with analogous graphical representations from previous years enhances our understanding of the shifting landscape of international trade. Such a temporal analysis sheds light on the dynamism inherent in trade relations, tracing the evolution of trading blocs and transformations in trade patterns over the years.

Additionally, using our predictive modelling can add a prospective dimension to this analysis as we can anticipate future trade trends and shifts in global trade configurations. The subsequent comparison of the projected trade network with the existing network would provide an intriguing exploration of international trade's potential trajectory, contributing significantly to our understanding of global economic development and growth patterns.

Moreover, an added value of our work lies in the capability to create more product-centric representations, be it at the level of the detailed six-digit Harmonized System (HS6) classification, the broader four-digit (HS4) or two-digit (HS2) categories, or any custom groupings derived from HS6 codes such as green trade or circular economy-associated goods. This specificity bears particular relevance for corporate entities or cluster organizations for whom comprehensive global trade maps may appear overly generalized.

Discussion The integration of the gravity model into the construction process of KGs enables the representation of essential factors influencing trade relationships and facilitates the prediction of future trade patterns.

The research findings highlight the novelty of using KG embeddings not only for link prediction purposes and demonstrate the potential of combining KG embeddings with traditional machine learning methods such as decision trees and GNNs. The results indicate improved prediction accuracy and additionally provide a step towards embedding explainability for knowledge representation purposes. The three-dimensional visualization of the generated embeddings offers a valuable tool for understanding and interpreting the underlying trade dynamics.

In conclusion, this research demonstrates combining KG embeddings and traditional machine learning methods shows promise for improving prediction performance. The findings contribute to the field of knowledge representation and offer practical implications for policymakers and stakeholders in navigating the complex landscape of international trade. By leveraging KGs and machine learning, we can gain deeper insights into trade patterns, anticipate the effects of policy changes, and make more informed decisions to foster economic and social development in the globalized economy.

Nevertheless, there are several avenues for future research. Firstly, exploring alternative embedding techniques and incorporating additional domain-specific features could enhance prediction accuracy. Secondly, investigating the interpretability and explainability of KG embeddings, for instance, in the context of international trade, would provide valuable insights and build trust in the decision-making process. Moreover, extending the approach to incorporate more diverse data types, such as transport, economic, and risk indicators, would enrich the knowledge representation and capture a comprehensive view of trade dynamics.

Chapter 5

Conclusions and Future Work

This section provides a comprehensive overview of the findings and elaborates on the future directions of this applied research thesis. It also summarizes the key results from the research work and includes a detailed review of publications that have had an impact.

5.1 Conclusions

This thesis work provides a comprehensive overview of the field of NeuroSymbolic AI regarding Knowledge Graph construction techniques from two diametrically opposed perspectives. On the one hand, traditional Semantic Web (SWT) technologies, starting from the first principles of Linked Data to powerful knowledge engines, gave rise to a vibrant search environment for describing human knowledge and making it freely available on the Web; on the other hand, Knowledge graph embeddings, a low-dimensional inductive learning technique for leveraging learning in interconnected data models, i.e., Knowledge Graphs (KGs).

Besides the extensive practical implications and real-life applications of this thesis, as demonstrated through the contributions. The aim of this thesis is to go beyond the conventional use of methodologies and approaches, and the research further advances the field of AI by providing a fresh perspective on frequently used techniques. This thesis

not only contributes to bringing down high-level research concepts and models to a more practical and real-life setting but also pushes the boundaries of theoretical AI research, offering novel insights and a refined understanding of the capabilities in what was considered a stable field.

5.1.1 Uptake from the Traditional Semantic work line

The integration of NeuroSymbolic AI for KBC, covered in the chapter 3, is designed for event detection by exploiting semantic annotation of data: a semantically enhanced framework for seismic event detection from data acquired from seismic sensor networks was presented in the chapter.

Besides ontology modeling, the framework accomplishes the classification/detection of seismic events to feed a knowledge base of new triple-based assertions, proposing a semantic ingestion data pattern from ML approaches and a new SWeMlS architecture pattern. The obtained results highlight the effectiveness of the proposed framework in terms of flexibility and high accuracy in integrating the two learning techniques. The synergy between Artificial Intelligence techniques and Semantic Web Technologies provides a solid and formal methodological infrastructure to perform specific and, at the same time, predictions in complex tasks.

In the smart manufacturing scenario, described in Section 3.7.3, the intention was to develop the closest to an industry-grade solution that brings value into an existing production line by implementing cutting-edge approaches and delivering a product complying with the standards of Industry 4.0. With the development of this platform, it was possible to demonstrate that a fast, semantic data-enhanced ecosystem can be installed in manufacturing and restrictive spaces and help add value to the quality control process.

5.1.2 Embeddings work line Uptakes

From the vector representation (embeddings), in Section 4.4, the construction of a domain-specific unsupervised knowledge graph was accomplished, given a natural language text. In particular, a case study

of a narrative text was designed, exploiting the special features of a prosaic text. The triples are semantically annotated by leveraging Semantic Web technologies to be linked and navigable in the Linked Data Cloud (LODCloud). The newly constructed KG can be processed by Knowledge Graph Embedding (KGE) models that exploit the potential of the collected triples by verifying and validating the knowledge thus generated and comparing it with the negative triples, providing the expected output from the proposed experimentation framework. In addition, performance is evaluated by cross-validation, scoring functions, and accuracy metrics on unseen triples.

The synthetic triple-generation method in Section 4.5 leverages the missing relations on the graph, defining a complement graph. These new triples show improvement in the mean rank computed from more than one KGE model and from HITS@N. The study investigated the effectiveness of synthetic triple generation for improving the performance in the knowledge graph embedding models. The results presented show that incorporating synthetic triples generated by the proposed method and in both scenarios tested leads to significant improvements in evaluation metrics, particularly MRR and HITS@N, for all four embedding models evaluated.

The contribution presented in Section 4.6 is the combination of using KG embeddings with traditional machine learning methods such as decision trees and GNNs. The results indicate improved prediction accuracy. The findings contribute to the field of knowledge representation and offer practical implications for policymakers and stakeholders in navigating the complex landscape of the chosen use case, i.e., international trade. By leveraging KGs and machine learning, we can gain deeper insights into trade patterns, anticipate the effects of policy changes, and make more informed decisions to foster economic and social development in the globalized economy. In particular, the integration of knowledge graphs with the gravity model reveals novel potential for constructing robust knowledge bases. As a final note in Section 4.6.3, a step towards embedding explainability for knowledge representation purposes was taken. The three-dimensional visualization of the generated gravity-influenced embeddings offers a valuable tool for understanding and interpreting the underlying trade dynamics.

5.2 Impacted Publications Review

Regarding publication in scientific and peer-review scenarios, this thesis covers eight (8) publications in various high-impact journals and top-level conferences. The published works are detailed in the following way.

Four (4) publications are devoted to the contributions mentioned in section 3, being discriminated in two journals, which highlights the "*IEEE Internet Of Things Journal*", with an *IF 10.2* with the invited paper "*Semantically Enhanced IoT-Oriented Seismic Event Detection: An Application to Colima and Vesuvius Volcanoes*" and two conferences, where it remarks, the International Semantic Web Conference (ISWC) holding the reputation of being the most prestigious conference in the semantic Web community globally, scoring *A1* for the Qualis ranking, in this venue one the Smart Manufacturing use case paper was presented with the title "*Enabling a Semantic Sensor Knowledge Approach for Quality Control Support in Cleanrooms*".

From the Vector Representation workstream stream, described in Chapter 4, an additional four (4) publications were made, being distributed into two journals, remaking the work presented in "*Journal of King Saud University - Computer and Information Sciences*, with an *IF of 9.0* with the paper "*Towards an explainability global trade model with a Gravity Embeddings approach*" and two conferences, being the European Semantic Web Conference (ESWC) the highlighted venue, being categorized as *A+* for the Qualis ranking, in this forum, the paper "*FAIR Knowledge Graph construction from text, an approach applied to fictional novels*", was presented.

5.2.1 Publications Not Related to the thesis

- Rodriguez, N., Barrero, J., & Rincon-Yanez, D. (2021). An alternative construction for the district institutional development index (IDID) of Bogota using DEA-CCR. 2021 7th Congreso Internacional de Innovacion y Tendencias En Ingenieria, CONIITI 2021, 10.1109/CONIITI53815.2021.9619602

- Rincon-Yanez, D., Gad-Elrab, M. H., Stepanova, D., Tran, K. T., Chu Xuan, C., Zhou, B., & Kharlamov, E. (2023). Addressing the

Table 5.1: Impacted (C)onferences and (J)journals, Impact Factor (IF), Publication year and Reference to the mentioned publication, ordered by publication date.)

Semantic Approaches Publications					
Name	Type	Rank	IF	Year	Reference
IEEE SSCI	C	B	–	2020	[93]
ISWC	C	A+	–	2021	[96]
IEEE IoTJ	J	Q1	10.2	2022	[94]
Applied Sciences	J	Q2	2.8	2022	[95]
Embedding Approaches Publications					
Name	Type	Rank	IF	Year	Reference
ESWC	C	A+	–	2022	[95]
Information	J	Q2	2.38	2023	[191]
KES	C	B	–	2023	[186]
JKSUCIS	J	Q1	9.0	2023	[186]

Scalability Bottleneck of Semantic Technologies at Bosch. ESWC 2023.

5.3 Contribution Summary

The following list showcases the achievements that have emerged from the research in the NeuroSymbolic AI integration for Knowledge graph Construction during the development of this thesis.

- The introduction of a NeuroSymbolic AI design pattern that extends the semantic data ingestion for reasoning purposes from the DL model, extending the event recognition task functionality.
- Design and development of an expert-validated event detection ontology schema based on the extension of SOSA and SSN standard ontologies.
- Introduction of a Feedforward Neural Network (CNN) for the event classification of semantically annotated sensor data, with capabilities of classifying background noise data.

- Implementation of a NeuroSymbolic AI system capable of semantically annotating live-stream data using the VEO ontology and classifying occurring events in a sensor network, with successful implementation in the Vulcanology Domain and Smart Manufacturing.
- Adoption of the Smart manufacturing implementation at the production level, demonstrating improvement in the production line quality by monitoring the environment in the cleanroom.
- Development and release of a technological tool for the Knowledge Graph Construction of a semantically annotated KG in RDF format from plain, prosaic text sources.
- A straightforward methodology for the knowledge graph construction from plain text based on similarity implementing techniques of synthetic triple generation and negative triple generation.
- Introduction of a synthetic triple generation algorithm, based on the concept of candidate triples, combined with KGE models, enhances the overall performance of the vector representation generation for link prediction tasks.
- Creation of the embeddings generation methodology leveraged by a physics model to be ingested in conventional machine learning models to enhance the model downstream task

The previously presented contributions in each respective line of work collectively push the NeuroSymbolic AI integration for the KGC research field forward, opening up new opportunities in different domains for performing more deep and seamless integration strategies for creating better and more efficient hybrid learning approaches.

5.4 Future Work

Initially, our NeuroSymbolic AI approach provides a forefront framework designed to be easily extended in a larger sensing domain and to various methods and classification algorithms. Regarding the specific sensor network, this approach can be augmented to include not only traditional station network approaches but also IoT-based applications and wearable devices to build a comprehensive knowledge base to support users in any event identification task.

The promising performance shown by our ML approach in background noise analysis applied to volcanoes with different activities (e.g., hydrothermal, volcanic, anthropogenic, etc.), in quiescent or in unrest phases, and with even different geological features (calderas, stratovolcano), described in Section3.7.2, makes us confident that the method can be extended to a large variety of volcanoes for extracting their fingerprints and significantly contributing to the study of the seismic/volcanic sources. Separate classification of background noise and transients opens the way to hybrid approaches that use specific classifiers integrated into decision models to employ both signals, even constructed from a dataset provided by a single seismic station. This would be successful in those areas that are not accessible to locate seismic networks or seismic arrays. Finally, our network enhanced by the NeuroSymbolic AI approach should recognize any anomalies in the background noise, thus indirectly evidencing changes in the dynamics of a volcano.

From the KG Embedding perspective, the work provides a general methodological approach to automatically build a knowledge base on a preferred domain given in text resources. This starting point allows the creation of different types of KBs, being leveraged to build ad-hoc knowledge bases tailored to specific domains supporting all sorts of Question-Answering systems that can act as validation systems when verification of the veracity and reliability (degree of truth) of a given statement is required, on a more concrete example, for fake news discovery or fact-checking. In the Knowledge Base construction, the generation and enrichment of triples by querying existing other Open Knowledge bases can be improved, along with the more challenging goal of enhancing the published tool to increase the degree of confidence in the build KB by implementing an automatic validation method in construction time.

Furthermore, regarding the synthetic triple generation method, the goal will be to reduce the complexity of the algorithm and perform evaluations with state-of-the-art benchmarks to compare the impact of learning downstream tasks on multiple KGE models. Furthermore, the quality of this initial synthetic triple-generation method can be improved by capturing different semantic relations and exploring deeper strategies by implementing additional techniques such as incorporating external knowledge sources, vocabularies, or simpler approaches such

as reinforcement learning.

Finally, the gravitational KBC approach is likely to explore alternative geometric embedding techniques and incorporate additional domain-specific features that could improve prediction accuracy. Studying the interpretability and explainability of KG embeddings in the context of international trade would provide valuable insights and enhance confidence in the decision-making process. In particular, extending the approach to incorporate more diverse types of data, such as transport, economic, and risk indicators, would enrich the knowledge representation and capture a comprehensive view of trade dynamics.

Bibliography

[1] A. Breit, L. Waltersdorfer, F. J. Ekaputra, M. Sabou, A. Ekelhart, A. Iana, H. Paulheim, J. Portisch, A. Revenko, A. ten Teije, and F. van Harmelen, "Combining Machine Learning and Semantic Web: A Systematic Mapping Study," *ACM Computing Surveys*, mar 2023.

[2] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A Review of Relational Machine Learning for Knowledge Graphs," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 11–33, jan 2016.

[3] B. Bazzanella, Y. Tzitzikas, V. Guidetti, A. Vecchia, E. Bellini, B. Sierman, D. Giaretta, S. Lambert, J. Lehtonen, and R. van Horik, "APARSEN Alliance for Permanent Access to the Records of ScienceNetwork," Project APARSEN, UK, Tech. Rep., 2013.

[4] M. Jarke, B. Neumann, Y. Vassiliou, and W. Wahlster, "KBMS Requirements of Knowledge-Based Systems," *Logic, Databases, and Artificial Intelligence*, pp. 391–195, 1978.

[5] E. Davis, "Knowledge Representation," in *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*, 2nd ed., J. D. Wright, Ed. Oxford: Elsevier, 2015, pp. 98–104.

[6] B. Abu-Salih, "Domain-specific knowledge graphs: A survey," *Journal of Network and Computer Applications*, vol. 185, p. 103076, 2021.

[7] S. Bonner, I. P. Barrett, C. Ye, R. Swiers, O. Engkvist, C. T. Hoyt, and W. L. Hamilton, "Understanding the performance of knowledge graph embeddings in drug discovery," *Artificial Intelligence in the Life Sciences*, vol. 2, p. 100036, 2022.

[8] M. Wang, J. Zhang, J. Liu, W. Hu, S. Wang, X. Li, and W. Liu, "Pdd graph: Bridging electronic medical records and biomedical knowledge graphs via entity linking," in *The Semantic Web – ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*. Springer, 2017, p. 219–227.

[9] S. K. Mohamed, A. Nounu, and V. Nováček, "Biological applications of knowledge graph embedding models," *Briefings in bioinformatics*, vol. 22, no. 2, pp. 1679–1693, 2021.

[10] M.-Y. Day, "Artificial intelligence for knowledge graphs of cryptocurrency anti-money laundering in fintech," in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. New York, NY, USA: Association for Computing Machinery, 2022, p. 439–446.

[11] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The World Wide Web Conference*. New York, NY, USA: Association for Computing Machinery, 2019, p. 3307–3313.

[12] S. Wu, M. Wang, D. Zhang, Y. Zhou, Y. Li, and Z. Wu, "Knowledge-aware dialogue generation via hierarchical infobox accessing and infobox-dialogue interaction graph network," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 3964–3970, main Track.

[13] D. Lukovnikov, A. Fischer, J. Lehmann, and S. Auer, "Neural network-based question answering over knowledge graphs on word and character level," in *Proceedings of the 26th International Conference on World Wide Web*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 1211–1220.

[14] M. Sabou, K. Höffner, S. Walter, E. Marx, R. Usbeck, J. Lehmann, and A.-C. Ngonga Ngomo, "Survey on challenges of question answering in the semantic web," *Semantic Web*, vol. 8, no. 6, p. 895–920, jan 2017.

[15] T. R. Besold, A. d'Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Kuehnberger, L. C. Lamb, D. Lowd, P. M. V. Lima, L. de Penning, G. Pinkas, H. Poon, and G. Zaverucha, "Neural-Symbolic Learning and Reasoning: A Survey and Interpretation," *Neuro-Symbolic Artificial Intelligence*, nov 2017.

[16] F. van Harmelen and A. ten Teije, "A Boxology of Design Patterns forHybrid Learningand Reasoning Systems," *Journal of Web Engineering*, vol. 18, no. 1, pp. 97–124, 2019.

[17] M. van Bekkum, M. de Boer, F. van Harmelen, A. Meyer-Vitali, and A. ten Teije, "Modular design patterns for hybrid learning and reasoning systems," *Applied Intelligence*, vol. 51, no. 9, pp. 6528–6546, sep 2021.

[18] T. Berners-Lee, "Linked Data," W3C Design Issues, jul 2006. [Online]. Available: https://www.w3.org/DesignIssues/LinkedData.html

[19] A. S. d'Avila Garcez, K. Broda, and D. M. Gabbay, "Neural-symbolic learning systems - foundations and applications," in *Perspectives in Neural Computing*, 2012.

[20] M. B. Ganapini, M. Campbell, F. Fabiano, L. Horesh, J. Lenchner, A. Loreggia, N. Mattei, F. Rossi, B. Srivastava, and K. B. Venable, "Thinking Fast and Slow in AI: The Role of Metacognition," 2023, pp. 502–509.

[21] A. McAfee and E. Brynjolfsson, "Big data: The management revolution," *Harvard Business Review*, vol. 90, no. 10, p. 4, 2012.

[22] P. Hitzler, M. . Krötzsch, and S. Rudolph, *Foundations of Semantic Web technologies*, 1st ed. CRC Press, 2010, vol. 1.

[23] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous Graph Neural Network," *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[24] L. Ehrlinger and W. Wöß, "Towards a definition of knowledge graphs," in *Procedings of 12th International Conference on Semantic Systems (SEMANTiCS2016)*, vol. 1695. CEUR-WS, 2016, pp. 1–4.

[25] J. Liu, Y. Yang, and H. He, "Multi-level semantic representation enhancement network for relationship extraction," *Neurocomputing*, vol. 403, pp. 282–293, aug 2020.

[26] B. Abu-Salih, M. Al-Tawil, I. Aljarah, H. Faris, P. Wongthongtham, K. Y. Chan, and A. Beheshti, "Relational Learning Analysis of Social Politics using Knowledge Graph Embedding," *Data Mining and Knowledge Discovery*, vol. 35, no. 4, pp. 1497–1536, jul 2021.

[27] A. Padia, K. Kalpakis, F. Ferraro, and T. Finin, "Knowledge graph fact prediction via knowledge-enriched tensor factorization," *Journal of Web Semantics*, vol. 59, p. 100497, dec 2019.

[28] S. Tiwari, F. N. Al-Aswadi, and D. Gaurav, "Recent trends in knowledge graphs: theory and practice," *Soft Computing*, vol. 25, no. 13, pp. 8337–8355, jul 2021.

[29] A. Hogan, E. Blomqvist, M. Cochez, C. D'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, *Knowledge Graphs*, ser. Synthesis Lectures on Data, Semantics, and Knowledge. Morgan & Claypool, 2021, vol. 12.

[30] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications," *IEEE Transactions on Neural Networks and Learning Systems*, no. 1, pp. 1–21, 2021.

[31] M. Krötzsch and G. Weikum, "Editorial," *Journal of Web Semantics*, vol. 37-38, pp. 53–54, mar 2016.

[32] A. Blumauer, "From Taxonomies over Ontologies to Knowledge Graphs - Semantic Web Company," jun 2016. [Online]. Available: https://semantic-web.com/from-taxonomies-over-ontologies-to-knowledge-graphs/

[33] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - The story so far," *International Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, jul 2009.

[34] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J. W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. t Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S. A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. Van Der Lei, E. Van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons, "Comment: The FAIR Guiding Principles for scientific data management and stewardship," *Scientific Data*, vol. 3, no. 1, pp. 1–9, mar 2016.

[35] N. Abbas, K. Alghamdi, M. Alinam, F. Alloatti, G. Amaral, C. D'Amato, L. Asprino, M. Beno, F. Bensmann, R. Biswas, L. Cai, R. Capshaw, V. A. Carriero, I. Celino, A. Dadoun, S. De Giorgis, H. Delva, J. Domingue, M. Dumontier, V. Emonet, M. van Erp, P. E. Arias, O. Fallatah, S. Ferrada, M. G. Ocaña, M. Georgiou, G. A. Gesese, F. Gillis-Webber, F. Giovannetti, M. G. Buey, I. Harrando, I. Heibi, V. Horta, L. Huber, F. Igne, M. Y. Jaradeh, N. Keshan, A. Koleva, B. Koteich, K. Kurniawan, M. Liu, C. Ma, L. Maas, M. Mansfield, F. Mariani, E. Marzi, S. Mesbah, M. Mistry, A. C. M. Tirado, A. Nguyen, V. B. Nguyen, A. Oelen, V. Pasqual, H. Paulheim, A. Polleres, M. Porena, J. Portisch, V. Presutti, K. Pustu-Iren, A. R. Mendez, S. Roshankish, S. Rudolph, H. Sack, A. Sakor, J. Salas, T. Schleider, M. Shi, G. Spinaci, C. Sun, T. Tietz, M. T. Dhouib, A. Umbrico, W. van den Berg, and W. Xu, "Knowledge Graphs Evolution and Preservation – A Technical Report from ISWS 2019," International Semantic Web Summer School, Tech. Rep., dec 2020.

[36] B. Lee, S. Zhang, A. Poleksic, and L. Xie, "Heterogeneous Multi-Layered Network Model for Omics Data Integration and Analysis," *Frontiers in Genetics*, vol. 0, p. 1381, jan 2020.

[37] B. Bouihi and M. Bahaj, "An UML to OWL based approach for extracting Moodle's Ontology for Social Network Analysis," *Procedia Computer Science*, vol. 148, pp. 313–322, jan 2019.

[38] D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jato, and F. Fdez-Riverola, "Web scraping technologies in an API world," *Briefings in Bioinformatics*, vol. 15, no. 5, pp. 788–797, sep 2013.

[39] C. Engel, S. Magnus, and J. Krause, "Using Twitter as source of traffic data - Concepts and experiences of the project AUSWEG," in *Procedia Computer Science*, vol. 110. Elsevier B.V., jan 2017, pp. 479–485.

[40] H. Tahaei, F. Afifi, A. Asemi, F. Zaki, and N. B. Anuar, "The rise of traffic classification in IoT networks: A survey," p. 102538, mar 2020.

[41] B. Shi and T. Weninger, "Open-World Knowledge Graph Completion," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1 SE - AAAI Technical Track: Knowledge Representation and Reasoning, apr 2018.

[42] K. Scharei, F. Heidecker, and M. Bieshaar, "Knowledge Representations in Technical Systems – A Taxonomy," *arXiv*, jan 2020.

[43] R. Rahman, B. Grau, and S. Rosset, "Graph-based relation validation method," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10180 LNAI. Springer Verlag, nov 2017, pp. 167–171.

[44] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, p. 112948, mar 2020.

[45] G. Brewka, S. Ellmauthaler, R. Gonçalves, M. Knorr, J. Leite, and J. Pührer, "Reactive multi-context systems: Heterogeneous reasoning in dynamic environments," *Artificial Intelligence*, vol. 256, pp. 68–104, mar 2018.

[46] A. Polleres, B. De Meester, P. Heyvaert, D. Arndt, A. Dimou, and R. Verborgh, "RDF graph validation using rule-based reasoning," *Semantic Web*, vol. 12, no. 1, pp. 117–142, jan 2020.

[47] Y. Shen, E. Hermans, T. Brijs, and G. Wets, "Fuzzy data envelopment analysis in composite indicator construction," *Studies in Fuzziness and Soft Computing*, vol. 309, pp. 89–100, 2014.

[48] S. Liang, K. Stockinger, T. M. de Farias, M. Anisimova, and M. Gil, "Querying knowledge graphs in natural language," *Journal of Big Data*, vol. 8, no. 1, p. 3, dec 2021.

[49] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, apr 2021.

[50] L. Nai, Y. Xia, I. G. Tanase, and H. Kim, "Exploring big graph computing, An empirical study from architectural perspective," *Journal of Parallel and Distributed Computing*, vol. 108, pp. 122–137, oct 2017.

[51] H. Paulheim, "Automatic Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods," *Semantic Web*, vol. 8, no. 3, pp. 489–508, dec 2015.

[52] C. Ré, A. A. Sadeghian, Z. Shan, J. Shin, F. Wang, S. Wu, and C. Zhang, "Feature Engineering for Knowledge Base Construction," ., jul 2014.

[53] V. Leone, G. Siragusa, L. Di Caro, and R. Navigli, "Building semantic grams of human knowledge," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 2991–3000.

[54] Y. Liu, T. Zhang, Z. Liang, H. Ji, and D. L. McGuinness, "Seq2RDF: An end-to-end application for deriving Triples from Natural Language Text," in *Proceedings of the 17th International Semantic Web Conference*. CEUR-WS, oct 2018, p. 4.

[55] A. Ratner, C. Ré, and P. Bailis, "Research for practice," *Communications of the ACM*, vol. 61, no. 11, pp. 95–97, oct 2018.

[56] G. Fenza, M. Gallo, V. Loia, D. Marino, and F. Orciuoli, "A cognitive approach based on the actionable knowledge graph for

supporting maintenance operations," in *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2020, Bari, Italy, May 27-29, 2020*. IEEE, 2020, pp. 1–7.

[57] G. Weikum and M. Theobald, "From information to knowledge: Harvesting entities and relationships from web sources," in *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. New York, New York, USA: ACM Press, 2010, pp. 65–76.

[58] T.-P. Nguyen, S. Razniewski, and G. Weikum, "Advanced Semantics for Commonsense Knowledge Extraction," in *Proceedings of the Web Conference 2021*. New York, NY, USA: ACM, apr 2021, pp. 2636–2647.

[59] T. Kliegr and O. Zamazal, "LHD 2.0: A text mining approach to typing entities in knowledge graphs," *Journal of Web Semantics*, vol. 39, pp. 47–61, aug 2016.

[60] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, and A. B. Rios-Alvarado, "OpenIE-based approach for Knowledge Graph construction from text," *Expert Systems with Applications*, vol. 113, pp. 339–355, dec 2018.

[61] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a Web of open data," in *International Semantic Web Conference*, vol. 4825. Springer, Berlin, Heidelberg, nov 2007, pp. 722–735.

[62] M. Brockmeier, Y. Liu, S. Pateer, S. Hertling, and H. Paulheim, "On-Demand and Lightweight Knowledge Graph Generation – a Demonstration with DBpedia," in *Proceedings of the Semantics 2021*, vol. 2941. CEUR-WS, jul 2021.

[63] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems*, ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, pp. 2787–2795.

[64] T. Ebisu and R. Ichise, "Generalized translation-based embedding of knowledge graph," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 941–951, may 2020.

[65] C. Wang, X. Liu, and D. Song, "Language Models are Open Knowledge Graphs," ., oct 2020.

[66] H. Turki, T. Shafee, M. A. Hadj Taieb, M. Ben Aouicha, D. Vrandečić, D. Das, and H. Hamdi, "Wikidata: A large-scale collaborative ontological medical database," *Journal of Biomedical Informatics*, vol. 99, p. 103292, nov 2019.

[67] P. N. Mendes, M. Jakob, and C. Bizer, "DBpedia: A multilingual cross-domain knowledge base," in *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012, pp. 1813–1817.

[68] V. Nastase, M. Strube, B. Börschinger, C. Zirn, and A. Elghafari, "WikiNet: A very large scale multi-lingual concept network," pp. 1015–1022, 2010.

[69] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A Core of Semantic Knowledge," in *Proceedings of the 16th international conference on World Wide Web - WWW '07*. New York, New York, USA: ACM Press, 2007, pp. 697–706.

[70] S. Fricke, "Semantic Scholar," *Journal of the Medical Library Association*, vol. 106, no. 1, jan 2018. [Online]. Available: http://jmla.pitt.edu/ojs/jmla/article/view/280

[71] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2016, pp. 855–864.

[72] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems*, ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, pp. 2787–2795.

[73] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding Entities and Relations for Learning and Inference in Knowledge Bases," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015, p. 12.

[74] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex Embeddings for Simple Link Prediction," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. PMLR, 2016, pp. 2071–2080.

[75] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1955–1961.

[76] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2018.

[77] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 2, jan 2021.

[78] Y. Du and I. Mordatch, "Implicit generation and modeling with energy-based models," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[79] G. Fenza, M. Gallo, V. Loia, D. M. Orciuoli, Francesco, and A. Volpe, "Semantic CPPS in Industry 4.0," *Advances in Intelligent Systems and Computing*, vol. 1151 AISC, pp. 1057–1068, nov 2020.

[80] X. Zou, "A Survey on Application of Knowledge Graph," *Journal of Physics: Conference Series*, vol. 1487, no. 1, p. 012016, mar 2020.

[81] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, dec 2012.

[82] A. Calabrese, M. Bevilacqua, and R. Navigli, "Fatality Killed the Cat or: BabelPic, a Multimodal Dataset for Non-Concrete Concepts," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 4680–4686.

[83] A. Moro, F. Cecconi, and R. Navigli, "Multilingual Word Sense Disambiguation and Entity Linking for Everybody," in *Proceedings of the 13th International Semantic Web Conference (ISWC 2014)*. Riva del Garda, Italy: CEUR-WS, 2014, pp. 25–28.

[84] A. Berven, O. A. Christensen, S. Moldeklev, A. L. Opdahl, and K. J. Villanger, "A knowledge-graph platform for newsrooms," *Computers in Industry*, vol. 123, p. 103321, dec 2020.

[85] G. D'aniello, M. Gaeta, F. Orciuoli, G. Sansonetti, and F. Sorgente, "Knowledge-based smart city service system," *Electronics (Switzerland)*, vol. 9, no. 6, pp. 1–22, jun 2020.

[86] R. LU, C. FEI, C. WANG, S. GAO, H. QIU, S. ZHANG, and C. CAO, "HAPE: A programmable big knowledge graph platform," *Information Sciences*, vol. 509, pp. 87–103, jan 2020.

[87] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Information Sciences*, vol. 497, pp. 38–55, sep 2019.

[88] X. Zhang and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion," *Information Processing and Management*, vol. 57, no. 2, p. 102025, mar 2020.

[89] K. Angele, D. Fensel, E. Huaman, E. Kärle, O. Panasiuk, U. Simsek, I. Toma, and A. Wahler, "Semantic Web Empowered E-Tourism," in *Handbook of e-Tourism*. Cham: Springer International Publishing, 2020, pp. 1–46.

[90] B. Gautam, O. Ramos Terrades, J. M. Pujadas-Mora, and M. Valls, "Knowledge graph based methods for record linkage," *Pattern Recognition Letters*, vol. 136, pp. 127–133, aug 2020.

[91] N. Kertkeidkachorn, R. Nararatwong, and R. Ichise, "UWKGM: A Modular Platform for Knowledge Graph Management," in *International Conference on Information and Knowledge Management, Proceedings*. New York, NY, USA: Association for Computing Machinery, oct 2020, pp. 3421–3424.

[92] S. Bonner, I. P. Barrett, C. Ye, R. Swiers, O. Engkvist, A. Bender, C. T. Hoyt, and W. Hamilton, "A Review of Biomedical Datasets Relating to Drug Discovery: A Knowledge Graph Perspective," ., feb 2021.

[93] D. Rincon-Yanez, E. D. Lauro, M. Falanga, S. Senatore, and S. Petrosino, "Towards a semantic model for IoT-based seismic event detection and classification," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, dec 2020, pp. 189–196.

[94] M. Falanga, E. De Lauro, S. Petrosino, D. Rincon-Yanez, and S. Senatore, "Semantically Enhanced IoT-Oriented Seismic Event Detection: An Application to Colima and Vesuvius Volcanoes," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9789–9803, jun 2022.

[95] D. Rincon-Yanez and S. Senatore, "FAIR Knowledge Graph construction from text, an approach applied to fictional novels," in *ESWC 2022*. Hersonissos, Greece: CEUR-WS, 2022, pp. 94–108.

[96] D. Rincon-Yanez, F. Crispoldi, D. Onorati, P. Ulpiani, G. Fenza, and S. Senatore, "Enabling a Semantic Sensor Knowledge Approach for Quality Control Support in Cleanrooms," in *ISWC 2021*, vol. 2980. CEUR-WS, 2021, pp. 1–3.

[97] A. Chatzimichail, E. Stathopoulos, D. Ntioudis, A. Tsanousa, M. Rousi, A. Mavropoulos, G. Meditskos, S. Vrochidis, and I. Kompatsiaris, "Semantic web and iot," in *Semantic Web and IoT*. Springer, Cham, 2021, pp. 3–33.

[98] S. Awadallah, D. Moure, and P. Torres-González, "An Internet of Things (IoT) Application on Volcano Monitoring," *Sensors*, vol. 19, no. 21, p. 4651, oct 2019.

[99] L. Spalazzi, G. Taccari, and A. Bernardini, "An Internet of Things ontology for earthquake emergency evaluation and response," in *2014 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, may 2014, pp. 528–534.

[100] D. Andročec, M. Novak, and D. Oreški, "Using semantic web for internet of things interoperability: A systematic review," *Int. J. Semant. Web Inf. Syst.*, vol. 14, no. 4, pp. 147–171, Oct. 2018.

[101] M. Cameranesi, D. Potena, C. Diamantini, and E. Storti, "GoAAL: An ontology for goal-oriented development of AAL environments," in *Proceedings of the ACM Symposium on Applied Computing*. Association for Computing Machinery, apr 2016, pp. 347–353.

[102] O. Berat Sezer, S. Z. Can, and E. Dogdu, "Development of a smart home ontology and the implementation of a semantic sensor network simulator: An Internet of Things approach," in *2015 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, jun 2015, pp. 12–18.

[103] M. Gaur, S. Shekarpour, A. Gyrard, and A. Sheth, "Empathi: An ontology for emergency managing and planning about hazard crisis," in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, 2019, pp. 396–403.

[104] S. Li, S. Chen, and Y. Liu, "A method of emergent event evolution reasoning based on ontology cluster and bayesian network," *IEEE Access*, vol. 7, pp. 15 230–15 238, 2019.

[105] J. O. Contreras, M. A. Ballera, and E. D. Festijo, "Ontology Learning using Hybrid Machine Learning Algorithms for Disaster Risk Management," in *Proceedings of the 2020 3rd International*

Conference on Signal Processing and Machine Learning. New York, NY, USA: ACM, oct 2020, pp. 13–20.

[106] J. Pulido, M. Aréchiga, E. M. R. Michel, G. Reyes, and V. Zobin, "On the finding process of volcano-domain ontology components using self-organizing maps," in *International Workshop on Self-Organizing Maps*. Springer, 2009, pp. 255–263.

[107] J. Myer, "The Volcanism Ontology (VO): Semantic Modeling of Volcanic Eruptions and Volcanoes," Ph.D. dissertation, Georgia State University, 2018.

[108] A. Sinha, D. Mcguinness, P. Fox, R. Raskin, K. Condie, B. Stern, B. Hanan, and D. Seber, "Towards a reference plate tectonics and volcano ontology for semantic scientific data integration," in *Geoinformatics 2007 — Data to Knowledge U.S. Geological Survey Scientific Investigations Report 2007-5199*, 01 2007, pp. 43–46.

[109] V. K. Papanikolaou, C. Z. Karakostas, and N. P. Theodoulidis, "A low-cost instrumentation system for seismic hazard assessment in urban areas," *Sensors*, vol. 21, no. 11, 2021.

[110] Q. Kong, R. Martin-Short, and R. M. Allen, "Toward Global Earthquake Early Warning with the MyShake Smartphone Seismic Network, Part 2: Understanding MyShake Performance around the World," *Seismological Research Letters*, vol. 91, no. 4, pp. 2218–2233, 05 2020.

[111] F. Finazzi, "The Earthquake Network Project: A Platform for Earthquake Early Warning, Rapid Impact Assessment, and Search and Rescue," *Frontiers in Earth Science*, vol. 14, pp. 1–7, Jul. 2020.

[112] R. Clayton, T. Heaton, M. Kohler, M. Chandy, R. Guy, and J. Bunn, "Community seismic network: A dense array to sense earthquake strong motion," *Seismological Research Letters*, vol. 86, pp. 1354–1363, 2015.

[113] Y. Wu, W. Liang, H. Mittal, W. Chao, C. Lin, B. Huang, and C. Lin, "Performance of a Low-Cost Earthquake Early Warning

System (P-Alert) during the 2016 ML 6.4 Meinong (Taiwan) Earthquake," *Seismological Research Letters*, vol. 87, no. 5, pp. 1050–1059, 08 2016.

[114] L. Braginskaya, V. Kovalevsky, A. Grigoryuk, and G. Zagorulko, "Ontological approach to information support of investigations in active seismology," in *2017 Second Russia and Pacific Conference on Computer Technology and Applications (RPC)*, 2017, pp. 25–29.

[115] G. Taccari, G. Bernardini, L. Spalazzi, M. D'Orazio, and W. Smari, "Earthquake emergencies management by means of semantic-based internet of things," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2015, vol. 151, pp. 318–327.

[116] A. M. Esposito, L. D'Auria, F. Giudicepietro, T. Caputo, and M. Martini, "Neural analysis of seismic data: applications to the monitoring of mt. vesuvius," *Annals of Geophysics*, vol. 56, no. 4, p. 0446, 2013.

[117] C. X. Ren, A. Peltier, V. Ferrazzini, B. Rouet-Leduc, P. Johnson, and F. Brenguier, "Machine learning reveals the seismic signature of eruptive behavior at piton de la fournaise volcano," *Geophysical Research Letters*, vol. 47, no. 3, p. e2019GL085523, 2020.

[118] R. Carniel, A. D. Jolly, and L. Barbui, "Analysis of phreatic events at ruapehu volcano, new zealand using a new som approach," *Journal of volcanology and geothermal research*, vol. 254, pp. 69–79, 2013.

[119] M. Masotti, S. Falsaperla, H. Langer, S. Spampinato, and R. Campanini, "Application of support vector machine to the classification of volcanic tremor at etna, italy," *Geophysical research letters*, vol. 33, no. 20, 2006.

[120] M. Titos, A. Bueno, L. García, C. Benítez, and J. C. Segura, "Classification of isolated volcano-seismic events based on inductive transfer learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, pp. 869–873, 2020.

[121] M. Titos, A. Bueno, L. García, M. C. Benítez, and J. Ibañez, "Detection and classification of continuous volcano-seismic signals with recurrent neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 1936–1948, 2019.

[122] G. Cortés, R. Arámbula, L. Gutiérrez, C. Benítez, J. Ibánez, P. Lesage, I. Alvarez, and L. Garcia, "Evaluating robustness of a hmm-based classification system of volcano-seismic events at colima and popocatepetl volcanoes," in *2009 IEEE International Geoscience and Remote Sensing Symposium*, vol. 2. IEEE, 2009, pp. II–1012.

[123] S. Scarpetta, F. Giudicepietro, E. C. Ezin, S. Petrosino, E. Del Pezzo, M. Martini, and M. Marinaro, "Automatic classification of seismic signals at Mt. Vesuvius volcano, Italy, using neural networks," *Bulletin of the Seismological Society of America*, vol. 95, no. 1, pp. 185–196, feb 2005.

[124] S. Falsaperla, S. Graziani, G. Nunnari, and S. Spampinato, "Automatic classification of volcanic earthquakes by using multi-layered neural networks," *Natural Hazards*, vol. 13, no. 3, pp. 205–228, 1996.

[125] H. Langer, S. Falsaperla, M. Masotti, R. Campanini, S. Spampinato, and A. Messina, "Synopsis of supervised and unsupervised pattern classification techniques applied to volcanic tremor data at mt etna, italy," *Geophysical Journal International*, vol. 178, no. 2, pp. 1132–1144, 2009.

[126] D. N. Green and J. Neuberg, "Waveform classification of volcanic low-frequency earthquake swarms and its implication at soufrière hills volcano, montserrat," *Journal of Volcanology and Geothermal Research*, vol. 153, no. 1-2, pp. 51–63, 2006.

[127] A. Ruano, G. Madureira, O. Barros, H. Khosravani, M. Ruano, and P. Ferreira, "Seismic detection using support vector machines," *Neurocomputing*, vol. 135, pp. 273 – 283, 2014.

[128] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D Convolutional Neural Networks and

Applications-A Survey," *Mechanical Systems and Signal Processing*, p. 20, 2019.

[129] M. Titos, A. Bueno, L. Garcia, and C. Benitez, "A deep neural networks approach to automatic recognition systems for volcano-seismic events," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 5, pp. 1533–1544, 2018.

[130] "Semantic Sensor Network Ontology - W3C Recommendation 19 October 2017," https://www.w3.org/TR/vocab-ssn/.

[131] A. Haller, K. Janowicz, S. J. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler, "The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation," *Semantic Web*, vol. 10, no. 1, pp. 9–32, jan 2018.

[132] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, "The SSN ontology of the W3C semantic sensor network incubator group," *Journal of Web Semantics*, vol. 17, pp. 25–32, dec 2012.

[133] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, and M. Lefrançois, "SOSA: A lightweight ontology for sensors, observations, samples, and actuators," *Journal of Web Semantics*, vol. 56, pp. 1–10, 2019.

[134] "Time Ontology in OWL," March 2017. [Online]. Available: https://www.w3.org/TR/owl-time/

[135] "Extensions to the Semantic Sensor Network Ontology - W3C Working Draft," January 2020. [Online]. Available: https://www.w3.org/TR/vocab-ssn-ext/

[136] F. Bianco, P. Cusano, S. Petrosino, M. Castellano, C. Buonocunto, M. Capello, and E. Del Pezzo, "Small-aperture array for seis-

mic monitoring of mt. vesuvius," *Seismological Research Letters*, vol. 76, no. 3, pp. 344–355, 2005.

[137] N. Varley, C. B. Connor, and J.-C. Komorowski, *Volcán de Colima: Portrait of a Persistently Hazardous Volcano*. Springer, 2019.

[138] T. Permana, T. Nishimura, H. Nakahara, and N. Shapiro, "Classification of volcanic tremors and earthquakes based on seismic correlation: application at Sakurajima volcano, Japan," *Geophysical Journal International*, vol. 229, no. 2, pp. 1077–1097, jan 2022.

[139] Irie and Miyake, "Capabilities of three-layered perceptrons," in *IEEE International Conference on Neural Networks*. IEEE, 1988, pp. 641–648 vol.1.

[140] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, may 2015. [Online]. Available: http://www.nature.com/articles/nature14539

[141] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, dec 2015.

[142] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[143] M. Gadermayr, A.-K. Dombrowski, B. M. Klinkhammer, P. Boor, and D. Merhof, "CNN cascades for segmenting sparse objects in gigapixel whole slide images," *Computerized Medical Imaging and Graphics*, vol. 71, pp. 40–48, jan 2019.

[144] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, p. 107398, apr 2021.

[145] F. Giudicepietro, M. Orazi, G. Scarpato, R. Peluso, L. D'Auria, P. Ricciolino, D. Lo Bascio, A. M. Esposito, G. Borriello, M. Capello, A. Caputo, C. Buonocunto, W. De Cesare, G. Vilardo,

and M. Martini, "Seismological Monitoring of Mount Vesuvius (Italy): More than a Century of Observations," *Seismological Research Letters*, vol. 81, no. 4, pp. 625–634, jul 2010.

[146] D. Rincon-Yanez, E. De Lauro, S. Petrosino, S. Senatore, and M. Falanga, "Identifying the Fingerprint of a Volcano in the Background Seismic Noise from Machine Learning-Based Approach," *Applied Sciences*, vol. 12, no. 14, p. 6835, jul 2022.

[147] C. Villalonga, H. Pomares, I. Rojas, and O. Banos, "MIMU-Wear: Ontology-based sensor selection for real-world wearable activity recognition," *Neurocomputing*, vol. 250, pp. 76–100, aug 2017.

[148] C. Johnson, Y. BenZion, H. Meng, and F. Vernon, "Identifying Different Classes of Seismic Noise Signals Using Unsupervised Learning," *Geophysical Research Letters*, vol. 47, no. 15, aug 2020.

[149] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2019, p. 105–113.

[150] D. Ruffinelli, S. Broscheit, and R. Gemulla, "You CAN teach an old dog new tricks! on training knowledge graph embeddings," in *ICLR 2020*, 2020.

[151] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–21, 2021.

[152] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: Association for Computing Machinery, 2008, p. 1247–1250.

[153] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Commun. ACM*, vol. 57, no. 10, p. 78–85, Sep 2014.

[154] K. Kolluru, V. Adlakha, S. Aggarwal, Mausam, and S. Chakrabarti, "Openie6: Iterative grid labeling and coordination analysis for open information extraction," in *EMNLP (1)*. Association for Computational Linguistics, 2020, pp. 3748–3761.

[155] J. L. Martinez-Rodriguez, I. Lopez-Arevalo, and A. B. Rios-Alvarado, "Openie-based approach for knowledge graph construction from text," *Expert Systems with Applications*, vol. 113, pp. 339–355, 2018.

[156] A. Bastos, A. Nadgeri, K. Singh, I. O. Mulang, S. Shekarpour, J. Hoffart, and M. Kaul, "Recon: Relation extraction using knowledge graph context in a graph neural network," in *Proceedings of the Web Conference 2021*. New York, NY, USA: Association for Computing Machinery, 2021, p. 1673–1685.

[157] T. Gui, Y. Zou, Q. Zhang, M. Peng, J. Fu, Z. Wei, and X. Huang, "A lexicon-based graph neural network for Chinese NER," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, nov 2019, pp. 1040–1050.

[158] P.-L. Huguet Cabot and R. Navigli, "REBEL: Relation Extraction By End-to-end Language generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 2370–2381.

[159] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun, "Neural relation extraction with selective attention over instances," in *Annual Meeting of the Association for Computational Linguistics*, 2016.

[160] X. Han, T. Gao, Y. Yao, D. Ye, Z. Liu, and M. Sun, "OpenNRE: An open and extensible toolkit for neural relation extraction," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*. Association for Computational Linguistics, 2019, pp. 169–174.

[161] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, "Representation learning of knowledge graphs with entity descriptions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016.

[162] P. Della Rocca, S. Senatore, and V. Loia, "A semantic-grained perspective of latent knowledge modeling," *Information Fusion*, vol. 36, pp. 52–67, 2017.

[163] X. Chen, N. Zhang, X. Xie, S. Deng, Y. Yao, C. Tan, F. Huang, L. Si, and H. Chen, "Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction," in *Proceedings of the ACM Web Conference 2022*. Association for Computing Machinery, 2022, p. 2778–2788.

[164] N. Zhang, X. Xu, L. Tao, H. Yu, H. Ye, X. Xie, X. Chen, Z. Li, L. Li, X. Liang, Y. Yao, S. Deng, Z. Zhang, C. Tan, F. Huang, G. Zheng, and H. Chen, "Deepke: A deep learning based knowledge extraction toolkit for knowledge base population," ., vol. abs/2201.03335, 2022.

[165] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, ser. AAAI'10. AAAI Press, 2010, p. 1306–1313.

[166] *WordNet: An Electronic Lexical Database*, ser. Language, Speech, and Communication. Cambridge, MA: MIT Press, 1998.

[167] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proceedings of the 16th International Conference on World Wide Web*. Association for Computing Machinery, 2007, p. 697–706.

[168] B. Jagvaral, W.-K. Lee, J.-S. Roh, M.-S. Kim, and Y.-T. Park, "Path-based reasoning approach for knowledge graph completion using cnn-bilstm with attention mechanism," *Expert Systems with Applications*, vol. 142, p. 112960, 2020.

[169] N. Jain, T.-K. Tran, M. H. Gad-Elrab, and D. Stepanova, "Improving knowledge graph embeddings with ontological reasoning,"

in *The Semantic Web – ISWC 2021*, A. Hotho, E. Blomqvist, S. Dietze, A. Fokoue, Y. Ding, P. Barnaghi, A. Haller, M. Dragoni, and H. Alani, Eds. Cham: Springer International Publishing, 2021, pp. 410–426.

[170] W. Chen, Y. Cao, F. Feng, X. He, and Y. Zhang, "Explainable sparse knowledge graph completion via high-order graph reasoning network," *ArXiv*, vol. abs/2207.07503, 2022.

[171] X. Chen, C. Liang, A. W. Yu, D. Song, and D. Zhou, "Compositional generalization via neural-symbolic stack machines," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2020.

[172] I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi, and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, jul 2020, pp. 6901–6914.

[173] H. Ren, H. Dai, B. Dai, X. Chen, M. Yasunaga, H. Sun, D. Schuurmans, J. Leskovec, and D. Zhou, "Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8959–8970.

[174] Y. Wang, F. Ma, and J. Gao, "Efficient Knowledge Graph Validation via Cross-Graph Representation Learning," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: ACM, oct 2020, pp. 1595–1604.

[175] D. Roth and W.-t. Yih, "A linear programming formulation for global inference in natural language tasks," in *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*. Association for Computational Linguistics, 2004, pp. 1–8.

[176] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun, "DocRED: A large-scale document-level relation extraction dataset," in *Proceedings of ACL 2019*, 2019.

[177] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, and L. Toldo, "Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports," *Journal of Biomedical Informatics*, vol. 45, pp. 885–892, 2012.

[178] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: ACM, 2020, pp. 7871–7880.

[179] M. Zhang, *Artificial higher order neural networks for economics and business*, 1st ed. Information Science Reference, 2009, vol. 1.

[180] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-Value Memory Networks for Directly Reading Documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 1400–1409.

[181] H. Keith and M. Thierry, "Chapter 3 - gravity equations: Workhorse,toolkit, and cookbook," in *Handbook of International Economics*, ser. Handbook of International Economics, G. Gopinath, E. Helpman, and K. Rogoff, Eds. Elsevier, 2014, vol. 4, pp. 131–195.

[182] Z. Fu, Y. Lin, Z. Liu, and W. Lam, "Fact discovery from knowledge base via facet decomposition," in *NAACL HLT 2019*, vol. 1, Stroudsburg, PA, USA, apr 2019, pp. 2891–2901.

[183] T. Ebisu and R. Ichise, "Graph pattern entity ranking model for knowledge graph completion," in *NAACL HLT 2019*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 988–997.

[184] L. Costabello, S. Pai, C. L. Van, R. McGrath, and N. McCarthy, "AmpliGraph: a Library for Representation Learning on Knowledge Graphs," *GitHub*, mar 2019.

[185] T. Caifang, R. Yuan, Y. Hualei, S. Ling, C. Jiamin, and W. Yutian, "Improving Knowledge Graph Completion Using Soft Rules and Adversarial Learning," *Chinese Journal of Electronics*, vol. 30, pp. 623–633, jul 2021.

[186] D. Rincon-Yanez, A. Mouakher, and S. Senatore, "Enhancing downstream tasks in Knowledge Graphs: A Complement Graph-based Approach Applied to Bilateral Trade," in *KES 2023*. Procedia Computer Science, 2023.

[187] G. Li, Z. Sun, L. Qian, Q. Guo, and W. Hu, "Rule-based data augmentation for knowledge graph embedding," *AI Open*, vol. 2, pp. 186–196, 2021.

[188] A. Borrego, D. Ayala, I. Hernández, C. R. Rivero, and D. Ruiz, "Generating Rules to Filter Candidate Triples for their Correctness Checking by Knowledge Graph Completion Techniques," in *Proceedings of the 10th International Conference on Knowledge Capture*, New York, NY, USA, sep 2019, pp. 115–122.

[189] B. Kotnis and V. Nastase, "Analysis of the impact of negative sampling on link prediction in knowledge graphs," *ArXiv*, vol. abs/1708.06816, 2017.

[190] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, S. Janarthanam, J. Andreas, D. Jurafsky, and Y. Choi, "Automated fact-checking with NLP: Challenges and opportunities," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Jun. 2018, pp. 7–18.

[191] G. Di Paolo, D. Rincon-Yanez, and S. Senatore, "A Quick Prototype for Assessing OpenIE Knowledge Graph-Based Question-Answering Systems," *Information*, vol. 14, no. 3, mar 2023.

[192] Z. Sun, H. Deng, X. Nie, J. Tang, and Y. Yang, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," in *(EMNLP-IJCNLP)*, 2019, pp. 119–128.

[193] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs," in *AAAI*, 2016, pp. 2305–2311.

[194] O. Cattaneo, G. Gereffi, and C. Staritz, *Global Value Chains in a Postcrisis World*. The World Bank, 2010. [Online]. Available: https://doi.org/10.1596/978-0-8213-8499-2

[195] S. Ponte, G. Gereffi, and G. Raj-Reichert, Eds., *Handbook on Global Value Chains*. Cheltenham, UK, Northampton, MA, USA: Edward Elgar Publishing, 2019.

[196] World Bank, *World Development Report 2020: Trading for Development in the Age of Global Value Chains*. Washington: World Bank Group, 2020.

[197] (2022) The cepii gravity database. 202211 version. CEPII. [Online]. Available: http://www.cepii.fr/CEPII/en/bdd_modele/bdd_modele_item.asp?id=8

[198] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering," *ACM Computing Surveys*, vol. 31, no. 3, sep 1999.

[199] Y. LeCun, S. Chopra, R. Hadsell, A. Ranzato, and F. J. Huang, "A tutorial on energy-based learning," 2006.

[200] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *International Conference on Learning Representations*, 2019.