

Towards Leveraging Link Prediction for Geospatial Data Integration

Albulen Pano*, Mattia Fumagalli, Davide Lanti and Diego Calvanese

Free University of Bozen-Bolzano, Faculty of Engineering, 39100 Bolzano, Italy

Abstract

Link prediction is a technique used to predict new relationships between entities in a given graph. There are several domains in which this technique is applied. Those span from social media friendship links suggestions to correlated products prediction. Nevertheless, the use of link prediction to support knowledge integration is still a subject of debate, especially in the context of geospatial data. In this paper, we aim to discuss the role and some of the potential benefits of link prediction in the context of geospatial data completion and integration. To this end, we aim to position and discuss the role of geospatial link prediction within the framework of Ontology-Based Data Access (OBDA), highlighting the potential contribution of link prediction in this field. Additionally, we present a series of preliminary experiments designed to predict relationships of “competition” among business activities within a specific geographic area. Finally, we explore how the injection of knowledge from information-rich schemas about concepts related to the geospatial domain can positively influence the accuracy of the prediction model.

Keywords

Knowledge graphs, Link prediction, Geospatial data, Knowledge completion, Knowledge integration

1. Introduction

Semantic geospatial applications, such as geographic search engines, heavily rely on knowledge graphs (KGs). These can be the output of the integration of several autonomous and independent data sources. For instance, the knowledge graph about a specific geographic area may result from the integration of data related to the buildings of that area, streets and even green areas. Geospatial knowledge graphs are commonly designed to support map navigation, enhance the visualization of geographic regions, and provide insights into spatial relationships. Additionally, they can be enriched with semantic information beyond purely spatial attributes. For instance, they may capture relations between locations frequently co-visited by tourists, highlight maximum occupancy rates of specific buildings, or represent other contextual associations that do not strictly depend on spatial properties.

A significant challenge in geospatial knowledge graphs is their extraction from diverse sources. These are often available only in unstructured (textual) or semi-structured data (e.g., JSON, XML). Moreover, they often result in incomplete representations and limited interoperability with other knowledge graphs containing related information. Addressing these limitations requires techniques for KG enrichment to enhance the density and depth of knowledge representations, thereby improving completeness and even enabling the derivation of new knowledge.

KG enrichment approaches can be broadly categorized into two main classes: (1) logic-based reasoning and (2) machine learning (ML)-based KG completion. The logic-based approach employs automated reasoning techniques over ontological axioms, as outlined by Baader et al. [1], or rule-based systems, such as those described by Horrocks et al. [2], to infer new triples that are not explicitly asserted in the knowledge graph. For geospatial data, the GeoSPARQL standard facilitates spatial reasoning by enabling the inference of geospatial relationships [3]. While logic-based approaches can yield highly accurate results, they are contingent on the availability of high-quality input data and explicitly defined ontological axioms or rules.

SEBD 2025: 33rd Symposium On Advanced Database Systems, Ischia (Italy), 16 Jun - 19 Jun 2025

*Corresponding author.

✉ albulen.pano@unibz.it (A. Pano); mattia.fumagalli@unibz.it (M. Fumagalli); davide.lanti@unibz.it (D. Lanti); diego.calvanese@unibz.it (D. Calvanese)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

In contrast, ML-based approaches offer greater flexibility in handling incomplete or noisy data, especially when axioms and rules are missing or difficult to formulate. These methods first transform the KG into a high-dimensional vector space using embedding techniques, after which link prediction algorithms rank candidate missing links based on learned patterns. However, most existing KG embedding models overlook spatial characteristics, leading to suboptimal performance in geospatial applications. In response to this limitation, Mai et al. [4] introduced SE-KGE, a location-aware KG embedding model that explicitly incorporates spatial information, such as point coordinates and bounding boxes, directly into the KG embedding space, thereby enhancing geospatial inference capabilities.

In this paper, building upon a pipeline for geospatial data integration proposed in a previous study, we aim to discuss the introduction of a new module that leverages the link prediction technique. We assume that this new module can serve as a support for the existing data integration components within the current pipeline. Furthermore, the link prediction module itself can benefit from the knowledge graph generated through the pipeline. As this is a discussion paper, the issues addressed herein are primarily introductory. However, to make the discussion more concrete, in addition to presenting the extended version of the integration pipeline incorporating the link prediction module, we also describe several tests conducted on an existing dataset, reusing established strategies for applying link prediction in the context of geospatial data.

The structure of the paper is as follows: in section 2 we introduce some works that are relevant for our proposal, in particular by focusing on similar approaches that leverage link prediction in the context of geospatial data. In section 3 we briefly discuss the overall approach of geospatial data integration upon which we want to plug-in a new link-prediction component and we describe the role of the link prediction module itself. In section 4 we discuss some preliminary experiments to explore the feasibility and the potential utility of our proposal. section 5 is about conclusion and future work.

2. Related Work

2.1. Geospatial Link Prediction

Link prediction is a technique used to predict relationships between nodes in a graph [5]. Typically, in its simplest setup, this technique involves transforming a graph into an adjacency matrix. This matrix is then used to train a predictive algorithm, which can later be employed to predict relationships in a new graph provided as test input (also encoded as an adjacency matrix). In this context, most link prediction algorithms are designed to address ranking problems by assigning a score proportional to the likelihood of a relationship between two nodes. A threshold is set by the algorithm or the user, and node pairs with scores exceeding this threshold are considered positive predictions. In this sense, link prediction can be framed as a binary classification problem.

This technique is widely used across various domains for specific tasks, such as suggesting friendship links in social networks, identifying hyperlinks between websites, or recommending related products based on browsing profiles in e-commerce. In recent years, the application of link prediction to geospatial data has gained traction, yielding significant results. For instance, Liu et al. [6] employed link prediction to suggest optimal geographic locations for opening commercial establishments within urban areas. Another example is provided by Mann et al. [7], where various supervised and unsupervised machine learning adaptations of this technique were tested in the context of sparsely interlinked geospatial knowledge graphs.

2.2. Geospatial Knowledge Graphs

Geospatial knowledge graphs are KGs containing geospatial objects, geometries, and their relationships. GeoSPARQL is an *Open Geospatial Consortium (OGC)* standard for representing and querying geospatial KGs [8]. The GeoSPARQL ontology introduces classes such as features, geometries, and their

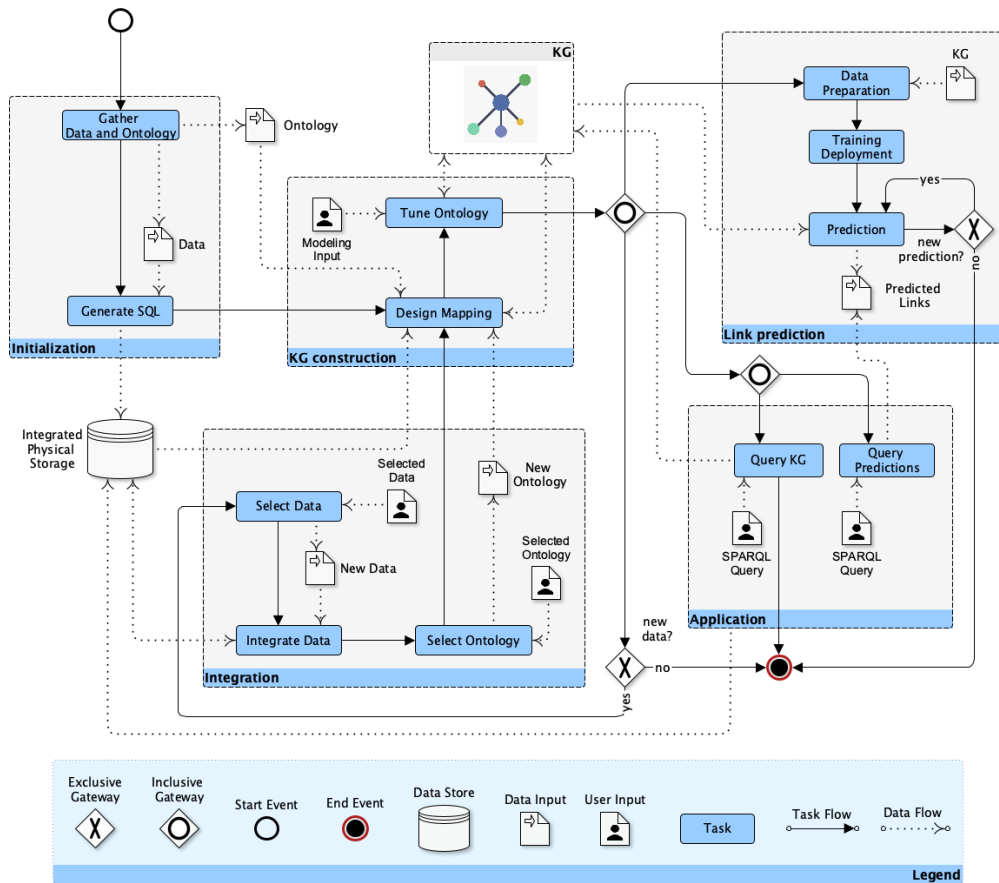


Figure 1: Overview of the proposed methodology.

representations using the *Geography Markup Language (GML)*¹ and *Well-Known Text (WKT)*² literals. It also includes vocabularies for topological relationships. Additionally, GeoSPARQL extends the standard SPARQL query language with topological functions for quantitative reasoning.

Geospatial KGs are often converted from geospatial data sources stored in spatial databases or other popular formats such as *Shapefiles*. The ontology-based data access (OBDA) paradigm provides a systematic approach to this conversion, by allowing end-users to access data sources through a domain ontology. Typically, the domain ontology imports the GeoSPARQL ontology and is semantically linked to the data sources via a *mapping* expressed in the R2RML language [9], standardized by the W3C.

One of the most well-known geospatial KG projects is LinkedGeoData [10], which primarily follows the OBDA approach to expose data from *Open Street Map (OSM)*³ as geospatial KGs.

3. Methodology

Our proposal builds upon a geospatial data integration pipeline previously described in a prior work [11]. Figure 1, using a Business Process Model (BPMN),⁴ presents a simplified version of this pipeline, incorporating a new set of tasks that represent the novel aspect we aim to discuss. The depicted pipeline primarily serves two purposes. First, it demonstrates how a KG can be generated to support query-answering services on geospatial data. Second, it illustrates how the KG can be evolved by

¹<https://www.ogc.org/publications/standard/gml/>

²<https://libgeos.org/specifications/wkt/>

³<https://www.openstreetmap.org/>

⁴Note that BPMN is a conceptual modeling language adopted to represent tasks and procedures within a system. For more information about BPMN, the authors refer the readers to [12].

incorporating new data and knowledge, thereby extending query-answering services. Concerning this second point, we aim to discuss how tasks related to link prediction can play a useful role.

As shown in the figure, the pipeline consists of five main phases: (i) *Initialization*, (ii) *KG Construction*, (iii) *Integration*, (iv) *Link Prediction*, and (v) *Application*. Each of these phases comprises tasks or steps that can receive and/or produce different types of data. The group labeled as ‘KG’ represents the resource that supports query-answering activities, which are described by the tasks in the application phase.

In this context, we do not delve into the details of what constitutes the KG. It suffices to note that it can be either a *Virtual Knowledge Graph* (VKG) or a *Materialized Knowledge Graph* (MKG). The VKG consists of two subcomponents: an *ontology* function and a *mapping* function, which are used to generate RDF triples on demand from physical storage. In contrast, representing the KG as an MKG eliminates the need for a virtualized pipeline for RDF [13] data, at the cost of increased storage requirements and the need to rematerialize RDF triples whenever the source data change. All these components can then evolve through the steps in the integration phase. For more information, we refer readers to [14, 15].

Returning to the phase descriptions, *Initialization* primarily aims to generate an integrated physical data storage from input data (e.g., *CityGML*⁵ data) while also providing an initial ontology to represent the data (e.g., the *CityGML* ontology⁶), as a base version of a knowledge graph. To create the target data repository, the input dataset is incorporated into a relational database (see *Generate SQL* in Figure 1), mapping each data row to entities and columns to specific information fields. This ensures that the data can be queried, retrieved, stored, and updated. Although various automated and complete solutions are available for this phase, some custom *ad hoc* refinements may be necessary. This is because the physical storage of this phase’s output must align with the technology used to generate the knowledge graph in the subsequent phase.

The second phase, *KG Construction*, aims to generate a KG that serves as a reference point for query-answering activities. This phase can be iterated multiple times. The first iteration occurs after the initialization phase, using as input the ontology selected in the initialization phase, the *Integrated Physical Storage* that houses the selected data, and the *mapping* required to link the two. In subsequent iterations, the input to this phase comes from the integration phase’s output. In both cases, the KG construction phase primarily deals with defining the ontology and its corresponding mappings, with three key objectives: (i) defining the set of concepts, relationships, and properties within the reference knowledge domain; (ii) capturing the semantics of stored information to enable enhanced reasoning and inference capabilities; and (iii) fostering interoperability among integrated data sources.

A crucial aspect is identifying an existing ontology that best captures the semantics of the selected data. Once the ontology is selected, a mapping phase follows, aligning the database generated in the initialization phase with the ontology concepts. If the information cannot be straightforwardly mapped, manual intervention is required, typically involving modifications to the selected ontology to adequately incorporate the information stored in the physical repository.

After completing the initialization and KG construction phases, a KG is ready to support query-answering activities. However, our approach also enables the integration of additional data sources, and here comes into play the *Integration* phase, allowing for the creation of an extended KG beyond its initial version. As described in prior work, this evolution is addressed in the integration phase, where the key tasks are: (i) user selection of new data sources, (ii) integration of the new data into the existing physical storage, and (iii) user selection of a new ontology or additional ontological information to account for the integrated data.

The new *Link Prediction* phase is designed to support the integration phase by training a predictive model capable of inferring new links within the KG. A crucial step in this process is *data preparation*, which involves key sub-steps such as *database generation*, *area discretization*, the *creation of new relations*, *KG construction*, and *ontology tuning*.⁷ The model can then be trained and deployed using a

⁵<https://www.ogc.org/publications/standard/citygml/>

⁶<https://smartcity.linkeddata.es/ontologies/cui.unige.chcitygml2.0.html>

⁷Note that some of these sub steps can be also handled in the data integration phase. Here, due to lack of space, we do not delve into this aspect.

partially extended version of the KG, incorporating newly inferred relationships (as demonstrated in the experimental example in the next section). Alternatively, training can be conducted on new graphs. Training can also be extended by leveraging *embeddings* to enhance predictive performance. The type of input graph depends on the adopted prediction model, which may support either link prediction alone or both link and node prediction. It should be noted that, at present, how to integrate newly predicted information remains an open question. One possible approach is to allow users to decide whether to incorporate the predicted information after reviewing the output.

Finally, the last phase of the entire process consists of the tasks grouped under *Application*, which is dedicated to using the output KG. In the existing solution, users are enabled, via a dedicated *ad hoc* interface, to query the KG using SPARQL queries, with results presented in textual or visual formats. With the introduction of the link prediction phase, users may also request information about newly predicted links within the KG.

4. Preliminary Experiment

We present below the preliminary experiment we conducted to explore the link prediction technique’s usefulness within the pipeline described in section 3. Specifically, the research questions we want to address are:

- **RQ1.** *How can link prediction be leveraged to complete a target KG?*
- **RQ2.** *How can the results of the link prediction model be made more reliable?*

All files and processes used in the experiment are available on a *GitHub* repository.⁸

4.1. Setup

The experiment was conducted on a high-performance computing (HPC) cluster equipped with *NVIDIA A100-SXM4-80GB GPUs*. A standard computer is insufficient due to the high computational cost of the link prediction task, especially in the context of geospatial data where graphs are rich in information. The library used to run the models was *Pykeen*,⁹ a Python package capable of running easily reproducible knowledge graph embedding (KGE) models. Pykeen (v. 1.11.0) was selected for its extensive library of *Knowledge Graph Embedding (KGE)* models and its simple code execution pipeline.

4.2. Resources

For this preliminary discussion, we limited the experiment to a single geospatial data source, *Open-StreetMap (OSM)*, an open and free map database with volunteered geographic information. The discussion of the role of the methodological integration phase’s iterations with additional data sources and ontologies, as discussed in section 3, is deferred to a more extended future work. We selected as the area of interest the city of Bolzano-Bozen (Italy) and its immediate surroundings by defining a bounding box with limits **11.3°E** to **11.4°E longitude** and **45.52°N** to **46.45°N latitude**. We retrieved an OSM dataset from *Geofabrik*¹⁰, which stores daily data dumps of OSM data and filtered it based on Bolzano’s coordinates. OSM relies on geographic information contributed by volunteers, resulting in a diverse array of key-value pairs—over 13,000¹¹ in Italy alone. While numerous OSM ontologies exist, we chose the widely recognized LinkedGeoData ontology, which encompasses more than 150 classes and helps to structure the knowledge base more effectively.

Although the LinkedGeoData ontology serves as a starting point for our analysis, it suffers from several pitfalls due to its relatively flat structure. It contains very few object properties making it difficult to discriminate specific subclasses, e.g., both `lgdo:Restaurant` and `lgdo:Bakery` are subclasses

⁸<https://github.com/D2G2Project/KGLinkPrediction>

⁹<https://pykeen.readthedocs.io/en/stable/>

¹⁰<https://www.geofabrik.de/geofabrik/>

¹¹<https://taginfo.geofabrik.de/europe:italy/keys>

of `lgdo:Amenity` and do not have any idiosyncratic properties to differentiate them. Therefore, for our link prediction task we choose an additional well-known vocabulary to enrich the graph to be used for training the link prediction model. Such a vocabulary is Schema.org, version 28.1.¹² It is important to note that the number of classes modelled decreases from 115 to 52 when moving from LinkedGeoData to Schema.org, as the latter is less expressive in its taxonomy for our use case. For instance, while Schema.org uses a single combined class `BarOrPub`, LinkedGeoData distinguishes between two separate classes `Bar` and `Pub`. However, this reduction in class granularity comes with a trade-off as Schema.org offers greater richness in properties. An example of this is the `hasMenu` property, which Schema.org applies to various food establishments but not bakeries.

4.3. Tests

We structured the experiment into two main tests, both following the same two-step process: *data preparation* and *model deployment*. In the second test, we introduced an additional step—*knowledge injection*—which can be considered a sub-step of *data preparation*.

In the first test, we focused on *data preparation* and *model deployment* without incorporating *knowledge injection*, primarily addressing **RQ1**. In contrast, the second test included *knowledge injection*, with a primary focus on **RQ2**. For both tests, we applied the same *parameters* and *evaluation metrics*.

4.3.1. Data preparation

Database generation. As a first step, to load OSM data for Bolzano we relied on a *PostgreSQL database*, version 17.3, and its geospatial extension PostGIS version 3.5. The geospatial extension is needed to render geometry data. A total of 532,794 entities were retrieved. Two sample rows and the first five columns of the **entities** table are provided in Table 1.

Table 1

Example entities stored in the PostgreSQL database.

osm_type	osm_id	class	geom	name
N	4234022289	Restaurant	POINT(11.1983051 46.6838314)	Restaurant - Cafe Georgenhof
N	2503251423	Bakery	POINT(11.1892331 46.6871877)	Bäckerei - Konditorei Höfler

Areas discretization. Secondly, following the approach adopted in the UrbanKG work [16], where the cities analyzed were divided into business areas based also on roads, we adopted a geographic discretization step, namely we divide the areas of our dataset into a finer group of sub-areas of interest. For that purpose, we used the OSM defined *Primary*, *Secondary* and *Tertiary highways* to segment the city of Bolzano into sub-areas. We then added to each record in the OSM relational table a link to the respective sub-area it belongs to. This procedure is mainly intended to improve the connection paths in the knowledge graph between geographically closer entities.

Competitive Relation. With this step, we introduced a new relation (note that this step can also be addressed in the *Integration phase* of the pipeline), namely the *competitive* relation, which we selected as the target link for prediction. For the purpose of this experiment, the competitive relation is initially defined to hold between any two entities belonging to the same OpenStreetMap (OSM) category, provided that their absolute distance from each other does not exceed 500 meters.

KG construction. Thirdly, we constructed the KG utilizing *Ontop v.5.1.1* [15],¹³ an open-source platform that provides support for querying over relational databases using Semantic Web technologies, specifically the RDF data model, SPARQL [17] query language, OWL 2 QL [18] ontology, and R2RML mapping language. To create the KG in Ontop, we designed mappings between the PostgreSQL database and the ontology. A mapping consists of three components: 1) mapping identifier, 2) relational source, and 3) target. The mapping identifier is any unique identifier. The source refers to an SQL query expressed over a relational database to retrieve data. The target is an RDF triple pattern(s) that uses the answer variables from the preceding SQL query as placeholders. A sample mapping is presented below for `rdf:type`:

¹²<https://github.com/schemaorg/schemaorg/tree/main/data/releases/28.1>

¹³<https://ontop-vkg.org/>

```

mappingId  OSM classes
target     lgd:entity/{osm_id} a lgdo:{class_name} .
source     SELECT "osm_id", "class_name"
           FROM public.entities LEFT JOIN public.classes ON entities.class = classes.
           class_id::TEXT

```

For these experiments, we limited mappings to `rdf:type`, `:competitive` as well as `:locateAt` and `:borderBy`. We generated the relations `:locateAt` for OSM points of interest (POIs) within these sub-areas and `:borderBy` for sub-areas that border one-another (i.e., share an edge) based on the geographic discretization described previously whereas we discuss `:competitive` separately further below.

Ontology tuning. The choice to add a business-oriented relation such as “competitive” to our graph made it necessary to further filter the LinkedGeoData classes used in the ontology. Specifically, we filtered classes where a relation in the context of commercial operations is meaningful. Therefore, we preserve classes like Bar, Restaurant, Hotel but remove classes like Museum and Community Center, leaving in total 115 classes. The filter is also applied to the corresponding classes in Schema.org.

With the specification of the ontology and mappings, we leveraged Ontop to generate a knowledge graph (KG) based on the relational OSM data. Finally, the virtual knowledge graph (VKG) can be transformed into a materialized knowledge graph (MKG) in RDF format.

4.3.2. Model Deployment

We implement *Knowledge Graph Embedding (KGE)* models to define the embeddings and learn new links. For this experiment, we focused on 2 transductive KGE models *TransE* [19] and *TransR* [20], but the experiments can also be applied to inductive models. Rather than use random initialization, we also leveraged pre-trained Space2Vec [4] geospatial embeddings as a beginning point. We adopted this option to check the role of the embeddings in performance improvement. Spatial awareness, i.e. make the embeddings of entities that are geographically closer also have closer embeddings in the n-dimensional space, can help the model achieve better convergence and performance and, perhaps, also allow the preservation of spatial relationships.

As a last point, in the experiment, no text embeddings were used. Bolzano-Bozen is a unique case due to its bilingual nature, and the issue of text embeddings for the names of the establishments needs to be analyzed in detail separately. This is an issue that can be revisited in the future.

4.3.3. Knowledge injection

This step was addressed in the second test of the experiment, where we primarily focused on answering **RQ2**. In this phase, triples containing object property information from the selected ontologies were added to the training dataset. As previously discussed, due to the simpler structure of the LinkedGeoData ontology—comprising only classes and data properties—no additional training data could be generated. However, for Schema.org, we were able to generate additional data to enrich the training phase of our pipeline.

Specifically, we imported new information from Schema.org concerning the classes of entities already present in the KG. This involved incorporating properties such as `<schema:hasMenu>` and associating them with existing classes like `<schema:FoodEstablishment>`. The goal was to enhance the discriminative information within the KG. For example, both `<schema:Restaurant>` and `<schema:Bakery>` are subclasses of `<schema:FoodEstablishment>`, but the `<schema:hasMenu>` property is characteristic only of the former, providing additional distinguishing information.

4.3.4. Parameters and Metrics

In both tests, we split the input data into training, test, and validation sets using an 8/1/1 ratio. The embedding dimensionality was set to 64, and we employed an early stopping strategy during training—meaning that training would halt if the loss did not decrease for 10 consecutive iterations. However, the total number of epochs was capped at 20.

For our analysis, we selected the evaluation metrics *Hits@10* and *Mean Reciprocal Rank (MRR)* for the link prediction task, both of which are commonly used in knowledge graph completion tasks. Hits@10 measures the percentage of cases where the correct entity appears within the top 10 ranked results. MRR, on the other hand, computes the average of the reciprocal ranks of the correct answers, assigning greater weight to higher-ranked predictions. Since link prediction is more effectively evaluated through ranking rather than pure classification, using these two metrics in conjunction provides a more comprehensive assessment.

Table 2

Results of Link Prediction Experiment

Model	Initialization	Ontology	Hits@10	MRR
TransE	Random	None	0.538	0.197
	Random	Schema.org	0.641	0.224
	Random	LinkedGeoData	0.463	0.168
	Space2Vec	None	0.612	0.219
	Space2Vec	Schema.org	0.694	0.250
	Space2Vec	LinkedGeoData	0.553	0.198
TransR	Random	None	0.648	0.213
	Random	Schema.org	0.624	0.266
	Random	LinkedGeoData	0.260	0.100
	Space2Vec	None	0.312	0.115
	Space2Vec	Schema.org	0.588	0.282
	Space2Vec	LinkedGeoData	0.324	0.095
RotatE	Random	None	0.808	0.498
	Random	Schema.org	0.781	0.497
	Random	LinkedGeoData	0.747	0.434
	Space2Vec	None	0.053	0.025
	Space2Vec	Schema.org	0.093	0.046
	Space2Vec	LinkedGeoData	0.039	0.020

4.4. Results and Discussion

The results of our experiment highlight that, given the current setup, the most effective approach for leveraging link prediction to infer new connections is through the adoption of *RotatE* [21] without embeddings and without incorporating knowledge injection or ontology-based information. This serves as evidence to start answering **RQ1**. However, when focusing on **RQ2**, we observe that using alternative models such as TransE and TransR demonstrates that the integration of embeddings and ontological data can influence performance outcomes.

The key takeaway from our findings is that, in our setting, link prediction is a viable strategy for identifying new relations with a reasonable level of accuracy, particularly when employing *RotatE*. Within the context of our methodology, these predicted links can be queried and potentially integrated into the knowledge graph to enhance its completeness and utility.

An open research question remains: can our data integration pipeline, when used in reverse, contribute to the construction of more robust knowledge graphs that, in turn, facilitate the development of more accurate predictive models. Further investigation is required to explore whether refining the KG through data integration could lead to improvements in link prediction performance.

5. Conclusion and Perspectives

In this discussion paper we provide some initial ideas on how to integrate link prediction over geospatial knowledge graphs into a knowledge graph construction pipeline. We demonstrate that it is feasible to add additional links to an existing knowledge graph to make it denser. Furthermore, we review the importance of the choice of the ontology and its transformation into the input data can have on model performance for link prediction.

Future work needs to be dedicated to testing a greater variety of KGE models, better ways to integrate the geospatial embeddings in model evaluation, as well as the choice of data to use for the construction of the KG and its respective training dataset. Due to the use of KGs at the heart of the architecture, integrating new data sources and therefore adding more data to the training phase can be performed in a systematic fashion.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

Acknowledgments

This research has been supported by the German Research Foundation (DFG) and the Autonomous Province of Bolzano-Bozen through its joint project “*Dense and Deep Geographic Virtual Knowledge Graphs for Visual Analysis - D2G2*” (grant number 500249124), by the HEU project CyclOps (grant agreement 101135513), by the Province of Bolzano and FWF through the project Ontegra (DOI 10.55776/PIN8884924), by the Province of Bolzano and EU through the project EFRE/FESR 1078 CRIMA, and by the Italian PRIN project S-PIC4CHU (grant agreement 2022XERWK9). This work has been carried out while Albulen Pano was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with Free University of Bozen-Bolzano.

References

- [1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*, 2 ed., Cambridge University Press, 2007.
- [2] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, M. Dean, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, W3C Member Submission, World Wide Web Consortium, 2004. URL: <https://www.w3.org/Submission/SWRL/>.
- [3] M. Perry, J. Herring, GeoSPARQL – A Geographic Query Language for RDF Data, OGC Implementation Standard OGC 11-052r4, Open Geospatial Consortium, 2012. URL: <http://www.opengeospatial.org/standards/geosparql>.
- [4] G. Mai, K. Janowicz, L. Cai, R. Zhu, B. Regalia, B. Yan, M. Shi, N. Lao, SE-KGE: A location-aware Knowledge Graph embedding model for geographic question answering and spatial semantic lifting, *Trans. in GIS* 24 (2020) 623–655. doi:10.1111/tgis.12629.
- [5] V. Martínez, F. Berzal, J.-C. Cubero, A survey of link prediction in complex networks, *ACM Computing Surveys* 49 (2016) 1–33.
- [6] Y. Liu, J. Ding, Y. Fu, Y. Li, UrbanKG: An urban knowledge graph system, *ACM Trans. on Intelligent Systems and Technology* 14 (2023) 60:1–60:25. doi:10.1145/3588577.
- [7] G. Mann, A. Dsouza, R. Yu, E. Demidova, Spatial link prediction with spatial and semantic embeddings, in: *Proc. of the 22nd Int. Semantic Web Conf. (ISWC)*, volume 14265, Springer, 2023, pp. 179–196. doi:10.1007/978-3-031-47240-4_10.
- [8] E. van Rees, Open Geospatial Consortium (ogc), *Geoinformatics* 16 (2013) 28.
- [9] S. Das, S. Sundara, R. Cyganiak, R2RML: RDB to RDF Mapping Language, W3C Recommendation, World Wide Web Consortium, 2012. URL: <http://www.w3.org/TR/r2rml/>.
- [10] C. Stadler, J. Lehmann, K. Höffner, S. Auer, LinkedGeoData: A core for a web of spatial open data, *Semantic Web* 3 (2012) 333–354.
- [11] L. Ding, G. Xiao, A. Pano, M. Fumagalli, D. Chen, Y. Feng, D. Calvanese, H. Fan, L. Meng, Integrating 3D city data through knowledge graphs, *Geo-spatial Information Science* (2024) 1–20. doi:10.1080/10095020.2024.2337360.
- [12] S. A. White, *Introduction to BPMN*, IBM Cooperation 2 (2004).
- [13] G. Schreiber, Y. Raimond, *RDF 1.1 Primer*, W3C Working Group Note, World Wide Web Consortium, 2014. URL: <http://www.w3.org/TR/rdf11-primer/>.
- [14] G. Xiao, L. Ding, B. Cogrel, D. Calvanese, *Virtual Knowledge Graphs: An overview of systems and use cases*, *Data Intelligence* 1 (2019) 201–223. doi:10.1162/dint_a_00011.
- [15] G. Xiao, D. Lanti, R. Kontchakov, S. Komla-Ebri, E. Güzel-Kalayci, L. Ding, J. Corman, B. Cogrel, D. Calvanese, E. Botoeva, The virtual knowledge graph system Ontop, in: *Proc. of the 19th Int. Semantic Web Conf. (ISWC)*, volume 12507 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 259–277. doi:10.1007/978-3-030-62466-8_17.
- [16] Y. Liu, J. Ding, Y. Fu, Y. Li, UrbanKG: An urban knowledge graph system, *ACM Trans. Intell. Syst. Technol.* 14 (2023). doi:10.1145/3588577.
- [17] S. Harris, A. Seaborne, SPARQL 1.1 Query Language, W3C Recommendation, World Wide Web Consortium, 2013. URL: <http://www.w3.org/TR/sparql11-query>.
- [18] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, *OWL 2 Web Ontology Language Profiles (Second Edition)*, W3C Recommendation, World Wide Web Consortium, 2012. URL: <http://www.w3.org/TR/owl2-profiles/>.
- [19] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: *Proc. of the 29th AAAI Conf. on Artificial Intelligence (AAAI)*, AAAI Press, 2015, pp. 2181–2187. doi:10.1609/AAAI.V29I1.9491.

- [20] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proc. of the 27th Annual Conf. on Neural Information Processing Systems (NIPS 2013), 2013, pp. 2787–2795. URL: <https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html>.
- [21] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, RotatE: Knowledge graph embedding by relational rotation in complex space, in: 7th Int. Conf. on Learning Representations (ICLR 2019), OpenReview.net, 2019. URL: <https://openreview.net/forum?id=HkgEQnRqYQ>.