TerraLink: Bridging Domains for Spatial Understanding

Federico Larrieu
federico.larrieu@colostate.edu
Department of Computer Science
Fort Collins, CO, USA

Eric Burton Samuel Martin
eric.burton.martin@colostate.edu
Department of Computer Science
Fort Collins, CO, USA

ABSTRACT

The TerraLink framework is introduced as the beginning steps towards a solution for enhancing spatial understanding through the integration of heterogeneous geospatial datasets into a unified knowledge graph enabling intuitive, human-language-based querying. Key contributions of this work include the development of a Spatial Resolution Mapping technique to help optimize data integration processes and reduce computational inefficiencies, allowing for a more customizable graph construction approach. Our knowledge graph demonstrates efficient performance in query processing, allows for the creation of custom motifs. Our framework aims to allow advancements in environmental monitoring and urban planning by enabling detailed analyses of spatial interactions that affect water quality and other ecological factors. Future directions for TerraLink involve adding more diverse datasets and improving upon its query functionalities to include novel spatial relationship types, thus broadening the scope of intuitive queries that reflect complex human thought processes.

CCS CONCEPTS

• **Information systems** → **Graph-based database models**; *Entity resolution*; *Document representation*; *Information retrieval query processing*; *Specialized information retrieval*; • **Computing methodologies** → *Distributed computing methodologies.*

KEYWORDS

Geospatial Analytics, Knowledge Graphs, Spatial Relationships, Distributed Storage, Distributed Processing, Spatial Data Integration, Graph-Based Querying, Spatial Statistics, Geographic Information Systems

ACM Reference Format:
Federico Larrieu and Eric Burton Samuel Martin. 2024. TerraLink: Bridging Domains for Spatial Understanding. In *Distributed Systems, January 18 – May 9, 2024, Fort Collins, CO, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Geospatial analytics allows us to observe natural phenomena or spatial interactions across domains. Whether it's planning urban infrastructure or understanding the impacts of industrial activities on water quality, geospatial analytics plays a large role in many decision-making processes. These analytics though, often require compute-intensive operations over large voluminous datasets that are often heterogeneous and require extensive costs, time, and domain expertise in order to create data pipelines to synthesize into a cohesive form.

Well, imagine a world where we can understand and interact with the space around us not just through maps, tables, and complex queries but through intuitive spatial relationships to help us identify new associations between spatially related data. That's the motivation behind our project. In this paper, we propose a system called TerraLink that leverages knowledge graphs in order to query data using intuitive spatial relationships. The system simplifies the integration and querying of diverse, heterogeneous geospatial datasets through the use of knowledge graphs. The knowledge graph that is produced provides spatial relationship mappings to records across a plethora of heterogeneous datasets, an example can be seen in Figure 1. By encoding these spatial relationships between multi-domain datasets into a knowledge graph and leveraging distributed processing frameworks and storage, we provide a platform that supports scalable spatial analytics and data retrieval while offloading expensive spatial computations into the graph structure.

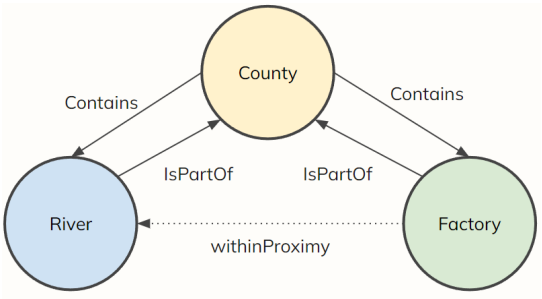

Figure 1: Example of our knowledge graph with spatial relationship semantics.

In this paper, we focus our attention on creating a way to help answer the following research question. How can we use spatial queries to help identify and analyze factors surrounding bodies of water that could potentially correlate with negative impacts on water quality? Numerous historical events illustrate the impact of nearby agricultural and industrial processes on water pollution, including Chesapeake Bay Pollution [1, 2], the ongoing issues in the Mid-Atlantic Region [3], Hudson River PCBs [4], the Mississippi River and Gulf of Mexico Hypoxic Zone (20th Century to Present)[5], Central California Agricultural Pollution (20th Century to Present)[6], and Salton Sea Pollution in California (1950s to Present) [7], among others.

CS555 '24, January 18 – May 9, 2024, Fort Collins, CO, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Understanding the spatial interactions at play in water quality is important as it allows us to observe environmental interactions and inform policy. Geospatial analytics provide a backbone on which policies can be constructed. If we understand the negative factors, then we can propose ways to mitigate further destruction of our water bodies. This paper will characterize the problem, look at current approaches to the problem, describe our approach, report our results, provide our analysis and finally conclude with our insights.

2 PROBLEM CHARACTERIZATION

The specific challenge addressed in this paper is conducting geospatial analysis across datasets from various domains, each in different formats and resolutions. This heterogeneity makes it difficult to derive meaningful insights without extensive preprocessing and integration efforts. The current process for doing such an analysis requires linking records in datasets to shared spatial bounds. For instance, the United States Census Bureau [8] offers a variety of cartographic boundaries such as Counties Within Congressional Districts, Divisions, Metropolitan and Micropolitan Statistical Areas, Related Statistical Areas, Nations, Regions, States, Urban Areas, ZIP Code Tabulation Areas, Tracts, Counties, Block Groups, and Blocks, to which demographic and economic data are tied. Integrating this census data with environmental datasets, which may be formatted differently, requires aligning them within shared spatial bounds and including foreign identifiers with these spatial boundaries. Spatial boundaries themselves are defined by key geometry types such as Point, LineString, Polygon, MultiPoint, MultiLineString, and MultiPolygon.

The project uses theories from spatial statistics, GIS, and landscape ecology. Spatial statistics focuses on how nearby data points often share similar values, which they call spatial autocorrelation. GIS principles guide the storage, manipulation, and analysis of geospatial data, with specific applications such as coordinate referencing systems and spatial indexing. Landscape ecology examines the influence of spatial patterns on ecological processes. Together, these foundations provide techniques and ideology that support the importance of our proposed framework.

We leverage modern big data technologies such as the Hadoop File System (HDFS) [9], Apache Spark [10], GraphFrames [11], and Apache Sedona [12] to address these challenges. HDFS manages the storage of large datasets, while Apache Spark supports distributed data processing. Apache Sedona provides comprehensive compute-intensive capabilities in handling spatial queries and transformations, and GraphFrames allows for advanced graph-based analytics. Together, these provide us with the tools necessary to enable the construction of a spatial knowledge graph using fundamental boundaries to map records across datasets from different domains through the integration of datasets with varying geometry types at different resolutions, while also supporting a wide array of SQL queries, easy motif creation, graph analytics, and graph representation learning projects, and potentially reducing search space. The resulting spatial knowledge graph is dynamic in the sense that datasets can be continuously integrated.

3 DOMINANT APPROACHES

The paper *"Extending the YAGO2 Knowledge Graph with Precise Geospatial Knowledge"* by Karalis, Nikolaos et al. [13] took multiple geospatial datasets and crafted a knowledge graph relationships relating to the geometry, population, name, and some spatial relationships. The work uses primarily already existing semantic web knowledge graphs though. With that said, this paper is producing work close to our end goals by allowing users to propse queries such as *"what is the city of Germany where two streams meet at a lake"*, or *"which are the neighboring municipalities of the municipality of Athens?"*. [13] We want to allow the same intuitive questioning but with the ability to query the impact of natural resources and ecosystems through the integration of the vast datsets found in the Urban Sustain Project. [14]

The paper, *"A Heterogeneous Geospatial Data Retrieval Method Using Knowledge Graph"* by Liu J, Liu H, et al. proposes a knowledge graph construction method to integrate heterogeneous geospatial data similar to our goal. [15] One of the main differences is that the knowledge graph is constructed from the bottom-up. Our proposed knowledge graph is built from the top down, like a tree structure. Although, we still support drill up or down queries. The method also retrieves entities belonging to related concepts. Another main difference is that they build nodes and relationships using general name semantics. We use spatial geometry and spatial queries.

We utilized the work by Alishiba Dsouza, Nicolas Tempelmeier, et al. [16] as a reference to help develop our methodology for our ontology and knowledge graph creation by following a process similar to Figure 2. The paper was an excellent resource but was creating a knowledge graph from a single data source which did not address our multi-domain integration goals.

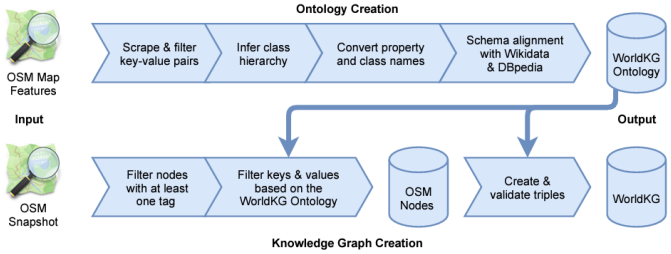

Figure 2: Example ontology and knowledge graph creation process [16]

4 METHODOLOGY

Engineering TerraLink to link voluminous geospatial datasets across domains generally required data collection, data preprocessing, big data frameworks, construction of a base knowledge graph, knowledge graph enrichment, SQL queries, and finally, analysis. In data collection, we decided to select datasets from different domains with different properties, standards, and geometry types. This is an important aspect as geospatial data does not come with the same purpose, information, or shape. Primarily, we only chose vector-based geometries, but adding modules for raster data would be possible. Adding datasets from different domains is important for capturing inter-domain interactions.

In choosing the datasets, there were two fundamental questions: What datasets would we use to build our spatial hierarchy? What datasets would be of interest, say, in water quality spatial interactions? This framework isn't specific to water quality problems but rather provides a generalized solution for spatial interactions. Many different questions could be answered. To satisfy the first question, we decided to use the hierarchical boundaries seen in Table 1 for the base spatial graph.

Dataset	Records	Size
Continents	7	3.2 KB
Countries	195	89.14 KB
Census States	102	38.77 MB
Census Counties	3221	103.97 MB
Census Tracts	85058	115.15 MB
Census Block Groups	242298	291.20
Total	330881	641.23 MB

Table 1: Summary of the hierarchical boundaries for the base spatial graph.

The base graph is not limited to our chosen hierarchy. As long as there is some spatial boundary, you can define your own hierarchy. To satisfy the second question, we selected to enrich the base dataset with the datasets seen in Table 2.

Dataset	Records	Size
Agricultural Areas	N/A	61.24 MB
Electrical Substations	68991	80.79 MB
Dams	14603	51.65 MB
General Manufacturing Facilities	324358	472.19 MB
Hydro-Carbon Gas Liquid Pipelines	95	188.71 KB
Natural Gas Compressor Stations	1768	3.18 MB
Natural Gas Pipelines	33806	34.62 MB
Natural Gas Processing Plants	840	1.8 MB
Natural Gas Storage Facilities	486	932.94 KB
Oil Refineries	155	328.02 MB
Power Plants	4514	15.61 MB
Total	449616	998.57 MB

Table 2: Summary of the hierarchical boundaries for the base spatial graph.

Not a lot of preprocessing was necessary due to the fact that most datasets followed the GeoJSON standard [17]. The only real need for data preprocessing was in transforming shapefiles to .geojson files and ensuring the geometries followed the EPSG:4326 coordinate referencing system standard. Storing voluminous datasets can be a large undertaking, especially when geospatial datasets can reach an extreme scale. To prepare for this, we decided to use HDFS as our storage system. Processing large-scale data also requires large-scale processing frameworks like Apache Spark. This framework allowed us to distribute computing across 10 computers. The spark context was configured with 10 workers, a 4 GB limitation per executor, 1025 MB memory overhead per executor, and enabled memory off-heap with a 500 MB limit. We increased the broadcast timeout to 1 hour and set our shuffle partitions to 600. We also leveraged Apache Sedona for graph construction and enrichment. This framework is used due to its powerful ability to extend Apache Spark for handling large-scale spatial data. It provides advanced spatial operations and queries, such as spatial joins and range queries, which are essential for efficiently processing and analyzing geospatial data distributed across different datasets. In our case, we used it to build the relationships (edges) between records. For instance, we could describe if a record was part of a block group by simply using the following SQL query:

```
SELECT * FROM parents, children WHERE ST_Contains(
parent_geometry, ST_Centroid(child_geometry));
```

This spatial SQL query was primarily used for building the base knowledge graph by using the centroid of the child boundary and for integrating point geometry dataset into the knowledge graph. In terms of handling other geometry such as LineString and Polygon variations, the conditions are as follows:

```
WHEN ST_Contains(p.parent_geometry, ST_Centroid(
c.child_geometry)) THEN 'Contains' OR WHEN ST_Intersects(
p.parent_geometry, c.child_geometry) THEN 'Intersects';
```

GraphFrames facilitates sophisticated graph analysis, enabling motif finding, subgraph algorithms, page rank, and edge degree analysis on large graphs. This is particularly useful for constructing and querying the proposed spatial knowledge graph, allowing for detailed exploration of relationships and patterns within the integrated geospatial data.

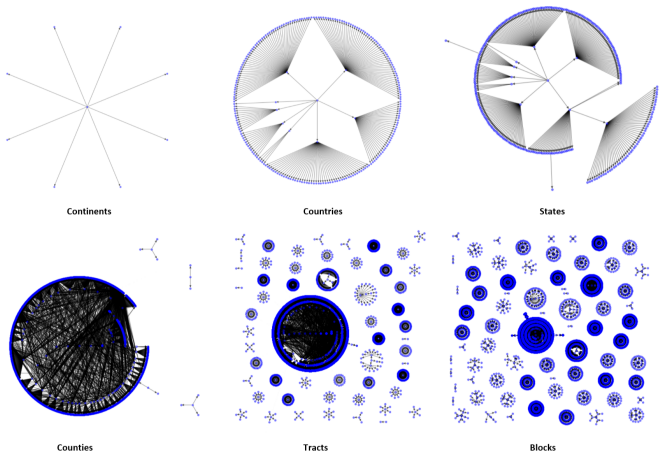

Figure 3: Visualization of each step towards building the complete base knowledge graph.

Constructing the base knowledge graph is the foundational component of this framework and the stages of the creation can be seen in Figure 3 It is what allows us to describe the spatial bounds and map data to those specified bounds. As mentioned above, we primarily used boundaries that describe a hierarchical structure of Earth. We started by introducing a root node, Planet Earth. We felt it was important to describe a root node to mimic a tree structure but

this isn't necessary. We then introduced geometry and properties for continents. These continents have an isPartOf relationship with the Earth node. We also included a Contains relationship with the Earth node to the continents. This allows us to have a bidirectional graph, which would allow drill down and up queries. Next, we introduced the next hierarchical relationship, Countries. To create these relationships programmatically, we would run a spatial query across the Continents and Countries datasets. The spatial query would define the centroid of the child dataset, in this case, Countries, and a query for Continent geometry that contained that centroid. This defines both the isPartOf and Contains relationships. The same process was applied from Countries to States, States to Counties, Counties to Tracts, and Tracts to block groups. The node id was explicitly specified depending on the properties that were available in each of the datasets. In some cases, it was a unique id, in other cases, it was the name. At the end of this process, we have the base knowledge graph in which we can start to integrate and map records from datasets.

Integrating datasets is the next quintessential aspect of this process. Leveraging the knowledge graph, we can spatially map records based on a defined spatial hierarchy. To integrate a dataset, we would follow a similar process to what was used to build the base knowledge graph. A bidirectional relationship, isPartOf and Contains, was applied to records using a spatial query. The only difference here was how we defined if a record had these relationships. The base knowledge graphs the geometries are pretty uniform and would only need the centroid to apply a relationship to a parent geometry. The same approach wouldn't work for all geometry cases. For instance, we also integrated a dataset that contained LineString geometry. Using the centroid of a line wouldn't accurately describe the spatial relationships of the line. A line could span many Block Groups, Tracts, Counties, States, or even Countries. Here we made a distinction. If the geometry type of a dataset was primarily point, then the centroid approach would work. In the case the geometry type was anything other than point, we would introduce a different spatial query that defined if the geometry intersected or was within the parent geometry.

5 EXPERIMENTAL BENCHMARKS

In order to assess the TerraLink framework, we executed a series of benchmarks aimed at evaluating its performance.

5.1 Graph Construction and Enrichment

Figure 4 displays the number of vertices and edges within the base and enriched knowledge graphs. The base graph comprised a substantial count of vertices and edges, which exponentially increased upon enrichment, signaling successful integration of additional datasets. Specifically, the enriched graph exhibited a growth of over 1.5 million edges, denoting a denser and more interconnected data network suitable for complex analytics.

5.2 Relationship Type Distribution

Analysis of relationship types among nodes revealed that the enriched graph had a significantly higher proportion of many-to-many relationships compared to the base graph, which predominantly consisted of one-to-one and one-to-many types, see Figure 5. This

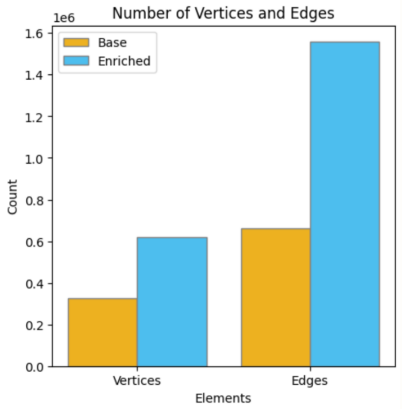

Figure 4: Edge and node (vertex) count in original base graph versus enriched graph.

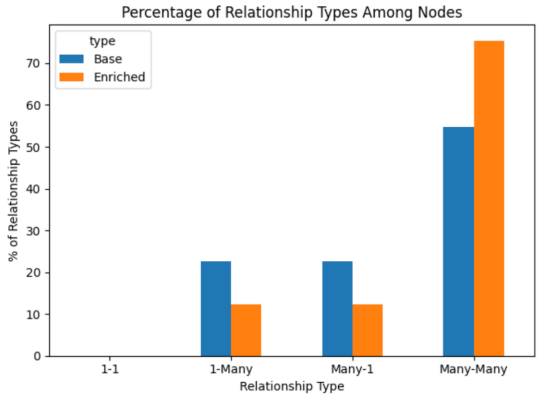

Figure 5: A comparison of the relationships between nodes within the base and enriched graph.

increase in many-to-many relationships shows the enriched graph's improved capability to model complex interactions between various geospatial entities.

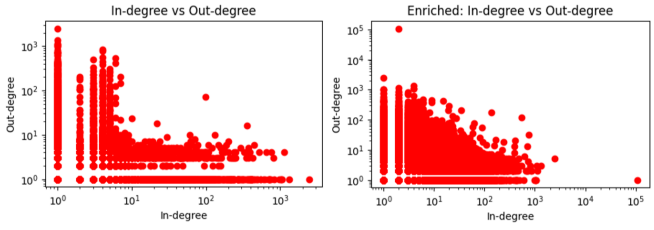

Figure 6: A comparison of the in vs out-degree of the base and enriched graphs.

5.3 Degree Distribution

Another method we used to assess the graph topology was to look into the in-degree versus out-degree distribution. The enriched

graph displayed a wider and more varied degree distribution as seen in Figure 6 indicating a more intricate structure in comparison to the base graph which is to be expected. Such a distribution is beneficial for graph traversals and allows for a more in-depth discovery of spatial correlations.

5.4 PageRank Benchmarking

Utilizing PageRank, an algorithm for measuring the importance of graph vertices, helped us identify an issue within our knowledge graph. The PageRank scores highlighted the most significant nodes, such as County_Los Angeles but it showed a large number of repeating counties such as County_Middlesex which led us to realize that we were seeing the affects of namespace collision since our counties don't have unique identifiers. This metric was supposed to highlight nodes with the highest potential of affecting or being affected by spatial phenomena within the graph context but rather helped us unveil a fixable issue. More detail on this is covered in challenges.

5.5 Query Performance

To test the querying potential of our graph we executed various SQL queries through GraphFrames, see Figure 7.

```
g.edges.filter("src == 'State_Colorado'")
g.edges.filter("src LIKE 'County_J%' AND Relationship == 'isPartOf'")
g.edges.filter("dst LIKE 'County_Wa%'")
g.edges.filter("src == 'State_Colorado' OR dst == 'State_Colorado'")
g.vertices.filter("Type == 'State'").show(100)
g.vertices.createOrReplaceTempView("vertices")
spark.sql("SELECT * FROM vertices WHERE id = 'County_Mineral'")
```

Figure 7: An example of various SQL filter queries performed on the knowledge graph.

5.6 Motif Discovery

Motif discovery tests were conducted to evaluate how we can identify recurring patterns within our knowledge graph which can be used to reveal the geospatial interconnections within the network.

```
motifs = g.find("(a)-[ab]->(b)")

# a is the source vertex,
# b is the destination vertex,
# ab is the edge between them
filtered_motifs = motifs.filter(
    (motifs['a']['Type'] == 'County') &
    (motifs['ab']['Relationship'] == 'isPartOf') &
    (motifs['b']['id'] == 'State_Colorado')
)
```

Figure 8: An example of creating a motif (sub-graph) from our knowledge graph.

5.7 Integration Execution Times

Since the knowledge graph needs to be easily updated, we analyzed the amount of time it took to integrate the various datasets which can be seen in Figure 9 The results showed that datasets with complex and abundant geometries that pass through many census blocks and tracts, such as 'Natural Gas Pipelines', took considerably longer to integrate, as opposed to less complex datasets like 'Oil Refineries'. This realization led us to create a method to integrate datasets according to a specified spatial hierarchy resolution which we discuss next.

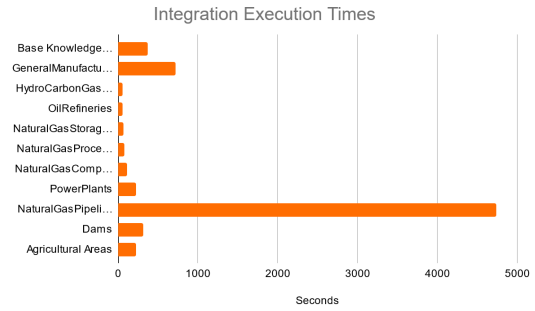

Figure 9: Execution time required to integrate datasets into the knowledge graph.

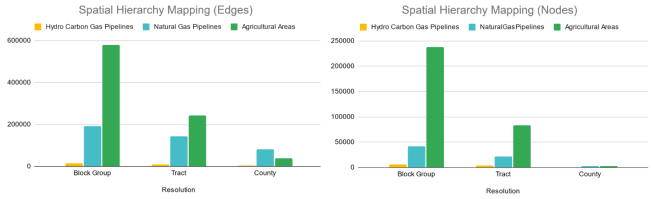

Figure 10: Plot showing the drastic reduction in the number of edges and node connections being created when controlling the integration resolution level.

5.8 Spatial Hierarchy Mapping

After obtaining the integration metrics we identified that datasets that contain large line data points like the 'Natural Gas Pipelines', pass through or within proximity of a very large number of the finest granularity spatial nodes (census blocks and tracts). This leads to an extremely large number of new edges and relationships between nodes and slows down computation. In order to address this, we developed a method of spatial hierarchy-based integration which would integrate a dataset to the desired hierarchical level such as Block Group, Tract, County, State, etc. The drastic reduction in the number of edges and node relationships can be seen in Figure 10.

6 CHALLENGES

In implementing this framework we came across several obstacles and thought about a possible threat to validity. In our approach, we planned to use the natural language properties as identifiers. There are some possible cases of namespace collision when records

contain the same ID. For instance, counties in different states contain the same name. Page rank indicated some namespace space collision because Walker County filled 15 entries. Walker County is in three different states; Alabama, Georgia, and Texas. Mitigating this would be trivial, as you could leverage hash functions for generating a unique identifier. There are examples of this being done with both GeoHash and H3. However, the only downside is the loss of natural language interpretation and queries. An intermediate solution would be appending the unique identifier to the node ID to ensure uniqueness but maintain interpretation.

7 INSIGHTS

In the initial phase of this project we aimed to introduce spatial information into data lakes. This was inspired by methodologies used in ="*Dataset Discovery in Data Lakes*" [18]. Their implementation of attribute relatedness was a useful approach to defining relationships between columns across datasets. We asked ourselves, "How can we facilitate spatial relationships across voluminous data?". It was once we began delving into the details of how to implement these ideas that we recognized we had to reduce our scope. . Our initial idea had the additional concept of being able to identify similar columns within datasets in a data lake using TURL [19], a framework that contextualizes representations on relational tables in an unsupervised manner through transformer embeddings in order to find conceptual relationships between columns and create new datasets. Although useful, it would be limited in defining more fine-grained relationships that exist within the data.

Getting spatial relationship across large voluminous geospatial datasets was initially a daunting problem. This is due to the exploding search space that would be necessary to accomplish this task. Regardless we decided to move forward with a reduced scope. Our new idea was to leverage a pre-computed knowledge graph that could cache spatial relationships. This graph would be queried instead of need to continuously run spatial queries. You could also run aggregation queries on children data nodes that share a common hierarchy node. This graph could also offer search space reductions for spatial relationships that are not yet covered by our knowledge graph. For instance, you could run a spatial query on a subset of data across datasets. This initial search space prune would provide the foreign identifiers of records across datasets. The identifiers could then be queried from the specified identification column in the original dataset tables. The results could also be cached in the knowledge graph.

A major discovery we were very proud of occurred when we ran the integration metrics and discovered that the Natural Gas Pipeline dataset took orders of magnitude longer to integrate than the other datasets. We had to contemplate the reasoning behind this and recognized that since the geometries in the Natural Gas Pipeline dataset were lines, they could cross through the spatial containers of many nodes, especially the census data due to the fine granularity of the tract and block zones. Seeing Figure 11 can put this into perspective.

So, in order to address this we came up with the idea of integrating datasets into the knowledge graph according to a specific level of granularity, a technique we call Spatial Hierarchy Mapping, as seen in section Spatial Hierarchy Mapping. This allowed us to integrate any dataset with granularity parameters that ensure no relational edges will be created below the granularity level. We were proud of this.

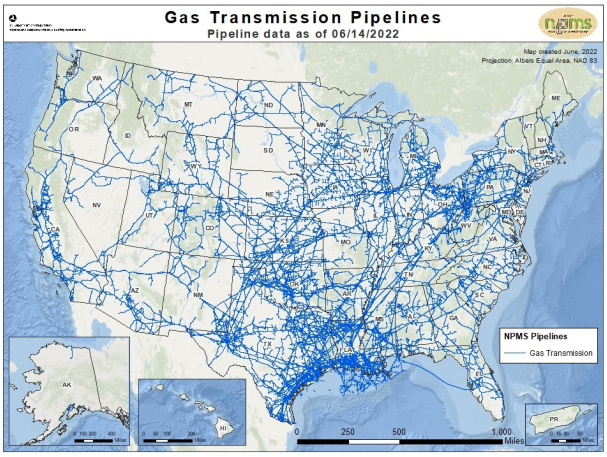

Figure 11: Visualization of the Natural gas transmission pipelines in the United States of America. [20]

8 FUTURE PROSPECTS

As we continue into the age of data, knowledge graphs will become more important than ever. Having studied the topic and listened to various podcasts covering knowledge graphs, we have been indoctrinated into the belief that knowledge graphs are the next big step in the coming future. The fact that they combine characteristics of databases, graphs, and knowledge bases into a dynamic topology that can answer complex queries across heterogeneous datasets is a game changer. But the most beautiful thing about them is that they attempt to mirror the web of connections inherent to human knowledge which makes them perfectly suited for the needs of the growing field of explainable AI since they can provide a framework that makes the 'why' behind AI decisions more transparent.

Since knowledge graphs consist of semantic relationships between various ontologies, they often require a significant amount of domain expertise and manual effort to create these relationships. Automated methods for relationship extraction and ontology building are in development but they are still naïve and struggle with the nuances and complexities of real-world knowledge. The maintenance and updating of knowledge graphs also present challenges, as they must evolve with the changing landscape of information and context they represent.

With all of these challenges though, comes the rise of the 'Knowledge Scientist' as a profession. New careers for people with expertise knowledge graphs are popping up and with the proliferation of data, the ever-growing need to extract information from that data, and the need to explain reasoning for choices will only increase the need for people with the skills to work with KG's. In the context of our project, we believe the evolution of knowledge

graphs promises to revolutionize how we interact with and understand spatial data. By enabling queries that intuitively reflect human thought processes—such as looking for "factories within 10 miles north of Horsetooth Reservoir" we can begin to unveil and build new relationships in spatial data.

9 CONCLUSION

Our research presents TerraLink, a framework designed to enhance spatial understanding through the integration of heterogeneous geospatial datasets into a single comprehensive knowledge graph. This project is the beginning steps towards addressing challenges in the domain of geospatial analytics, such as the integration of heterogeneous and diverse geospatial datasets and using intuitive human language to query the data.

Spatial Data Integration: By leveraging Apache Spark, Apache Sedona, GraphFrames, and HDFS we were able to successfully integrate 17 datasets into a single knowledge graph. Currently, we have added two relationships 'isPartOf' and 'contains' but by using Apache Sedona we are poised to add over 60 relationships through the Sedona toolkit.

Addressing Computational Challenges: We introduced a technique called Spatial Resolution Mapping to reduce computational inefficiencies when integrating and make the resulting query space more readable. Our approach reduced the complexity of integration and increased the customization of building the graph by allowing granularity-based integrations.

Enhancements in Query Performance and Data Enrichment: The enriched knowledge graph demonstrated a excellent query performance and allowed for custom motifs to be generated allowing users to utilize a subgraph and query that for even faster query times.

Practical Implications and Use Cases: The practical applications of TerraLink, if completed, could be used to address our original question "How can we use spatial queries to help identify and analyze factors surrounding bodies of water that could potentially correlate with negative impacts on water quality?" By providing a user-friendly and intuitive framework for querying and analyzing spatial relationships which can be leveraged to help policymakers and researchers make more informed decisions.

Future Directions: Moving forward, we aim to expand TerraLink's capabilities to include more diverse datasets and develop more advanced query functionalities. We aim to add the more novel idea of cardinal and topological relationships to the graph so users can perform queries such as "factories at a higher elevation and within 10 miles north of Horsetooth Reservoir" and find answers to our original question in the introduction.

ACKNOWLEDGMENTS

Thanks to Dr. Shrideep Pallickara at Colorado State University for the amazing class and also to the GTA's who only added to the enjoyment. We would also like to thank the Urban Sustain Project for supplying such vast datasets which were crucial to the success of this project.

REFERENCES

[1] Q. Zhang, P. J. Tango, R. R. Murphy, M. K. Forsyth, R. Tian, J. Keisman, and E. M. Trentacoste, "Chesapeake bay dissolved oxygen criterion attainment deficit: Three decades of temporal and spatial patterns," *Frontiers in Marine Science*, vol. 5, 2018.

[2] D. Szepeso, J. Malarkey, and L. Harmon, *Chesapeake Bay Program Technical Studies: A Synthesis*. Washington, D.C, United States of America: United States Environmental Protection Agency, 1982. Available from The EPA Agency.

[3] A. C. Lewis, M. J. Evans, J. Methven, N. Watson, J. D. Lee, J. R. Hopkins, R. M. Purvis, S. R. Arnold, J. B. McQuaid, L. K. Whalley, M. J. Pilling, D. E. Heard, P. S. Monks, A. E. Parker, C. E. Reeves, D. E. Oram, G. Mills, B. J. Bandy, D. Stewart, H. Coe, P. Williams, and J. Crosier, "Chemical composition observed over the mid-atlantic and the detection of pollution signatures far from source regions," *Journal of Geophysical Research: Atmospheres*, vol. 112, no. D10, 2007.

[4] H. Feng, J. Kirk Cochran, H. Lwiza, B. J. Brownawell, and D. J. Hirschberg, "Distribution of heavy metal and pcb contaminants in the sediments of an urban estuary: The hudson river," *Marine Environmental Research*, vol. 45, no. 1, pp. 69–88, 1998.

[5] N. N. Rabalais and R. E. Turner, "Gulf of mexico hypoxia: Past, present, and future," *Limnology and Oceanography Bulletin*, vol. 28, no. 4, pp. 117–124, 2019.

[6] M. Almaraz, E. Bai, C. Wang, J. Trousdell, S. Conley, I. Faloona, and B. Z. Houlton, "Agriculture is a major source of no x pollution in california," *Science advances*, vol. 4, no. 1, p. eaao3477, 2018.

[7] B. A. Jones and J. Fleck, "Shrinking lakes, air pollution, and human health: Evidence from california's salton sea," *Science of the Total Environment*, vol. 712, p. 136490, 2020.

[8] "Cartographic boundary files - shapefile." https://www.census.gov/geographies/mapping-files/time-series/geo/carto-boundary-file.html, 2018. Last accessed 28-APR-2024.

[9] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, pp. 1–10, 2010.

[10] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache spark: a unified engine for big data processing," *Commun. ACM*, vol. 59, p. 56–65, oct 2016.

[11] A. Dave, A. Jindal, L. E. Li, R. Xin, J. Gonzalez, and M. Zaharia, "Graphframes: an integrated api for mixing graph and relational queries," in *Proceedings of the fourth international workshop on graph data management experiences and systems*, pp. 1–8, 2016.

[12] J. Yu, Z. Zhang, and M. Sarwat, "Spatial data management in apache spark: the geospark perspective and beyond," *GeoInformatica*, vol. 23, pp. 37–78, 2019.

[13] N. Karalis, G. Mandilaras, and M. Koubarakis, "Extending the yago2 knowledge graph with precise geospatial knowledge," in *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*, pp. 181–197, Springer, 2019.

[14] "Urban sustain project." https://urban-sustain.org/, 2024. Last accessed 28-APR-2024.

[15] J. Liu, H. Liu, X. Chen, X. Guo, Q. Zhao, J. Li, L. Kang, and J. Liu, "A heterogeneous geospatial data retrieval method using knowledge graph," *Sustainability*, vol. 13, no. 4, p. 2005, 2021.

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

[17] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and S. Hagen, "The GeoJSON Format." RFC 7946, Aug. 2016.

[18] A. Bogatu, A. A. Fernandes, N. W. Paton, and N. Konstantinou, "Dataset discovery in data lakes," in *2020 ieee 36th international conference on data engineering (icde)*, pp. 709–720, IEEE, 2020.

[19] X. Deng, H. Sun, A. Lees, Y. Wu, and C. Yu, "Turl: Table understanding through representation learning," 2020.

[20] "Where are gas pipelines located?." https://pipeline101.org/topic/where-are-gas-pipelines-located/, 2022. Last accessed 28-APR-2024.