UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Semantic Data Management

Graph-Based Metadata Management

DigiScan360

Authors

Narmina Mahmudova
narmina.mahmudova@estudiantat.upc.edu

MD Kamrul Islam
md.kamrul.islam@estudiantat.upc.edu

Supervisors

Oscar Romero
Anna Queralt

June 11, 2024

Table of Contents

1 Introduction

In the rapidly evolving landscape of electronic manufacturing and retail, staying ahead of the competition requires real-time insights and strategic intelligence. Traditional methods of data collection and competitive analysis often involve lengthy processes and substantial resources, making them inadequate for the dynamic needs of modern businesses. DigiScan360 addresses this challenge by offering a cutting-edge solution designed to provide a comprehensive, 360-degree view of the competitive landscape, delivering actionable recommendations and insights with unprecedented speed and accuracy.
This project aims to enhance DigiScan360's competitive intelligence capabilities by developing an advanced data integration framework using GraphDB and knowledge graphs. By integrating data from multiple local sources into a unified global schema, we can provide a holistic view of the market and competitive landscape. This approach not only improves the accuracy and speed of data processing but also ensures that our clients have access to the most relevant and up-to-date information. Through automated processes, we collect, transform, and integrate data, creating a robust and scalable system that supports comprehensive analytics and reporting. This initiative is a testament to our commitment to innovation and excellence, positioning DigiScan360 as a leader in competitive intelligence solutions for the electronic manufacturing and retail sectors.
Source code of the project: GitHub Repo Link

1.1 Significance

The significance of this project for DigiScan360 cannot be overstated, as it represents a pivotal advancement in our mission to revolutionize competitive advantage analysis. By developing an advanced data integration framework using GraphDB and knowledge graphs, we are transforming the way electronic manufacturers and retailers access and utilize market intelligence. This project enables us to deliver real-time, comprehensive insights that are both accurate and actionable, directly addressing the dynamic needs of our clients. The seamless integration of diverse data sources into a unified global schema enhances our analytical capabilities, allowing us to provide a deeper, more nuanced understanding of market dynamics, customer preferences, and competitor strategies. This not only strengthens our clients' strategic decision-making but also solidifies DigiScan360's position as a leader in competitive intelligence solutions. Ultimately, this project is a testament to our commitment to innovation, ensuring that our clients stay ahead of the curve in an ever-evolving industry landscape.

2 Project Purpose Statement (M1)

The purpose of incorporating graph-based solutions in DigiScan360 is to enhance our data governance framework, ensuring robust and ethical management of metadata. By utilizing graph-based metadata management, we can seamlessly integrate diverse data sources, offering a dynamic and interconnected view of data relationships. This approach not only strengthens data integrity and compliance with privacy regulations but also enhances transparency and traceability throughout the data lifecycle.
Graph-based metadata management enables us to map complex data ecosystems efficiently, facilitating better data discovery, lineage, and classification. This ensures that our competitive intelligence tool operates on a foundation of well-governed, high-quality data, ultimately leading to more reliable insights and strategic guidance for our clients. Emphasizing data governance through graph-based solutions underscores our commitment to ethical practices and sustainable data management, reinforcing trust and confidence among our users.

3 Choice of Graph Family (M2)

We chose **knowledge graphs** for DigiScan360 to enhance our data governance and management capabilities. Knowledge graphs offer a dynamic and interconnected way to organize and visualize complex data relationships, making it easier to integrate and manage diverse data sources. How Knowledge Graphs Help Our Project

1. **Improved Data Governance:** Knowledge graphs ensure robust data governance by providing clear data lineage, enhancing transparency, and ensuring compliance with privacy regulations.
2. **Enhanced Data Discovery:** They facilitate better data discovery and classification, enabling users to find and understand relevant data quickly.
3. **Efficient Data Integration:** By mapping complex data ecosystems, knowledge graphs enable seamless integration of various data sources, improving data quality and reliability.

4. **Better Insights:** This foundation of well-governed data leads to more accurate and actionable insights, empowering our clients to make informed strategic decisions.

4 Graph Schema (M3)

4.1 Global Schema Design

The global schema model depicted in the image represents a sophisticated network of interconnected entities central to our competitive advantage analysis tool. At the core of the schema are key entities such as Consumer, Tweet, Product, Review, and the equivalent classes Seller and Brand.
Each entity is linked through relationships that reflect real-world connections. For instance, the Consumer entity is connected to the Tweet entity through the tweets relationship, indicating that consumers post tweets. These tweets, in turn, can mention products or brands, as shown by the mentionsBrand relationship between the Tweet and Brand entities. The Product entity is enriched with detailed attributes such as product_name, product_price, product_description, and product_url, providing comprehensive information about each product. Additionally, products are reviewed, as indicated by the reviews relationship connecting the Product and Review entities. Reviews contain attributes like review_id, review_title, review_body, review_rating, and review_date, offering a thorough understanding of customer feedback.
The Seller and Brand entities, which are defined as equivalent classes, include attributes such as user_id, name, username, created_at, url, followers_count, friends_count, and verified, describing the brand's presence and influence. This entity is connected to the Product entity through the belongsTo relationship, highlighting which seller is associated with which products.

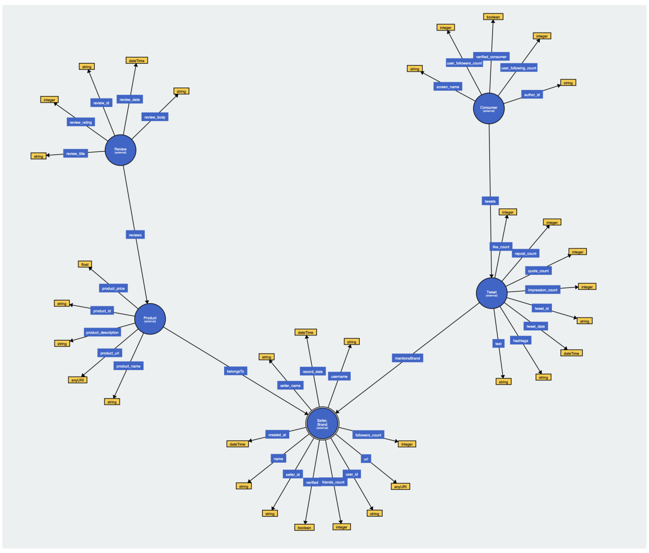


Figure 1: Global Schema

4.2 Local Schema Design

4.2.1 Brand Tweet

The brand tweet schema encompasses essential attributes and relationships for representing brands and their presence on social media platforms. It includes properties such as `name`, `username`, `followers_count`, and `verified`, which detail the brand's identity and credibility. Additionally, the schema defines the `mentionsBrand` relationship, indicating when a tweet references a specific brand. These components enable the structured representation of brand-related data within RDF graphs, facilitating comprehensive analysis and understanding of brand engagement and interaction on social media.

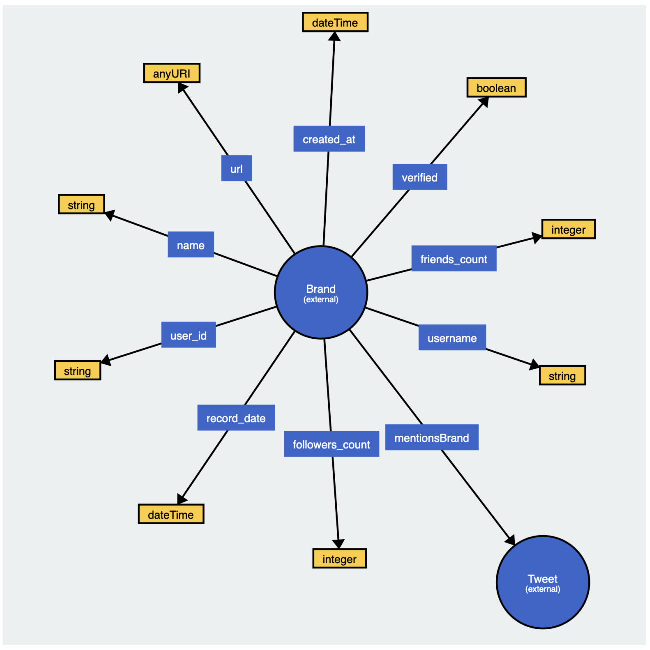


Figure 2: Local Brand Tweet Schema

4.2.2 Consumer Tweet

The consumer tweet schema outlines essential attributes and relationships for describing consumers and their tweets in RDF format. It includes properties like `tweet_id`, `timestamp`, and `text` to represent tweet characteristics, and `author_id` and `screen_name` for consumer profile information. Additionally, it defines the `tweets` relationship to connect consumers with their posted tweets. This schema enables structured representation and analysis of consumer behavior and engagement on social media platforms within RDF graphs.

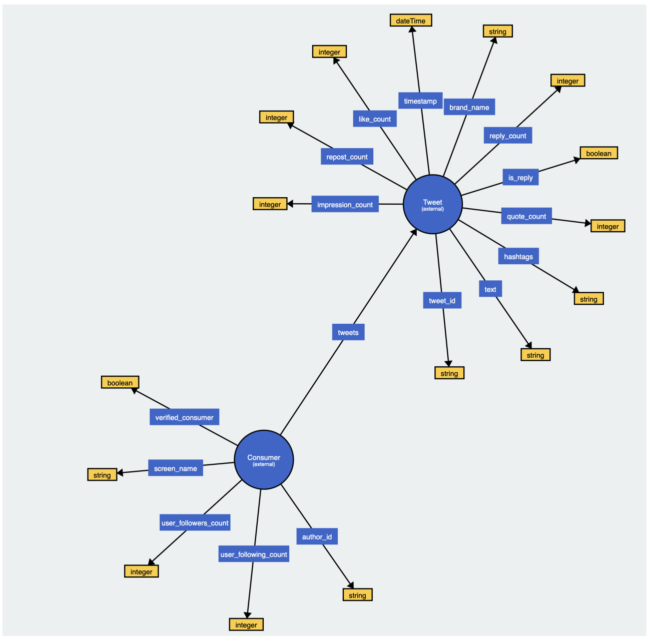


Figure 3: Local Consumer Tweet Schema

4.2.3 Products

The product schema defines essential attributes and relationships for representing products, brands, and sellers in RDF format. It includes properties such as `product_id`, `price`, `product_description`, and `rating` to capture product details and ratings. Additionally, properties like `brand_name` signify brand information associated with the product.

Relationships like belongsTo establish connections between products and sellers, facilitating the organization of product offerings. This schema enables structured representation and analysis of product-related data, supporting e-commerce applications and market analysis within RDF graphs.

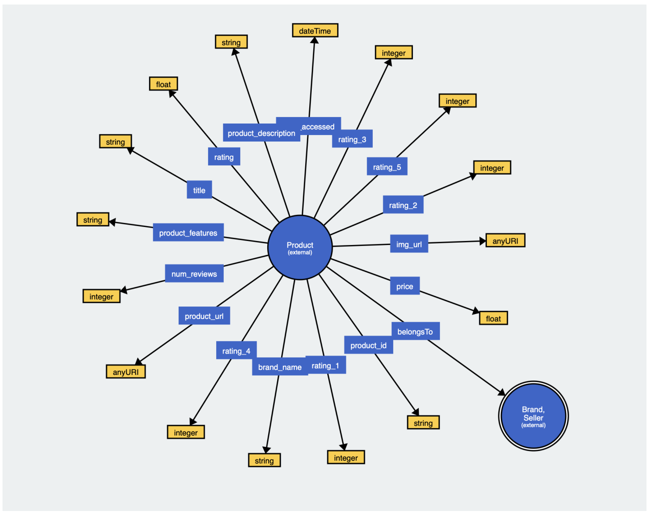


Figure 4: Local Product Schema

4.2.4 Reviews

The review schema outlines essential attributes and relationships for representing reviews and products. It includes properties such as review_id, title, and body to capture review details, along with num_helpful to quantify helpfulness. Additionally, properties like product_id denote the associated product for the review. The schema defines relationships like reviews to connect reviews with their respective products.

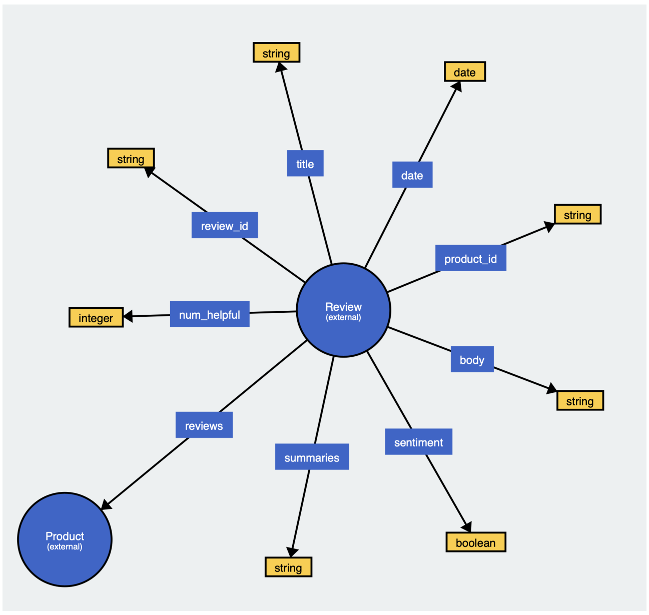


Figure 5: Local Reviews Schema

5 Data Integration and Mappings (M4 & M5)

The automation of data integration within DigiScan360 is a critical component designed to streamline and enhance the efficiency of competitive intelligence processes for electronic manufacturers and retailers. This section outlines the automated workflow, aligned with the project's primary objective to provide real-time, actionable insights. Figure 6 illustrates the BPMN model of our overall Graph-Based Metadata Management Process.

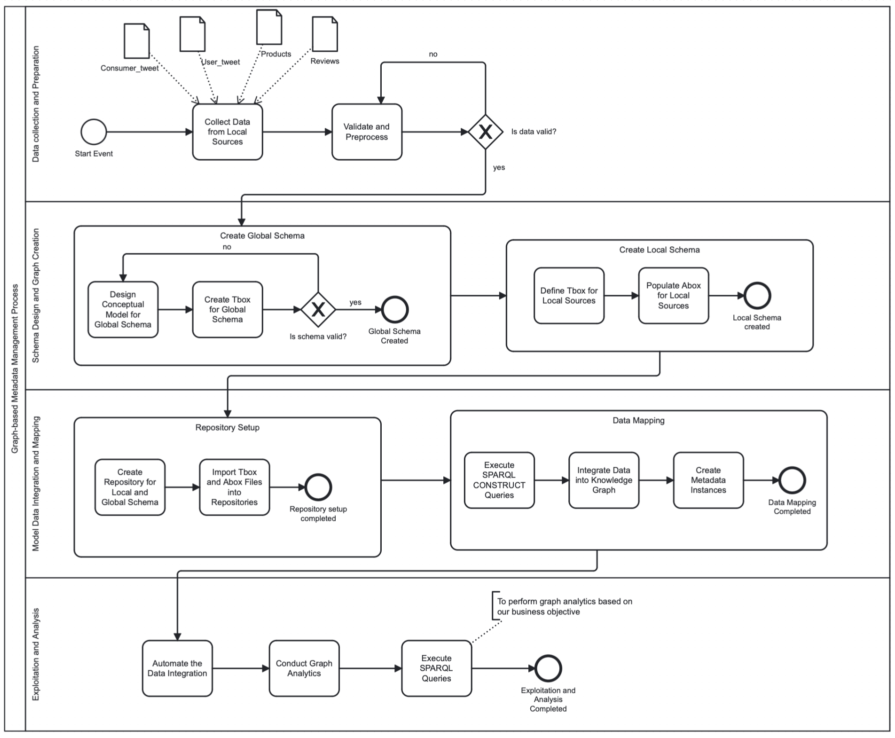

Graph-based Metadata Management Process
Data collection and Preparation
Consumer_tweet
User_tweet
Products
Reviews
Start Event
Collect Data from Local Sources
Validate and Preprocess
no
Is data valid?
yes
Schema Design and Graph Creation
Create Global Schema
no
Design Conceptual Model for Global Schema
Create Tbox for Global Schema
yes
Is schema valid?
Global Schema Created
Create Local Schema
Define Tbox for Local Sources
Populate Abox for Local Sources
Local Schema created
Model Data Integration and Mapping
Repository Setup
Create Repository for Local and Global Schema
Import Tbox and Abox Files into Repositories
Repository setup completed
Data Mapping
Execute SPARQL CONSTRUCT Queries
Integrate Data into Knowledge Graph
Create Metadata Instances
Data Mapping Completed
Exploitation and Analysis
To perform graph analytics based on our business objective
Automate the Data Integration
Conduct Graph Analytics
Execute SPARQL Queries
Exploitation and Analysis Completed

Figure 6: Graph-based Metadata Management Process

5.1 Data Sources

To effectively populate our knowledge graph, we selected several local data sources based on our business objective namely, Consumer Tweet Data, Product Data, Review Data, and User Tweet Data (Headphone Brands)

5.2 Key Techniques and Processes

1. **Data Collection and Validation**

   The process begins with the collection of data from various local sources, including Consumer Tweets, User Tweets, Products, and Reviews. This initial phase ensures that comprehensive data is gathered to form a robust foundation for further analysis. The collected data undergoes preprocessing to ensure accuracy and reliability for our graph model. Any data failing validation is flagged and sent back for re-evaluation, ensuring only high-quality data progresses through the workflow.

2. **Schema Design and Graph Creation**

   Once validated, the data is integrated into a structured schema. This involves creating a global schema that serves as a unified framework for all data sources. The global schema is meticulously designed and validated to ensure it accurately represents the conceptual model of the data. Concurrently, local schemas are developed for each data source, defining the specific attributes and relationships within each dataset. This dual approach ensures both local specificity and global coherence.

3. **Model Data Integration and Mapping** With the schemas in place, the next phase involves setting up repositories for both local and global schemas. The Terminology Box (TBox) and Assertion Box (ABox) files, which contain

schema definitions and data instances respectively, are imported into these repositories. The core of the integration process is the execution of SPARQL CONSTRUCT queries. These queries transform local data into a format that conforms to the global schema, enabling seamless integration into the knowledge graph. The integration is automated, ensuring that updates are performed efficiently and consistently.
The SPARQLWrapper library is employed to execute SPARQL CONSTRUCT queries against the local repositories. This Python library acts as a wrapper for interacting with SPARQL endpoints, facilitating the extraction and transformation of data. Once the data is transformed, it is collected in RDF format.
To insert the transformed data into the global repository, HTTP POST requests are used. These requests are made using the requests library in Python, which handles the transmission of RDF data to the GraphDB instance. The HTTP POST requests ensure that the data is securely and accurately inserted into the global schema repository.

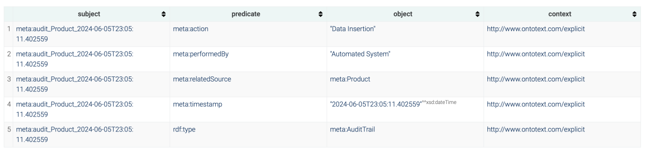
4. **Metadata Creation and Management** As part of the integration, metadata instances are created to track the provenance and actions performed on the data. This metadata is crucial for maintaining transparency and auditability, providing a clear trail of data transformations and integrations. The rdflib library in Python is used to create and manage this metadata, allowing for the creation and manipulation of RDF graphs. The metadata is stored in a dedicated repository, ensuring it is easily accessible for future reference and compliance purposes. This comprehensive approach ensures that data from various sources is consistently transformed and integrated into a robust knowledge graph, supporting advanced analytics and strategic insights.

5. **Exploitation and Analysis**

   The final phase involves the exploitation and analysis of the integrated data. Automated data integration ensures that the knowledge graph is continually updated with the latest information, providing a real-time, dynamic view of the competitive landscape. Graph analytics using SPARQL queries are conducted to make sure our global schema is working and also to extract meaningful insights, supporting strategic decision-making. SPARQL queries are executed to retrieve specific information, enabling detailed analysis and reporting.

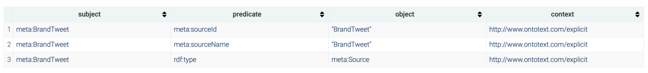
6 Metadata Management (M6)

Metadata plays a crucial role in tracking data provenance, maintaining audit trails, and facilitating reinforced learning. The generation, storage, and reuse of metadata are integrated into the data integration and mapping processes to ensure transparency and accountability. Metadata generation occurs throughout various stages, including data collection, schema creation, data integration, and reinforced learning. During data collection, metadata is generated to capture details about data sources, such as source ID, name, and timestamps. As we create global and local schemas, metadata documents design decisions, validation results, and versioning information. During data integration, metadata tracks the mapping and transformation activities by recording actions, timestamps, and responsible entities. This metadata ensures reproducibility and traceability during graph analytics and reinforced learning processes.
Graph-based metadata management significantly enhances our company's operations by providing a structured and interoperable framework for handling metadata. By using RDF triples to store metadata instances, we enable seamless querying and interoperability across different systems. This structured approach allows for detailed tracking of data provenance, ensuring that every data transformation and integration step is transparent and accountable.
Metadata is stored in a dedicated metadata repository structured to accommodate various types, such as source tracking, audit trails, schema versions, and analytical results. This repository is crucial for maintaining an organized record of all metadata, making it easily accessible for future reference and compliance purposes. The graph-based nature of this repository ensures that metadata can be queried efficiently, supporting various analytical and operational needs.
Metadata reuse supports several enhanced processes. For audit trails and provenance, metadata provides detailed records of data integration activities, enabling traceability. In reinforced learning, historical metadata informs and optimizes future analyses by aiding in parameter tuning and model validation. Metadata also facilitates schema evolution by documenting changes and ensuring consistency over time. Additionally, metadata enables automated data integration tasks by providing a comprehensive catalog of sources, mappings, and transformations, thus reducing manual intervention. This comprehensive metadata management approach not only supports our core operations but also enhances our ability to provide robust, real-time insights to our clients. By maintaining detailed and accessible metadata records, we ensure that our data integration processes are reproducible and transparent, which is essential for building trust and credibility with our clients. Furthermore, the ability to query and analyze metadata helps us continually refine and improve our data integration and analytics processes, driving innovation and efficiency.
Figures 7 and 8 illustrate the Audit Trails for the Product class and Metadata sources tracking for the Tweet class, respectively.



	subject	predicate	object	context
1	meta:audit_Product_2024-06-05T23:05:11.402559	meta:action	"Data Insertion"	http://www.ontotext.com/explicit
2	meta:audit_Product_2024-06-05T23:05:11.402559	meta:performedBy	"Automated System"	http://www.ontotext.com/explicit
3	meta:audit_Product_2024-06-05T23:05:11.402559	meta:relatedSource	meta:Product	http://www.ontotext.com/explicit
4	meta:audit_Product_2024-06-05T23:05:11.402559	meta:timestamp	"2024-06-05T23:05:11.402559"^^xsd:dateTime	http://www.ontotext.com/explicit
5	meta:audit_Product_2024-06-05T23:05:11.402559	rdf:type	meta:AuditTrail	http://www.ontotext.com/explicit

Figure 7: Audits Trails for Product class



	subject	predicate	object	context
1	meta:BrandTweet	meta:sourceId	"BrandTweet"	http://www.ontotext.com/explicit
2	meta:BrandTweet	meta:sourceName	"BrandTweet"	http://www.ontotext.com/explicit
3	meta:BrandTweet	rdf:type	meta:Source	http://www.ontotext.com/explicit

Figure 8: Metadata Source Tracking for Tweet Class

7 Proof of Concept (PoC) Implementation

7.1 Tools Selection and Justification

- **Graph Database/Triplestore: GraphDB**
  - **Justification**: GraphDB is selected for its robust RDF triplestore capabilities, support for SPARQL querying, efficient storage, and management of RDF data. It is ideal for handling large-scale semantic data integration tasks and ensuring data consistency across various sources.
- **Graph Framework: RDFLib**
  - **Justification**: RDFLib is chosen for its simplicity and flexibility in creating and manipulating RDF graphs. It supports parsing, serialization, and querying of RDF data, which aligns with the project's requirements for mapping local data schemas to a global schema.
- **Query Engine: SPARQLWrapper**
  - **Justification**: SPARQLWrapper provides an easy-to-use interface for executing SPARQL queries and interacting with SPARQL endpoints. It simplifies the process of sending queries to the triplestore and handling results, crucial for automating the data integration process.

7.2 PoC Setting

The PoC demonstrates the integration of data from local repositories into a global schema using GraphDB, RDFLib, and SPARQLWrapper. The process involves the following steps:

1. **Setup GraphDB Triplestore**
   - **Configuration**: Install GraphDB and configure it to host the necessary repositories, including local data repositories (`local_consumer_tweet`, `local_product_data`, `local_reviews_data`, `local_user_tweet`) and a global schema repository (`global_schema`).
2. **Data Transformation and Integration Using RDFLib**
   - **SPARQL CONSTRUCT Queries**: Define queries to map local schema data to the global schema.
   - **RDF Graph Creation**: Use RDFLib to create RDF graphs, adding triples for entities like `Consumer`, `Tweet`, `Product`, and `Review`.
   - **Metadata Management**: Implement metadata creation to track the source and audit trails of the data.
3. **Query Execution and Data Insertion with SPARQLWrapper**
   - **Execute Queries**: Use SPARQLWrapper to execute CONSTRUCT queries on local repositories.
   - **Collect RDF Data**: Collect query results and prepare them for insertion into the global repository.
   - **Insert Data**: Insert RDF data and metadata into the global schema repository in GraphDB.

7.3 Justification for Tool Choices

GraphDB is chosen for its robust support for RDF and SPARQL, making it ideal for semantic data integration. RDFLib provides a flexible and easy-to-use interface for creating and manipulating RDF graphs, which is essential for the project. SPARQLWrapper simplifies the process of executing SPARQL queries and handling results, facilitating efficient data integration and query execution.

8 Conclusion

The project demonstrated the feasibility and efficiency of using graph-based solutions for integrating diverse data sources in the competitive intelligence domain. DigiScan360 showcased its ability to provide dynamic, real-time analysis of market conditions and competitor activities, emphasizing the importance of ethical data management to ensure transparency and trust. Lessons learned include the critical need for designing scalable schemas and integration processes to handle increasing data volumes, the benefits of automating data integration to enhance efficiency and reduce manual errors, and the necessity of intuitive visualization tools to make insights accessible and actionable for end-users. Potential areas for future work involve incorporating machine learning models to enhance predictive analytics and trend forecasting, expanding the range of data sources to include IoT devices and more extensive social media platforms, developing more user-friendly interfaces for better interaction with the knowledge graph and visualization tools, and further strengthening data privacy measures and compliance with evolving regulations to maintain user trust.