Transforming Unstructured Historical and Geographic Data into Spatio-Temporal Knowledge Graphs

by

Basel Shbita

A Dissertation Presented to the
FACULTY OF THE USC GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(COMPUTER SCIENCE)

May 2024

To my parents, Ahmad and Nineli, my sister, Lina, and my partner, Maisa,

for their unwavering support and encouragement throughout this journey.

Acknowledgements

This dissertation marks the end of my doctoral studies. The PhD journey was long but rewarding and would not have been successful without the guidance, help, and support of many special people in my life.

First, I would like to thank my advisor, Professor Craig A. Knoblock, for believing in me and giving me the opportunity to be part of the Center on Knowledge Graphs at USC's Information Sciences Institute (ISI). I would also like to express my gratitude to Professors Cyrus Shahabi, John P. Wilson, Jay Pujara, and Yao-Yi Chiang (of the University of Minnesota), who were part of my qualification examination committee and thesis committee, and for providing insightful and sincere feedback on my work over the past several years.

During my PhD journey, I had the opportunity to work with many talented individuals. I would like to thank my colleagues and collaborators at USC-ISI: Professors Pedro Szekely, Filip Ilievski, Muhammad Rostami, and Jon May. I also want to extend my appreciation to my past collaborators at the University of Colorado Boulder, Dr. Johannes H. Uhl and Professor Stefan Leyk.

Additionally, I would like to thank all the staff at ISI, my fellow PhD students, and friends for their never-ending support throughout my doctoral studies: Binh, Fandel, Minh, Sami, Elizabeth, Myrl, Negar, Ehsan, Carlos, and of course, Karen.

Finally, I would like to thank my family. I will always be grateful to my parents, Ahmad and Nineli, my sister, Lina, and my partner, Maisa, for everything they have offered me and for always supporting every decision I made while chasing my dreams.

Table of Contents

List of Tables

List of Figures

Abstract

This dissertation presents a comprehensive approach to the transformation, integration and semantic enrichment of historical spatial and spatio-temporal data into knowledge graphs. The dissertation encompasses four core contributions: one, the automated generation of knowledge graphs from digitized historical maps for analyzing geographical changes over time; two, the generation of geo-feature label taxonomies from *OpenStreetMap*, the richest publicly available information source on geo-graphic entities on the web; three, the integration of spatial and semantic context embeddings for accurate geo-entity recognition and semantic typing; and four, the creation of a comprehensive knowledge graph for the analysis of historical data from digitized archived records. I introduce innovative methodologies and practical tools to support researchers from diverse fields, enabling them to derive meaningful insights from historical and geographic data. My approach is demonstrated through various applications, such as analyzing geospatial changes over time in USGS (United States Geological Survey) historical maps of transportation networks and wetlands, comparing geo-feature taxonomies generated from different regions in the world, automatic semantic typing of unlabeled georeferenced spatial entities, and constructing a spatio-temporal knowledge graph from digitized historical mineral mining data. The dissertation combines semantic web technologies, data mining, representation learning, and semantic modeling to build comprehensive knowledge graphs that support geospatial and temporal analyses.

Chapter 1

Introduction

1.1 Motivation

Understanding historical and geographic data plays a crucial role in geoscience research, impacting various domains like resource management, environmental conservation, and the broader domain of scientific discovery. These contain valuable information on activities and physical phenomena across time and space, and are increasingly available in digital data formats [1–5]. The challenge lies in transforming vast amounts of unstructured and semi-structured unlabeled data into coherent, structured knowledge that aids in decision-making and analysis [6, 7]. Traditional methods face difficulties in managing the complexity and diversity of data types and contexts requiring a sophisticated approach to integrate spatial, temporal, and semantic dimensions effectively [8, 9].

The representation, characterization, and linking of physical phenomena are central to intelligent behavior. As more historical and geographic data is digitized, there is a growing demand for geospatial artificial intelligence (AI) tools to support Geographic Information Systems (GIS), enhancing map and geo-data understanding, and spatial decision-making [10–12].

In the realm of geospatial data, we encounter a rich array of information, including but not limited to topographic, geographic, and economic elements that span both natural environments and human-made features. These data come in various forms, from digitized map archives to economic technical reports that offer insights into activities like transportation, mining and other phenomena like climate change. The diversity in data types, including scanned maps, digitized

geo-features, structured web data, and semi-structured data such as tables, presents significant analytical challenges. It requires a designated approach to effectively combine and make sense of these elements, emphasizing the need for sophisticated methods to identify the spatial and temporal dimensions involved in different problem settings. Our goal is not just about collecting information but also about understanding the interactions in the data, how it is relating to other data, and how we can represent it in a meaningful and effective way.

Recent advances in AI, computational tools, and web standards have paved the way for innovative approaches to data analyses and representation. Knowledge Graphs [13, 14] (KGs) have become instrumental in this context, offering a robust framework for representing, querying, and understanding diverse data sets in a coherent and interoperable manner. KGs are large semantic networks of entities, their attributes, and the relationships between them, often visualized as graphs and offering a robust framework for organizing the massive amounts of data across various domains. KGs facilitate the integration of diverse data sources, enabling more coherent and comprehensive data integration and analysis.

In the realm of KGs, the Semantic Web [15] provides the foundation for creating, storing, and querying the resulting linked data[16]. Technologies such as RDF [17] (Resource Description Framework), SPARQL [18] (a query language for RDF data), and OWL [19] (Web Ontology Language) enable the representation of knowledge in a machine-readable format, which is crucial for automated reasoning and data integration. The integration of KGs with Semantic Web technologies facilitates complex queries that span multiple domains and functionalities (such as data transformation), enhancing data discoverability, interoperability, and usability, as well as enhancing the ability to publish and link structured data on the web.

The evolution of KGs and the Semantic Web reflects a broader trend towards more dynamic, interconnected, and semantically rich data environments. These technologies are not just tools for data management; they are transforming how we understand and interact with the vast landscape of public and digitized information, especially in the domain of geo-data [20–22]. By creating

networks of knowledge that mirror the complexity and interconnectedness of the real world, KGs and the Semantic Web are paving the way for advanced analytics and downstream applications.

Our goal is to transform and enrich digitized historical geo-data into an expressive and interoperable format by harnessing open data, Semantic Web principles, and novel methodologies that rely on recent technological advancements and tools. This transformation enables comprehensive analysis over time and space, facilitating the fusion of data from various sources into a unified spatio-temporal and contextual view. This dissertation presents innovative methodologies and practical tools exploring the potentials of geo-semantics and spatio-temporal data integration, employing Semantic Web and AI technologies to create an effective, scalable, and accessible spatio-temporal knowledge framework.

While I primarily focus on historical and geographic data, the methodologies I present are broadly applicable to various spatial data types. These techniques can be used in contexts such as real-time environmental monitoring, urban planning, and disaster response. Our approach can be adapted to analyze contemporary datasets like satellite imagery or urban infrastructure data, enhancing the utility and applicability of our framework across a wide set of applications.

1.2 Research Objectives

The overarching goal of this dissertation is to bridge the gap between digitized historical and geographic data and contemporary analytical frameworks, transforming unstructured digitized geo-data into structured, semantic, and queryable spatio-temporal KGs. The specific research objectives are:

- **Automated Knowledge Graph Construction**: Develop methodologies for the automated generation of KG components (entities and relations) from digitized data (e.g., historical map sheets and digitized economic reports), and doing so automatically.

- **Contextual Geo-Entity Recognition and Semantic Typing**: Integrate spatial and semantic context embeddings to enhance geo-entity recognition and semantic typing, ensuring accurate retrieval and representation of geospatial entities in the KG.
- **Semantic Enrichment and Analysis**: Apply Semantic Web technologies to enrich the KG with additional data sources, including the task of entity and geo-entity resolution, thereby increasing its analytical depth and breadth for advanced spatio-temporal analyses.
- **Adherence to Linked Open Data Principles**: Implement and promote Linked Open Data (LOD) principles to ensure that the constructed KGs are accessible, interlinkable, and reusable, facilitating a more open and connected web of geospatial data.

These objectives aim to contribute to the fields of GIS, the Semantic Web, and AI, providing researchers and decision-makers with powerful tools and techniques to process, analyze, and interpret historical geospatial data. The anticipated outcome is a set of practical methodologies and tools that enable the transformation of historical and geographic data into dynamic and queryable KGs in the form of linked data that adheres to LOD principles.

1.3 Thesis Statement

This thesis provides tools and techniques for the automated understanding and transformation of unstructured and semi-structured geospatial and historical data from heterogeneous sources into a standardized representation of expressive and interoperable spatio-temporal knowledge graphs. The thesis presents methodologies that both integrates the data with other sources on the web and leverages web knowledge for enhanced data analysis.

1.4 Approach

My approach adopts a multi-faceted approach to address the challenges of data transformation and integration. I focus on the automatic conversion of unstructured and semi-structured historical

and geographic digitized data into structured, queryable knowledge representations. This process requires the development of advanced methods for data understanding (data parsing, decomposition, and classification), contextualization (entity resolution and linking), and transformation (data representation), enabling the creation of spatio-temporal KGs that encapsulate the "meaning" and complex interrelations of geographic and historical entities.

1.4.1 Contributions of the Research

The primary contributions of this research are as follows:

- An automatic methodology for transforming vectorized (digitized) topographic contemporary and historical maps into web-publishable spatio-temporal knowledge graphs, facilitating seamless geo-feature change analysis over time and space (Chapter 2).
- An unsupervised, automatic method for building lightweight geo-feature taxonomies from *OpenStreetMap* data, enhancing geo-data semantics and categorization (Chapter 3).
- An embedding technique that captures spatial and semantic contexts for accurate geo-entity typing and classification, thus enabling its integration in the KG and enhancing the interpretability of geographic data for accruate entity retrieval (Chapter 4).
- A methodology for the construction and semantic enrichment of a knowledge graph centered on mineral mining data, demonstrating advanced spatial, temporal, and quantitative data integration and analysis capabilities (Chapter 5).

1.5 Outline of the Thesis

The rest of the thesis is organized as follows, with its structure, including the inputs, interrelationships, and core contributions of each chapter, visually depicted in Figure 1.1. The figure illustrates how these chapters collectively address the broader problem and showcases the evaluated use cases to demonstrate their practical utility.

Chapter 2 (marked in green) details the process of converting vectorized topographic contemporary and historical map sheets into spatio-temporal KGs. This chapter not only addresses the dynamic representation of geo-features over time but also establishes the foundational structure of the KG and the aspect of geo-data representation, such as geo-featurs extracted from hisotrical map, to enable its interoperability and representation as a resource on the web.

Chapter 3 (marked in purple) presents the construction of geospatial feature taxonomies using publicly available *OpenStreetMap* data. This taxonomy of labels is designed to enhance more complex data analyses, as demonstrated in the subsequent Chapter 4.

Chapter 4 (marked in yellow) presents a novel methodology for embedding geo-referenced vector data. This process integrates geometric, spatial, and semantic aspects to enable the classification and typing of geo-entities. This chapter bridges a methodological gap by providing means to accurately identify vectorized (digitized) geospatial entities with their respective types (e.g., `wetland`), assumed as known in Chapter 2, thereby enhancing the accuracy of the KG, particularly in recognizing and contextualizing geo-features and assigning them with fine-grained types.

Chapter 5 (marked in red) transitions from the focus on primarily geo-encoded vector data in the preceding chapters, moving towards integrating textual and historical data associated with geo-referenced spatial entities. The chapter explores how digitized historical mineral reports can be semantically modeled, enriched and integrated into spatio-temporal KGs, illustrating the process of transforming textual and spatial data into a structured and insightful knowledge base.

Collectively, these four chapters showcase a comprehensive approach to geospatial data integration and analysis, highlighting the contributions to the field and their adaptability across different domains, exemplifying their potential to power different downstream tasks.

Appendix A (marked in blue) is an auxiliary to our main contribution in Chapter 5, where the process of identifying units in textual scientific data is elaborated. This complements the discussions in Chapter 5 on grounding quantitative measurement data, such as ore tonnage, to enable automatic data transformation often needed in downstream applications. Chapter 6 concludes with the contributions of this thesis and discusses future research.

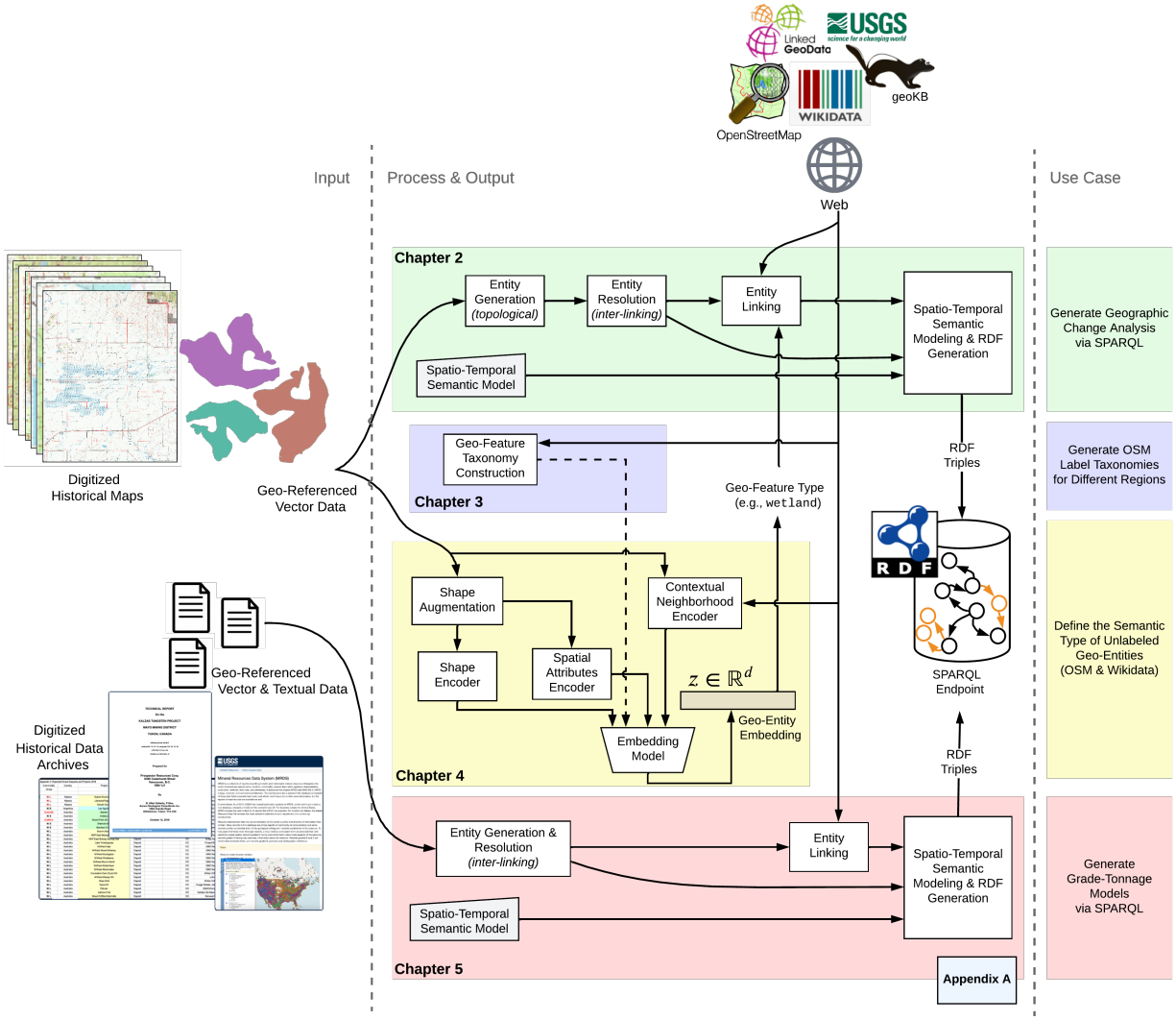


Figure 1.1: Flowchart depicting the structure of the thesis, illustrating the input, interrelationship, and core contributions of each chapter. It highlights how these chapters collectively address the broader problem and showcases the use cases evaluated to demonstrate their practical utility.

Chapter 2

Building Spatio-Temporal Knowledge Graphs from Digitized Historical Maps

In this chapter, I introduce the foundation of my thesis by showcasing how vectorized topographic historical maps are transformed into comprehensive spatio-temporal knowledge graphs (KGs) [23]. This transformation is essential for capturing the dynamic nature of geographical features and their historical changes in a Semantic Web-friendly representation. I discuss the techniques involved in transforming vector geographic features into structured knowledge, including location-based entity linking through reverse-geocoding, which plays a crucial role in the contextualization of geo-entities. Supported by a case study using United States Geological Survey (USGS) historical maps, I evaluate the approach on specific examples like railroad networks and wetland reductions. I also demonstrate how the automatically constructed KG and linked data enable effective querying and visualization of changes over different points in time. This foundation sets the stage for subsequent chapters by establishing the methodologies for data representation and the interoperability of the generated KGs on the web.

2.1 Motivation

Historical map archives contain valuable geographic information on both natural and human-made features across time and space [6, 7] and are increasingly available in systematically acquired digital data formats [2, 3]. The spatio-temporal data extracted from these documents are important since they can convey when, where and what changes took place [8]. For example, this type of data enables the detection of long-term changes in railroad networks or studying the evolution of wetlands within the same region over time and thus can support decision-making related to the development of transportation infrastructure or questions related to land conservation, landscape ecology, or long-term land development and human settlement. Many applications assessing geographic changes over time typically involve searching, discovering, and manually identifying, digitizing, and integrating relevant data. This is a difficult and laborious task that requires domain knowledge and familiarity with various data sources, data formats and working environments, and the task is often susceptible to human error [24].

Generally, there are two types of geospatial data, namely, raster data and vector data. Recent technological advances facilitate the efficient extraction of vectorized information from scanned historical maps [1, 2, 4–6, 8]. Vector data provide a compact way to represent real-world features within a GIS. Every geographic feature can be represented using one of three types of geometries: (i) points (depict discrete locations such as addresses, wells, etc.), (ii) lines (used for linear features such as roads, rivers, etc.) or (iii) polygons (describe enclosed areas like waterbodies, islands, etc.). This work tackles the core problem of detecting how geometries change over time, focusing on linear and polygonal features.

Linked geospatial data has been receiving increased attention in recent years as researchers and practitioners have begun to explore the wealth of geospatial information on the web [11, 12, 25]. In addition, the need for tracing geographic features over time or across documents has emerged for different applications [8]. Furthermore, growing attention has been paid to the integration and contextualization of the extracted data with other datasets in a GIS [26, 27].

To better support analytical tasks and understand how map features change over time, we need more than just the extracted vector data from individual maps. We need to tackle the following challenges:

1. **Entity generation and interlinking.** Generate and interlink the geospatial entities ("building block" geometries originating from the vector data) that constitute the desired features (i.e., railroad lines or wetland areas) and can exist across map editions.

2. **External geo-entity linking.** Contextualize and link the generated entities to external resources to enable data augmentation and allow users to uncover additional information that does not exist in the original map sheets.

3. **Representation.** Represent the resulting data in a structured and semantic output that can be easily interpreted by humans and machines, and adheres to the principles of the Semantic Web.

Previous work on creating linked data from historical geospatial information has focused on the problem of transforming a single instance of a map sheet or a formatted geospatial data source into RDF [28–30]. However, this line of work does not address the issue of entity interlinking that is essential for building linked geospatial data for the task of change analysis with a semantic relationship between geospatial entities across map editions of the same region, which could be part of a large collection of topographic map sheets. Similar work is concerned with only a specific type of geometry, such as points, as in [26, 31], or is limited to a specific physical feature (i.e. flooded areas [32] or wildfires [10]). Our work does not impose such limitations.

Our approach is not only helpful in making the RDF data widely available to researchers but also enables users to easily answer complex queries, such as investigating the interrelationships between human and environmental systems. Our approach also benefits from the open and connective nature of linked data. Compared to existing tools such as PostGIS[1], which can only handle

[1] `https://postgis.net/`

queries related to geospatial entities and relationships within local databases, linked data can utilize many widely available knowledge sources on the web, such as *OpenStreetMap* (OSM) [20], GeoNames [21], and LinkedGeoData [22], to enable rich semantic queries.

Providing contextual knowledge can help explain or reveal interactions of topographic changes to further spatio-temporal processes. For example, the external-linking enables augmentation of the contained geographic information with data from external KBs, such as historical population estimates. Once we convert the map data into linked data, we can execute SPARQL queries to identify the changes in map features over time and thus accelerate and improve spatio-temporal analysis tasks. Using a semantic representation that includes geospatial attributes, we can support researchers to query and visualize changes in maps over time and allow the development of data analytics applications that could have great implications for environmental, economic, or societal purposes.

Problem Definition The task we address here is as follows: Given geographic vector data extracted from multiple map editions of the same region, we aim to automatically construct a knowledge graph depicting all the geographic features that represent the original data, their relations (interlinks), and their semantics across different points in time. Using the constructed knowledge graph, we enable tackling more specific downstream analysis tasks. These may include the visualization of feature changes over time and the exploration of supplementary information (e.g., population data, elevation data, etc.) related to the region originating from an external knowledge base. As an example, consider the maps in Figure 2.1 where changes in the New Albany (OH) and Chicago (IL) railroad system have occurred between 1886 and 1904. Figure 2.2 shows the railroad line changes between the different map editions. Line segments that have been added are marked in red and line segments that have been removed are marked in blue. Assuming we have the data available as vector data (which can be generated from scanned maps using the approaches mentioned earlier), our task in such a setting would be to construct a knowledge graph describing the shared line segments that are shared across these maps or unique in individual maps with a conventional semantic representation for the railroad line segments in each map edition. This

would include objects from a list of common geographic features (points, lines, or polygons), their geolocational details, and their temporal coverage to allow easy analysis and visualization.

Figure 2.1: New Albany (OH) and Chicago (IL) railroad system maps in 1886 (left) and 1904 (right).

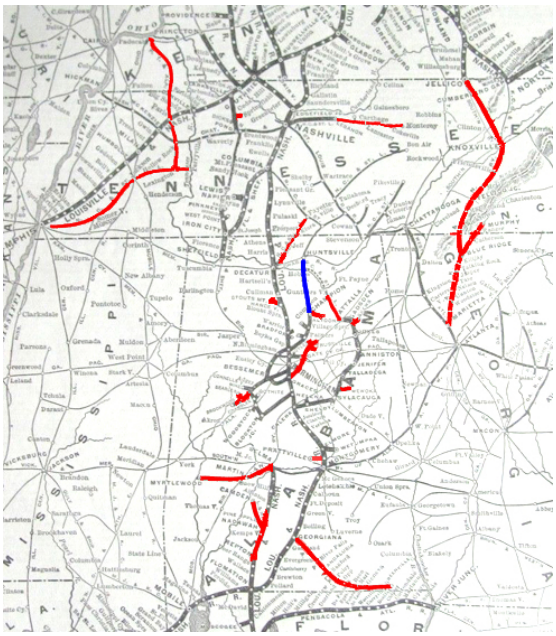


Figure 2.2: Visual representation of the change in the New Albany (OH) and Chicago (IL) railroad system between the years 1886 and 1904; additions are in red, removals are in blue.

Significance This chapter presents a fully automatic end-to-end approach for building a contextualized spatio-temporal knowledge graph from a set of vectorized geographic features extracted from topographic historical maps, given the feature type. We tackle the core challenges we mentioned earlier by presenting:

1. An algorithm to identify and partition the original vector data into interlinked geospatial entities (i.e., "building block" geometries) that constitute the desired geographic features across map editions (entity generation and interlinking task).

2. A method to identify and retrieve geospatial entities from a publicly available knowledge base (external geo-entity linking task).

3. A semantic model to describe the resulting spatio-temporal data in a structured and semantic output that can be easily interpreted by humans and machines in a form of a knowledge graph that adheres to linked data principles (representation task).

4. A thorough evaluation for each of the tasks above by applying our method to five relevant datasets that span two types of geographic features: railroads (line-based geometry) and wetlands (polygon-based geometry). We also make the source code, the original datasets, and the resulting data publicly available[2], and publish the resulting knowledge graph as linked data, along with a designated SPARQL endpoint with compliant IRI dereferencing. The resulting knowledge graph characteristics are described in table 2.1.

Table 2.1: Resulting knowledge graph characteristics.

SPARQL endpoint	`https://linked-maps.isi.edu/sparql`
Linked Data explorer	`https://linked-maps.isi.edu`
Size	2 classes 8 relations 630 nodes (340 non literals) 1514 edges

[2] `https://github.com/usc-isi-i2/linked-maps`

2.2 Building Spatio-Temporal Knowledge Graphs

2.2.1 Preliminaries

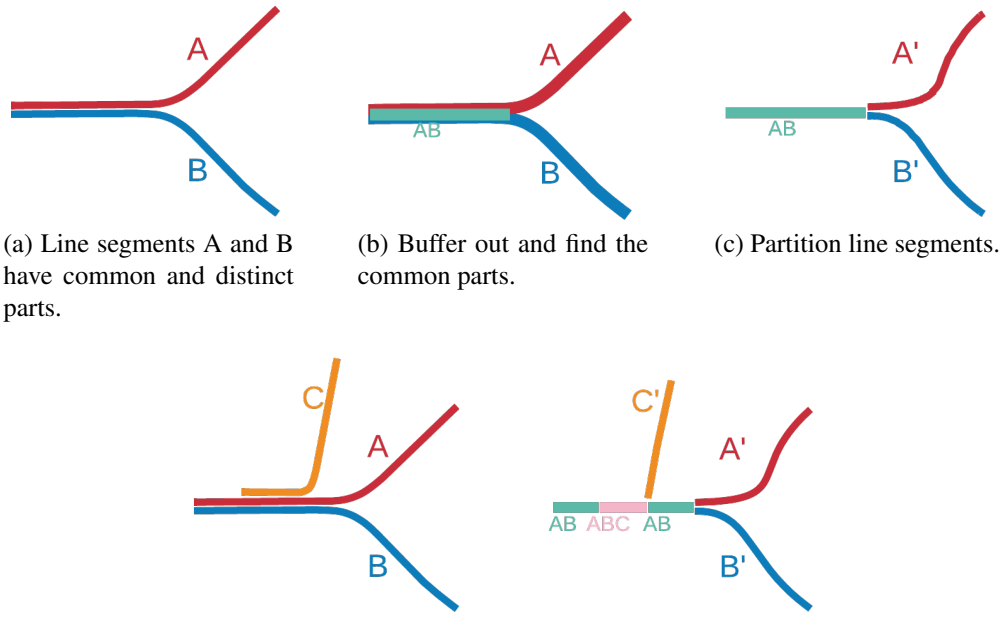
Before we discuss our proposed method, it is essential to define certain preliminaries and the terminology used in this chapter. A geographic feature refers to a collection of geometries (points, lines, or polygons), which together represent an entity or phenomenon on Earth. In this chapter, a geographic feature is represented using a collection of "building block" geometries. Given a set of geographic features pertaining to the same region from different points in time, each generated "building block" is essentially the largest geographic feature part that is either shared across different points in time or unique for a specific point in time. These blocks can be either lines or areas in the case of linear geographic features or polygon geographic features, respectively. For example, in Figure 2.3a, `A` and `B` are line building blocks. Each building block may be decomposed into smaller building blocks. For example, if a part of `A` and `B` represents the same entity, `A` and `B` are decomposed into three building blocks: `A'`, `B'`, and `AB`. `AB` represents the shared geometry detected from `A` and `B`, and `A'` and `B'` represent the unique parts in `A` and `B`, respectively. Similarly, each color area in Figure 2.4 (`A'` in red, `B'` in blue, and `AB` in green) is an area building block (giving a total of three building blocks). Each building block geometry is encoded as Well-Known Text [3] (WKT) representation (e.g., `MULTILINE` or `MULTIPOLYGON` textual format) and corresponds to a geospatial entity in our resulting KG.

2.2.2 Overview of the Approach

The proposed pipeline for the construction of the knowledge graph consists of several major steps as illustrated in Figure 2.5. These steps can be summarized as follows:

1. Automatically partition the input geographic feature originating from the vector data into building block geometries (i.e., geospatial entities) using a spatially-enabled database service (e.g., PostGIS) (see Section 2.2.3).

[3] `https://www.ogc.org/standard/wkt-crs/`



(a) Line segments A and B have common and distinct parts.

(b) Buffer out and find the common parts.

(c) Partition line segments.

(d) Line segment C is added.

(e) Final line partitioning.

Figure 2.3: Illustration of the geometry partitioning to building blocks for a line geometry: spatial buffers are used to identify the same line segments considering potential positional offsets of the data.

Figure 2.4: Illustration of a geometry partitioning result for a polygon geometry: each color represents a different building block. A single building block may contain disconnected areas.

2. Perform external entity linking by utilizing a reverse-geocoding service to map the geospatial entities to existing instances in an open knowledge base (e.g., *OpenStreetMap*) (see Section 2.2.4).

3. Construct the knowledge graph by generating RDF triples following a pre-defined semantic model using the data we generated in the previous steps (see Sections 2.2.5 and 2.2.6).

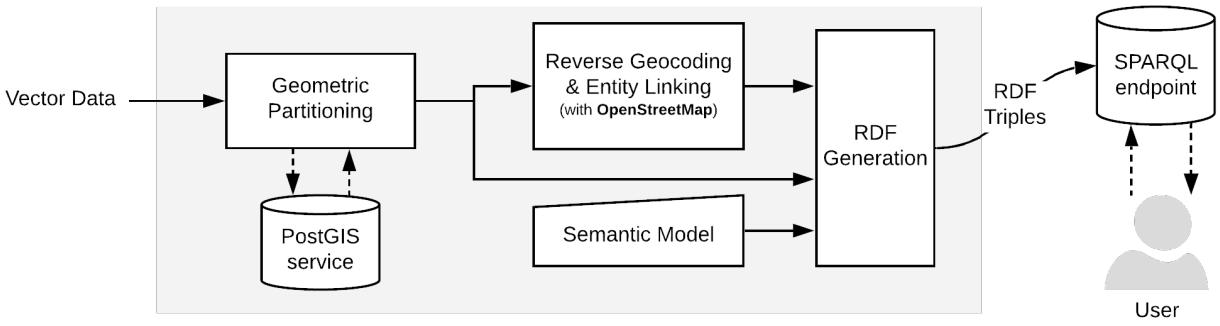


Figure 2.5: Pipeline for constructing spatio-temporal linked data from vector data.

Once the RDF data is deployed, users can easily interact with the building block geometries (geospatial entities), the geographic features and metadata to perform queries (Section 2.2.7). These allow end-users to visualize the data and support the development of spatio-temporal downstream applications.

2.2.3 Generating Building Blocks and Interlinking

The first task in our pipeline is the generation of building block geometries that can represent the various geographic features (e.g., railroad networks or wetlands) across different map editions in a granular and efficient fashion. This task can be classified as a common entity matching/linking and entity "partitioning" task. Given the geometries from different map editions of the same region, we want to identify which parts of those geometries coincide and thus represent the same parts of the feature. This allows us to generate building block geometries that are more granular and can be used to represent the common and the distinct parts (changes) of the geographic features.

Consider a simplified example consisting of linear features from two map editions (Figure 2.3a), where line A is from an older map edition and line B is from the latest map edition with a part of the feature that has been changed. We split the lines into building block lines A', B', and AB based on the intersection of the lines, as shown in Figures 2.3b and 2.3c. When a third source (another map edition also containing the feature), C, is added, a similar partitioning process is executed as shown in Figures 2.3d and 2.3e. Another example is seen in Figure 2.4, where we show an illustration of a geometry partitioning result for a polygon-based feature (i.e., wetland

Algorithm 1: The feature partitioning and interlinking algorithm.

Input: a set $\mathscr{M}$ of feature geometries for different map editions of the same region (vector data)
Output: a directed acyclic graph $\mathscr{G}$ of building block geometries (nodes) and their relations (edges)

1 **foreach** $i \in \mathscr{M}$ **do**
2 $\mathscr{F}_i$ = set of geometries in i;
3 $\mathscr{G}$.add($i \mapsto F_i$);
4 $\mathscr{L}$ = list of current leaf nodes in $\mathscr{G}$;
5 **foreach** $k \in \mathscr{L}$ **do**
6 $\mathscr{F}_k$ = set of geometries in k;
7 $\mathscr{F}_\alpha = \mathscr{F}_i \bigcap \mathscr{F}_k$;
8 $\mathscr{G}$.add($\alpha \mapsto F_\alpha$) ; set i, k as direct predecessors of α;
9 $\mathscr{F}_\gamma = \mathscr{F}_k \setminus \mathscr{F}_\alpha$;
10 $\mathscr{G}$.add($\gamma \mapsto F_\gamma$) ; set k as direct predecessor of γ;
11 $\mathscr{F}_\delta = \mathscr{F}_i \setminus (\bigcup_{j \in \mathscr{L}} F_j)$;
12 $\mathscr{G}$.add($\delta \mapsto F_\delta$) ; set i as direct predecessor of δ;

data). Similar to the line partitioning process described above, the polygon partitioning generates a collection of building block areas (as seen in Figure 2.4).

As we mentioned in Section 2.2.2, we use a spatially-enabled database service to simplify handling data manipulations of geospatial objects. PostGIS is a powerful PostgreSQL extension for spatial data storage and query. It offers various functions to manipulate and transform geographic objects in databases. To handle our task efficiently and enable an incremental addition of map sheets over time, we implemented Algorithm 1. The algorithm performs the partitioning task by employing several PostGIS Application Programming Interface (API) calls over the geometries of our lines or polygons in the database. In the case of line geometries, we buffer each line segment to create two-dimensional areas before applying any geospatial operation described below.

In detail, the procedure works as follows. The **for** loop in line 1 iterates over each of the map editions to extract the feature geometry (as seen in line 2 and stored in $\mathscr{F}_i$) to create the initial "building block" geometry (line 3, denoted as node i and added to graph $\mathscr{G}$, which eventually will hold our final set of building blocks and record their relations in a data structure). Line 4 retrieves the leaf nodes from graph $\mathscr{G}$ to list $\mathscr{L}$. In the first iteration list $\mathscr{L}$ is empty. In the next iterations it will include "leaf" nodes. These are nodes that represent the most fine-grained building blocks

computed so far. A and B in Figure 2.3a correspond to k and i respectively (in iteration 2 of the algorithm when executed over the data in the Figure 2.3). We then perform the following over the newly added building block i:

1. For each "leaf" in $\mathscr{L}$ we execute:

   (a) **Geometry intersection.** k's geometry is stored in $\mathscr{F}_k$ (line 6) and then used to compute the matched geometry parts between i and k to generate the geometry $\mathscr{F}_\alpha$ (line 7) and create the new building block α, a direct successor of nodes i and k (line 8). α (iteration 2) corresponds to AB in Figure 2.3c.

   (b) **Geometry difference (local "subtraction").** In line 9, we compute the geometry in k that is not in i, resulting in the geometry $\mathscr{F}_\gamma$ corresponding to the new building block γ, now a direct successor of k (line 10). γ (iteration 2) corresponds to B' in Figure 2.3c.

2. **Geometry union-difference (global "subtraction").** Once we finish going over the list of leaves, we compute the unique geometries that exist in i (the last added map edition in $\mathscr{M}$) by subtracting the union of the geometries of the leaf node intersections (with previous processed maps) from the original map block i (as described in line 11), resulting in the geometry $\mathscr{F}_\delta$ corresponding to the new building block δ, now a direct successor of node i (line 12). δ (iteration 2) corresponds to A' in Figure 2.3c.

In the worst-case scenario, graph $\mathscr{G}$ will grow as a balanced binary tree. For each added map edition, we increase $\mathscr{G}$'s depth by one and split the geometry of each leaf node into two parts (shared and unique, with respect to the lastly added node). For M map editions, we get 2^M leaf nodes. Assuming that the average number of vectors per feature is V, each leaf node computation will introduce at most V new vectors. With the assumption that the computation cost for a pair of vectors is constant, the expected time complexity of Algorithm 1 is $O(2^M MV)$.

The relations between the nodes in graph $\mathscr{G}$ carry a semantic meaning between the different building blocks (a node is contained in its predecessors and contains its successors) and will play a

critical role in the RDF generation and query mechanism since they represent the relations between the building blocks across different points in time of the same region.

2.2.4 Reverse-Geocoding and Geo-Entity Linking

Volunteered geographic information platforms [33] are used for collaborative mapping activities with users contributing geographic data. *OpenStreetMap* (OSM) is one of the most pervasive and representative examples of such a platform and operates with a certain, albeit somewhat implicit, ontological perspective of place and geography more broadly. OSM suggests a hierarchical set of tags[4] that users can choose to attach to its basic data structures to organize their map data. These tags correspond to geographic feature types that we will query (i.e., wetland, railroad, etc.).

Additionally, a growing number of OSM entities are being linked to corresponding Wikipedia articles, Wikidata [34] entities and feature identifiers in the USGS Geographic Names Information Service (GNIS) database. GNIS is the U.S. federal government's authoritative gazetteer. It contains millions of names of geographic features in the United States.

Our proposed method for the enrichment of the generated geospatial entities (i.e., building block geometries) with an external resource is built upon a simple geo-entity linking mechanism with OSM. This is again a task of entity matching/linking; this time it is with an entity in an external knowledge base.

The method is based on reverse-geocoding, which is the process of mapping the latitude and longitude measures of a point or a bounding box to an address or a geospatial entity. Examples of these services include the GeoNames reverse-geocoding web service[5] and OSM's API.[6] These services support the identification of nearby street addresses, places, areal subdivisions, etc., for a given location.

The geo-entity linking process is depicted in Algorithm 2 and illustrated in Figure 2.6. We start with individual building block geometries of known type (T in Algorithm 2). In the case

[4] `https://wiki.openstreetmap.org/wiki/Map_features`
[5] `http://www.geonames.org/export/reverse-geocoding.html`
[6] `https://wiki.openstreetmap.org/wiki/API`

Algorithm 2: The geo-entity linking algorithm.

Input: building block geometry s, number of samples N, feature type T
Output: list $\mathscr{L}$ of *OpenStreetMap* instances in s

1 B_s = bounding box wrapping s;
2 $\mathscr{L}$ = reverse-geocoding(B_s, T); // returns *OpenStreetMap* instances of type T in B_s
3 **for** 1...N **do**
4 $\quad e$ = randomly sample a `Point` in s;
5 $\quad E$ = reverse-geocoding(e, T);
6 $\quad \mathscr{L}$.add(E);
7 remove instances with a single appearance in $\mathscr{L}$;
8 return $\mathscr{L}$;

of the data we present later in the evaluation in Section 2.3, we start with the building blocks of geometries of **railroads** or **wetlands** (seen in blue in Figure 2.6), so we know the feature type we are searching for. Each input building block geometry, s, is an individual node in graph $\mathscr{G}$ from Section 2.2.3. We first generate a global bounding box for s by finding its northmost, southmost, eastmost, and westmost coordinates. The OSM service takes the resulting box as input to execute a reverse-geocoding API call that locates instances of type T on the external knowledge base, as described in lines 1-2. Some of these instances do not share any geometry parts with our inspected building block. As a heuristic, we randomly sample a small number of coordinate pairs (`Points`), corresponding to the number of entities composing s (N ranges from 10 to 85 in our datasets, as presented in Section 2.3.1); thus, we gain more confidence in the detected instances, as seen in lines 4-6 in Algorithm 2 and in red in Figure 2.6. Finally, we reduce the list by removing the matching candidates in the external KB that have a single appearance, thus filtering out entities that are not likely part of the enclosed geometry of our geospatial entity. Each one of the resulting instances is used in later stages to enrich the knowledge graph we construct with additional semantics and metadata from the external knowledge base.

Figure 2.7 shows an example of a scanned topographic map (seen in the background), which we used to extract its corresponding wetland vector data, alongside two OSM instances detected using our geo-entity linking method. The labels in Figure 2.7 (i.e., `Four Mile Cove Ecological Preserve` and `Six Mile Cypress Slough Preserve`) correspond to the `name` attribute of each

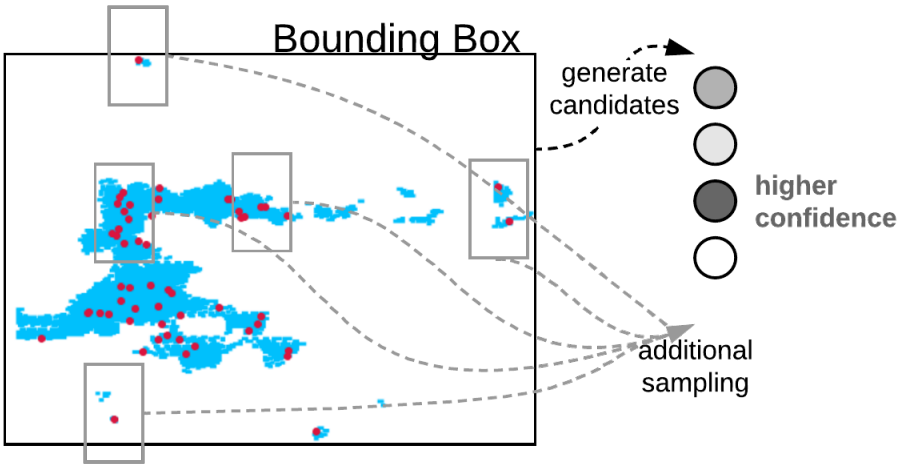


Figure 2.6: The method for acquiring external knowledge base instances.

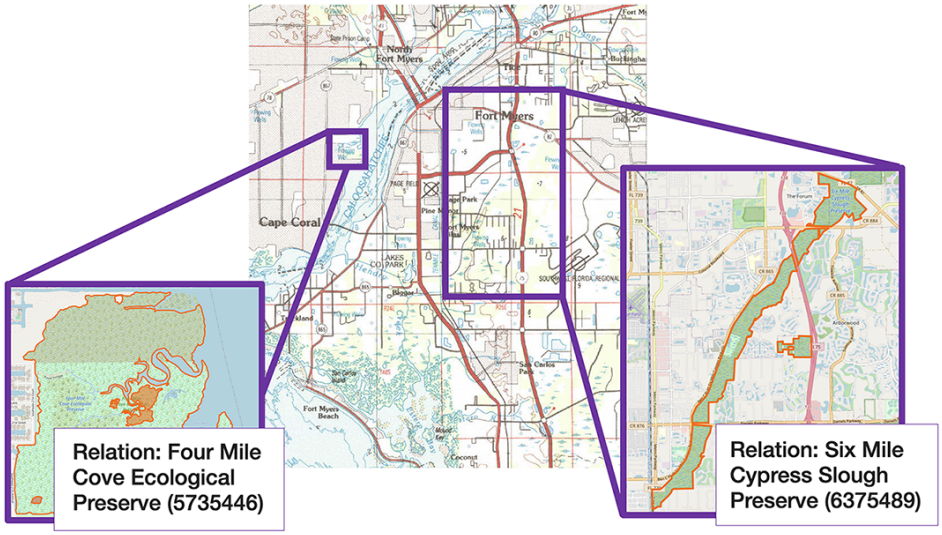


Figure 2.7: An example of two OSM instances (enclosed in purple) and their `name` labels detected using our geo-entity linking method over a scanned topographic map (seen in the back).

entity. These labels are part of a set of attributes we use to augment our resulting data with information that did not exist in the original dataset.

The time complexity of Algorithm 2 depends on the number of samples N we choose and on the bounding box calculation. Each API call has a constant cost. As well, the bounding box calculation has a constant cost. Thus, the expected time complexity of Algorithm 2 is $O(N)$. As we mentioned in Section 2.2.3, in the worst-case scenario, graph $\mathscr{G}$ will contain 2^M leaf nodes (for M map editions); thus, the total expected time complexity of this step in the pipeline is $O(2^M N)$.

2.2.5 Semantic Model

As a generic model or framework, RDF can be used to publish geographic information. Its strengths include its structural flexibility, particularly suited for rich and varied forms for metadata required for different purposes. However, it has no specific features for encoding geometry, which is central to geographic information. The OGC GeoSPARQL [35] standard defines a vocabulary for representing geospatial data on the web and is designed to accommodate systems based on qualitative spatial reasoning and systems based on quantitative spatial computations. To provide a representation with useful semantic meaning and universal conventions for our resulting data, we define a semantic model that builds on GeoSPARQL.

Our approach towards a robust semantic model is motivated by the OSM data model, where each feature is described as one or more geometries with attached attribute data. In OSM, `relations` are used to organize multiple `nodes` or `ways` into a single entity. For example, an instance of a bus route running through three different `ways` would be defined as a `relation`.

Figure 2.8 shows the semantic model we describe in this section. In GeoSPARQL, the class type `geo:Feature` represents the top-level feature type. It is a superclass of all feature types. In our model, each instance of this class represents a single building block extracted from the original vector data.

By aggregating a collection of instances of the class `geo:Feature` with a property of type `geo:sfWithin` we can construct a full representation for the geometry of a specific geographic feature in a given point in time. Similarly, we can denote the decomposition to smaller elements using the property `geo:sfContains`). The use of these properties enables application-specific queries with a backward-chaining spatial "reasoner" to transform the query into a geometry-based query that can be evaluated with computational geometry. Additionally, we use the property `geo:sfOverlaps` with subjects that are instances from OSM to employ the web as a medium for data and spatial information integration following linked data principles. Furthermore, each instance has at least one property of type `dcterms:date` to denote the different points in time in which the building block exists. Each of the aforementioned properties has a cardinality of 1:n,

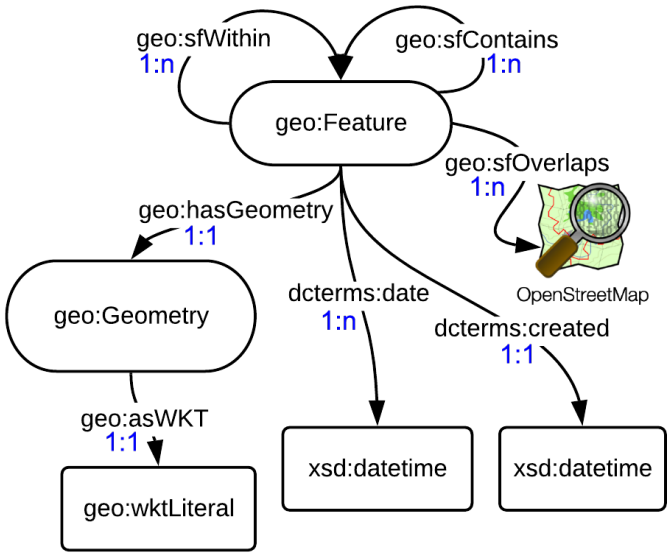


Figure 2.8: Semantic model for the linked data. Nodes cardinality is shown in blue next to each edge.

meaning that multiple predicates (relations) of this type can exist for the same building block. The property `dcterms:created` is used to denote the time in which this building block was generated. `dcterms` stands for the Dublin Core Metadata Initiative[7] metadata model, as recommended by the World Wide Web Consortium (W3C).[8]

Complex geometries are not human-readable as they consist of hundreds or thousands of coordinate pairs. Therefore, we use dereferenceable URIs to represent the geospatial entity instead. Using a named node in this capacity means that each entity has its own URI as opposed to the common blank-node approach often used with linked geospatial entities. Each URI is generated using a hash function (the MD5 message-digest algorithm, arbitrarily chosen) on a plain-text concatenation of the feature type, geometry, and its temporal extent, thus providing a unique URI given its attributes. The geometry contains absolute coordinates, thus rules out the possibility of hash clashes. Each building block instance (geospatial entity) holds a property of type `geo:hasGeometry` with

[7] `https://www.dublincore.org/specifications/dublin-core/dcmi-terms/`
[8] `https://www.w3.org/`

a subject that is an instance of the class `geo:Geometry`. This property refers to the spatial representation of a given feature. The class `geo:Geometry` represents the top-level geometry type and is a superclass of all geometry types.

In order to describe the geometries in a compact and human-readable way we use the WKT format for further pre-processing. The `geo:asWKT` property is defined to link a geometry with its WKT serialization and enable downstream applications to use SPARQL graph patterns.

Figure 2.9 shows how the spatio-temporal data, resulting from the previous steps, is mapped into the semantic model (from Figure 2.8) to generate the final RDF graph. The first column, titled `gid`, corresponds to the local URI of a specific node (building block geometry). The columns titled `predecessor_id` and `successor_id` correspond to the local URIs of the nodes composed-of and composing the specified `gid` node, respectively. All the three node entities are of type `geo:Feature`. The data in the `wkt` column contains the geometry WKT representation. It is linked to the building block node via an entity of type `geo:Geometry`, as we described above. The rest of the attributes (`year`, `time_generated`, and `OSM_uri`) are stored as literals, following the semantic model we presented in Figure 2.8.

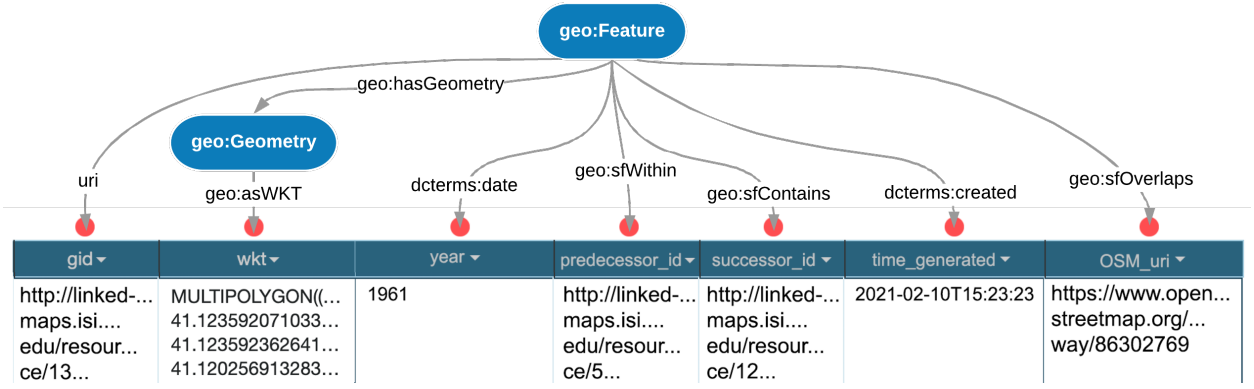


Figure 2.9: Mapping of the generated spatio-temporal data into the semantic model.

2.2.6 Incremental Linked Data

Linked Data and Semantic Web technologies can effectively maximize the value extracted from open, crowdsourced, and proprietary big data sources. Following the data extraction and acquisition tasks described in Sections 2.2.3 and 2.2.4, and the semantic model described in Section 2.2.5, we can now produce a structured standard ontologized output in a form of a knowledge graph that can be easily interpreted by humans and machines, as linked data. In order to encourage reuse and application of our data in a manageable manner, we need to make sure that the linked data publication process is robust and maintainable.

This hierarchical structure of our directed acyclic graph $\mathscr{G}$ (introduced in Algorithm 1) and its metadata management allows us to avoid an update across all the existing published geographic vector data (in linked data) and instead handle the computations incrementally once a new representation of the feature from a subsequent map edition is introduced.

The approach we present is complete and follows the principles of Linked Open Data by:

1. Generating URIs as names for things, without the need to modify any of the previously published URIs once further vector data from the same region is available and processed.

2. Maintaining existing relations (predicates) between instances (additional relations may be added, but they do not break older ones).

3. Generating machine-readable structured data.

4. Using standard namespaces and semantics (e.g., GeoSPARQL).

5. Linking to additional resources on the web (i.e., *OpenStreetMap*).

2.2.7 Querying

The semantic model presented in Section 2.2.5 and its structure provide a robust solution enabling a coherent query mechanism to allow a user-friendly interaction with the linked data.

In order to elaborate on the query construction idea, we describe the elements that are needed for a general query "skeleton" from which we can establish more complicated queries to achieve different outcomes as required. Listing 2.1 shows a query (i.e., the "skeleton" query) that retrieves all the leaf node building blocks (i.e., the most granular building blocks). As shown in Listing 2.1, we first denote that we are interested in a `geo:Feature` that has a geometry in `WKT` format which gets stored in the variable `?wkt` as shown in lines 3-4 (the variable we visualize in Figures 2.14a, 2.14b, 2.15a, and 2.15b). Line 5 restricts the queried building blocks (`geo:Features`) to leaf nodes only (in graph $\mathscr{G}$), thus retrieving the most granular building blocks. This is done by discarding nodes that with predicate of type `geo:sfContains`, which means that we retrieve only leaf nodes.

```
1 SELECT ?f ?wkt
2 WHERE {
3  ?f a geo:Feature ;
4     geo:hasGeometry [ geo:asWKT ?wkt ] .
5  FILTER NOT EXISTS { ?f geo:sfContains _:_ } }
```

Listing 2.1: Our SPARQL query "skeleton".

This is important due to the way graph $\mathscr{G}$ "grows": as we mentioned previously, every time we add a new representation of the feature from a subsequent map edition, we decompose the existing leaf nodes (most granular building blocks) to a new layer of leaf blocks (newer, smaller and more granular building blocks, if subject to decomposition) and its metadata migrates to the lowest level of nodes (new leaves). This property makes our solution robust and suggests an efficient way of querying, avoiding the need to "climb up" the graph for more complicated ("composed") blocks.

If, for example, we are interested to see the entire geographic feature in a specific point in time, we can add the clause {`?f dcterms:date <...> .`} inside the `WHERE` block (lines 2-5). If we are interested to see the changes from a different time, we can add an additional clause {`MINUS { ?f dcterms:date <...> . }`} as well. The syntax and structure of the query allows an easy adaptation for additional tasks such as finding the distinct feature parts from a specific time or finding the feature parts that are shared over three, four or even more points in time or map editions. The nature of our knowledge graph provides an intuitive approach towards writing simple and complex queries.

2.3 Evaluation and Discussion

We evaluate and analyze our methods using qualitative and quantitative methods over two types of geographic features: railroads (line-based geometry) and wetlands (polygon-based geometry). In this section, we present the results, measures, and outcomes of our pipeline when executed on the following datasets:

Railroad data We tested two datasets of vector railroad data (encoded as `MULTILINE`s) extracted from the USGS historical topographic map archive,[9][10] using the extraction methods of Duan et al. [1]. Each dataset covers a different region and includes map sheets for different points in time. The railroad data originates from a collection of historical maps for:

1. Bray, California (denoted as **CA**) from the years 1950, 1954, 1958, 1962, 1984, 1988, and 2001 (the original raster maps are shown in Figure 2.10).

2. Louisville, Colorado (denoted as **CO**) from the years 1942, 1950, 1957 and 1965.

Wetland data We tested three datasets of vector wetland data (encoded as `MULTIPOLYGON`s) that were similarly extracted from the USGS historical topographic map. Again, each of these map sheets covers a different region and spans different points in time. The wetland data originates from a collection of historical maps for:

1. Bieber, California (denoted as **CA**) from the years 1961, 1990, 1993, and 2018 (the original raster maps are shown in Figure 2.11).

2. Palm Beach, Florida (denoted as **FL**) from the years 1956, 1987 and 2020.

3. Duncanville, Texas (denoted as **TX**) from the years 1959, 1995 and 2020.

Our primary goal in this section is to show that our proposal provides a complete, robust, tractable, and efficient solution for the production of linked data from vectorized historical maps.

[9] `https://viewer.nationalmap.gov`
[10] `http://historicalmaps.arcgis.com/usgs/`

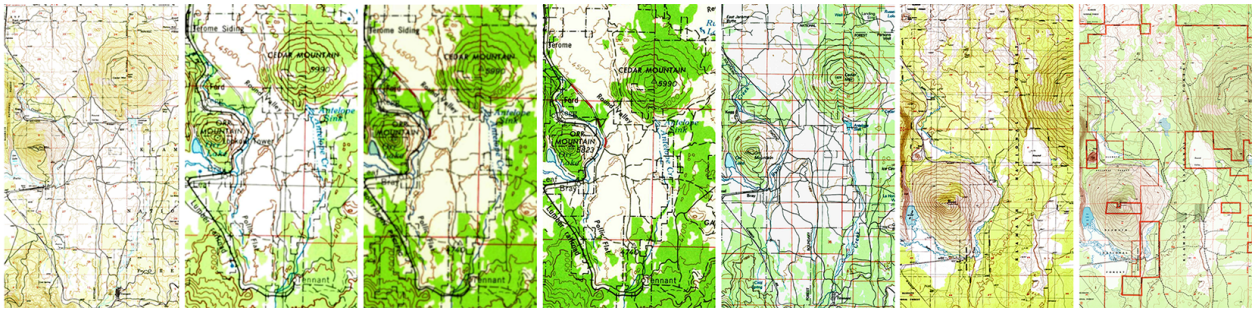


Figure 2.10: Historical maps of Bray, California from 1950, 1954, 1958, 1962, 1984, 1988 and 2001 (left to right, respectively).

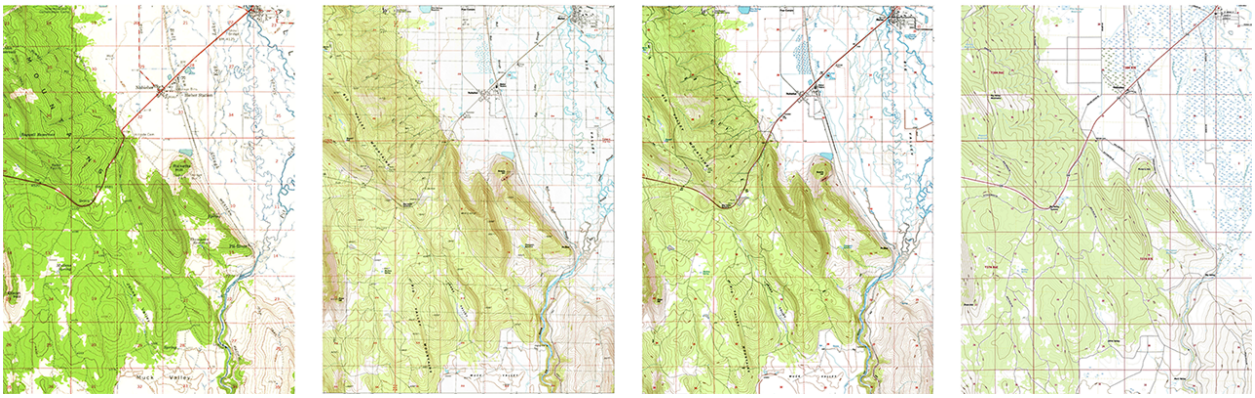


Figure 2.11: Historical maps of Bieber, California from 1961, 1990, 1993 and 2018 (left to right, respectively).

2.3.1 Evaluation on the Feature Partitioning

In order to evaluate the performance of this task, we look into the runtime and the number of generated nodes (in graph $\mathscr{G}$) for each region and feature type (executed on a 16 GB RAM machine @ 2.9 GHz Quad-Core Intel Core i7). The number of vector features in the geographic feature geometry (column '# Vectors'), resulting runtimes (column 'Runtime', measured in seconds) and total number of nodes following each sub-step of an addition of another map sheet feature geometry (column '# Nodes') are depicted in Tables 2.2 (CA) and 2.3 (CO) for the railroad data, and in Tables 2.4 (CA), 2.5 (FL) and 2.6 (TX) for the wetland data.

As seen in Tables 2.2, 2.3, 2.4, 2.5 and 2.6, we observe that for both types of geographic features, the building block geometries extracted from these maps vary in terms of "quality". That is, they have a different number of vector lines that describe the geographic feature and each one has a

Table 2.2: Partitioning statistics for CA railroads.

Year	# Vectors	Runtime (s)	# Nodes
1954	2,382	<1	1
1962	2,322	36	5
1988	11,134	1,047	11
1984	11,868	581	24
1950	11,076	1,332	43
2001	497	145	57
1958	1,860	222	85

Table 2.3: Partitioning statistics for CO railroads.

Year	# Vectors	Runtime (s)	# Nodes
1965	838	<1	1
1950	418	8	5
1942	513	5	8
1957	353	4	10

different areal coverage (the bounding box area for each feature geometry is reported in Table 2.7). This is caused by the vector extraction process and is not within the scope of this chapter. We also acknowledge that the quality and scale of the original images used for the extraction affects these parameters, but we do not focus on such issues. We treat these values and attributes as a ground truth for our process.

Table 2.4: Partitioning statistics for CA wetlands.

Year	# Vectors	Runtime (s)	# Nodes
1961	12	<1	1
1993	17	<1	5
1990	27	6	11
2018	9	6	24

Table 2.5: Partitioning statistics for FL wetlands.

Year	# Vectors	Runtime (s)	# Nodes
1987	184	<1	1
1956	531	180	5
2020	5,322	1,139	13

Table 2.6: Partitioning statistics for TX wetlands.

Year	# Vectors	Runtime (s)	# Nodes
1959	8	<1	1
1995	6	<1	5
2020	1	1	10

First, we note that the growth of the number of nodes in graph $\mathscr{G}$ is not exponential in practice due to the way the given geographic features actually change over time. Furthermore, the runtime of each sub-step is also tractable and runs only once when a new set of geometries is inserted from a new map edition. As expected, the first two map editions (for all areas) generate results within less than a minute for railroads and less than three minutes for wetlands, requiring at most three computations: one geometry intersection between two building block geometries and two additional subtractions: a local and a global one (as explained in Section 2.2.3). By inspecting Tables 2.2, 2.3, 2.4, 2.5 and 2.6, we observe that the partitioning runtime depends mostly on two factors: the number of vectors in the geometries and the number of processed maps, as we expected. The more geometry elements we have and the more geometries exist, the more operations we need to run.

These results are not surprising because "leaves" in the graph will only be partitioned in case it is "required," that is, they will be partitioned to smaller unique parts to represent the geospatial data they need to compose. With the addition of new map sheet feature geometries, we do not necessarily add unique parts since changes do not occur between all map editions. This shows that the data processing is not necessarily becoming more complex in terms of space and time, thus, providing a solution that is feasible and systematically tractable.

Additionally, by comparing the first three rows in Tables 2.2 and 2.5, we notice that the computation time over a polygon geometry is significantly slower than that of a buffer-padded line geometry. This is despite the railroads having a larger number of feature vectors. Further, by examining Tables 2.4, 2.5 and 2.6, we notice that a bigger number of feature vectors in the case of a polygon geometry causes a significant increase in processing time. These observations are predictable, as the computation time is expected to grow when dealing with more complex geometries like polygons covering larger areas that may include several interior rings in each enclosed exterior

Table 2.7: Geo-entity linking results; Area is in square kilometers.

		Area	Precision	Recall	F_1
Railroads	CA-baseline	420.39	0.193	1.000	0.323
	CA		0.800	0.750	**0.774**
	CO-baseline	132.01	0.455	1.000	0.625
	CO		0.833	1.000	**0.909**
Wetlands	CA-baseline	224.05	0.556	1.000	0.714
	CA		1.000	1.000	**1.000**
	FL-baseline	27493.98	0.263	1.000	**0.417**
	FL		0.758	0.272	0.400
	TX*	16.62	-	-	-

(similar to "holes in a cheese"), compared to the simple buffer-padded line geometries (sort of a "thin rectangle").

2.3.2 Evaluation on Geo-Entity Linking

In the process of linking our data to *OpenStreetMap*, we are interested in the evaluation of the running time and correctness (precision, recall, and F1, which is the harmonic mean of precision and recall) of this task.

As we expected, the running time is strongly dependent on the number of nodes in graph $\mathscr{G}$ (exponentially dependent on the number of the processed maps) and the block's areal coverage, which affects the number of samples using the *OpenStreetMap* API. The API response time averages 3 seconds for each sample. For the railroad data, the execution time for the set of maps from the CA region took approximately an hour (85 nodes) and only a few minutes for CO (10 nodes). This is not surprising as the CA region covers a bigger area and a larger number of nodes. We observe similar behavior in the wetland data. The FL execution time took approximately 2 hours (13 nodes), as it has the largest areal coverage (as seen in Table 2.7), while the other regions (CA, TX) took approximately 30 minutes to finish. This provides a feasible solution to a process that runs only once for a given set of geometries from different map editions.

Due to the absence of a gold standard for this mapping task, we had to manually inspect and label the sets of instances found in each bounding box that we query for each building block

geometry. The measure we present here is in terms of entity (instance) coverage. Precision and recall are calculated according to the labeled (type) instances that are available on *OpenStreetMap* and make up the inspected geographic feature (i.e., railroad or wetland). Figures 2.12 and 2.13 show an example of such instances, with their corresponding tags in OSM's graphic interface. Figure 2.12 shows a wetland instance on OSM marked as a `SwampMarsh` (with its corresponding GNIS code) and matching an active wetland area in our data (a common building block from all the editions of the CA wetland data). Figure 2.13 shows a railroad instance on OSM marked as `abandoned` and matching an abandoned railroad segment in our data (a unique building block from the 1950 edition of the CA railroad data). This shows our ability to enrich and link our graph to external resources on the web.

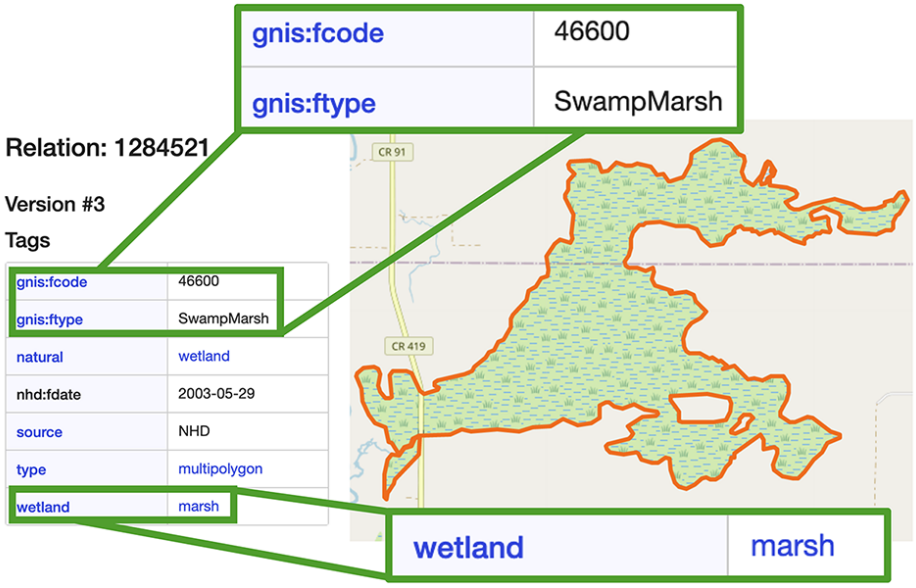


Figure 2.12: Screenshot of a wetland instance on *OpenStreetMap* matching an active area corresponding to an instance we generated from the CA wetland data.

We have set up a baseline for comparison with our geo-entity linking method. The baseline approach returns the set of all instances found in the bounding box. This is the list of candidates we generate in the first step of our method, without the additional sampling and removal steps we have described in Section 2.2.4.

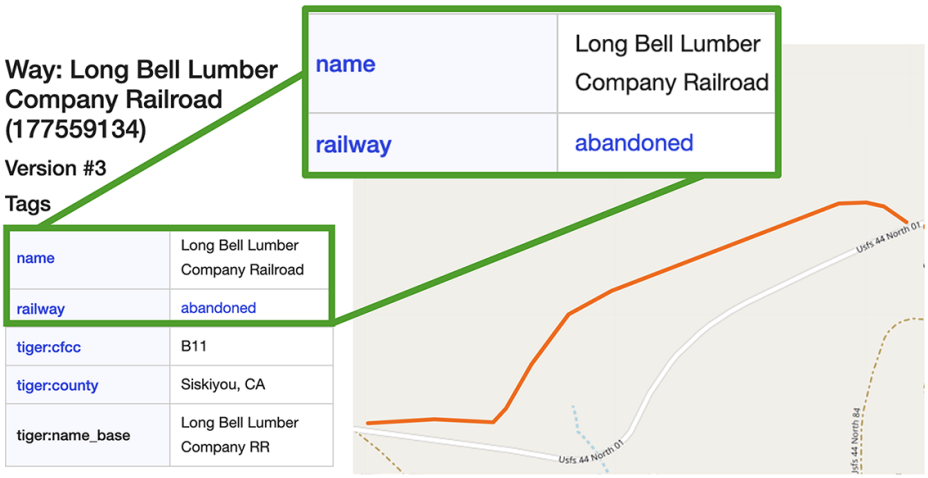


Figure 2.13: Screenshot of a railroad instance on *OpenStreetMap* matching an abandoned rail segment corresponding to an instance we generated from the CA railroad data.

The precision, recall, and F_1 scores of each method over each dataset are shown in Table 2.7. For each geographic feature and each region, we report the baseline result and our method's result. We also present the bounding box area for each dataset (in square kilometers), as it is an important factor in the geo-entity linking evaluation. The bigger the area, the more sampling points we require. The first 4 rows correspond to the railroad data (in CA and CO). The rest corresponds to the wetland data. Note, that results for the TX wetlands have been omitted in this part of the evaluation due to a complete absence of labeled data in OSM covering that area.

Due to the varying geometries, areal coverage, and available data in the external knowledge base for each region, and as expected, our measure shows different scores for each dataset. In 3 out of 4 datasets, our method achieves much higher F_1 scores than the baseline (0.774 and 0.909 compared to 0.323 and 0.625 respectively in the railroad data; and 1.000 compared to 0.714 in the CA wetland data) and achieves an acceptable score for this task. In the FL wetland dataset, we achieve lower F_1 scores for both methods (baseline and ours). This is not surprising as the area of coverage is significantly bigger than in all other datasets, requiring us to generate a bigger number of samples in order to capture all the relevant instances on OSM. Nonetheless, further examination of the FL wetland results shows that the low F_1 score of the baseline is due to the fact that it only considers the global bounding box (thus the high recall, but low precision). On the other hand, our

method achieves a higher precision score but a much lower recall, compared to the baseline. This is a crucial point in geographic applications, as many systems consider the precision to be more important than recall due to a low false-positive tolerance [36].

2.3.3 Evaluation on Querying the Resulting Data

We execute several query examples over the knowledge graph we constructed in order to measure our model in terms of query time, validity, and effectiveness. For the generated railroad data, we had a total of 914 triples for the CA dataset and 96 triples for the CO dataset. For the wetland data, we had a total of 270 triples for the CA dataset, 149 for the FL dataset, and 85 for the TX dataset.

Larger areas do not necessarily mean that we will have more triples. The number of triples depends on the geometry similarity and difference in the given data. Despite the FL wetland data covering a significantly larger area than other datasets, it did not cause notable triples growth or query performance degradation, as we show in Table 2.8.

The generated RDF triples would be appropriate to use with any Triplestore. We hosted our triples in Apache Jena.[11] Jena is relatively lightweight, easy to use, and provides a programmatic environment.

Table 2.8 shows the query-time performance results (average, minimum and maximum). In the first type of query we want to identify the feature parts that remain unchanged in two different map editions (different time periods) for each region (e.g., Listing 2.2). Each row with a label starting with `SIM-` in Table 2.8 corresponds to this type of query (the label suffix determines the tested region). We executed a hundred identical queries for each feature type and each area across different points in time to measure the robustness of this type of query.

We repeated the process for a second type of query to identify the parts of the feature that were removed or abandoned between two different map editions for each region (i.e., Listing 2.3). Each row with a label starting with `DIFF-` in Table 2.8 corresponds to this type of query.

[11] `https://jena.apache.org/`

```
SELECT ?f ?wkt WHERE {
  ?f a geo:Feature ;
    geo:hasGeometry [ geo:asWKT ?wkt ] ;
    dcterms:date 1962^^xsd:gYear ;
    dcterms:date 2001^^xsd:gYear .
  FILTER NOT EXISTS { ?f geo:sfContains _:_ } }
```

Listing 2.2: Query similar feature geometries in both 1962 and 2001.

```
SELECT ?f ?wkt WHERE {
  ?f a geo:Feature ;
    geo:hasGeometry [ geo:asWKT ?wkt ] ;
    dcterms:date 1962^^xsd:gYear .
  FILTER NOT EXISTS { ?f geo:sfContains _:_ }
  MINUS { ?f dcterms:date 2001^^xsd:gYear . } }
```

Listing 2.3: Query feature geometries present in 1962 but not in 2001.

The third type of query retrieves the parts of the feature that are unique to a specific edition of the map (i.e., Listing 2.4). Each row with a label starting with `UNIQ-` in Table 2.8 corresponds to this type of query.

```
SELECT ?f ?wkt WHERE {
  ?f a geo:Feature ;
    geo:hasGeometry [ geo:asWKT ?wkt ] ;
    dcterms:date 1958^^xsd:gYear .
  FILTER NOT EXISTS { ?f geo:sfContains _:_ }
  ?f dcterms:date ?date . }
GROUP BY ?f ?wkt
HAVING (COUNT(DISTINCT ?date) = 1)
```

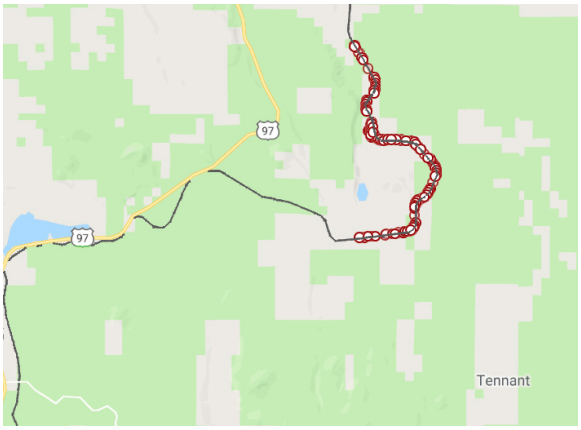
Listing 2.4: Query unique feature geometries from 1958.

Looking at Table 2.8, we notice that the average query times are all in the range of 10-48(ms) and do not seem to change significantly with respect to the number of map editions we process or the complexity of the query we compose. The query time results corresponding to the wetland data are slightly slower, but not significantly, comparing to the railroad data. This may be explained by the longer literal encoding of the WKT geometry for polygons, thus slower retrieval time comparing the the line encoding.

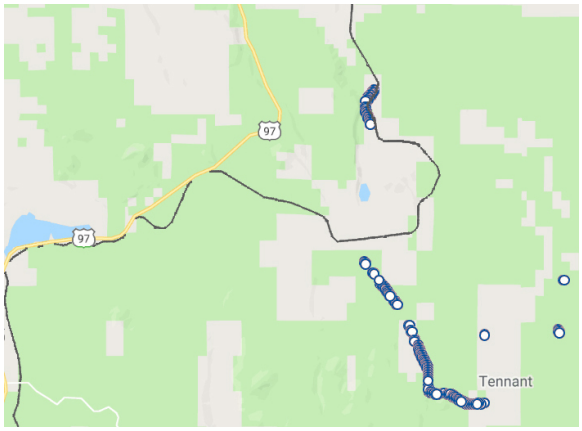
Table 2.8: Query time statistics (in milliseconds).

		Average	Min	Max
Railroads	SIM-CA	12	10	18
	SIM-CO	11	9	20
	DIFF-CA	10	8	20
	DIFF-CO	10	9	14
	UNIQ-CA	14	8	28
	UNIQ-CO	15	9	17
Wetlands	SIM-CA	22	18	34
	SIM-FL	35	18	55
	SIM-TX	21	12	44
	DIFF-CA	25	16	43
	DIFF-FL	32	18	60
	DIFF-TX	21	11	30
	UNIQ-CA	24	18	44
	UNIQ-FL	48	38	73
	UNIQ-TX	14	12	40

In order to evaluate the validity of our graph we observe the visualized results of the query presented in Listing 2.2 when executed over the CA railroad data, which are shown in Figure 2.14a. The figure shows in red the unchanged building block geometries between the years 1962 and 2001. We notice that the geometries we retrieve qualitatively match what we observe in the original vector data (the line marked in black over the maps in Figures 2.14a and 2.14b represents the current railway, which has not changed since 2001). The results of the query presented in Listing 2.3 are shown in Figure 2.14b, again, when executed over the CA railroad data. Figure 2.14b shows in blue the parts of the railroad that were abandoned between 1962 to 2001. Comparably, we perform a similar type of query and visualization for the CA wetland data, between the years 1961 and 2018. Figure 2.15a shows the similar parts in both editions (in red). Figure 2.15b shows the parts of the wetland (swamp) that were present in 1961 but are not present in 2018 (in dark blue). Again, this matches our qualitative evaluation based on the original vector files (the light blue marks that are part of the map's background depict the current swamp, thus validating our results qualitatively). The query results establish high confidence in our model, showing that we can easily and effectively answer complex queries in a robust manner.

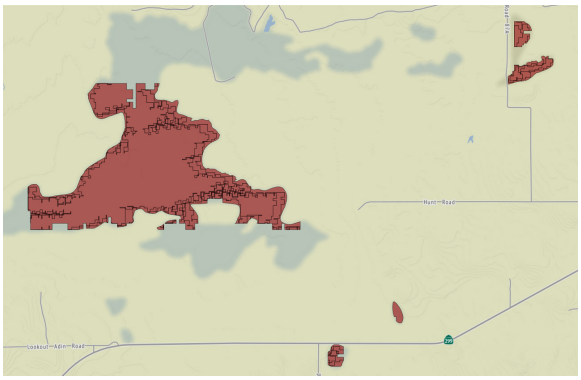


(a) The parts of the railroad in Bray (CA) that are similar in 1962 and 2001 (marked in red).

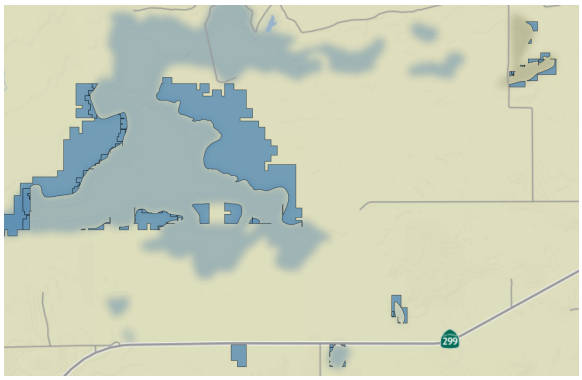


(b) The parts of the railroad in Bray (CA) that are present in 1962 but are not present in 2001 (marked in blue).

Figure 2.14: Example of railroad system changes over time, generated via SPARQL.



(a) the parts of the Big Swamp in Bieber (CA) that are similar in 1961 and 2018 (marked in red).



(b) the parts of the Big Swamp in Bieber (CA) that are present in 1961 but are not present in 2018 (marked in blue), emphasizing its decline throughout time.

Figure 2.15: Example of wetland changes over time, generated via SPARQL.

Overall, the evaluation shows that our approach is feasible and effective in terms of processing time, completeness and robustness. The partitioning process runs only once for newly added resources, and does not require re-generation of "old" data since our approach is incremental. In case a new map edition emerges for the same region, we only need to process the newly added geometry. Thus, data that has been previously published will continue to exist with a proper URI and will be preserved over time.

2.4 Related Work

Much work has been done on mapping geospatial data into RDF graphs. Kyzirakos et al. [29] developed a semi-automated tool for transforming geospatial data from their original formats into RDF using R2RML mapping. Usery et al. [30] presented a method for converting point and other vector data types to RDF for supporting queries and analyses of geographic data. The transformation process presented in these papers does not address linking the data across multiple sources or linking the source data with additional knowledge bases on the Semantic Web as described in this chapter.

Annotating geospatial data with external data on the web is used for contextualization and the retrieval of relevant information that cannot be found in the source data. This line of research has been addressed in different studies. Vaisman et al. [31] studied the problem of capturing spatio-temporal data from different data sources, integrating these data and storing them in a geospatial RDF data store. Eventually, these data were enriched with external data from LinkedGeoData [22], GeoNames [21], and DBpedia [37]. Smeros et al. [38] focus on the problem of finding semantically related entities lying in different knowledge bases. According to them, most approaches on geo-entity linking focus on the search for equivalence between entities (same labels, same names, or same types), leaving other types of relationships (e.g. spatial, topological, or temporal relations) unexploited. They propose to use spatio-temporal and geospatial topological links to improve the process.

The discovery of topological relations among a simplified version of vector representations in geospatial resources has been studied as well. Sherif et al. [39] presented a survey of 10 point-set distance measures for geospatial link discovery. SILK [38] computes topological relations according to the DE-9IM [40] standard. In contrast, RADON [41] is a method that combines space tiling, minimum bounding box approximation, and a sparse index to calculate topological relations between geospatial data efficiently. These approaches work on matching geometry to integrate vector data from two sources in a single matching process. In contrast, our approach can systematically match multiple vector datasets iteratively without performing duplicate work.

Current work on geospatial change analysis spans the construction of geospatial semantic graphs to enable easier search, monitoring and information retrieval mechanisms. Perez et al. [42] computed vegetation indexes from satellite image processing and exposed the data as RDF triples using GeoSPARQL [35]. The changes in these indexes are used to support forest monitoring. Similar to that approach, Kauppinen et al. [43] collected statistical data from open data sources to monitor the deforestation of the Amazon rainforest by building temporal data series translated into RDF. Kyzirakos et al. [10] used open knowledge bases to identify hot spots threatened by wildfire. However, this line of work does not address an incremental process of geospatial change over time. In this work, we incorporate a temporal dimension to the geospatial semantic graphs and present a pipeline for an automatic incremental geographic feature analysis over time.

Chapter 3

Constructing Geospatial Feature Taxonomies from *OpenStreetMap* Data

This section presents a method for constructing a lightweight taxonomy of geospatial features using *OpenStreetMap* (OSM) data [44]. Leveraging the OSM data model, our process mines frequent tags to efficiently produce a structured hierarchy, enriching the semantic representation of geo-features. This data-driven taxonomy supports various geospatial analysis applications. Accompanying the methodology, we release the source code of our tool and demonstrate its practical application with tailored taxonomies for California (US) and Greece, emphasizing our approach's adaptability and scalability.

3.1 Motivation

OpenStreetMap[1] (OSM) has emerged as a community-driven initiative to provide free and open access to global spatial data, making it the richest publicly available information source on geographic entities worldwide. However, using OSM data in downstream applications is challenging due to the large scale of OSM, the heterogeneity of entity annotations, and the absence of a standardized ontology to describe entity semantics [45]. Our taxonomy supports applications from automated navigation systems, which require precise geographical feature recognition for route

[1] `https://www.openstreetmap.org/`

optimization, to the classification of remotely sensed data, enhancing both the integration and utility of OSM data in sophisticated GIS applications.

Leveraging the concept of Volunteered Geographic Information (VGI) [46], OSM relies on user contributions to map the geometries and attributes of both natural and urban features. While OSM has proven to be a valuable resource, certain limitations hinder its full potential [47]. The utility of OSM data heavily relies on the consistent tagging of geographical entities by its users, as the platform does not impose restrictions on tag choices. Instead, OSM encourages its contributors to follow a set of best practices for annotation, leading to a highly heterogeneous landscape of tags. The number of tags and the level of detail for individual OSM entities is highly variable. Figure 3.1 provides an illustration of various building types within a neighborhood, selected from OSM, showcasing the most specific tags associated with them. However, OSM lacks a system that establishes relationships between these tags, hindering the extraction of valuable insights. As a result, the lack of clear semantics not only hinders the interoperability of OSM data with other datasets but also severely limits its usability in various applications. To overcome these limitations, it is crucial to establish a comprehensive taxonomy extractor from this dynamic data. This will enable better integration with other datasets and facilitate the effective utilization of the data for diverse scientific and practical purposes.

We address the limitations above and unlock the full potential of OSM data by proposing an approach to structure a taxonomy of geo-feature types from a given OSM data dump. We demonstrate this approach by creating a comprehensive and well-defined taxonomy of geospatial features derived from *OpenStreetMap* (OSM) data, and via a tool that we have made available as open source[2]. This taxonomy will enable users to understand the connections and categorization of different types of features, facilitating detailed analysis and utilization of the data. This approach enables better integration of OSM data into machine learning models and broadens its application, unlocking new opportunities to harness OSM's rich informational spectrum in diverse domains.

[2] `https://github.com/basels/osm-taxonomy`

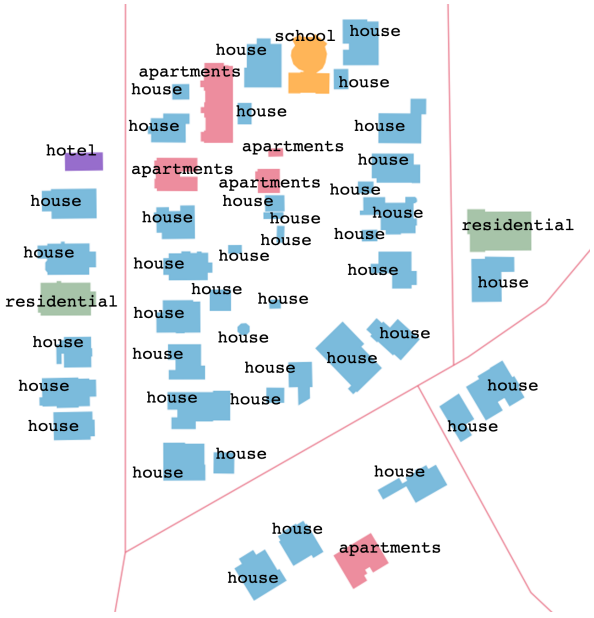


Figure 3.1: Simplified illustration of a neighborhood within *OpenStreetMap* with different `building` feature sub-types, depicting instances with type `house` in blue, `apartments` in red, `residential` in green, `school` in orange, and `hotel` in purple.

3.2 Constructing OSM Taxonomies

3.2.1 The *OpenStreetMap* Data Model

To understand our proposed approach, it is crucial to elucidate the structure of the OSM data. Initiated in 2004 as a collaborative project, OSM strives to generate a publicly accessible vector map encompassing the entire world. Remarkably successful, the project has over 10 million registered users as of March 2023. In the OSM data model, each feature is represented as one or more geometries (nodes, ways, and relations) with attached attribute data, which contains meaningful information for the taxonomy construction. Attribute information is stored as tags associated with geographic entities in the form of key-value pairs. As OSM does not prescribe a fixed set of tags, meticulous filtering becomes imperative to include only pertinent information. The comprehensiveness and diversity of features available in OSM can exhibit substantial regional variations due to the contributions of volunteers. While this data model is adequate for numerous applications,

it lacks a meaningful structure between tags or their interrelationships, which constitutes the focal point of this work.

3.2.2 Identification of Meaningful Tags

The initial phase of our methodology centers around the preprocessing of OSM and reducing the set of its attribute data into a meaningful one. This dataset consists of a wide range of tags contributed by OSM users, encompassing both suggested and self-defined tags [48]. To illustrate the magnitude of this tag diversity, let us consider a recent snapshot of OSM data for California from March 2023, where we encountered an overwhelming 3,000 unique tags. Given the extensive and heterogeneous nature of these tags, it becomes imperative to establish a concise and representative set of target labels that would serve as the foundation for constructing the taxonomy.

During the identification process, we encounter two distinct challenges. First, we need to address the issue of frequent tags that are non-informative. For instance, the `name` tag, which typically provides the name of a geo-entity (e.g., "The Ritz-Carlton"), or the `maxspeed` tag, commonly associated with road features to indicate the maximum allowable driving speed. To address this issue, we identify the most commonly used tags from a user-centric viewpoint and manually curate a set of "blacklisted" tags. This list is included with the tool and can be easily modified to accommodate different domains or specific user preferences.

Secondly, we confront the challenge of infrequent tags that may possess informative characteristics but are inadequately represented within the dataset. To mitigate this issue, we apply a frequency cutoff threshold, effectively filtering out less common and idiosyncratic tags, and focusing exclusively on the most prevalent and significant ones. For instance, consider the key-value pair `leisure=sauna` which describes a specific subtype of leisure. In the recent California OSM snapshot, this pair appeared fewer than 10 times. Consequently, it was not considered for inclusion in the final taxonomy. Through these meticulous processes, we strike a delicate balance between inclusiveness and practicality, ensuring that the resulting taxonomy faithfully represents the prominent geo-features while avoiding an unwieldy and unmanageable taxonomy structure.

3.2.3 Establishment of Hierarchical Relationships

We present our methodology for establishing taxonomic parent-child relationships among various geo-features using the OSM data. The objective is to construct a hierarchical taxonomy of labels based on frequent tag assignments. To accomplish this task, we implemented Algorithm 3, which takes the OSM snapshot data as input and produces a desired taxonomy tree data structure.

The algorithm operates as follows. Following the initial processing and removal of the undesirable tags (as described in Section 3.2.2), we iterate through the dataset, creating a key-value path counter to identify commonly occurring tag assignments (lines 3-8). These paths serve as the foundation for defining parent-child relationships within the tree structure, thereby shaping the taxonomic hierarchy. For instance, consider the set of tags {`highway=service`, `service=driveway`} which forms the path `highway--service--driveway`. In this case, the unique parent-child paths are `highway--service` and `service--driveway`.

Subsequently, we insert paths into the tree, prioritizing the most frequent ones (line 9). To maintain consistency and address any ambiguities, we handle instances where multiple paths may conflict with the evolving tree structure by favoring the more frequently occurring path and omitting the less common one (line 10). Moreover, when integrating a parent-child path into the tree, if a child tag appears under different parent tags for distinct entities, we replicate it with a unique identifier (line 13). For instance, the tag (key or value) `residential` may pertain to both `highway` and `building`; in such cases, they would be distinctly labeled as `residential_highway` and `residential_building`, respectively.

By following this process, we construct a taxonomy tree that encompasses a comprehensive representation of geo-features within the OSM dataset. The resulting taxonomy allows for a more nuanced understanding of their interrelationships.

Algorithm 3: Constructing a lightweight taxonomy.

```
   Input: osmDataset
   Output: taxonomyTree
 1 taxonomyTree = create_empty_tree();
 2 tagPathsCounter = Counter();
 3 for entity in osmDataset do
 4     tags = entity.get_tags(); // key-value pairs
 5     filteredTags = filter_tags(tags);
 6     if filteredTags is not empty then
 7         tagPath = create_tag_path(filteredTags);
 8         tagPathsCounter[tagPath]++;
 9 for (tagPath,count) in tagPathsCounter.sort(order=descending) do
10     if is_path_consistent_with_tree(tagPath, taxonomyTree) then
11         parent, child = extract_parent_and_child(tagPath);
12         if parent is not null and child is not null then
13             insert_parent_child_pair(taxonomyTree, parent, child);
14 return taxonomyTree;
```

3.3 Evaluation and Discussion

To assess the effectiveness of our proposed method for constructing a lightweight taxonomy of geographic features using OSM data, we conducted an experiment utilizing two comprehensive OSM datasets in the form of `.osm` dump (snapshot) files. Our evaluation consists of a qualitative analyses, providing insights into the resulting taxonomies generated from each dataset and comparing them on a surface level. The first dataset we employed comprised the complete California (US) OSM geo-data snapshot from March 2023[3], encompassing approximately 150 million OSM instances. Among these instances, approximately 10 million contained at least one tag, with 1 million being nodes, 9 million being ways, and around 68,000 being relations. The number of tags assigned to each instance varied from 1 to 16, with an average of 2.3 tags per geographic entity. The second dataset consisted of the complete Greece OSM snapshot from March 2023[4], which included approximately 40 million OSM instances. Around 2 million contained at least one tag, with 266,000 being nodes, 1.7 million being ways, and around 18,000 being relations. Similarly,

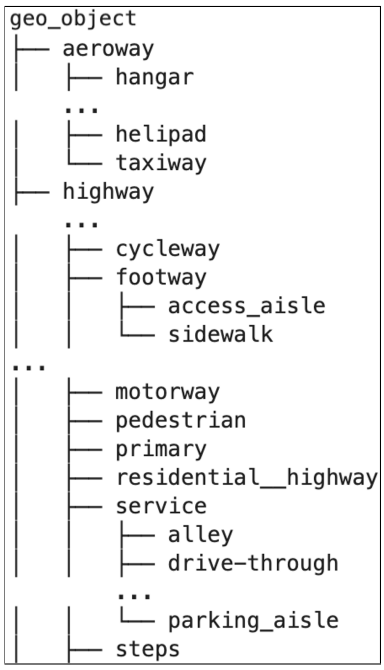
[3] `https://download.geofabrik.de/north-america/us/california.html`
[4] `https://download.geofabrik.de/europe/greece.html`

the number of tags assigned to each instance ranged from 1 to 13, with an average of 2.1 tags per geographic entity. Each dataset comprised around 3,000 unique labels. For both datasets, we established a minimum threshold of 500 instances per tag for the purpose of our analysis.

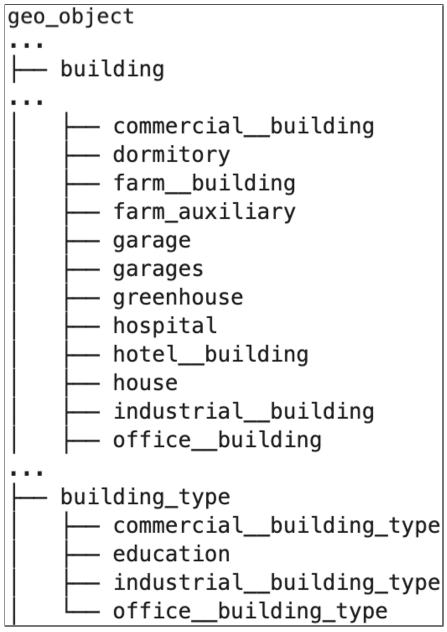
To evaluate the resulting taxonomies, we performed a qualitative analysis, examining them from a user perspective to assess their coherence and utility. The qualitative analysis revealed several positive findings regarding the constructed taxonomies. In both cases, the taxonomies successfully captured the essential characteristics of the geographic features within the OSM datasets, providing a structured and organized representation. A snippet from the resulting taxonomy generated from the California dataset is depicted in Figure 3.2a, demonstrating the hierarchical relationships between tags that facilitated the classification of diverse types of geo-features, such as `aeroway` and `highway`. This hierarchical structure enhanced the comprehension and interpretation of the roles and functions of these features. Furthermore, the taxonomy facilitated the differentiation of various sub-types within the same feature category, such as `cycleway` and `footway`, as well as different types of `service` ways, including `alley` and `drive-through`, as illustrated in the same figure.

From a human perspective, the resulting taxonomies exhibited both accurate and inaccurate taxonomic relations. It accurately captured hierarchical relationships between categories in certain domains, such as transportation (e.g., `highway - residential_highway`, `aeroway - taxiway`) and amenities (e.g., `amenity - restaurant`, `leisure - park`). The resulting relationships reflected intuitive groupings and aligned with human understanding. However, certain inaccuracies were observed in the taxonomy, likely stemming from its automatic generation. Figure 3.2b illustrates an example where the taxonomic relation between `building` and `building_type` is redundant and does not provide additional meaningful information.

Furthermore, we conducted a surface-level comparison between the resulting taxonomies derived from both datasets. This comparison highlighted how different geographical locations, such as countries or states, can yield distinct results. Figure 3.3 presents a textual comparison between the resulting taxonomies, demonstrating the differences between California and



(a) Snippet from the California taxonomy showing accurate hierarchical relationships.



(b) Snippet from the resulting California taxonomy showing redundant taxonomic terms and relations.

Figure 3.2: Taxonomy snippets from the resulting California (US) dataset.

Greece. For example, the `historic` category in Greece encompasses types of features, such as `archaeological_site` and `castle`, which are either uncommon or nonexistent in California. Furthermore, the usage of the `internet_access=wlan` tag by OSM users in Greece was much more prevalent compared to the California dataset. Additionally, the presence of the `kerb` tag category, encompassing different types of the feature (e.g., `flush` and `raised`), was observed in the taxonomy resulting from the California dataset but not in the Greece dataset. These variations in the taxonomies could be attributed to factors such as tagging style, cultural differences, and historical context. The full taxonomy text files generated from both experimental datasets are also available in our repository.[5]

The inaccuracies observed in the taxonomy can be attributed to the limitations of automatic generation. While automated approaches can be efficient, they often lack the contextual understanding and domain knowledge possessed by humans. The absence of human judgment and

[5] `https://github.com/basels/osm-taxonomy/tree/main/data`

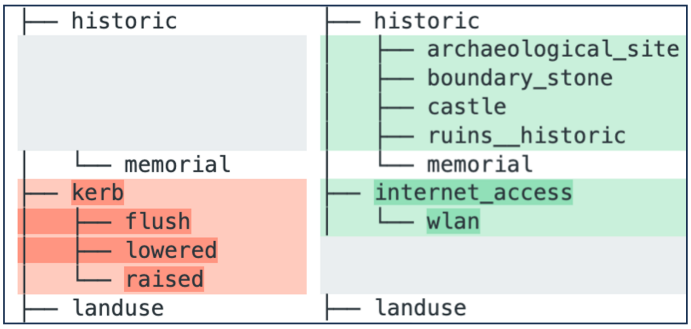


Figure 3.3: Textual comparison of snippets from two resulting taxonomies: California (US) on the left and Greece on the right.

expert curation during the automatic generation process can result in inconsistencies and illogical relationships within the taxonomy.

To enhance the accuracy of automatically generated taxonomies, while still benefiting from the efficiency of the automated process, it is useful to integrate human oversight and expert input at key stages. Combining automated techniques with strategic human validation and refinement can help identify and rectify inaccuracies without undermining the automation's extensive groundwork. Moreover, incorporating domain-specific knowledge and user feedback can further improve the quality and coherence of the generated taxonomy.

The semantic representation of the taxonomy offers a meaningful utility for OSM, addressing the limitations associated with unstructured tags, noise, inconsistencies, and the requirement of domain knowledge within the OSM suggested schema, which is vast and constantly evolving. Consequently, it provides a comprehensive framework for categorizing different types of features.

3.4 Related Work

Various approaches have been employed to construct ontologies suitable for geographical data, introducing more structure. Sun et al. [49] have developed a three-level ontology for geospatial data that, although potentially reusable, requires completion and quality assessment through manual work. Similarly, Codescu et al. [50] have created OSMonto, an ontology for *OpenStreetMap* tags that facilitates the exploration of tag hierarchies and relationships with other ontologies, but

also requires manual effort. In contrast, our research initially surpasses ontology development by automatically constructing a lightweight taxonomy as a foundational step, which is then refined with minimal human intervention. This balance of our approach being primarily automatic while still benefiting from human expertise not only sets it apart from these works but also leads to a more targeted representation of geospatial features, enhancing the analysis and utilization of OSM data.

In the domain of leveraging OSM data and constructing structured taxonomies for geospatial features, WorldKG [51] is a geographic knowledge graph that provides a comprehensive semantic representation of geographic entities from OSM. Dsouza et al. [52] further leveraged WorldKG to develop a neural architecture that exploits a shared latent space for effective tag-to-class alignment of OSM entities. Building on these pivotal contributions, our methodology enriches this line of research by dynamically constructing taxonomies that can assimilate OSM data from any time frame, ensuring an up-to-date and adaptive representation, thus underlining the enduring significance of data alignment and structure.

Chapter 4

Contextual and Spatial Embeddings for Geo-Entity Typing

In Chapter 2, we laid the groundwork by developing a methodology to transform vectorized topographic historical maps into spatio-temporal knowledge graphs (KGs). This effort primarily focused on capturing the dynamic changes of specific geographical features, such as railroads and wetlands, over time. The known types of these features are pivotal for the retrieval of candidates for entity typing and resolution tasks, such as the one described earlier in Section 2.2.4. In Chapter 3, we presented a method to utilize crowd-sourced data such as *OpenStreetMap* to organize geo-feature labels into a structured geo-feature taxonomy.

In this chapter, I present a novel methodology for embedding geo-referenced vector data, addressing the challenge of fully interpreting these raw representations. By utilizing representation learning techniques, this approach leverages geometric, spatial, and semantic contexts via neighboring spatial entities, enabling the precise identification and classification of geo-entity types. The methodology was evaluated using *OpenStreetMap* data, achieving an F_1 score of approximately 0.85 in linking geo-referenced vector data with their appropriate semantic types in Wikidata, thereby outperforming existing state-of-the-art models.

This chapter bridges the initial data transformation described in Chapter 2 and sets the stage for further advancements in data integration based on entity embeddings. Such process will facilitate more sophisticated data integration, including the structured knowledge in the form of a label taxonomy that we acquire in Chapter 3. In turn, this enhances the granularity and accuracy of

our KGs, and allows for more precise spatial queries and improved decision-making in further downstream applications.

4.1 Motivation

While technological advancements have made significant improvements, analyzing individual features and entities, even in digitized vector format, remains notably time-consuming [2, 9]. Figure 4.1 provides some examples of vectorized geo-entities extracted from different maps and studies, categorized by their types. The figure highlights the various shapes and footprints that could offer some spatial knowledge than can be exploited for data understanding. Accurately capturing and encoding these geo-entities enables easier data understanding such as geo-entity typing and linking. This can support further GIS applications with the integration of resources on the web, and additional tasks, such as domain-specific semantic typing and knowledge graph construction.

(a) Bodies of water (b) Waterways (c) Residential buildings

Figure 4.1: Examples of geo-instance shapes and footprints, encoded as vector data and categorized by type.

The multi-dimensionality of the geo-referenced vector data, including its spatial and contextual aspects, challenges data integration systems that require automatic understanding, such as those involving digitized maps and remote sensing data [2, 12, 23]. Organizing, classifying, and linking this data is complex, necessitating strategies for semantic- and spatially-aware interpretation and alignment of geo-entities to enhance their utility across scientific domains.

Studies are exploring the abundant geospatial information available on the web for enhanced data understanding and entity-linking with geospatial entities on the web [11, 12, 44]. *OpenStreetMap*[1] (OSM), emerging as a significant open knowledge base on the web, houses an expansive repository of crowd-sourced geospatial data obtained through collective efforts of a vast network of contributors. With its rich yet non-ontologized tagging system, OSM encompasses semantic annotations and a wealth of metadata. Figure 4.2 shows an example of a geo-instance on OSM with the user-assigned tags `natural=water` and `water=reservoir`.

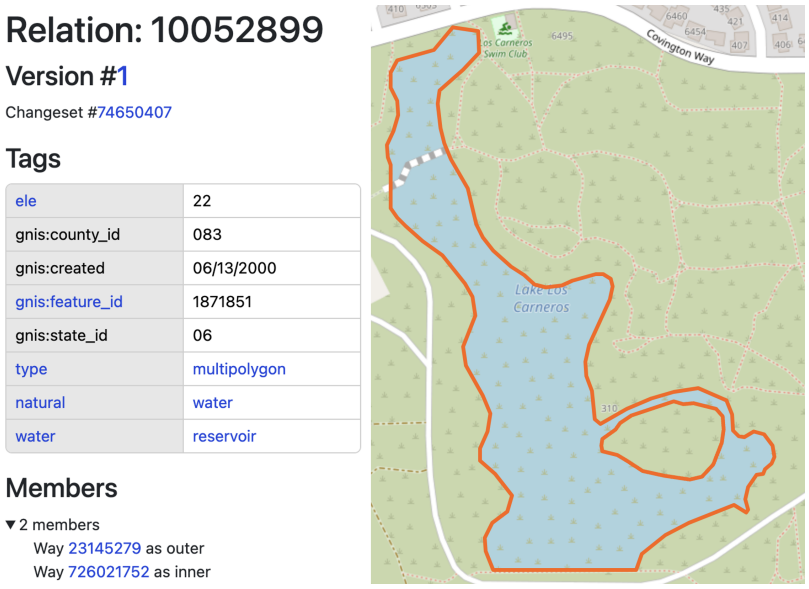


Figure 4.2: An *OpenStreetMap* instance depicting a geographic feature labeled with key tags `natural=water` and `water=reservoir`, offering vital crowd-sourced information for data understanding, structuring, and integration.

In this paper, we present a novel approach using machine learning and exploiting open geospatial data to embed geo-entities, exemplified by the classification of OSM geo-instances into their matching semantic types on OSM and on Wikidata [34]. The representation of a geo-entity faces challenges from its multi-dimensional nature, including its proximity to various other geo-entities. For example, the location of a building relative to other structures, roads, or natural features like rivers, can influence its function and size. We explore the use of representation learning and embedding techniques in geospatial data analysis. Embedding aims to automatically learn a mapping

[1] `https://www.openstreetmap.org/`

function $f : O_{geo} \mapsto \mathbb{R}^d$ from an object of interest to its vector representation. Here, d denotes the dimensionality of the latent space, where the entity is projected. The resulting representation can be used in various geospatial tasks, such as remote sensing classification and historical data linkage, as we mentioned earlier.

This approach has shown success across various domains in natural language processing and computer vision, where vector representations are created for words [53, 54], sentences [55], documents [56], and knowledge graphs [57, 58], using neural networks trained on large corpora of text. Similarly, convolutional neural networks (CNNs) enable robust data understanding for applications like image classification, object detection, and segmentation [59, 60].

Our approach combines geometric, spatial, and semantic neighborhood context encodings to train a model that generates robust geo-entity embeddings for geo-entity typing tasks. The methodology is based on self-supervised learning within a contrastive learning framework [61, 62], employing a CNN and neural architecture for the embedding model. In our implementation, we optimize the loss function by selectively weighting negative instances based on a taxonomy matrix describing the relationships between tags in semi-structured tag data such as OSM, as well as employ shape augmentation to enhance model generalization.

In order to evaluate the effectiveness of our approach, we conducted experiments on a real-world dataset containing geo-referenced vector data. Our evaluation involved classifying geo-entities into their respective Wikidata classes and OSM tags and classes.

Significance. This chapter makes the following contributions:

1. We present a novel self-supervised embedding method for geo-entities that combines geometric, spatial, and semantic contexts. We employ open data from the web, particularly *OpenStreetMap*, to characterize the geo-entity context.

2. We implement a weighted contrastive learning framework for our model, incorporating a taxonomy-informed loss function that assigns weights to negative pairs based on their dissimilarity within a specified taxonomy.

3. We evaluate our method on two datasets that span a diverse set of geographic features. We also make our source code and data publicly available[2] as a contribution to the broader research community.

4.2 Embedding Geo-Referenced Vector Data

The task at hand can be classified as a geospatial entity embedding and representation challenge to enable geo-entity typing and classification. Ultimately, we wish to classify geo-referenced, Well-Known Text (WKT) representation of geospatial objects or entities into a set of semantic types in a given dataset. WKT is a text-based format used to represent geometric objects, such as points, lines, and polygons, in geospatial data [63]. A geo-referenced WKT contains a sequence of spatial coordinates of the object in vector format along with additional attributes such as the geometry type (e.g., `LINE`, `MULTILINE`, `POLYGON`, `MULTIPOLYGON`). The geo-referenced representation captures the multi-dimensionality of the data, providing the neighborhood context through its geo-coordinates and spatial and shape data through its scale and the layout of the vector data itself. Constructing a robust representation for geo-entities poses a challenge in accurately capturing the spatial, geometric, and semantic context of geo-entities and their interrelations while leveraging available open data and knowledge that we can exploit. Figure 4.3 shows a visualization of the embedding architecture operating over the input datat in WKT format.

4.2.1 Representation Learning Model

Our architecture pivots on three main components: the shape encoder, the spatial attributes encoder, and the contextual neighborhood encoder, as depicted in Figure 4.3. The shape encoder aims to capture the geometric characteristics of the geospatial object, while the spatial attributes encoder extracts measurable attributes, such as area and length, employing standard spatial computational methods. Conversely, the neighborhood encoder generates a feature vector based on

[2] `https://github.com/basels/GeoEntityContextNet`

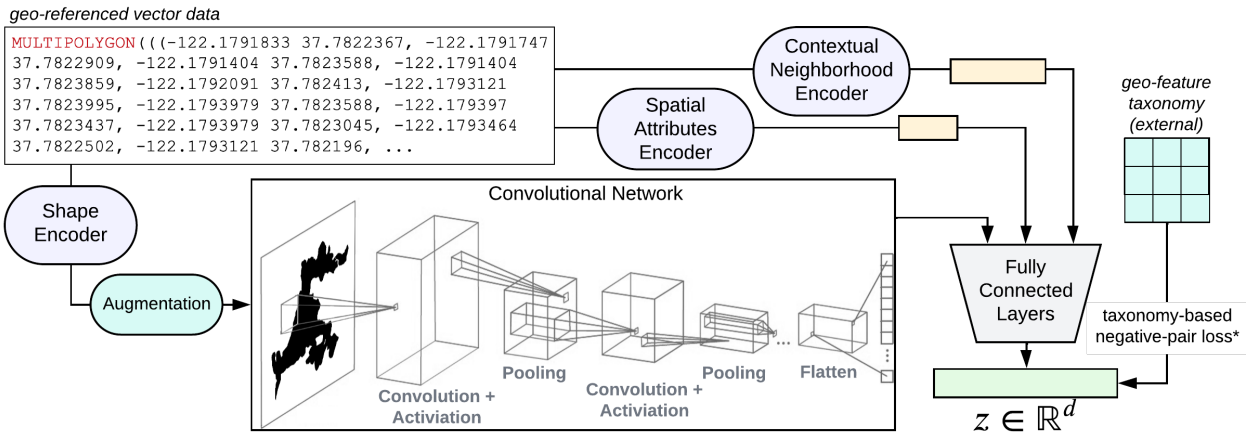


Figure 4.3: Illustration of the geo-entity encoding and embedding architecture, integrating shape, spatial, and neighborhood information, with auxiliary components depicted in blue and the resulting output, representing the latent vector, in green.

the semantic types of the neighboring geo-entities relative to the entity under consideration. To comprehensively represent the geospatial entity, we utilize the output from each encoder to train the primary embedding model.

4.2.1.1 Extracting Geometric and Spatial Features

Our approach holistically encodes the geo-entity's shape information, ensuring that it is not constrained by memorizing the positions of training examples. Addressing the heterogeneity and variable length of vector data, we generate a "footprint" outline for each entity to learn its shape characteristics. The WKT representation is discretized into a fixed 200×200 binary two-dimensional array, serving as a single-channel binary raster, and identified as the minimum resolution that adequately depicts lines and multi-lines, rendering visually perceptible. As depicted in Figure 4.3, we incorporate augmentation during training by applying various image transformations such as resizing, sharpness adjustments, rotations, and flips, to enhance model robustness and ensure generalization across diverse geo-entities.

Simultaneously, spatial attributes encoding focuses on extracting and integrating size and length attributes of the geo-entities using established geospatial tools.[3] Integrating the size and

[3] `https://shapely.readthedocs.io/en/stable/`

length attributes is crucial for the final embedding model, as the geometric shape encoding component only considers the feature's shape while ignoring its scale. By considering both shape and size attributes, we capture the nuances of the differences in features in their final embedding.

4.2.1.2 Neighborhood Contextual Semantic Encoding

To efficiently materialize the neighborhood context of a specified geo-entity, our encoder embeds the relative positions of each neighboring feature for the target entity, ensuring comprehensive encapsulation of the data.

Figure 4.4 provides a visual illustration, delineating an anchor feature (e.g., `school`, highlighted in orange), and its neighborhood context — a collection of geo-features surrounding it at varying distances and with different type labels shown in different colors. We employ a "bag-of-features" vector encoding to capture the spatial relationships among the geospatial entities, using a distance-based encoding method to generate a "bag-of-distances" feature vector. This feature vector encodes the relative shortest distance to every recognized geo-type within the neighborhood, preserving relative distance and directionality information between entities, and serveing as an additional input to the model, along with the geometric and spatial features.

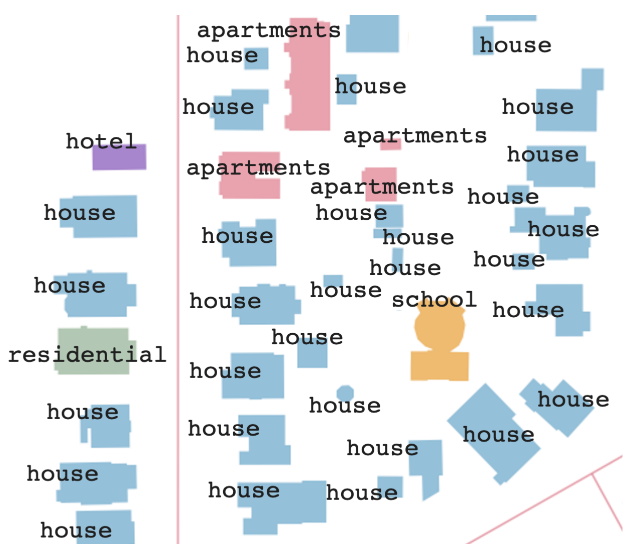


Figure 4.4: Illustration of a neighborhood, with anchor entity (`school`) in orange. Surrounding features include `house` in blue, `apartments` in red, `residential` in green, and `hotel` in purple.

To comprehensively construct a neighborhood encoding, a spatial knowledge base or database are essential. The knowledge base is merely used to fetch the entities in the context "window". In this work, we specifically utilized OSM to retrieve neighboring geo-instances — including nodes, ways, and relations — within a defined distance threshold (a model hyperparameter) from the geo-referenced center of the entity.

4.2.2 Taxonomy-Guided Contrastive Learning

Integrating taxonomic information about geo-feature types into the learning framework can act as an auxiliary tool to encode additional semantic knowledge. This taxonomic knowledge, which could come in the form of an ontology, is not only beneficial in offering a systematic classification but also instrumental in a learning setting, where distinguishing between varied geo-feature types, such as `commercial` and `residential` buildings or `motorway` and `primary` highways, can be leveraged to distinguish between different types of "negatives" in a contrastive learning framework. Furthermore, recognizing sub-type relationships, like `beach` being a subtype of `natural`, can help with learning better generalizations in the vector representations of the parent features. In this context, distances between "leaves" in the taxonomy tree are leveraged to create a dissimilarity measure for each leaf-pair, providing a tangible and meaningful way to quantify the penalty between samples in the loss function. This incorporates a more structured and hierarchically informed representation of data, enabling the quantification of similarities and dissimilarities between samples Figure 4.5 illustrates a simplified example of this concept.

Navigating the hierarchical maze of OSM tags and labels presents a nuanced challenge due to their diverse granularity. Given the dynamic set of tags in OSM, rigorous filtering is essential for selecting beneficial labels for taxonomy-guided self-supervised training within our framework. To this end, we developed a lightweight taxonomy of OSM tags [44], as detailed in Chapter 3. This taxonomy, constructed using OSM data, employs the OSM data model to mine frequent tags, thus creating a multi-level hierarchy that enriches the semantic representation of geo-features. This data-driven approach not only organizes geospatial data into a well-defined hierarchy but also

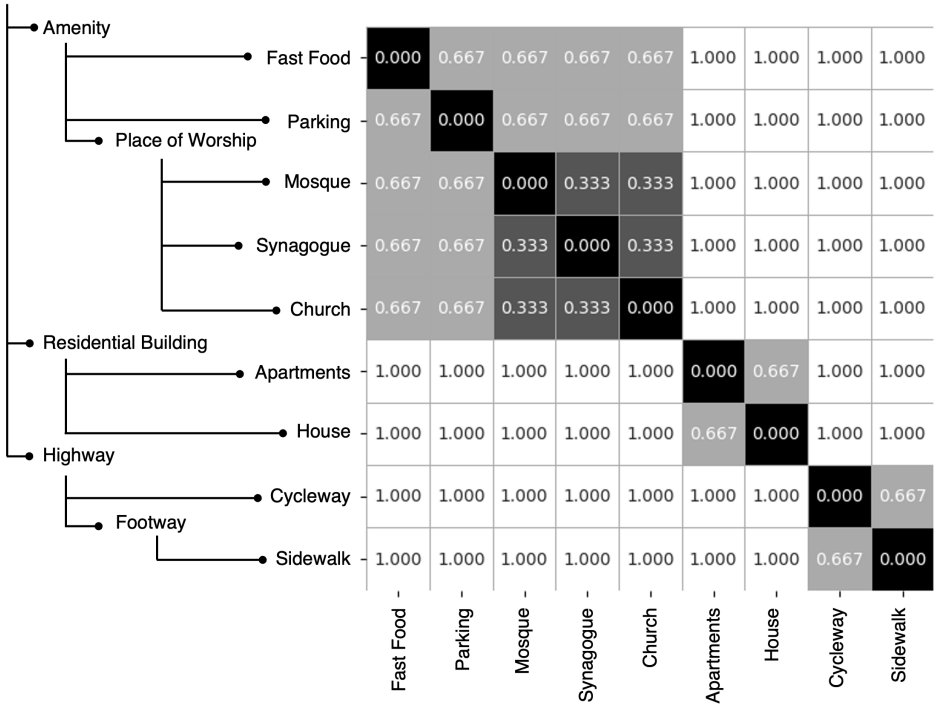


Figure 4.5: A simplified example of a taxonomy matrix employed within the loss function, illustrated with accompanying sample weights to convey the concept. On the left, the taxonomy tree demonstrates the hierarchical relationships among chosen geo-features; on the right, the dissimilarity matrix quantifies their respective taxonomic distances. Diagonal elements denote similar entities with zero dissimilarity, while off-diagonal elements quantify dissimilarity, reflecting the varying taxonomic distances between different entities.

supports a wide range of geospatial analysis applications. The methodology behind this taxonomy construction is detailed in Chapter 3, the source code of this tool is released, and its practical application is demonstrated with tailored taxonomies for regions like California (US) and Greece, underscoring the adaptability and scalability of this approach. We also incorporate the work by Dsouza et al. [52] to link some of the target labels to corresponding classes in Wikidata.

We incorporate the taxonomy weights into the Normalized Temperature-scaled Cross Entropy loss function [64, 65], which we define as follows. For each anchor entity e_q in a given batch, the taxonomy-aware loss is calculated with respect to the positive and negative samples in the set, and is given by:

$$L_q = -\log \frac{\exp(sim(e_q, e_+)/\tau)}{\sum_{i=0}^{K} \exp(sim(e_q, e_i) \cdot w_{q,i}/\tau)} \tag{4.1}$$

where e_+ is a positive sample, $sim(e_i, e_j)$ is the cosine similarity between the normalized embeddings of entities e_i and e_j. The temperature τ scales the similarity scores. The sum is over one positive and K negative samples. $w_{q,i}$ is the weight representing the taxonomic distance of labels between e_q and a negative sample e_i. The taxonomic weight $w_{i,j}$ is defined by the relative distance within the taxonomy tree as:

$$w_{i,j} = \frac{d_{\text{tree}} - d_{i,j}}{d_{\text{tree}}} \tag{4.2}$$

where d_{tree} is the depth of the taxonomy tree, and $d_{i,j}$ is the depth of the common ancestor of entities i and j. This normalization ensures that weights adjust the influence of negative samples in the loss function to reduce penalty of misclassifying entities to similar but still incorrect classes.

In the grand scheme of the embedding model, three distinct data inputs are combined to train a mapping function. This function learns to differentiate various geo-instances in a low-dimensional vector space based on their respective types in the taxonomy. Consequently, the resultant embeddings can drive a classifier that effectively discriminates between target semantic types, as demonstrated in Section 4.3.

4.3 Evaluation and Discussion

We evaluate the effectiveness of our proposed geo-entity embedding approach by training a model under various settings of our methodology and comparing it to two baselines, including the state-of-the-art (SotA) in geo-entity embedding. Each model was evaluated through a classification and semantic typing task using two distinct datasets. The objective is to explore how different types of information affect the performance in our approach, as an ablation study, and to test our best performing model against other systems, aiming to gain insights into the generalizability of the model and its proficiency in the overall task of semantic typing.

4.3.1 Experiment Setup

Data. Consistently across all settings, our model was trained using the same data, which encompassed 200,000 OSM instances from the California OSM snapshot[4]. We utilized linear and polygonal features, whilst excluding discrete point-based features.[5] Currently, this comprehensive dataset encapsulates around 150 million instances, of which about 10 million contain at least one tag. Instances were tagged with 1 to 16 labels, resulting in an average of 2.3 tags per instance. While the dataset originally featured over 3,000 unique OSM tags, this was filtered down to 75 following the process described in Section 4.2.2. Additionally, we utilized tools[6] provided by the OSM community to perform basic data conversions of the WKT format. The neighborhood contextual semantic encoding, encapsulated in a flat vector of size 278 captures the array of feature types its surrounding area. A single binary channel represents the shape, while the spatial attributes are encoded as literals.

The classification test datasets utilized were crafted by separately sampling from OSM. We ensured that the geo-instances in the test datasets were not present in the training data. The first dataset, WD-2k, comprises 2,146 instances with direct mapping to their Wikidata classes (based on the Wikidata instance labeled by OSM users), covering 11 distinct classes. The second dataset, OSM-16k, consists of 16,059 instances that span 18 OSM "classes" (most fine-grained tag per instance). Both datasets are publicly available via our repository.[7]

The resulting embedding were tested using Support Vector Classification, which rendered the best results comparing with other classifiers like Random Forest, K-Nearest Neighbors, and Logistic Regression. Model evaluation was measured in precision, recall, and F_1 scores, utilizing 8-fold cross-validation to divide the data into mutually exclusive subsets (87.5% training; 12.5% testing).

[4] `https://download.geofabrik.de/north-america/us/california.html`

[5] This exclusion is due to the lack of geometric or spatial value in point features, given their zero-dimensional nature, which negates the possibility of measuring attributes such as length, area, or shape. Point features are commonly used to represent intangible entities like locations or place names, whereas linear and polygonal features contain information on physical phenomena with spatial extent, to this reason, point data were omitted from our evaluation.

[6] `https://wiki.openstreetmap.org/wiki/Software_libraries`

[7] `https://github.com/basels/GeoEntityContextNet/tree/main/data`

Experimental Settings. We evaluate our model performance under varying conditions using four variant settings. The first setting focused solely on shape information, excluding any neighborhood information or spatial attributes, while the second setting incorporated both shape and spatial data, adding a spatial encoder to include its area and length. We included shape, spatial, and contextual neighborhood data in the third setting but did not consider taxonomic relations. The fourth and final setting combined shape, spatial, semantic, and taxonomic data to further enhance model performance.

Model Training. The model's hyperparameters were systematically determined through an iterative process of experimentation. A neighborhood size of 15" degrees (equivalent to approximately 450 meters or 1,500 ft) emerged as optimal for semantic contextualization. A learning rate of 10^{-5} and weight decay of 0.05 were chosen to facilitate model stability throughout training. To accommodate the computational constraints of the available hardware resources, which included four NVIDIA GeForce RTX 2080 Ti GPUs and an Intel i7 CPU, providing 4,352 cores and 11 GB DDR6 memory per GPU, the batch size was established at 32. The model was trained for 100 epochs. Additionally, the hyperparameter d, representing the dimensionality of the latent vector, was set to 300, to maintain a consistent metric space and enable a fair evaluation against the SotA baseline model.

Baselines. To establish robust baselines for our study, we include two additional settings. First, we utilize GeoVectors [66] as a baseline, a pre-trained corpus of OSM embeddings, given its standing as the nearest SotA model trained to navigate analogous challenges of embedding geo-entities. GeoVectors was trained by leveraging two models: a neural location model for spatial relations and a pre-trained word embedding model to encode semantic similarities based on tags.

Additionally, we explore the capabilities of Large Language Models (LLMs) in a zero-shot classification setting to assess their performance with geographic data. Specifically, we used natural language queries to provide the transformer-based model, GPT-3.5 Turbo [67], with classification candidates and their descriptions, alongside the geo-referenced input vector data in its source WKT format, to generate an answer regarding the semantic type.

We evaluate the effectiveness of our proposed geo-entity embedding method by training a model under different settings. Each setting was subsequently tested through a classification task using two distinct datasets. The objective was to explore how different types of information affect performance, striving to yield insights into the model's generalizability and overall task of semantic typing.

4.3.2 Results and Discussion

We present the results of our experiments across the settings described above and discuss their implications for the effectiveness of our proposed method for semantic typing.

4.3.2.1 Overall Performance.

Table 4.1 shows the results for each setting across both datasets. In our baseline, Setting 1, we solely relied on geometric shape data for classification, which resulted in F_1 scores of 0.501 for WD-2k and 0.492 for OSM-16k. Introducing the spatial attribute encoder in Setting 2, the scores elevated to 0.525 and 0.513, respectively.

Remarkably, when both contextual neighborhood data and geo-entity type taxonomy were incorporated (Settings 3 and 4), performance was significantly boosted. Setting 4, which combines all these inputs, yielded the most impressive results: F_1 scores of 0.850 for WD-2k and 0.856 for OSM-16k. Interestingly, the peak precision was observed in Setting 3, where taxonomic data was omitted. This phenomenon suggests that in our non-guided contrastive learning, treating all negatives uniformly — as opposed to a weight-based approach in Setting 4 — results in finer distinctions between all entity types. This could be explained by the higher total (negative) loss per

Table 4.1: Summary of results for semantic-type classification in all experimental settings, across both datasets.

		WD-2k			OSM-2k		
Setting	Method	Precision	Recall	F_1	Precision	Recall	F_1
1	$\text{Ours}_{\texttt{shape}}$	0.497	0.506	0.501	0.473	0.512	0.492
2	$\text{Ours}_{\texttt{shape+spatial}}$	0.506	0.545	0.525	0.491	0.536	0.513
3	$\text{Ours}_{\texttt{full}}$	**0.850**	0.823	0.836	**0.877**	0.725	0.794
4	$\text{Ours}_{\texttt{full w/taxonomy}}$	0.849	**0.852**	**0.850**	0.858	**0.854**	**0.856**
	GPT-3.5-Turbo	0.198	0.209	0.121	0.145	0.063	0.026
	GeoVectors [66]	0.819	0.834	0.826	0.833	0.815	0.824

epoch, as observed in Setting 3 compared to Setting 4. These findings underscore the nuanced balance and interplay between incorporating varied data sources and managing the complexity of the learning environment to produce the embeddings, ultimately driving the performance and precision of the final geo-entity classification task.

A comparison with the SotA model shows that our model outperforms SotA on both WD-2k and OSM-16k dataset. Notably, our model achieved better results on OSM-16k, where classification was aligned with OSM tags — a logical outcome given the model's training on a this data source. This distinction is even more significant considering the added complexity in the OSM-16k task, with 18 classes versus 11 in WD-2k, entailing a robust representation across the data. However, the GPT-3.5 Turbo, in a zero-shot setting, scored lower with an F_1 score of 0.121 on the WD-2k dataset and only 0.026 on the OSM-16k dataset, highlighting challenges in adapting LLMs to spatial semantic tasks without a domain-specific and tailored training.

It is important to note that our model was constructed without embedding direct semantic information or OSM tags about the geo-entity in the self-learning process, focusing solely on geometric, spatial, and neighborhood contexts. In contrast, GeoVectors incorporated such semantic data, including Wikidata connections, subtly giving them an advantage. Ultimately, the results show our method's advantage.

4.3.2.2 Analysing the Optimal Setting.

Figure 4.6 shows the per-class confusion matrix results for the WD-2k dataset utilizing our method's optimal setting (Setting 4). An initial analysis indicates that our model exhibits exceptional performance across most classes, notably achieving the highest scores for `light rail line`, `limited-access road`, and `stream`. This implies that our model more adeptly distinguishes linear features than polygon-based features. Various factors could contribute to the fact that `light rail line` secured the highest recall score among other linear features. This could be due to the distinctive geometric and spatial characteristics of light rail lines, which often display a "twisting", elongated shape and inhabit distinct environments compared to other linear features in urban areas.

There were some challenges in differentiating between particular classes. For instance, 47.8% of `school` features are misclassified as `high school`, and 27.6% as `park`. While `high school` (Q9826) and `school` (Q3914) are distinct in the labeling scheme, a human annotator might perceive one as a subclass of the other, rendering the task potentially redundant. The model's capability to classify and meaningfully capture numerous fine-grained `high school` instances is noteworthy, lending qualitative confidence to its ability to differentiate between nuanced types that may exist under a broader, shared geo-feature taxonomy. The similarity between `school` and `park` may originate from their shared attributes (e.g., similar shape footprints and neighborhood environment of features), posing a challenge to accurate classification without further entity-specific knowledge.

Additional observations indicate that the model occasionally misinterprets `lake` instances as `reservoir`, a plausible error given the similar footprints and environmental roles of these water bodies. Likewise, `street` is confused with `limited-access road` 11.2% of the time, a mistake potentially stemming from geometric similarities and proximities to analogous geo-feature types.

Figure 4.7 shows the per-class confusion matrix results for the OSM-16k dataset utilizing our method's optimal setting (Setting 4). The results suggest that the model effectively captures the defining characteristics and contexts of almost all 18 geo-feature types, with particularly strong true

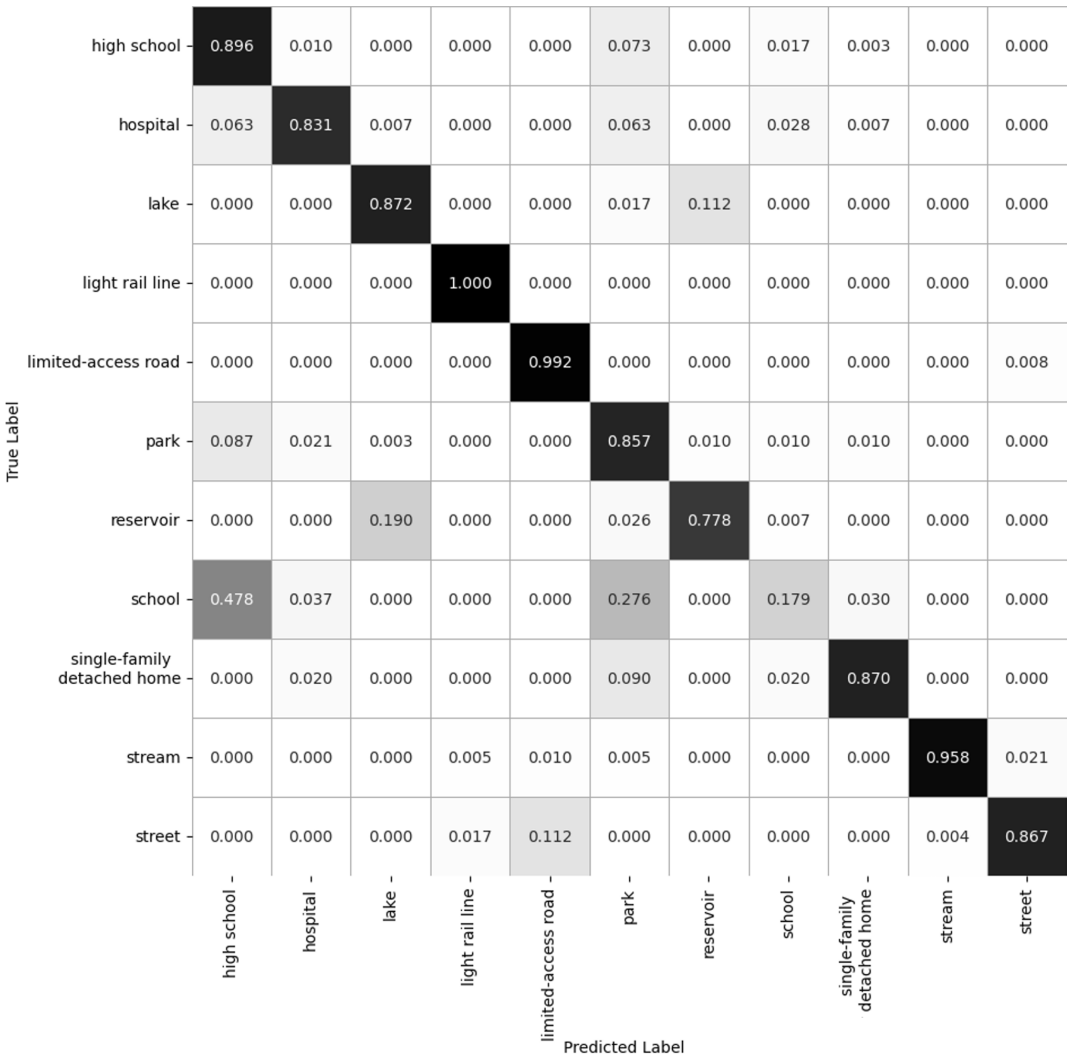


Figure 4.6: Confusion matrix illustrating classification results of geo-entities to Wikidata types using the WD-2k dataset, employing the model derived from the optimal setting (Setting 4). The matrix aggregates results across all mutually exclusive subsets of tests.

positive rates as indicated by the high scores (dark shades) along the diagonal. Certain classes, such as `track_leisure`, `beach`, and `golf_course`, show a high degree of predictive accuracy. However, some classes like `commercial_landuse` and `parking_amenity` demonstrate significant confusion with other amenity and building types, often being misclassified as `retail_building` and `retail_landuse`, respectively. This could point to an overlap in the feature space or insufficient differentiation between these feature types.

The dashed outlines around entity clusters in the confusion matrix in Figure 4.7 represent groups with a common tag "ancestor", highlighting the taxonomic hierarchy. Notably, confusion

between entities is more frequent within these clusters than between them, indicating the model's proficiency in distinguishing general tags (buildings vs. natural features) than closely related tags (building types).

Enhancements to our method can be incorporated via supplementary information. For instance, integrating satellite imagery and aerial photography could furnish additional details regarding polygon land use and environmental characteristics, facilitating improved differentiation between similar classes.

4.3.2.3 Visualizing the Latent Space

Furthering our understanding of the model's performance, we employed t-Distributed Stochastic Neighbor Embedding (t-SNE) to plot the embeddings of 10,000 geo-entities from OSM, as depicted in Figure 4.8. Additionally, Figure 4.9 provides a comparative view to the detailed tags illustrated in Figure 4.8, showcasing labels of the identical data points at the highest level of the OSM tag taxonomy. Each figure displays notable separation among various classes. Evaluating ground truth labels at a higher taxonomy level unveils noteworthy clustering, supplying additional qualitative evidence supporting the model's generalizability. The t-SNE plot shows that the clusters representing distinct classes have minimal overlap, affirming the model's capacity to discern inherent patterns in the data. However, it is vital to note that t-SNE, a two-dimensional representation suited for visualizing high-dimensional datasets through dimensionality reduction, may incur some loss of information during projection. Nonetheless, the visualization serves as a valuable tool to assess the quality of the embeddings and gain insight into the interrelations among diverse classes.

In summary, our proposed method for semantic embedding utilizing multi-faceted learning has yielded encouraging results, adeptly capturing spatial, geometric, and neighborhood information about geo-entities. The precision, recall, and F_1 scores, in conjunction with the confusion matrix and t-SNE visualization, illuminate the strengths and potential areas for refinement within our method, thereby guiding future enhancements.

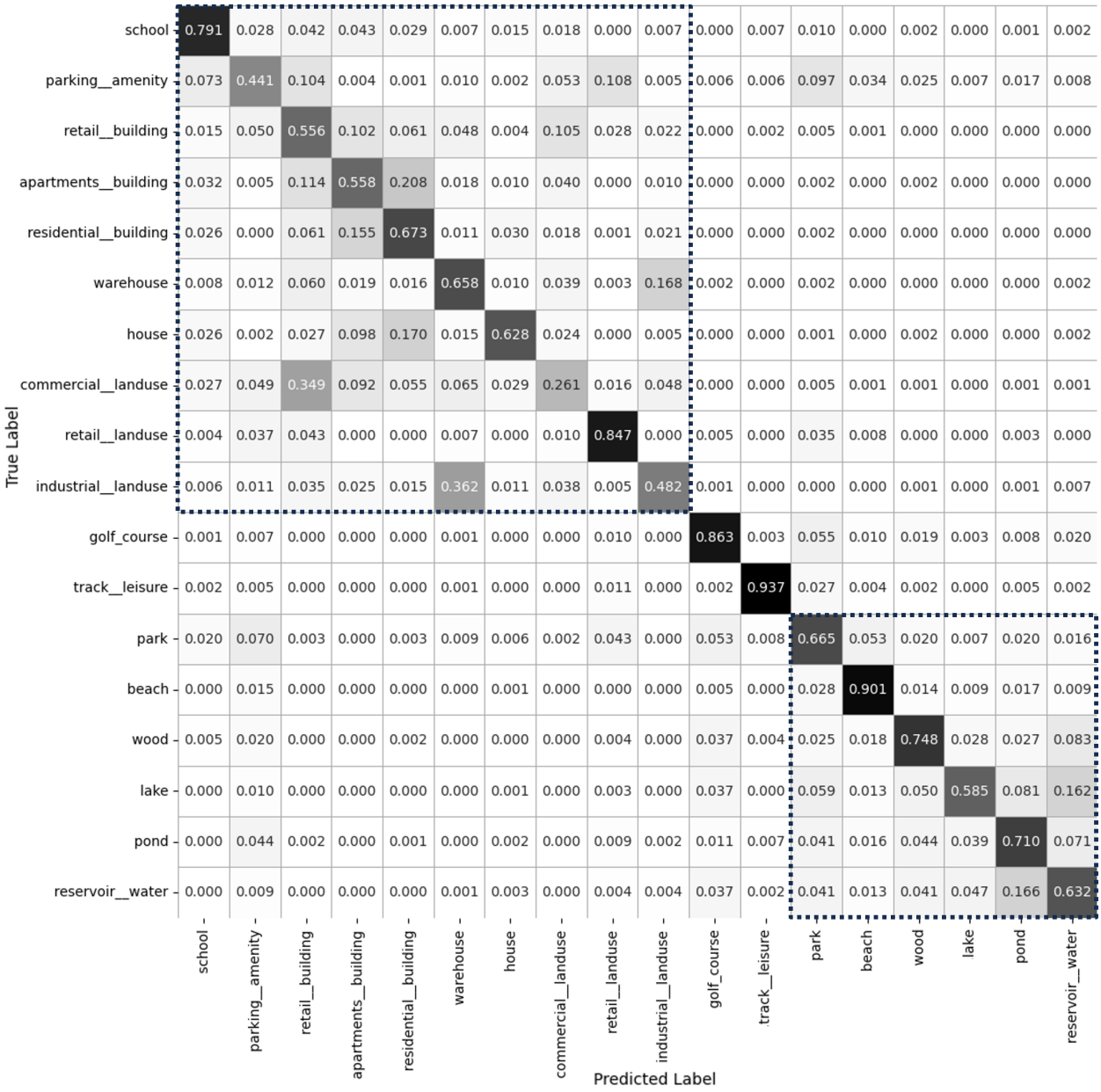


Figure 4.7: Confusion matrix illustrating classification results of geo-entities to *OpenStreetMap* types using the OSM-16k dataset, employing the model derived from the optimal setting (Setting 4). The matrix aggregates results across all mutually exclusive subsets of tests.

4.4 Related Work

The semantics of geospatial information is a rich domain that demands special attention within the web. Although GIS interoperability research has addressed fundamental issues regarding the geometry of geospatial features, recent surveys indicate that current approaches do not effectively

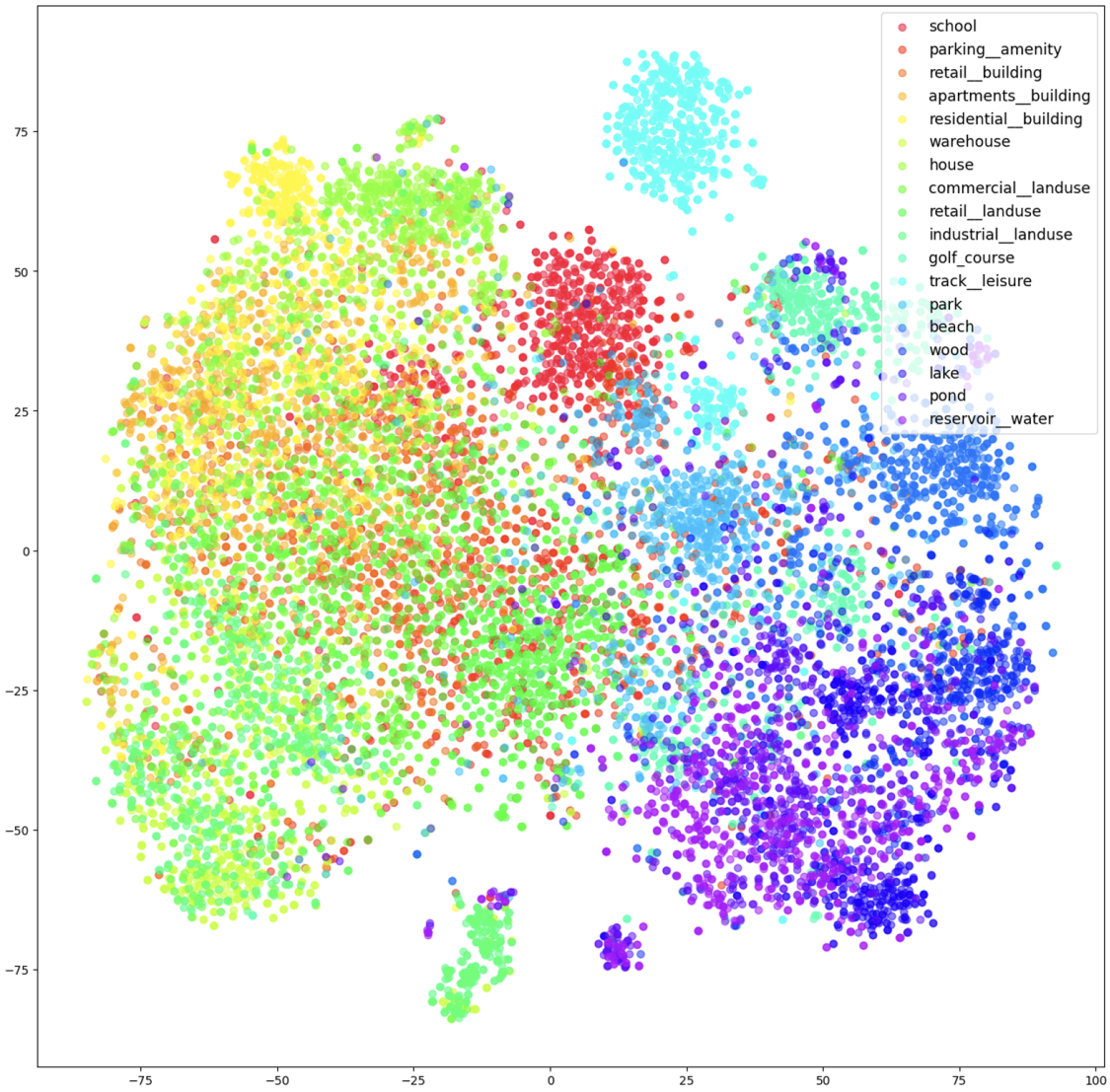


Figure 4.8: t-SNE visualization of embeddings derived from a 10k sample of OSM geo-entities from the California snapshot, generated using our model, and labeled according to the most fine-grained OSM tag. The colors signify the ground-truth labels attributed to each instance.

address the utilization of specific semantics by users for performing tasks that leverage geospatial data [68–70]. Despite these challenges, research on geospatial semantics has seen significant growth in recent years.

The use of machine learning for geospatial data classification has gained significant attention, with convolutional neural networks (CNNs) being a popular approach. Castelluccio et al. [71] proposed a CNN-based approach for land use classification using remote sensing images, and Li et al. [72] developed a CNN-based framework for automatic recognition of building footprints.

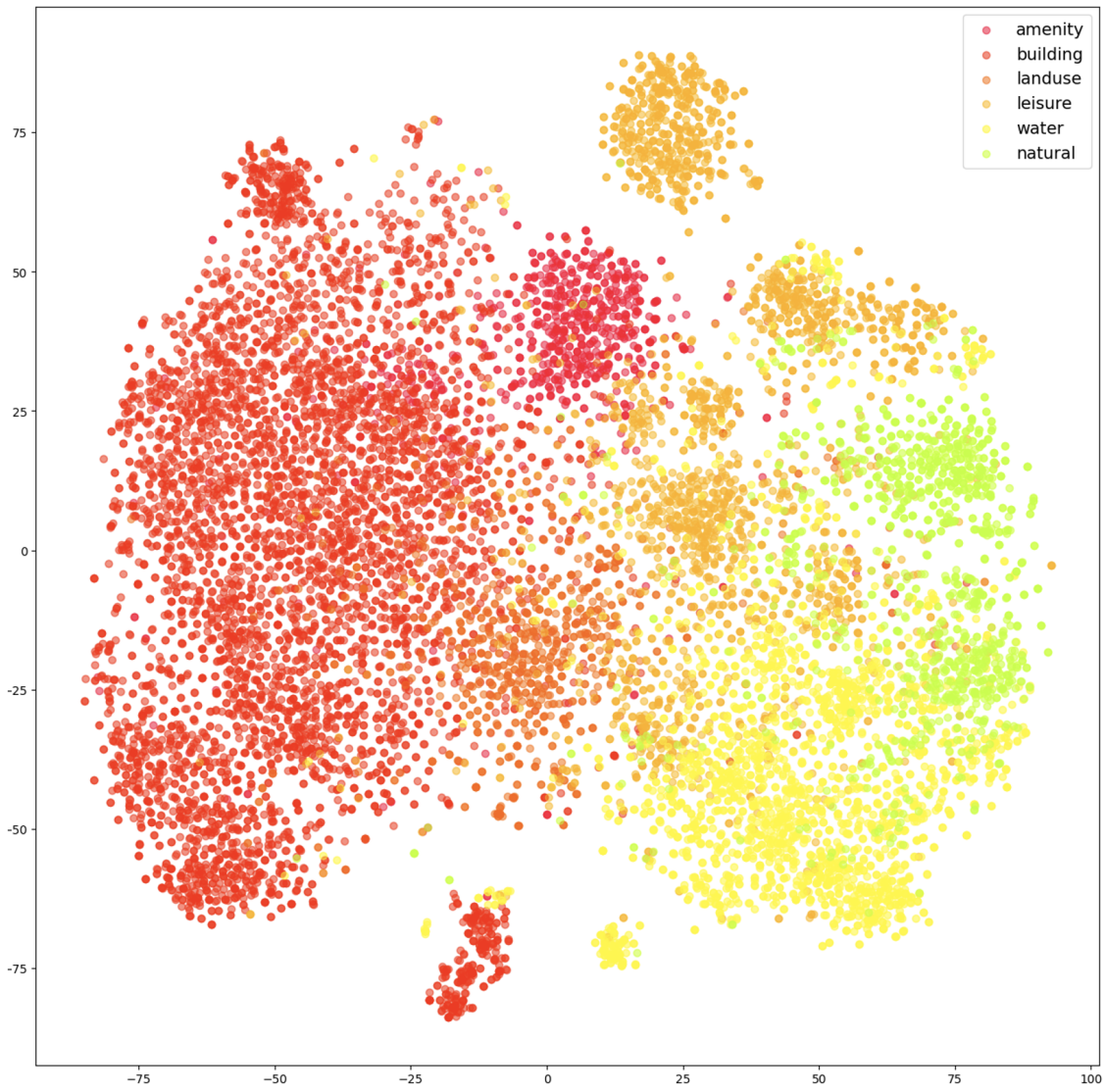


Figure 4.9: t-SNE visualization utilizing the same data in Figure 4.8, showcasing 10k OSM samples. Here, entities are labeled according to the highest-level OSM tags. Different colors distinctly categorize the respective high-level ground-truth labels assigned to each instance.

Dsouza et al. [52] proposed a neural architecture that capitalizes upon a shared latent space for tag-to-class alignment for OSM entities. Klemmer et al. [73] developed a GNN-based approach for context-aware vector encoding of geographic coordinates, Kaczmarek et al. [74] proposed a GNN-based method for spatial object classification using topology, and Xu et al. [75] used a GCN-based approach that incorporates spatial context and aggregates information of adjacent nodes within the graph for urban land-use classification. Yan et. al. [76] developed an approach that combines multiple features extracted from the boundary of a geospatial object to obtain a cognitively compliant

shape encoding. Our work is concerned with a learning task that incorporates multiple sources of information for use in NNs, specifically open data, such as OSM, to improve geospatial data representation and classification.

Geospatial embedding techniques have been explored for geospatial data analysis. Tempelmeier et al.[66] pioneered the development of GeoVectors, offering a pre-trained OSM embeddings corpus that we referenced earlier. Additionally, Jenkins et al. [77] proposed a method for unsupervised representation learning of spatial data via multimodal embedding. Another example is SpaBERT [78], a spatial language model that provides a general-purpose representation of geo-entities based on named neighboring entities in geospatial data, which can be helpful for geo-entity typing. Moreover, Qiu et al. [79] introduced a method that employs geospatial distance to optimize knowledge embedding for a Geographic Knowledge Graph (GeoKG) to help refine latent representations of geo-entities and geo-relations. In contrast to the approaches mentioned above, our work leverages geometric properties of geospatial features, including their shape, as part of the input signals, along with other information, to optimize the embedding process.

Incorporating open data, such as *OpenStreetMap*, for geo-entity representation has received limited attention. Woźniak and Szymański [80] proposed a method to embed OSM regions. This method is not directly comparable since it does not embed arbitrary OSM region entities, it instead decomposes space and embeds each grid cell by the tags contained in it to learn vector representations.

Although these studies have made significant contributions to geospatial data analysis, most do not consider incorporating open data. Our proposed approach builds upon these existing techniques by embedding geospatial and semantic data harvested from the web, into a continuous semantic space, enabling a more comprehensive latent representation.

Chapter 5

From Digitized Reports to Spatio-Temporal Knowledge Graphs

In Chapters 2, 3, and 4, we laid the foundation for transforming raw geo-referenced spatial and map data into structured form and semantically classifying it through a holistic embedding methodology. We explored the transformation of vectorized topographic historical maps into spatio-temporal knowledge graphs (KGs) and enhanced the semantic understanding of geo-entities within these graphs for a more accurate representation.

This chapter marks a transition from focusing on raw geospatial data in vector form to integrating textual and historical data related to geo-referenced spatial entities, specifically historical mining data [81]. We integrate quantitative data of interest, encoded via designated ontological classes, into the KG to enable comprehensive spatio-temporal analysis. Leveraging a custom ontology and semantic web technologies, we transform scattered and heterogeneous archival records into a semantically rich, temporally and spatially aware KG. This allows us to perform advanced temporal and spatial analyses, notably through SPARQL queries, demonstrating the KG's capability in processing and analyzing historical mining data. We illustrate this through a case study that uses a single spatio-temporal SPARQL query to generate direct results for a downstream application: constructing grade-tonnage models for critical minerals such as nickel and zinc.

This chapter demonstrates the significant potential of spatio-temporal KGs, constructed following our methodologies, in processing and analyzing complex datasets. These KGs provide a robust framework that supports complex query execution and offers insightful historical analyses, thereby enhancing the capability to understand and leverage historical geospatial data effectively.

5.1 Motivation

Understanding historical and quantifiable data pertaining to geographic locations, such as mining data is a pursuit of geoscience research and a necessity for informed decision-making in resource management and environmental conservation. The ability to accurately analyze and interpret this data is crucial for identifying new sources of critical minerals, understanding past resource utilization, and aiding in future project development. With the increasing demand for mineral resources [82, 83], there is a growing need to draw upon historical mining data to make informed decisions about current and future mining projects.

Historical mining data is often heterogeneous in nature, existing in varied forms and scattered across numerous archival sources. In many cases, Subject Matter Experts (SMEs) and organizations such as the United States Geological Survey (USGS) are pivotal in organizing these data. They bring indispensable knowledge to critical tasks, such as mineral assessments [84]. However, historical mining data is scattered across multiple sources - ranging from quantitative ore details in mine reports to spatial layers in existing databases - lacks structured organization, and suffers from issues such as quality, accuracy, and completeness [85, 86].

Emphasizing the value of transforming historical records into structured, queryable formats, KGs offer an effective solution for such transformation, combining expressivity, interoperability, and standardization in the semantic web stack, thus providing a strong foundation for querying and analysis.

The evolution and significance of the integration of geospatial data on the web and its extension to linked data has been extensively discussed by Janowicz et al. [69]. Recent technological advances [6] have greatly facilitated the integration of geospatial data from historical archives and maps, transforming these diverse datasets into structured, coherent knowledge bases [7, 9, 12, 23, 27]. A similar transformation is useful in addressing the challenges of mining data analysis, as it enables the consolidation of disparate data sources into a single, accessible, and simple linked data representation - a KG that can be materialized into triples, i.e., RDF data.

To address this task, this chapter presents a methodology for the construction, modeling, and augmentation of a KG dedicated to historical mining data. Our approach involves creating a KG from diverse data sources, formulating a custom domain ontology (semantic model) to represent the data, and utilizing open knowledge from the web to enrich and contextualize data about mineral commodities. Moreover, by integrating additional attributes from geospatial databases, such as MRDS [87], we support spatial queries and visualizations pertaining to specific geo-locations, allowing the development of dedicated downstream applications.

The resulting KG unlocks significant value in generating grade and tonnage plots for mineral resources and commodities. These plots are pivotal, illustrating the relationship between the grade (mineral content) of a deposit and the available tonnage (quantity of ore), which are crucial for assessing the economic viability of mining projects and for accurate reporting of mineral resources and reserves. Figure 5.1 showcases a grade-tonnage model derived from our KG, highlighting the technology's ability to support sophisticated analyses and facilitate the extraction of domain insights with ease.

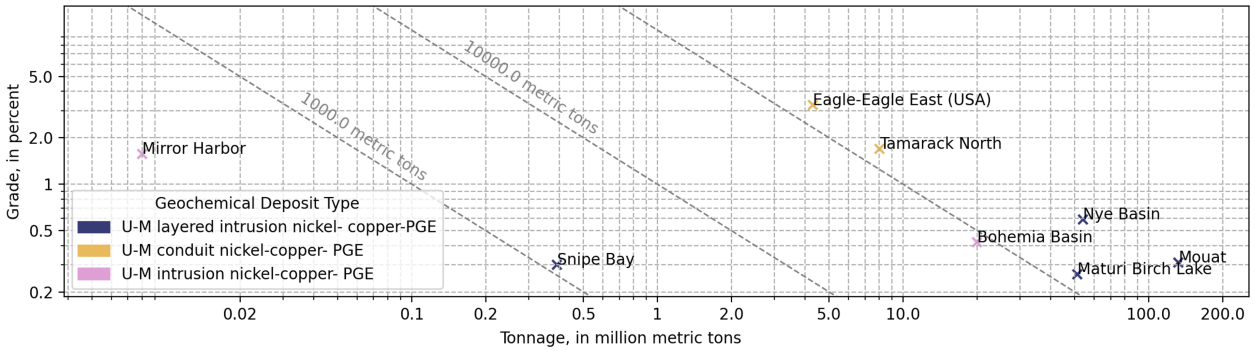


Figure 5.1: Grade-tonnage model of nickel mineral deposits built from a KG query (SPARQL) response, categorized by their Critical Minerals Mapping Initiative (CMMI) deposit classification. Specific sites are marked to illustrate the variability in grade and tonnage among these deposit types.

Through a well-crafted SPARQL [18] query, grade and tonnage data for various minerals can be efficiently retrieved from the KG. SPARQL is a powerful RDF query language and protocol designed for querying and manipulating RDF data (triples) in the KG, enabling users to precisely extract and analyze information based on specific criteria and the most current data. Furthermore,

the KG's flexible structure allows users to tailor queries to specific minerals, geographic areas, or time frames, showcasing its adaptability and comprehensive support for the economic assessments of mining projects.

We evaluate the application of our KG through a detailed analysis of historical mining sites, focusing on two critical minerals: zinc and nickel. Our evaluation demonstrates the KG's ability to generate grade and tonnage plots using SPARQL queries to visualize this data. It highlights the flexibility and robustness of our system in handling complex queries across multiple dimensions. For example, by aggregating mineral sites by their past resource classifications, geochemical deposit classification, or by restricting it to specific geographic regions. Additionally, we conduct a rigorous evaluation of our entity linking method, demonstrating its effectiveness in accurately matching commodities to their corresponding entities in an external knowledge base.

Significance We list the contributions of this chapter as follows:

1. We present a pipeline for the integration of extracted quantitative, spatial, and semantic information from historical mining data archives, resulting in an integrated KG.

2. We introduce a method to identify and retrieve instances of a given type from a publicly available KG, specifically entity matching commodities with open linked data.

3. We assess the applicability of the resulting KG by designing spatio-temporal queries to automatically generate grade-tonnage models for two critical minerals: zinc and nickel.

4. We make the resulting KG publicly available in the form of linked data (queryable RDF via a SPARQL endpoint).[1]

[1] `https://minmod.isi.edu/sparql`

5.2 Constructing the Knowledge Graph

This section describes the methodological framework for constructing the KG for historical mining data. Our methodology is characterized by the formulation of a unique semantic model (Section 5.2.1), linking with external linked data on the web (Section 5.2.2), and finally the materialization of the data into a KG (Section 5.2.3). Each component is meticulously planned to ensure the integration of semantic relationships, spatial data, and temporal dimensions, thereby facilitating a robust analysis of historical and contemporary mining data.

5.2.1 Defining the Semantic Model

Central to the semantic enrichment and structural integrity of our KG is developing a custom ontology and semantic model tailored to the unique characteristics and relationships inherent to historical mining data. This model is designed to capture the domain-specific attributes and complex relationships essential for accurately representing mining information about mineral commodities, such as mineral sites and inventories.

Utilizing RDF as a foundational framework, our approach leverages its structural flexibility and suitability for representing diverse metadata forms, including using existing standards that adhere to universally accepted conventions. This is particularly beneficial for supporting spatial queries, a capability enhanced by integrating the OGC GeoSPARQL standard [35]. The use of this standard enriches our model by providing a vocabulary for representing geospatial data on the web, facilitating qualitative spatial reasoning and quantitative spatial computations.

As illustrated in Figure 5.2, the semantic model delineates the primary entities and their relationships. The model identifies `:MineralSite` entities, characterized by spatial information (`:LocationInfo`) encoded using GeoSPARQL namespace and Well-Known Text (WKT) notation for describing geometries. Mineral deposits, representing natural occurrences of minerals, are linked to mineral sites. Each site may contain multiple `:MineralInventory` items, representing

the quantity (`:Ore`) and `:Grade` of commodities (`:Commodity`) present. The model allows for aggregating these commodities by their `:ResourceCategory` (enumeration from a predefined list, e.g., indicated, measured, inferred), and the association of each site with a specific `:DepositType`, that is also selected from a predefined list, according to the CMMI standards [88].

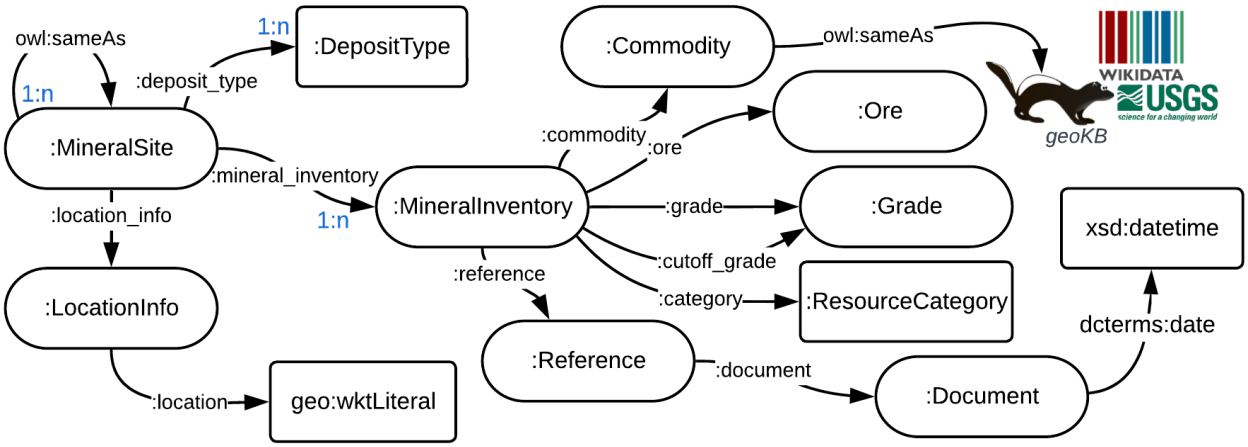


Figure 5.2: Semantic model of the mining data structure. Cardinalities (in blue) show one-to-many node relationships. Circular nodes represent instances, rectangular nodes represent literals (or enumerations). The ":" denotes our namespace.

Including missing information or expert inputs on deposit types underscores the KG's ability to facilitate data classification and aggregation by spatial or attribute data. Each inventory item is associated with a `:Reference` to `:Document` provenance, ensuring data veracity and enabling effective interaction with the KG. `owl:sameAs` enhances the integration with external sources (e.g., geoKB) and within our KG (`:MineralSite` instances). Furthermore, inventory items are tagged with `dcterms:date` properties to indicate temporal aspects, adhering to the Dublin Core Metadata Initiative and W3C recommendations, enhancing the model's comprehensiveness.

5.2.2 Entity Linking via SPARQL

Our approach to enhancing the KG with links going to additional sources involves a simple entity linking process with geoKB[2], the Geoscience Knowledge Base developed by the USGS. This process is pivotal in enriching our KG with validated and scientifically relevant data, leveraging the

[2] `https://geokb.wikibase.cloud/`

extensive earth systems science portfolio within geoKB. Figure 5.3 demonstrates how the nickel mineral is depicted in geoKB, highlighting the depth of linked data that enriches our KG with details on mineral species and their historical classifications.

Figure 5.3: Illustration of the nickel mineral species in geoKB, showcasing the depth of information our entity linking method accesses, enriching our KG with metadata from external sources.

At the heart of our methodology is a constrained search within geoKB, using SPARQL queries designed to find commodities classified under certain instance types. This approach narrows the scope of potential matches to enhance their relevance. Listing 5.1 presents a query aimed at fetching candidate mineral commodity instances of nickel from geoKB, marking the first step in our entity linking process. The query filters for entities that are instances of (`P1`) mineral commodities (`Q406`) as seen in line 3 (where `gkbt` and `gkbi` represent namespaces for predicates and instances within geoKB, respectively). The `FILTER` clause (line 4) conducts a case-insensitive search to align the entity labels with the commodity string in question, exemplifying our strategy for extracting semantically related instances.

Following retrieval, we apply the Jaccard [89] similarity measure for set-based comparison between the commodity strings and geoKB entity labels. This process, based on the intersection over

```
SELECT ?entity ?entityLabel WHERE {
  ?entity rdfs:label ?entityLabel.
  ?entity gkbt:P1 gkbi:Q406. # instance of mineral commodity
  FILTER(CONTAINS(LCASE(?entityLabel), "nickel")) }
```

Listing 5.1: A SPARQL query example targeting the nickel mineral commodity in geoKB serving as a foundational step for entity linking.

the union of the derived sets, helps determine the top instance among the candidates. The selected instance is then leveraged to enrich our KG, infusing it with additional semantics and metadata from sources like Wikidata [34], the Geoscience Ontology [90], and geoKB itself. Consequently, our KG is augmented with comprehensive information on mineral species and historical mineral classifications, among other data, thereby enhancing its semantic richness and inter-connectedness.

5.2.3 Transforming the Data into Triples

Our methodology for populating the KG leverages data sourced from semi-structured (tabular format) and structured (JSON files) data related to mineral mining. The initial step in the KG construction involves meticulous data cleaning and normalization, including entity deduplication and URI (Uniform Resource Identifier) mapping to ensure each entity, such as a mineral site, inventory, or document, is uniquely identifiable and accessible on the web.

URIs are generated using a hash function (e.g., MD5) to create a de-referenceable URI from the unique combination of the defining attributes of an instances, including feature type, location data, and temporal information. For instance, a `:MineralInventory` instance's URI is constructed using a concatenation of its commodity URI, category, the referring mineral site URI, and any associated document URIs. A similar approach is taken for `:MineralSite` and `:Document` URIs, using their respective source identifiers and bibliographic data.

In our KG, entities such as commodities and deposit types are linked to external knowledge bases or predefined lists, like the CMMI, while other entities are represented as blank nodes within our namespace. The transformation of data into RDF triples is facilitated by automated tools like D-REPR [91] and SAND [92], ensuring seamless integration and update of information.

Geospatial data are materialized using dedicated namespaces, notably GeoSPARQL, which augments the KG's capacity for spatial analysis. Additionally, the ontology supports OWL [19] for representing complex inter-entity relationships and attributes.

A validation layer, crucial for maintaining data integrity and consistency, is implemented through an automated system using SHACL [93]. This validation ensures that our KG not only accurately represents the data but also adheres to the predefined semantic model (Section 5.2.1), enabling reliable and sophisticated queries.

5.3 Evaluation and Discussion

Our evaluation framework includes qualitative and quantitative analyses over a KG built from a dataset covering two mineral commodities, focusing on the KG's adherence to the semantic model and completeness (Section 5.3.1), entity linking with geoKB (Section 5.3.2), and its utility and performance in advanced data analysis (Section 5.3.3), particularly in generating grade and tonnage models.

The extensive dataset we use covers over 50 NI 43-101 technical reports (International Strategic Mineral Inventory reports) on nickel (2001 to 2021) and zinc (2002 to 2019), supplemented by spatial data from the MRDS (Mineral Resources Data System) and USMIN (US Mineral Deposit Database) databases. Sources such as Mudd and Jowitt's compiled work on zinc from 2017 [94], and on nickel from 2022 [95], provide additional data and extensive coverage on various commodities as well, offering a rich blend of geospatial, geological, and economic data from various global locations.

5.3.1 Evaluation on the Semantic Model

Our evaluation confirms the KG's adherence to the semantic model, reflecting accurate domain representation in compliance with RDF standards for enhanced query performance and data interoperability. Our resulting KG characteristics are described in Table 5.1.

Table 5.1: Historical mining data knowledge graph characteristics.

Characteristic	Count
Total Triples	2,397,708
Distinct Classes	16
Instances (Non-literals)	226,267
Geospatial Instances	2,884
Blank Nodes	1,518,981

The resulting KG hosts a significant number of instances and blank nodes, suggesting a rich network of connected data. Diving deeper, the KG encapsulates vital entities including `:MineralSite`, `:MineralInventory`, `:Commodity`, `:DepositType`, and `:Reference`. The entities form a network that reflects the complex interactions and interplay between various facets of mining data. For instance, `:MineralSite` entities are geospatially positioned through `:LocationInfo` relationships, while each site can encompass multiple `:MineralInventory` records, detailing the reserves in terms of quantity and grade for each `:Commodity`. Moreover, the KG adheres to CMMI standards for deposit classification and manages data provenance through `:Reference` links to detailed source citations, as expected.

The granular representation of the data in the resulting graph, with 1,112 zinc and 1,132 nickel reserve and resource measurements, alongside 3,809 zinc and 2,021 nickel mineral site instances, demonstrates the KG's quantitative depth and utility. This structured model captures key attributes, ensuring interoperability and alignment with semantic data representation best practices. The approach we present is complete and follows linked data principles by:

- Generating URIs as names for things, without modifying previously published identifiers.
- Maintaining existing relations (predicates) between instances ("backward compatibility").
- Generating machine-readable structured data.
- Using standard namespaces and semantics (e.g., OWL, Dublin Core, GeoSPARQL).
- Linking to additional resources on the web (e.g., geoKB).

5.3.2 Evaluation on Entity Linking

To evaluate the effectiveness of our proposed entity-linking method, we conducted an evaluation using a dataset of 135 extracted commodity labels mapped by human experts to geoKB. This dataset serves as a benchmark for determining the success of our entity linking method.

To provide a comprehensive evaluation, we contrasted our approach against three distinct baseline methods within geoKB. The first two baselines involve a generalized, string search strategy with SPARQL, then label comparison — one utilizing the Jaro [96] string similarity measure and the other employing the Jaccard measure — based purely on textual relevance. The third baseline adopts a constrained search strategy (instance based) combined with the Jaro similarity measure, similar in part to our proposed method, refining the search scope yet still leveraging Jaro's well-regarded efficiency in measuring string similarities for short texts, such as names.

This multifaceted comparison shows substantial gains in matching accuracy with our proposed method — instance-based constrained search followed by Jaccard similarity measure for set-based comparisons — over the baselines. Specifically, our method has demonstrated significant performance enhancements, showcasing the advantage of a constrained search strategy combined with the precision of Jaccard similarity. This approach not only refines the selection of potential matches but also ensures a high degree of textual similarity between the commodity strings and the linked geoKB entities. The results are summarized in Table 5.2.

Table 5.2: Evaluation results for the entity linking experiments with geoKB.

Method	**MRR**	**Hits@1**	**Hits@3**	**Hits@5**
String search, then Jaro	0.557	0.459	0.659	0.659
String search, then Jaccard	0.648	0.637	0.659	0.659
Instance search, then Jaro	0.801	0.689	0.926	0.956
Instance search, then Jaccard (proposed)	**0.940**	**0.904**	**0.978**	**0.978**

The results underscore the superior performance of our proposed entity linking approach, with a Mean Reciprocal Rank (MRR) of 0.940 and impressive Hits@1, Hits@3, and Hits@5 rates of 0.904, 0.978, and 0.978 again, respectively. These metrics notably surpass those of the baseline methods, thereby affirming the utility of combining constrained search with the Jaccard similarity

measure. The MRR value, being substantially higher than that of the baselines, indicates that our method consistently identifies the most relevant geoKB entity at the top rank. The high Hits@1 value signifies that the correct entity is identified as the top match in a significant majority of cases, a critical metric for applications relying on precision. Similarly, the near-perfect Hits@3 and Hits@5 scores suggest that if the top match isn't the exact entity, it is highly likely to be within the top 3 or 5 candidates, offering a valuable safety net for ensuring data quality in the KG. These results collectively justify our method's design, which meticulously tailors the search and comparison phases to optimize for both accuracy and relevance in entity linking.

5.3.3 Evaluation on Querying the KG

To assess the query performance, utility, and effectiveness of our KG in extracting relevant information for creating grade-tonnage models, we executed a series of SPARQL queries. These queries, aimed at testing the KG's ability to retrieve grade and tonnage data under diverse constraints and scenarios, were conducted using RDF triples hosted on Apache Jena[3], a lightweight and programmable environment with geospatial query support. The baseline query, shown in Listing 5.2, is crucial for fetching grade and tonnage data along with site and inventory identifiers, and serves as the foundation for developing more complex queries. By building on this foundational query, we introduced three distinct constraint types in our subsequent queries — textual, temporal, and spatial — thereby not only testing the KG's flexibility in meeting varied query requirements but also showcasing its capability to provide precise and contextually relevant data across different analytical dimensions.

In the first type of query we retrieve ore and tonnage data for a specific commodity. This query aims to retrieve ore grade and tonnage data for all inventories associated with a specified commodity. It demonstrates the KG's capability to filter data based on commodity type, which is essential for users interested in specific mineral insights. Listing 5.3 shows the added clause needed to retrieve entries for a given mineral commodity name, nickel in the example.

[3] `https://jena.apache.org/`

```
SELECT ?ms ?mi ?ms_name ?mi_cat ?ore ?grade
WHERE {
    ?ms :mineral_inventory ?mi .
    OPTIONAL { ?ms rdfs:label|:name ?ms_name . }
    ?mi :category ?mi_cat .
    ?mi :ore [ :ore_value ?ore;
               :ore_unit  ?ore_unit] .
    ?mi :grade [ :grade_value ?grade;
                 :grade_unit  ?grade_unit] . }
```

Listing 5.2: Baseline SPARQL query for grade and tonnage data.

```
?mi :commodity/:name "nickel"@en .
```

Listing 5.3: SPARQL clause for filtering by commodity type: this clause filters inventory items to retrieve data specific to the `nickel` commodity, demonstrating how to tailor queries for particular mineral.

In the second type of query we retrieve ore and tonnage data with an emphasis on a temporal constraint on document provenance, from which the data originated. This query filters ore and tonnage data based on the publication date of the source documents. Such a query is useful for researchers interested in how grade and tonnage estimates could have changed over time or analyze specific data within a specific timeframe. Listing 5.4 shows the added clause needed to retrieve inventory items pertaining to specific time ranges. In this case we are fetching inventories from documents published between the year 2000 to 2010.

```
?mi :reference/:document [ dcterms:date ?date ] .
FILTER(?date >= "2000"^^xsd:gYear && ?date <= "2010"^^xsd:gYear) .
```

Listing 5.4: SPARQL clause for temporal filtering: this clause applies a temporal filter to select inventory items based on their document's publication year between 2000 and 2010, showcasing the KG's ability to analyze historical data over a specific time range.

Utilizing Apache Jena's support for GeoSPARQL, the third query retrieves tonnage data from inventories at mineral sites within a certain distance from a given point. It exemplifies the KG's spatial querying capabilities, which are crucial for geographical analyses and decision-making. Listing 5.5 shows the added clause needed to retrieve mineral sites, with inventory items, that are

within a specific distance from a given point data in WKT format. In this example we are searching for mines that are within 500 miles from given coordinates.

```
1 ?ms :location_info/:location ?loc_wkt .
2 FILTER(geof:distance(?loc_wkt, "POINT(-118.57 47.56)"^^geo:
     wktLiteral, unit:mile) < 500)
```

Listing 5.5: SPARQL clause for spatial proximity filtering: this clause leverages GeoSPARQL to find mineral sites within a 500-mile radius of a specified point, exemplifying spatial querying capabilities for geographical analysis. The `geof` and `unit` namespaces are standard namespaces utilized for specifying distance measurements and units, respectively.

Table 5.3 presents a summary of the query-time performance, including average, minimum, and maximum times, effectively showcasing the efficiency of our KG when operating under various query constraints. This efficiency is underscored by the execution of hundreds of similar queries across a diverse range of values for each constrained scenario, further demonstrating the robustness and adaptability of our system in handling retrieval tasks.

Table 5.3: Query time statistics (in milliseconds).

Query Constraint Type	**Avg**	**Min**	**Max**
Textual	450	369	649
Temporal/Numeric	438	388	607
Spatial	708	501	811

The results outlined in Table 5.3 showcase the KG's performance across different query constraints, with query times measured in milliseconds. The average query time for textual constraints was notably efficient at 450 ms, reflecting the rapid response to straightforward textual searches. Temporal queries, with an average time of 438 ms, highlight the KG's adept handling of quantitative and temporal data retrieval, facilitating temporal analysis. Spatial queries, while more computationally intensive due to the nature of geospatial data processing, still performed admirably with an average time of 708 ms. This demonstrates the system's capacity to efficiently manage spatial reasoning tasks, a crucial aspect for mining data analysis where geographical context is vital.

These results are not only indicative of the KG's robust performance but also validate our methodological choices and architecture. The swift response times, especially for spatial queries,

are a testament to the efficiency of integrating GeoSPARQL and our custom semantic model, facilitating advanced spatial analyses. Furthermore, the accurate retrieval of information across all query types confirms the KG's utility in supporting complex queries for critical tasks such as generating grade-tonnage models, as we demonstrate in Figure 5.1.

The application of SPARQL queries against our KG exemplifies the invaluable insights gained from the fusion of semantic web technology with spatial visualization techniques, enabling the straightforward interpretation of otherwise complex geographic data. For example, Figure 5.4 presents a detailed visualization of nickel mineral sites across the United States, categorized according to CMMI standards and overlaid on a topographic map, demonstrating the expansive coverage of our KG. By integrating this classification with other geospatial data, such as geological formations and stratigraphy information, we can significantly enhance the multi-dimensional analytical capabilities available to SMEs, allowing for predictive modeling of mineral potential and helping to identify unexplored areas with high resource prospects.

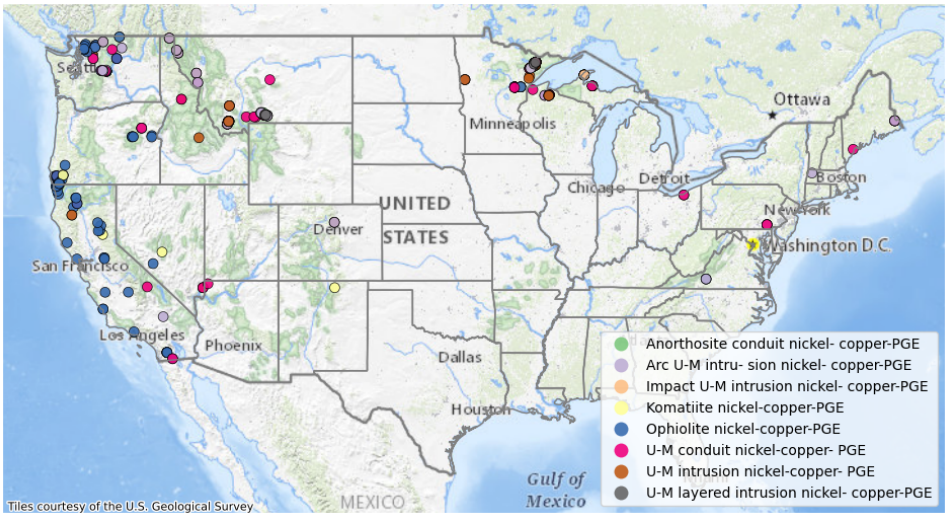


Figure 5.4: Nickel mineral sites in the US by CMMI classification on a topographic map background. Example of a spatial visualization based on data derived from the KG and retrieved via SPARQL, showcasing nickel mine distribution.

By structuring historical and current mining data within a KG, we enable powerful query capabilities through SPARQL, facilitating the retrieval and representation of complex data sets easily and quickly. The query results above establish high confidence in our model, showing that we can

easily and effectively answer complex queries in a robust manner. Furthermore, the integration of our commodity data with geoKB enhances our KG's utility by enabling federated SPARQL queries, which allow us to fetch additional data from external sources such as Wikidata. This capability significantly broadens the scope of our analysis, providing access to a wealth of information that complements our existing datasets. Overall, we demonstrated that our approach and the proposed pipeline can be effectively used to automatically construct effective and contextualized open KGs and linked data from historical and contemporary mining data, as well as support both temporal and spatial analysis.

5.4 Spatio-Temporal Analysis via SPARQL

We highlight the innovative application of semantic web technology within spatio-temporal KGs through a single SPARQL query designed to generate a grade and tonnage model from the mining data. The query, detailed in Listing 5.6, filters the data specifically for mineral sites with nickel commodities (line 19) and combines the integration of spatial and temporal constraints (lines 20-21) within the SPARQL framework, demonstrating how quantitative transformations can be seamlessly performed within the query environment, leveraging GeoSPARQL support. The query performs complex calculations and aggregations directly within the query environment. By selecting various attributes related to mineral sites, including names, deposit types, and geographical coordinates (WKT Geometry), the query aggregates measured, indicated, and inferred tonnages and ore grades. These aggregations are then used to compute total tonnage and contained metal, culminating in the calculation of the total grade of the mineral sites and deposits.

A key feature of this query is its use of spatial and temporal constraints to filter mineral sites by selecting the ones that are within a 500-mile radius of a specified point (line 20) and further narrows down the data to those originating from documents dated between 2000 and 2010 (line 21), showcasing the query's ability to integrate geospatial and temporal data with semantic web technologies. Beyond simple proximity, the query can utilize GeoSPARQL predicates such

as `overlaps`, `intersects`, or `within` to explore complex spatial relationships in spatial data, enhancing analytical capabilities. For instance, to demonstrate alternative spatial queries, line 20 could be replaced with `FILTER(geof:sfWithin(?loc_wkt, "POLYGON(...)"))` to retrieve mineral sites within a specified polygon. The calculated metrics, such as total tonnage and grade, are essential for analyzing the mineral wealth of a region. This precision in data manipulation within the query underscores the powerful analytical capabilities of KGs, allowing for targeted and meaningful analysis of the mineral resources.

Furthermore, Appendix A complements this technical discussion by introducing an unsupervised approach to further enhance our understanding and processing of quantitative data within KGs. This approach, described in the appendix, focuses on identifying and annotating textual occurrences of units in source data, linking them with their corresponding instances in the QUDT (Quantities, Units, Dimensions, and Types) ontology [97]. This linkage is explicitly utilized in our SPARQL query (line 25), as evidenced in the transformation and computation of mineral quantities, demonstrating how external ontologies can facilitate automatic quantitative transformations within the KG. This enhancement facilitates automatic quantitative transformations within the KG, enriching the semantic web technology framework used in this context. The methodology outlined in the appendix, supported by preliminary results, demonstrates the capability of this approach to automate and streamline the unit conversion process, thereby supporting the analytical power and insight gained from the spatio-temporal KG representation constructed in our work.

This SPARQL query exemplifies the advanced capabilities of spatio-temporal KGs constructed following our methodologies. The ease with which these complex data manipulations are performed within a single query not only demonstrates the technical sophistication of our approach but also underscores the practical value of our research in facilitating precise and meaningful analysis of mining data. This aspect of the thesis is a testament to the innovative integration of semantic, spatial, temporal, and quantitative data, providing a robust framework for generating actionable insights in the domain of geospatial-temporal knowledge discovery.

5.5 Related Work

Recent advancements in geology and earth science data analysis have been significantly propelled by the application of machine learning techniques, which have enabled the enhancement of data mining and extraction for geology and mineral data [98]. These developments have shown considerable promise in various applications, ranging from prospectivity mapping to knowledge organization in the natural sciences [99, 100]. However, the full utilization of semantic and spatial relationships in historical mining data remain largely underexplored, indicating a gap in the current research landscape.

In the domain of geoscientific KGs, our work complements existing knowledge bases such as GeoKB and the Geoscience Ontology [90], by addressing the nuanced gaps in the semantic enrichment and spatial analysis of historical mining data, areas often overlooked in the broader context of such applications. This gap presents a unique opportunity to contribute to the field by leveraging semantic web technologies with spatial and temporal data analysis to enrich our understanding of historical mining activities and their implications for contemporary and future mining endeavors.

```
SELECT
  ?ms ?ms_name ?deposit_name ?loc_wkt ?total_tonnage_measured ?
     total_tonnage_indicated ?total_tonnage_inferred ?
     total_contained_measured ?total_contained_indicated ?
     total_contained_inferred (?total_tonnage_measured + ?
     total_tonnage_indicated + ?total_tonnage_inferred AS ?total_tonnage)
     (?total_contained_measured + ?total_contained_indicated + ?
     total_contained_inferred AS ?total_contained_metal) (IF(?
     total_tonnage > 0, ?total_contained_metal / ?total_tonnage, 0) AS ?
     total_grade)
WHERE {
  {
    SELECT ?ms ?ms_name ?deposit_name ?country ?loc_wkt
           (SUM(?tonnage_measured) AS ?total_tonnage_measured)
           (SUM(?tonnage_indicated) AS ?total_tonnage_indicated)
           (SUM(?tonnage_inferred) AS ?total_tonnage_inferred)
           (SUM(?contained_measured) AS ?total_contained_measured)
           (SUM(?contained_indicated) AS ?total_contained_indicated)
           (SUM(?contained_inferred) AS ?total_contained_inferred)
    WHERE {
        ?ms :deposit_type [ rdfs:label ?deposit_name ] .
        ?ms :mineral_inventory ?mi .
        OPTIONAL { ?ms rdfs:label|:name ?ms_name . }
        ?ms :location_info/:location ?loc_wkt .
        ?mi :category ?mi_cat .
        ?mi :reference/:document [ dcterms:date ?date ] .
        ?mi :commodity [ :name "nickel"@en ] .
        FILTER(geof:distance(?loc_wkt, "POINT(-118.57 47.56)"^^geo:
           wktLiteral, unit:mile) < 500)
        FILTER(?date >= "2000"^^xsd:gYear && ?date <= "2010"^^xsd:gYear) .
        ?mi :ore [ :ore_value ?ore_val_raw; :ore_unit ?ore_unit] .
        ?mi :grade [ :grade_value ?grade_val; :grade_unit ?grade_unit] .
        BIND(IF(bound(?ore_val_raw), ?ore_val_raw, 0) AS ?ore_val_pre)
        BIND(IF(?ore_unit = <http://data.nasa.gov/qudt/owl/unit#MetricTon
           >, ?ore_val_pre / 1e6, ?ore_val_pre)) AS ?ore_val)
        BIND(IF(CONTAINS(LCASE(STR(?mi_cat)), "measured"), ?ore_val, 0) AS
            ?tonnage_measured)
        BIND(IF(CONTAINS(LCASE(STR(?mi_cat)), "indicated"), ?ore_val, 0)
           AS ?tonnage_indicated)
        BIND(IF(CONTAINS(LCASE(STR(?mi_cat)), "inferred"), ?ore_val, 0) AS
            ?tonnage_inferred)
        BIND(IF(CONTAINS(LCASE(STR(?mi_cat)), "measured") && ?grade_val >
           0, ?ore_val * ?grade_val, 0) AS ?contained_measured)
        BIND(IF(CONTAINS(LCASE(STR(?mi_cat)), "indicated") && ?grade_val >
            0, ?ore_val * ?grade_val, 0) AS ?contained_indicated)
        BIND(IF(CONTAINS(LCASE(STR(?mi_cat)), "inferred") && ?grade_val >
           0, ?ore_val * ?grade_val, 0) AS ?contained_inferred)
    }
    GROUP BY ?ms ?ms_name ?deposit_name ?loc_wkt }
```

Listing 5.6: A SPARQL query to generate grade and tonnage model for nickel commodities, constraind by spatial distance from a given point, and a given time-range for the documents from which the data originates.

Chapter 6

Conclusion and Future Directions

6.1 Conclusions

This thesis articulates methodologies for the construction of spatio-temporal knowledge graphs from digitized historical and geographic data, thus rendering such data machine-understandable, analytically valuable, publicly available, and cross-domain interoperable (Semantic Web-compliant). The methodological framework and techniques presented here enable the automatic transformation of digitized disparate, unstructured, and semi-structured historical datasets into structured, semantic, and spatially and temporally indexed KGs. This transformation process leverages different data parsing and data understanding techniques for entity generation from map and text data (entity recognition), as well as entity resolution and linking, semantic embedding techniques to categorize and contextualize geo-entities, and the integration of these elements within a robust semantic framework to ensure interoperability and enhance query capabilities as a resource on the web.

The resulting KGs enable advanced querying capabilities, supporting historical geographic studies across different domains or tasks of interest, and establishing a solid foundation for interdisciplinary research and practical applications, paving the way for innovative research in the realm of GIS, Semantic Web, and AI. Published as linked data, these KGs adhere to Semantic Web and open data principles, promoting data sharing, availability, and reuse across different domains and

applications, thereby maximizing their impact in the digital age. This approach not only democratizes access to historical and geographic analysis but also lays the groundwork for integrating these structured data with additional digital resources, enhancing the comprehensiveness and granularity of geospatial analytics.

6.2 Contributions

The core contributions of this work are detailed through the methodologies, evaluations, and findings discussed in Chapters 2, 3, 4, and 5. Additionally, a complementary auxiliary tool is presented as standalone research artifact in Appendix A.

In Chapter 2, I detailed a novel methodology for constructing spatio-temporal KGs from digitized historical maps. This work presented the automated modeling, linking, and semantic structuring of geographic and topographic features over time, enabling historical change analysis in a machine-readable format through a spatio-temporal queries to the KG. By employing techniques like reverse-geocoding and geo-entity linking, I enhanced the interconnectivity and contextual relevance of the derived entities in the KG within the data itself and with external data on the web, thus enabling richer, semantic-driven queries and analyses.

Chapter 3 described the organization of crowd-sourced geo-data into a structured hierarchy to support deeper data understanding and integration of geo-instance labels, detailing the construction of geospatial feature taxonomies from *OpenStreetMap* data.

Chapter 4 introduced a novel self-supervised embedding method for embedding geo-referenced vector data, integrating geometric, spatial, and semantic contexts. The embedding process, which includes neighborhood contextual semantic encoding and taxonomy-guided contrastive learning, provides a comprehensive approach of data representation for capturing spatial and semantic relationships among geo-entities, leveraging open data sources like *OpenStreetMap* to bridge the gap in automated geo-data understanding and integration. This methodology enhances geo-entity typing and classification in similar settings, surpassing state-of-the-art models [66]. The resulting

embeddings enable the classification of unlabeled geo-referenced data, predict its semantic type, and allow the retrieval of candidate entities from Open KBs (such as OSM or Wikidata [34]). This process complements the integration of external entities, as discussed in Chapter 2 (Section 2.2.4), into the resulting KG as linked data, enhancing data interoperability and precision (type granularity and instance retrieval).

Finally, Chapter 5 exemplified the modeling and representation of geo-data and geo-related data via spatio-temporal KGs through an application centered around historical mineral mining data. This chapter demonstrates how digitized mining reports coupled with spatial databases could be transformed into structured knowledge, enabling advanced spatio-temporal and quantitative analyses via a single query. The integration of these datasets into a dynamic and semantic KG demonstrates the versatility and scalability of the proposed methods in handling diverse data types and complex analytical tasks, like the generation of grade-tonnage models for mineral sites within a specific spatial and/or temporal context.

In Appendix A, I elaborate on the process of identifying units in scientific (textual) data, complementing the grounding of quantitative measurement data (e.g., ore tonnage) and its integration in the historical mineral mining data KG, to enable the automatic transformation often required when generating the data, as discussed in Chapter 5.

Collectively, these contributions represent a significant advancement in the field of geospatial and historical data analysis and knowledge graph construction. They not only provide new tools and techniques for data scientists and researchers but also pave the way for future innovations in semantic web technologies and geospatial artificial intelligence.

6.3 Future Directions

The methodologies and findings presented in this dissertation open several avenues for future work. For Chapter 2, extending the automatic segmentation and inter-linking process to accommodate

rapidly changing contemporary maps and diverse geographic feature transformations will be crucial for the data modeling piece (e.g., a large body of water drying out to become represented as a line, or a river causing a flood and generating a polygonal feature). Investigating adaptive hyperparameters for different tasks, such as geometry partitioning, potentially automated through optimization algorithms, along with enhancing geo-entity linking by incorporating additional knowledge bases like Yago2Geo [101], could significantly enhance the comprehensiveness of the data.

As an extension to the work presented in Chapter 4, future direction could focus on refining the geo-entity embedding process through subword information and deep-learning attention mechanisms, extending the contextual understanding. Thus, integrating additional textual knowledge from open knowledge bases would enrich the representations, enabling more accurate geo-entity typing and linking.

Chapter 5 on the other hand, sets the stage for expanding the KG to encompass a wider array of critical minerals and historical datasets. Leveraging advanced machine learning techniques could unravel deeper insights from the data, aiding in economic decision-making and strategic planning in the mining sector and beyond. The adaptability of this approach to other domains can be readily achieved by simply adjusting the semantic model, demonstrating the potential for a wide and versatile range of applications.

Developing more sophisticated semantic models, especially for the methods in Chapters 2 and 5 would also be beneficial, allowing for a more accurate representation of complex historical and geographic data. Future studies could explore how spatio-temporal KGs can support additional diverse fields such as archaeology, environmental science, and cultural heritage preservation.

Broadly, the increasing availability of diverse scientific and web data offers fertile ground for advancing data modeling in GIS. Future research could explore integrating multiple modalities of data, such as satellite imagery, sensor data, and historical archives, to model dynamic geographical phenomena more comprehensively. Enhanced data models could then incorporate sophisticated hyperparameter optimization techniques to adapt more dynamically to rapidly changing

geographic features, thereby providing more accurate predictions and insights for environmental and urban planning applications.

In line with the evolution of KGs, there is significant potential to expand their linkage across a broader range of knowledge bases and domains (e.g., archaeology and environmental sciences). This expansion would not only enrich the KGs with diverse datasets but also enable their application in multidisciplinary research areas. This would facilitate a bettter understanding and representation of multi-domain data, supporting dynamic semantic modeling for evolving real-world applications.

References

1. Duan, W., Chiang, Y., Knoblock, C. A., Leyk, S. & Uhl, J. *Automatic generation of precisely delineated geographic features from georeferenced historical maps using deep learning* in *Proceedings of the 22nd International Research Symposium on Computer-based Cartography and GIScience (Autocarto/UCGIS)* (eds Freundschuh, S. & Sinton, D.) (UCGIS.org, 2018), 59–63.

2. Uhl, J. H., Leyk, S., Chiang, Y.-Y., Duan, W. & Knoblock, C. A. Automated extraction of human settlement patterns from historical topographic map series using weakly supervised convolutional neural networks. *IEEE Access* **8,** 6978–6996. doi:10.1109/ACCESS.2019.2963213 (2019).

3. Uhl, J. H. & Duan, W. *Automating information extraction from large historical topographic map archives: new opportunities and challenges* in *Handbook of Big Geospatial Data* (eds Werner, M. & Chiang, Y.-Y.) (Springer, Cham, 2021), 509–522. doi:10.1007/978-3-030-55462-0{_}20.

4. Maduekwe, N. I. A GIS-based methodology for extracting historical land cover data from topographical maps: illustration with the nigerian topographical map series. *KN-Journal of Cartography and Geographic Information* **71,** 105–120. doi:10.1007/s42489-020-00070-z (2021).

5. Lin, F., Knoblock, C. A., Shbita, B., Vu, B., Li, Z. & Chiang, Y.-Y. *Exploiting Polygon Metadata to Understand Raster Maps-Accurate Polygonal Feature Extraction* in *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems* (2023), 1–12.

6. Chiang, Y.-Y., Leyk, S. & Knoblock, C. A. A survey of digital map processing techniques. *ACM Computing Surveys (CSUR)* **47,** 1–44. doi:10.1145/2557423 (2014).

7. Uhl, J. H., Leyk, S., Li, Z., Duan, W., Shbita, B., Chiang, Y.-Y., *et al.* Combining remote-sensing-derived data and historical maps for long-term back-casting of urban extents. *Remote Sensing* **13,** 3672. doi:10.3390/rs13183672 (2021).

8. Chiang, Y.-Y., Duan, W., Leyk, S., Uhl, J. H. & Knoblock, C. A. *Using historical maps in scientific studies: applications, challenges, and best practices* doi:10.1007/978-3-319-66908-3 (Springer, Cham, 2020).

9. Chiang, Y.-Y., Chen, M., Duan, W., Kim, J., Knoblock, C. A., Leyk, S., *et al. GeoAI for the Digitization of Historical Maps* in *Handbook of Geospatial Artificial Intelligence* (CRC Press, 2023), 217–247.

10. Kyzirakos, K., Karpathiotakis, M., Garbis, G., Nikolaou, C., Bereta, K., Papoutsis, I., *et al.* Wildfire monitoring using satellite images, ontologies and linked geospatial data. *Journal of Web Semantics* **24,** 18–26. doi:10.1016/j.websem.2013.12.002 (2014).

11. Bone, C., Ager, A., Bunzel, K. & Tierney, L. A geospatial search engine for discovering multi-format geospatial data across the web. *International Journal of Digital Earth* **9,** 47–62. doi:10.1080/17538947.2014.966164 (2016).

12. Li, Z., Chiang, Y.-Y., Tavakkol, S., Shbita, B., Uhl, J. H., Leyk, S., *et al. An automatic approach for generating rich, linked geo-metadata from historical map images* in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (eds Gupta, R., Liu, Y., Tang, J. & Prakash, B. A.) (Association for Computing Machinery, New York, NY, USA, 2020), 3290–3298. doi:10.1145/3394486.3403381.

13. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., *et al.* Knowledge graphs. *ACM Computing Surveys (Csur)* **54,** 1–37 (2021).

14. Kejriwal, M. Knowledge graphs: A practical review of the research landscape. *Information* **13,** 161 (2022).

15. Shadbolt, N., Berners-Lee, T. & Hall, W. The semantic web revisited. *IEEE intelligent systems* **21,** 96–101 (2006).

16. Bizer, C., Heath, T. & Berners-Lee, T. *Linked data: The story so far* in *Semantic services, interoperability and web applications: emerging concepts* (IGI global, 2011), 205–227.

17. Consortium, W. W. W. *et al.* RDF 1.1 Primer (2014).

18. Consortium, W. W. W. *et al. SPARQL 1.1 overview* tech. rep. (World Wide Web Consortium, 2013).

19. McGuinness, D. L., Van Harmelen, F., *et al.* OWL web ontology language overview. *W3C recommendation* **10,** 2004 (2004).

20. Haklay, M. & Weber, P. Openstreetmap: user-generated street maps. *IEEE Pervasive Computing* **7,** 12–18. doi:10.1109/MPRV.2008.80 (2008).

21. Wick, M. & Vatant, B. The geonames geographical database. *Online at http://geonames.org* (2012).

22. Auer, S., Lehmann, J. & Hellmann, S. *Linkedgeodata: adding a spatial dimension to the web of data* in *The Semantic Web - ISWC 2009* (eds Bernstein, A., Karger, D. R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., *et al.*) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009), 731–746. doi:10.1007/978-3-642-04930-9{_}46.

23. Shbita, B., Knoblock, C. A., Duan, W., Chiang, Y.-Y., Uhl, J. H. & Leyk, S. Building Spatio-Temporal Knowledge Graphs from Vectorized Topographic Historical Maps. *Semantic Web* **14,** 527–549. doi:10.3233/SW-222918 (2023).

24. Leyk, S., Boesch, R. & Weibel, R. A conceptual framework for uncertainty investigation in map-based land cover change modelling. *Transactions in GIS* **9,** 291–322. doi:10.1111/j.1467-9671.2005.00220.x (2005).

25. Athanasiou, S., Hladký, D., Giannopoulos, G., García-Rojas, A. & Lehmann, J. GeoKnow: making the web an exploratory place for geospatial knowledge. *ERCIM News* **96,** 119–120 (2014).

26. Athanasiou, S., Giannopoulos, G., Graux, D., Karagiannakis, N., Lehmann, J., Ngomo, A.-C. N., *et al. Big POI data integration with linked data technologies* in *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019* (eds Herschel, M., Galhardas, H., Reinwald, B., Fundulaki, I., Binnig, C. & Kaoudi, Z.) (OpenProceedings.org, 2019), 477–488. doi:10.5441/002/edbt.2019.44.

27. Alirezaie, M., Längkvist, M., Sioutis, M. & Loutfi, A. Semantic referee: a neural-symbolic framework for enhancing geospatial semantic segmentation. *Semantic Web* **10,** 863–880. doi:10.3233/SW-190362 (2019).

28. Bernard, C., Plumejeaud-Perreau, C., Villanova-Oliver, M., Gensel, J. & Dao, H. *An ontology-based algorithm for managing the evolution of multi-level territorial partitions*

in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (eds Kashani, F. B., Hoel, E. G., Güting, R. H., Tamassia, R. & Xiong, L.) (Association for Computing Machinery, New York, NY, USA, 2018), 456–459. doi:10.1145/3274895.3274944.

29. Kyzirakos, K., Vlachopoulos, I., Savva, D., Manegold, S. & Koubarakis, M. *GeoTriples: a tool for publishing geospatial data as RDF graphs using R2RML mappings* in *Joint Proceedings of the 6th International Workshop on the Foundations, Technologies and Applications of the Geospatial Web, TC 2014, and 7th International Workshop on Semantic Sensor Networks, SSN 2014, co-located with 13th International Semantic Web Conference (ISWC 2014)* (eds Kyzirakos, K., Grütter, R., Kolas, D., Perry, M., Compton, M., Janowicz, K., *et al.*) **1401** (CEUR-WS.org, 2014), 33–44.

30. Usery, E. L. & Varanka, D. Design and development of linked data from the national map. *Semantic Web* **3,** 371–384. doi:10.3233/SW-2011-0054 (2012).

31. Vaisman, A. & Chentout, K. Mapping spatiotemporal data to RDF: a SPARQL endpoint for brussels. *ISPRS International Journal of Geo-Information* **8,** 353. doi:10 . 3390 / ijgi8080353 (2019).

32. Kurte, K. R. & Durbha, S. S. *Spatio-temporal ontology for change analysis of flood affected areas using remote sensing images* in *Proceedings of the Joint Ontology Workshops 2016 Episode 2: The French Summer of Ontology co-located with the 9th International Conference on Formal Ontology in Information Systems (FOIS 2016)* (eds Kutz, O., de Cesare, S., Hedblom, M. M., Besold, T. R., Veale, T., Gailly, F., *et al.*) **1660** (CEUR-WS.org, 2016).

33. Goodchild, M. F. Citizens as sensors: the world of volunteered geography. *GeoJournal* **69,** 211–221. doi:10.1007/s10708-007-9111-y (2007).

34. Vrandečić, D. & Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM* **57,** 78–85. doi:10.1145/2629489 (2014).

35. Car, N. J. & Homburg, T. GeoSPARQL 1.1: Motivations, details and applications of the decadal update to the most important geospatial LOD standard. *ISPRS International Journal of Geo-Information* **11,** 117 (2022).

36. Nagy, G. & Wagle, S. Geographic data processing. *ACM Computing Surveys (CSUR)* **11,** 139–181. doi:10.1145/356770.356777 (1979).

37. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. & Ives, Z. *DBpedia: a nucleus for a web of open data* in *The Semantic Web* (eds Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., *et al.*) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007), 722–735. doi:10.1007/978-3-540-76298-0_52.

38. Smeros, P. & Koubarakis, M. *Discovering spatial and temporal links among RDF data* in *Proceedings of the Workshop on Linked Data on the Web, LDOW 2016, co-located with 25th International World Wide Web Conference (WWW 2016)* (eds Auer, S., Berners-Lee, T., Bizer, C. & Heath, T.) **1593** (CEUR-WS.org, 2016).

39. Ahmed Sherif, M. & Ngonga Ngomo, A.-C. A systematic survey of point set distance measures for link discovery. *Semantic Web* **9,** 589–604. doi:10.3233/SW-170285 (2018).

40. Clementini, E., Sharma, J. & Egenhofer, M. J. Modelling topological spatial relations: strategies for query processing. *Computers & Graphics* **18,** 815–822. doi:10.1016/0097-8493(94)90007-8 (1994).

41. Sherif, M. A., Dreßler, K., Smeros, P. & Ngomo, A.-C. N. *Radon–rapid discovery of topological relations* in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (eds Singh, S. P. & Markovitch, S.) (AAAI Press, 2017), 175–181.

42. Pérez-Luque, A., Pérez-Pérez, R., Bonet-Garcıa, F. J. & Magaña, P. An ontological system based on MODIS images to assess ecosystem functioning of natura 2000 habitats: a case study for quercus pyrenaica forests. *International Journal of Applied Earth Observation and Geoinformation* **37,** 142–151. doi:10.1016/j.jag.2014.09.003 (2015).

43. Kauppinen, T., de Espindola, G. M., Jones, J., Sánchez, A., Gräler, B. & Bartoschek, T. Linked brazilian amazon rainforest data. *Semantic Web* **5,** 151–155. doi:10.3233/SW-130113 (2014).

44. Shbita, B. & Knoblock, C. A. *Automatically Constructing Geospatial Feature Taxonomies from OpenStreetMap Data* in *2024 IEEE 18th International Conference on Semantic Computing (ICSC)* (2024), 208–211.

45. Touya, G. & Reimer, A. Inferring the scale of OpenStreetMap features. *OpenStreetMap in GIScience: Experiences, research, and applications,* 81–99 (2015).

46. Kunze, C. & Hecht, R. Semantic enrichment of building data with volunteered geographic information to improve mappings of dwelling units and population. *Computers, Environment and Urban Systems* **53,** 4–18 (2015).

47. Vargas-Munoz, J. E., Srivastava, S., Tuia, D. & Falcao, A. X. OpenStreetMap: Challenges and opportunities in machine learning and remote sensing. *IEEE Geoscience and Remote Sensing Magazine* **9,** 184–199 (2020).

48. Minghini, M. & Frassinelli, F. OpenStreetMap history for intrinsic quality assessment: Is OSM up-to-date? *Open Geospatial Data, Software and Standards* **4,** 1–17 (2019).

49. Sun, K., Zhu, Y., Pan, P., Hou, Z., Wang, D., Li, W., *et al.* Geospatial data ontology: the semantic foundation of geospatial data integration and sharing. *Big Earth Data* **3,** 269–296 (2019).

50. Codescu, M., Horsinka, G., Kutz, O., Mossakowski, T. & Rau, R. Osmonto-an ontology of openstreetmap tags. *State of the map Europe (SOTM-EU)* **2011,** 23–24 (2011).

51. Dsouza, A., Tempelmeier, N., Yu, R., Gottschalk, S. & Demidova, E. *Worldkg: A world-scale geographic knowledge graph* in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), 4475–4484.

52. Dsouza, A., Tempelmeier, N. & Demidova, E. *Towards Neural Schema Alignment for OpenStreetMap and Knowledge Graphs* in *International Semantic Web Conference* (2021), 56–73.

53. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

54. Pennington, J., Socher, R. & Manning, C. D. *Glove: Global vectors for word representation* in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), 1532–1543.

55. Reimers, N. & Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

56. Le, Q. & Mikolov, T. *Distributed representations of sentences and documents* in *International conference on machine learning* (2014), 1188–1196.

57. Wang, Q., Mao, Z., Wang, B. & Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* **29,** 2724–2743 (2017).

58. Wang, Z., Li, J., Liu, Z. & Tang, J. *Text-enhanced representation learning for knowledge graph* in *Proceedings of International joint conference on artificial intelligent (IJCAI)* (2016), 4–17.

59. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25** (2012).

60. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770–778.

61. Le-Khac, P. H., Healy, G. & Smeaton, A. F. Contrastive representation learning: A framework and review. *Ieee Access* **8,** 193907–193934 (2020).

62. Gao, T., Yao, X. & Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

63. Herring, J. *et al.* Opengis® implementation standard for geographic information-simple feature access-part 1: Common architecture [corrigendum] (2011).

64. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. *A simple framework for contrastive learning of visual representations* in *International conference on machine learning* (2020), 1597–1607.

65. Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems* **29** (2016).

66. Tempelmeier, N., Gottschalk, S. & Demidova, E. *GeoVectors: A Linked Open Corpus of OpenStreetMap Embeddings on World Scale* in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), 4604–4612.

67. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., *et al.* GPT-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

68. Hu, Y. Geospatial semantics. *arXiv preprint arXiv:1707.03550* (2017).

69. Janowicz, K., Scheider, S., Pehle, T. & Hart, G. Geospatial semantics and linked spatiotemporal data–Past, present, and future. *Semantic Web* **3,** 321–332 (2012).

70. Janowicz, K., Gao, S., McKenzie, G., Hu, Y. & Bhaduri, B. GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond. *International Journal of Geographical Information Science* **34,** 625–636. doi:10.1080/13658816.2019.1684500 (2020).

71. Castelluccio, M., Poggi, G., Sansone, C. & Verdoliva, L. Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint arXiv:1508.00092* (2015).

72. Li, Z., Zhang, S. & Dong, J. Suggestive Data Annotation for CNN-Based Building Footprint Mapping Based on Deep Active Learning and Landscape Metrics. *Remote Sensing* **14,** 3147 (2022).

73. Klemmer, K., Safir, N. S. & Neill, D. B. *Positional encoder graph neural networks for geographic data* in *International Conference on Artificial Intelligence and Statistics* (2023), 1379–1389.

74. Kaczmarek, I., Iwaniak, A. & Świetlicka, A. Classification of Spatial Objects with the Use of Graph Neural Networks. *ISPRS International Journal of Geo-Information* **12,** 83 (2023).

75. Xu, Y., Zhou, B., Jin, S., Xie, X., Chen, Z., Hu, S., *et al.* A framework for urban land use classification by integrating the spatial context of points of interest and graph convolutional neural network method. *Computers, Environment and Urban Systems* **95,** 101807 (2022).

76. Yan, X., Ai, T., Yang, M. & Tong, X. Graph convolutional autoencoder model for the shape coding and cognition of buildings in maps. *International Journal of Geographical Information Science* **35,** 490–512 (2021).

77. Jenkins, P., Farag, A., Wang, S. & Li, Z. *Unsupervised representation learning of spatial data via multimodal embedding* in *Proceedings of the 28th ACM international conference on information and knowledge management* (2019), 1993–2002.

78. Li, Z., Kim, J., Chiang, Y.-Y. & Chen, M. SpaBERT: A Pretrained Language Model from Geographic Data for Geo-Entity Representation. *arXiv preprint arXiv:2210.12213* (2022).

79. Qiu, P., Gao, J., Yu, L. & Lu, F. Knowledge embedding with geospatial distance restriction for geographic knowledge graph completion. *ISPRS International Journal of Geo-Information* **8,** 254 (2019).

80. Woźniak, S. & Szymański, P. *Hex2vec: Context-Aware Embedding H3 Hexagons with OpenStreetMap Tags* in *Proceedings of the 4th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery* (2021), 61–71.

81. Shbita, B., Sharma, N., Vu, B., Lin, F. & Knoblock, C. A. *Constructing a knowledge graph of historical mining data* in *6th International Workshop on Geospatial Linked Data (GeoLD) Co-located with the 21st Extended Semantic Web Conference (ESWC 2024)* (2024).

82. Schulz, K. J. *Critical mineral resources of the United States: economic and environmental geology and prospects for future supply* (Geological Survey, 2017).

83. Fortier, S. M., Nassar, N. T., Lederer, G. W., Brainard, J., Gambogi, J. & McCullough, E. A. *Draft critical mineral list—Summary of methodology and background information—US Geological Survey technical input document in response to Secretarial Order No. 3359* tech. rep. (US Geological Survey, 2018).

84. Green, C. J., Lederer, G. W., Parks, H. L. & Zientek, M. L. *Grade and tonnage model for tungsten skarn deposits—2020 update* tech. rep. (US Geological Survey, 2020).

85. Day, W. C. *The Earth Mapping Resources Initiative (Earth MRI): Mapping the Nation's critical mineral resources* tech. rep. (US Geological Survey, 2019).

86. Hofstra, A. H., Lisitsin, V., Corriveau, L., Paradis, S., Peter, J., Lauzière, K., *et al. Deposit classification scheme for the Critical Minerals Mapping Initiative Global Geochemical Database* tech. rep. (US Geological Survey, 2021).

87. McFaul, E., Mason, G., Ferguson, W. & Lipin, B. *US Geological Survey mineral databases; MRDS and MAS/MILS* tech. rep. (US Geological Survey, 2000).

88. Kelley, K. D., Huston, D. L. & Peter, J. M. Toward an effective global green economy: The critical minerals mapping initiative (CMMI). *SGA News* **8,** 1–5 (2021).

89. Jaccard, P. The distribution of the flora in the alpine zone. 1. *New phytologist* **11,** 37–50 (1912).

90. Brodaric, B. & Richard, S. M. *The geoscience ontology* in *AGU Fall Meeting Abstracts* **2020** (2020), IN030–07.

91. Vu, B., Pujara, J. & Knoblock, C. A. *D-REPR: a language for describing and mapping diversely-structured data sources to RDF* in *Proceedings of the 10th International Conference on Knowledge Capture* (2019), 189–196.

92. Vu, B. & Knoblock, C. A. *SAND: A Tool for Creating Semantic Descriptions of Tabular Sources* in *European Semantic Web Conference* (2022), 63–67.

93. Consortium, W. W. W. *et al. Shapes constraint language (SHACL)* tech. rep. (World Wide Web Consortium, 2017).

94. Mudd, G. M., Jowitt, S. M. & Werner, T. T. The world's lead-zinc mineral resources: scarcity, data, issues and opportunities. *Ore Geology Reviews* **80,** 1160–1190 (2017).

95. Mudd, G. M. & Jowitt, S. M. The new century for nickel resources, reserves, and mining: Reassessing the sustainability of the devil's metal. *Economic Geology* **117,** 1961–1983 (2022).

96. Jaro, M. A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association* **84,** 414–420 (1989).

97. Chalk, S., Hodgson, R. & Ray, S. *QUDT toolkit: Development of framework to allow management of digital scientific units* in *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY* **253** (2017).

98. Qun, Y., Linfu, X., Yongsheng, L., Rui, W., Bo, W., Ke, D., *et al.* Mineral prospectivity mapping integrated with geological map Knowledge graph and geochemical data: A Case Study of gold deposits at Raofeng area, Shaanxi Province. *Ore Geology Reviews,* 105651 (2023).

99. Zhu, Y., Zhou, W., Xu, Y., Liu, J., Tan, Y., *et al.* Intelligent learning for knowledge graph towards geological data. *Scientific Programming* **2017** (2017).

100. Wang, C., Ma, X., Chen, J. & Chen, J. Information extraction and knowledge graph construction from geoscience literature. *Computers & geosciences* **112,** 112–120 (2018).

101. Karalis, N., Mandilaras, G. & Koubarakis, M. *Extending the YAGO2 knowledge graph with precise geospatial knowledge* in *The Semantic Web – ISWC 2019* (eds Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., *et al.*) (Springer, Cham, 2019), 181–197. doi:10.1007/978-3-030-30796-7_12.

102. Shbita, B., Rajendran, A., Pujara, J. & Knoblock, C. A. *Parsing, Representing and Transforming Units of Measure* in *Proceedings of the Conference on Modeling the World's Systems* (Pittsburgh, PA, 2019).

103. Krishnan, S., Wang, J., Wu, E., Franklin, M. J. & Goldberg, K. Activeclean: Interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment* **9,** 948–959 (2016).

104. Turk, M. J., Smith, B. D., Oishi, J. S., Skory, S., Skillman, S. W., Abel, T., *et al.* yt: A multi-code analysis toolkit for astrophysical simulation data. *The Astrophysical Journal Supplement Series* **192,** 9 (2010).

105. Pebesma, E. J., Mailund, T. & Hiebert, J. Measurement Units in R. *R J.* **8,** 486 (2016).

106. Chambers, C. & Erwig, M. *Dimension inference in spreadsheets* in *2008 IEEE Symposium on Visual Languages and Human-Centric Computing* (2008), 123–130.

107. Abraham, R. & Erwig, M. *Header and unit inference for spreadsheets through spatial analyses* in *2004 IEEE Symposium on Visual Languages-Human Centric Computing* (2004), 165–172.

108. Ochsenbein, F., Bauer, P. & Marcout, J. The VizieR database of astronomical catalogues. *Astronomy and Astrophysics Supplement Series* **143,** 23–32 (2000).

109. Simons, B., Yu, J., Cox, S., Piantadosi, J., Anderssen, R. & Boland, J. *Defining a water quality vocabulary using QUDT and ChEBI* in *Proceedings of the 20th International Congress on Modelling and Simulation* (2013), 2548–2554.

110. Hennessy, M., Oentojo, C. & Ray, S. *A framework and ontology for mobile sensor platforms in home health management* in *2013 1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS)* (2013), 31–35.

111. Dejanović, I., Milosavljević, G. & Vaderna, R. Arpeggio: A flexible PEG parser for Python. *Knowledge-based systems* **95,** 71–74 (2016).

112. Fisher, M. & Rothermel, G. *The EUSES spreadsheet corpus: a shared resource for supporting experimentation with spreadsheet dependability mechanisms* in *Proceedings of the first workshop on End-user software engineering* (2005), 1–5.

Appendices

A Identifying Units in Scientific Data

This section presents an unsupervised approach that identifies units in source data and provides a corresponding semantic representation using a standardized ontology for units of measurement [102]. As well as a method that enables scientists to perform automatic data transformations via unit conversions. The preliminary results demonstrate that the method can be used to automatically capture units in source data achieving an F_1-score of 0.48 in unit detection and parsing, and an accuracy of 62% in the semantic representation and transformation.

A.1 Motivation

An important task in data normalization and representation is the identification of units that are associated with quantitative data. Unit identification is challenging because it requires having some domain knowledge about the process that produced the data. For example, a scientist who wishes to integrate a hydrology model which associates groundwater-withdrawal units in gallons with an agricultural watering model that presumes input units in liters would need to manipulate the data to allow the composition of the data in the required units.

Today, scientists' approach to handling incompatibility in units is usually via a one-time transformation. If they associate data with the incorrect units, they might be unable to use the dataset and resort to ad-hoc strategies that harm the transparency and reproducibility of the results. The diversity in disciplines, domains and conventions in different regions around the world poses an additional challenge to this problem. Considering the vast amount of data presently used in modeling infrastructure this problem becomes intractable and tedious and is often susceptible to human error.

One potential solution to the difficulties of manual unit detection and conversion is the use of automated systems. Unfortunately, automating unit detection is a difficult task. Frequently, units appear in files within datasets in a textual representations that is not easily recognized. These text strings usually contain a formula-like form: unit abbreviations, exponents, and additional elements

which represent an atomic or a compound unit. An atomic unit is a single unit symbol which may be modified by additional elements such as exponents or prefixes (i.e. `GHz`, `m^2`) whereas compound units are composed from two or more atomic units with some relationship between them (i.e. `A/cm^2`). Textual representation is not sufficient and does not carry any semantic or dimensional meaning and may require additional investigation if one needs to perform transformations and data alignment. Additionally, lexical conventions such as capitalizations are very important when representing units to resolve semantic ambiguities. For example: the text string `S` would stand for `Siemens` (electric conductance unit) where `s` would represent a unit of type `second` (time unit). Supporting SI (International System of Units) prefixes complicates the problem even more, adding an additional layer of combinatorial complexity.

Previous research [103–105] has developed approaches to allow easier data representation, cleaning and transformation for handling numerical and measurement data. However, previous research still depends on human interaction in early data processing stages. Published ontologies are a beneficial resource that can be used for a faster process of data cleaning. NASA has published an ontology called QUDT (Quantity, Unit, Dimension and Type) [97] which defines the base classes and attributes used for modeling physical quantities, units of measure, and their dimensions. The main drawback of existing unit ontologies is the amount of effort required to represent datasets, there is no readily an available tool to generally link data to the units used in the datasets. Thus it is intuitive to extend and integrate existing ontologies into a framework to enable an automatic and fast process of data understanding, normalization and transformation.

Problem Definition The problem we address is given arbitrary dataset files, in some common textual format (e.g. `json`), find textual explicit mentions of measurement units and map them to the QUDT units ontology [97]. Using the semantic representation we want to produce a structured standard ontologized output that can be easily interpreted by humans and machines. Then, we want to leverage the ontology to assist users in common modeling transformations such as complex compound units conversion. As an example, consider the compound unit `g/t` (marked in

a red box) seen in the observed table shown in Figure 6.1. A conventional semantic representation for this string is shown in Listing 6.1 where each base unit is an element of the list labeled with `qudt:_hasPart_` and its URI is introduced within the key labeled `qudt:quantityKind`. Besides the unit ontology, each part has additional attributes such as `qudt:exponent`. Additionally, `qudt:abbreviation` and `qudt:hasDimension` describe the properties of each element in the unit and are computed for the overall normalized compound unit, and the attribute labeled `qudt:conversionMultiplier` can be easily utilized for a unit conversion service.

Table 17-2: Concentrate Composition from 2006 Piloting

Grade	Assays					
	Cu (%)	Ni (%)	S (%)	Au (g/t)	Pt (g/t)	Pd (g/t)
Concentrate	7.16-10.1	1.66-2.20	18.4-21.5	0.65-1.28	1.17-1.59	5.76-6.71

Figure 6.1: A compound unit in source data (semi-structured table in a `pdf` file).

```
{
    qudt:hasDimension: "",
    qudt:abbreviation": "g tonne-1",
    ..
    qudt:hasPart: [
    {
        qudt:hasDimension: "M",
        qudt:quantityKind: "http://data.nasa.gov/qudt/owl/unit#Gram",
        qudt:conversionMultiplier: 0.001,
        qudt:conversionOffset: 0.0,
        qudt:symbol: "g"
      },
      {
        ccut:exponent: "-1",
        qudt:hasDimension: "M",
        qudt:quantityKind: "http://data.nasa.gov/qudt/owl/unit#MetricTon",
        qudt:conversionMultiplier: 1000.0,
        qudt:conversionOffset: 0.0,
        qudt:symbol: "t"
      }
    ]
}
```

Listing 6.1: partial JSON representation for `g/t`.

Figure 6.2: An example of a detected compound unit and its representation.

Significance In this chapter, we present an approach to automatically detect, parse, normalize and represent compound and atomic units of measure in a data source. Using the semantic representation we are able to support scientists by providing the ability to perform unit conversions that are less prone to error. To demonstrate our idea, we implemented a prototype system, called

CCUT (**C**anonicalization **C**ompound **U**nit Representation and **T**ransformation), which uses grammar tools to automatically parse the different components in a unit found in textual data in files and map them to elements of a standard ontology defined by NASA [97] to form a structured semantic output. The output depicts the different relationships, attributes and semantics of units and allows users to safely perform a transformation between units. Our method was tested on spreadsheets and can be easily deployed over a range of quantitative data resources and thus accelerate and improve the modeling process in any scientific domain.

A.2 Related Work

Chambers and Erwing [106] presented a reasoning system for inferring dimension information in spreadsheets. Their system aims to check the consistency of formulas and detect errors in spreadsheets. Abraham and Erwig [107] developed a VBA-based add-on for excel which enables the detection of errors and which is based on a set of rules for automatic header inference. Although these systems do not require any user intervention for their operation, they do not offer a semantic representation or any conversion services as we present in our work.

Existing frameworks such as VizieR [108], the yt Project [104] and Measurement-units-in-R [105] attempt to deal with the problem of unit representation and conversion by giving users the option to enforce a unit of measure for a given fixed set of data. This enables one to add, subtract, multiply, and divide using quantities and dimensional arrays. When used in expressions, some of these platforms automatically convert units, and simplify them when possible. Measurement-units-in-R gives the user the flexibility to expand beyond predefined units but it requires an initial user definition and understanding of data. Our work differs by providing users the ability to capture the semantics behind units given a string without any initial definitions since it uses a standard and structured representation defined by NASA.

The NASA QUDT ontology [97] is being adopted in many scientific research projects [109, 110]. Of the established and well-governed unit of measure ontology options, QUDT is well-aligned with our understanding of the relationships between measurements and units of measure.

A.3 Parsing, Representing, and Transforming Units of Measure

Our approach tackles three core problems, these are illustrated in Figure 6.3 and can be summarized as follows:

1. Identify and parse the individual prefixes, atomic units, their exponents and multipliers which compose a string of a compound unit. As shown in the first transition (marked as **Parse**) in the figure (Section A.3.1).

2. Map each atomic unit to its correct ontology in the schema. As shown in the second transition (marked as **Map**) in the figure (Section A.3.2).

3. Compute the dimension of the compound unit and construct a normalized representation of the compound unit with attributes that are required for transformation. As shown in the third transition (marked as **Infer**) in the figure (Section A.3.3).

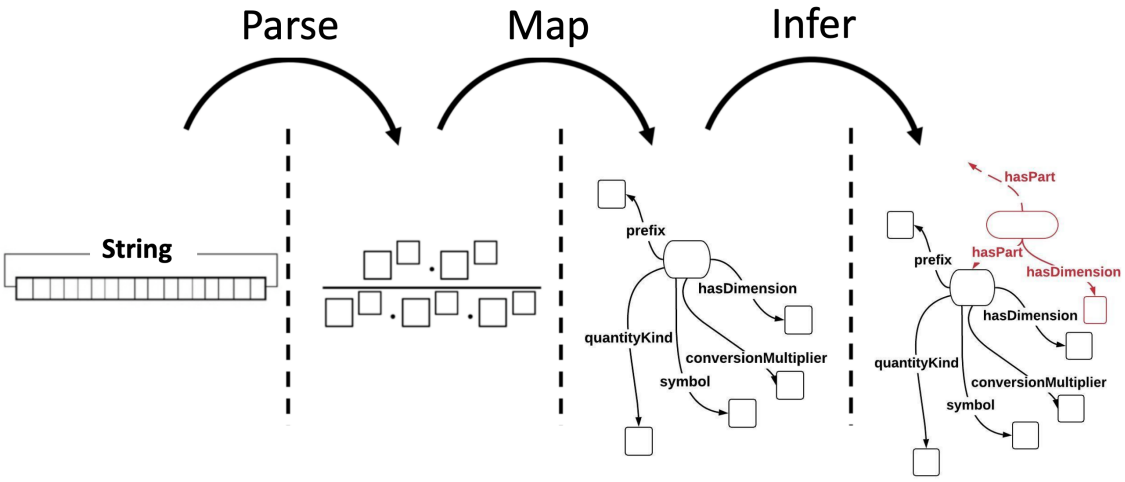


Figure 6.3: Flowchart describing our approach.

A.3.1 Parsing

Given a string that contains units, we want to generate a structured form which represents that unit and provide a set of relationships between its components. In this stage, there are a few problems we need to solve.

First, in the QUDT ontology there are some units that are missing a complete definition and have partial dimension information. In order to address this issue, we created a closed set of meaningful units and defined base dimension classes which were derived from the ontology.

Second, atomic units may be accompanied with unit prefixes (such as `kilo`, `micro` or `mega`), and may have different variants that are used to symbolize the same prefix. For example: a prefix of type micro (10^{-6}) may show up in text as `micro`, `mu` or μ. Additionally, there are some cases in which our parser can run into ambiguous cases. For example: the unit string `min` can be interpreted as `minute` (time unit) or as a combination of the prefix `m` as in `milli` (10^{-3}) and the unit `inch` that is abbreviated as `in` (length unit). To handle these issues we manually encoded 20 prefix classes that characterize unit prefixes in addition to their well-known different variants, and implemented an iterative joint matching algorithm for $\{$*prefix, unit*$\}$ pairs. The algorithm gives higher confidence to a mapping of a single atomic unit and iteratively keeps searching for a joint match in case of a failure in the first attempt until the two elements are mapped to a well-defined terms correspondingly.

Last, we must consider the different characters which represent the relations between the atomic units (i.e. fractions, exponents, etc) in a string. To tackle this issues, we implemented a well defined grammar, that is, a specification for how to read a language of compound and atomic units. We used Arpeggio [111] as a grammar parser. Arpeggio is a recursive descent parser with backtracking and memoization that is based on PEG (Parsing Expression Grammar) formalism. Table 6.1 depicts some grammar rules we have defined.

Table 6.1: Some grammar rules.

Character	**Rule**
`/`	Split to numerator & denominator
`^`	Interpret as an exponent
`( and )`	Split to canonical form
`-`	Interpret as a negative sign
`.`	Interpret as a floating point

A.3.2 Structured Unit Representation

While the structured representation is more informative, it still does not capture a semantic meaning. In order to provide the semantic meaning we have to rely on ontologies that define universal conventions for units. The QUDT ontology defines a unit symbol (`qudt:symbol`) that is associated with a unit ontology instance (`qudt:Quantity`) that has a unique URI. This unit instance is associated with the symbol using a relation of type `qudt:QuantityKind`. This relation is an instance in itself and is associated to some dimension (`qudt:Dimension`). Given the above, we are able to map an atomic unit string to its semantic type and find its dimensions. Utilizing the additional grammar output terms (i.e. exponents) enables us to normalize the compound unit and present an interpretable representation.

As mentioned earlier in Section A.3.1 we implemented a tight integration between the prefix classes and the unit classes. This provides an efficient solution and insures a proper semantic representation. Since each element is linked to a Uniform Resource Identifier (URI) and is uniquely identified by it, our solution provides a cost-free representation.

A.3.3 Transforming Compound and Atomic Units

In this stage, our goal is to enable arbitrary transformations between units using our semantic structured representation which we have already generated and mapped. A dimensions-based approach encoded in the QUDT ontology relates each unit to a system of base units using numeric factors. For example, any measurement of length can be expressed as a number multiplied by the unit `meter` (the SI base for the length dimension). Given that, and the set of exponents, prefixes and multipliers derived from the grammar and applied over a set of fundamental dimensions, we are able to generate the required calculation to perform unit conversions of same dimension. We use the conversion multiplier and offset, which are multiplied and added to quantities to convert from the current unit to the corresponding SI unit. So, if m_1, o_1 and m_2, o_2 are the conversion multipliers and offsets for U_1 and U_2 respectively, m_{p1}, o_{p1} and m_{p2}, o_{p2} are the conversion multipliers and offsets of their prefixes respectively, and α and β are their exponents respectively, then the proper

conversion is according to Equation A.1. When a conversion is desired between two SI units, their offsets are equal to zero by definition and therefore we get a simplified form as in Equation A.2. Thus, we can offer a transformation service to users via an additional service endpoint in the system to enforce correct and safe conversions.

$$U_2 = \frac{\left(\frac{(U_1 \cdot m_{p1}^{\alpha} + o_{p1}) \cdot m_1^{\alpha} + o_1 - o_2}{m_2^{\beta}}\right) - o_{p2}}{m_{p2}^{\beta}} \tag{A.1}$$

$$U_2 = U_1 \frac{(m_1 \cdot m_{p1})^{\alpha}}{(m_2 \cdot m_{p2})^{\beta}} \tag{A.2}$$

A.4 Evaluation and Discussion

We have employed the EUSES spreadsheet corpus [112] and implemented an `xls` file reader for the purpose of testing against our API endpoints of the CCUT system. The corpus contains 1,345 files and an overall number of 5,891 spreadsheets collected from different sources. We randomly sampled 30 files that included 112 spreadsheets. Explicit unit strings were present only in 31 spreadsheets. We used the sampled set as a testing set and for which we created a validation file to compare the dimensional analysis and the different elements to (including their URIs and normalized form). In the given set 267 compound units (and a total of 530 atomic units) were observed and manually annotated in the validation set. In addition, we defined an in-code python dictionary to keep track of the observed compound unit strings which have the same normalized dimension in order to test the accuracy of our transformation service between each pair of units in the same dimension.

CCUT detected and parsed a total of 882 atomic elements (328 true positives; 554 false positives) and misdetected 150 elements (false negatives), generating a total precision of 0.37, a recall of 0.69 and an overall F_1-score of 0.48. Out of the valid compound units (true positives) 62.12% were normalized correctly (overall dimension representation was precise). A total of 11 distinct (compound) dimension groups were identified, out of which 5 groups included more than a single

distinct string representation, providing us a total of 42 transformation test cases of pairs which had a 100% accuracy in the transformation calculation. This is normally what we expected in the transformation tests since these compound units were accurately captured and represented.

We examined cases where we were unable to map the correct unit and discovered that in some cases the units were detected in irrelevant text. Some of the detected units were mistakenly extracted from abbreviations for other entities or organizations. These issues caused an overall low precision score. In other cases there is an ambiguity because the same abbreviation is used in multiple units (i.e. `L` stands both for `liter`, a volume unit, and `lambert`, a luminance unit). Our system does not currently support a principled mechanism for using context to make a more intelligent prediction. In some cases we simply did not have the correct unit in our knowledge base which affected our recall.