Linked Open Data and Knowledge Graph

Final Project Presentation

OpenAlex

Habiba Naeem, Rafael Barros

Technology
Arts Sciences
TH Köln

Agenda

OpenAlex

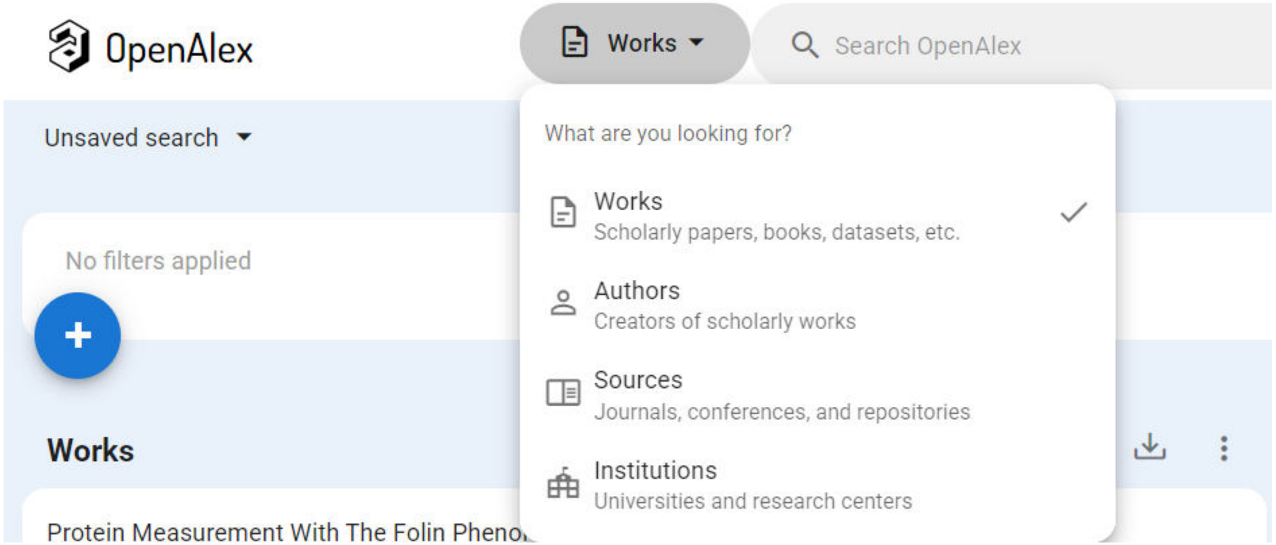
- A large-scale, open-source platform of scholarly works, authors, institutions, concepts etc
- Designed to replace Microsoft Academic Graph (MAG) after it was discontinued

Cabecera de la plataforma web OpenAlex. Clic para acceder

https://www.lluiscodina.com/openalex-scopus/

OpenAlex Scope

- Millions of Works (papers, proceedings etc.)
- Works, Authors, Institutions, Venues (journals/conferences), Concepts (subjects/topics)



What are you looking for?

Works
Scholarly papers, books, datasets, etc.

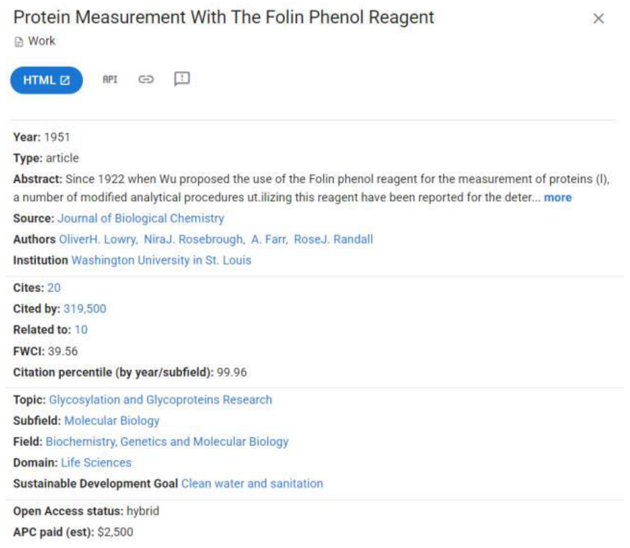
Authors
Creators of scholarly works

Sources
Journals, conferences, and repositories

Institutions
Universities and research centers

Why did we choose OpenAlex?

- **Rich Metadata**: Includes metadata for authors, works, venues, institutions, and research topics
- **Ideal for Knowledge Graphs**: Structured data that naturally maps to nodes and edges
- **API Access:** Offers a robust API for easy data retrieval and integration into research projects.
- **Open Licensing:** Operates under an open license, ensuring unrestricted access and use.
- **Use Cases:** Ideal for bibliometric analysis, research trend exploration, and knowledge graph building.



Protein Measurement With The Folin Phenol Reagent

Work

Year: 1951
Type: article
Abstract: Since 1922 when Wu proposed the use of the Folin phenol reagent for the measurement of proteins (l), a number of modified analytical procedures ut.ilizing this reagent have been reported for the deter... more
Source: Journal of Biological Chemistry
Authors OliverH. Lowry, NiraJ. Rosebrough, A. Farr, RoseJ. Randall
Institution Washington University in St. Louis

Cites: 20
Cited by: 319,500
Related to: 10
FWCI: 39.56
Citation percentile (by year/subfield): 99.96

Topic: Glycosylation and Glycoproteins Research
Subfield: Molecular Biology
Field: Biochemistry, Genetics and Molecular Biology
Domain: Life Sciences
Sustainable Development Goal Clean water and sanitation

Open Access status: hybrid
APC paid (est): $2,500

What does OpenAlex do?

The ancient Library of Alexandria aimed to create a universal collection of scholarship, indexed using the first library catalog, the Pinakes. They're working towards the same goal, but making it completely open.

- Their data is free and reusable, available via bulk download or API,
- their code is fully open-source;
- they're governed by a sustainable and transparent nonprofit.

OpenAlex believes the global research system is one of humankind's most beautiful creations. They aim to make that whole beautiful creation available to everyone, everywhere.

Data Extraction possibilities – Breakdown Summary

Features	API Access	Dataset Snapshots
Data Freshness	Real-Time updates	Static at export
Data Volume	Paginated, limited	Full database
Latency	Higher for large pulls	Local, faster access
Technical Skill	Moderate	Advanced
Scalability	Limited by rate limits	Supports big data
Storage Needs	Minimal	High

Entities and Relationships covered

1. Work – AUTHORED_BY → Author
2. Author – AFFILIATED_WITH → Institutions
3. Work – PUBLISHED_IN → Year
4. Work – HAS_Topic → Topics
5. Work – BELONGS_TO → Domain
6. Work – CITES → Work Yet to be covered..!

Why these Relationships?

- Captures the most critical scholarly links (authorship, institutional affiliation, venue, topical concepts, citations).
- Provides a solid core for building a **Scholarly Knowledge Graph**.

Additional possible extensions:

- Cited_by relationships (larger scale, more API calls) - problem emerged here
- Concept-to-Concept hierarchies (e.g., broader/narrower concepts)
- Author-to-Institution direct links

Methods and Material used

Approach used:

- Python scripts in **Google Colab.**
- Accessed circa 5,000 Works (25 pages × 200 items/page) using the OpenAlex API.
- Implemented a small delay (`time.sleep`) to avoid rate-limit issues.

Methods and Material

Core Python Libraries:

- requests: for API calls
- pandas: for data manipulation (DataFrames, CSV export)

Methods and Material

Data Modeling:

- Stored each record as `(subject, relationship, object)`
- Used unique OpenAlex IDs for Works (e.g., `Wxxxx`), Authors (`Axxxx`) etc.

Methods and Material

Knowledge Graph Construction:

- Built a large list of **(subject, relationship, object)** triples
- Exported to CSV for further analysis in Neo4j and **graph visualization** (Gephi)

Data Preprocessing – Cleaning and Preparing Relationships

- **Duplicates Removal**: Ensured no repeated triples;
- **Null Checks**: Filtered out any rows missing subject/relationship/object;
- **Simple Schema**: Kept relationships minimal and consistent (e.g., “CITES,” “AUTHORED_BY,” etc.);

Data Preprocessing

Transforming for Gephi:

- Created two main files: nodes.csv (all unique IDs + types) and edges.csv (source, target, relationship)
- This structure is **Gephi-friendly** for network visualization

Data Preprocessing

Potential Extensions:

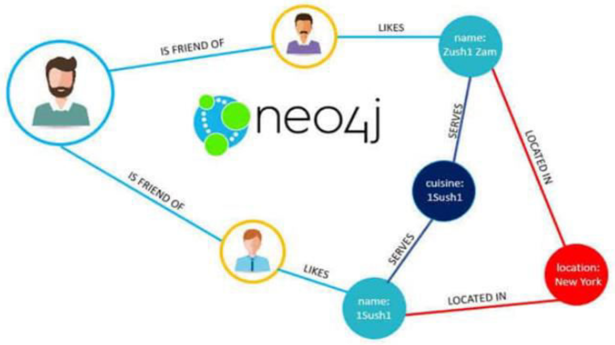
- Additional columns for weighting edges (e.g., citation frequency);
- Enhanced node metadata (e.g., year published, author name);

Introduction to Neo4j

Neo4j is a graph database platform designed to store, query, and analyze data as a network of nodes and relationships, reflecting real-world connections. Unlike traditional relational databases, Neo4j uses a graph-native architecture, making it highly efficient for handling connected data. With its intuitive Cypher query language, it enables users to explore complex relationships and gain insights quickly, making it ideal for applications like fraud detection, recommendation systems, and social network analysis.

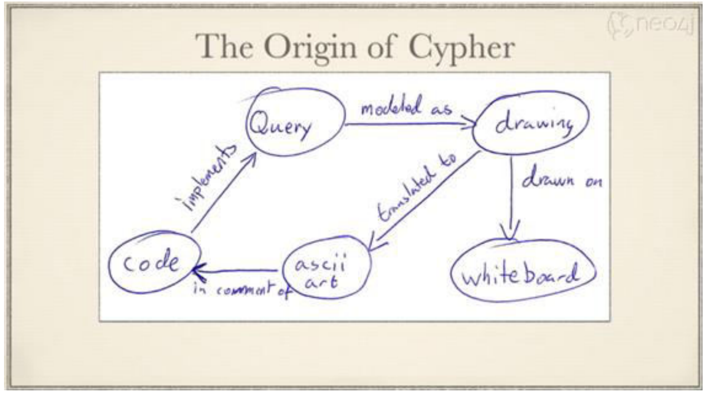
Why did we choose Neo4j?

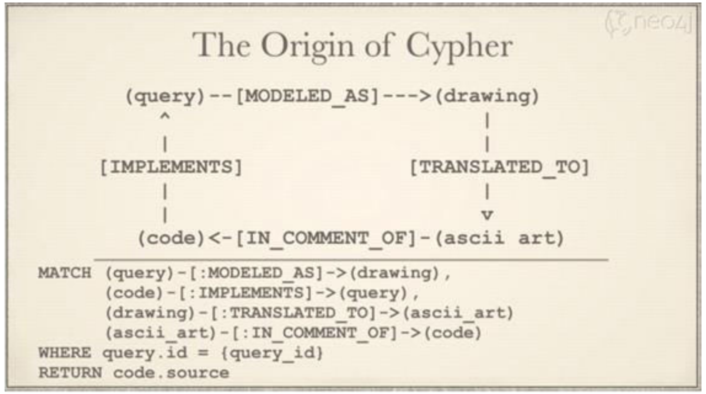
- Graph-native database that mirrors real-world connections.
- User-friendly Cypher query language for complex analysis.
- Delivers real-time insights for highly connected data.
- Scalable and efficient for large datasets.
- Flexible and adapts to evolving data models.
- Supported by a strong community and learning resources.



Introduction to Cypher Queries

- Designed for querying graph data with a simple, declarative syntax.
- Supports pattern matching to explore nodes and relationships.
- Enables efficient path traversal and complex data queries.
- Includes built-in functions for filtering, aggregation, and analysis.
- Easily integrates with Neo4j and popular programming languages.




The Origin of Cypher

```
(query)--[MODELED_AS]--->(drawing)
   ^                         |
   |                         |
[IMPLEMENTS]           [TRANSLATED_TO]
   |                         |
   |                         v
(code)<-[IN_COMMENT_OF]-(ascii art)

MATCH  (query)-[:MODELED_AS]->(drawing),
       (code)-[:IMPLEMENTS]->(query),
       (drawing)-[:TRANSLATED_TO]->(ascii_art)
       (ascii_art)-[:IN_COMMENT_OF]->(code)
WHERE  query.id = {query_id}
RETURN code.source
```

https://docs.nebula-graph.io/3.3.0/1.introduction/0-1-graph-database/#the_creation_of_cypher

Steps to create knowledge graph in Neo4j.

Step 1: Creating Database in Neo4j

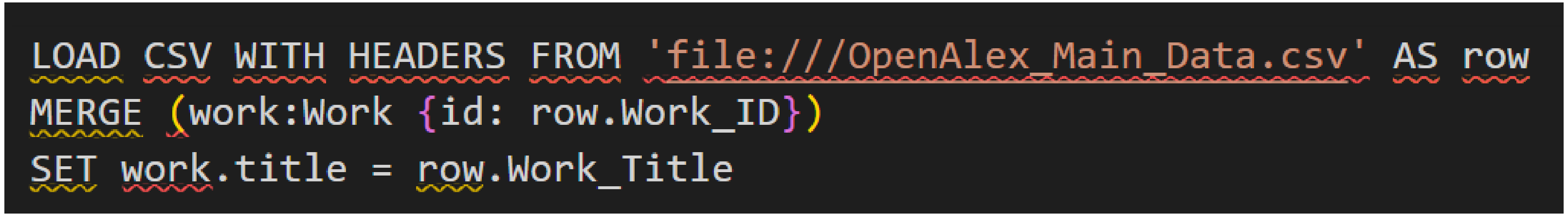
Step 2: Preparing data for import

Step 3: Upload csv files to Neo4j

Let's take an example of creating a knowledge graph of Work Published_In Year relationship.

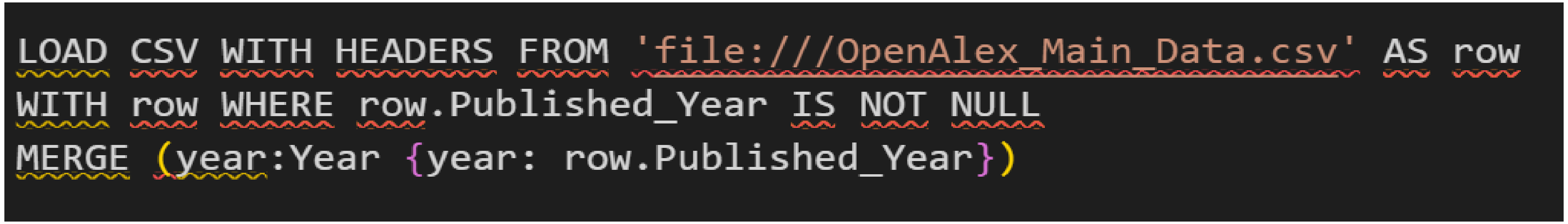
Steps to create knowledge graph in Neo4j.

Step 4: Creating Work nodes



```
LOAD CSV WITH HEADERS FROM 'file:///OpenAlex_Main_Data.csv' AS row
MERGE (work:Work {id: row.Work_ID})
SET work.title = row.Work_Title
```

Step 5: Creating Year nodes

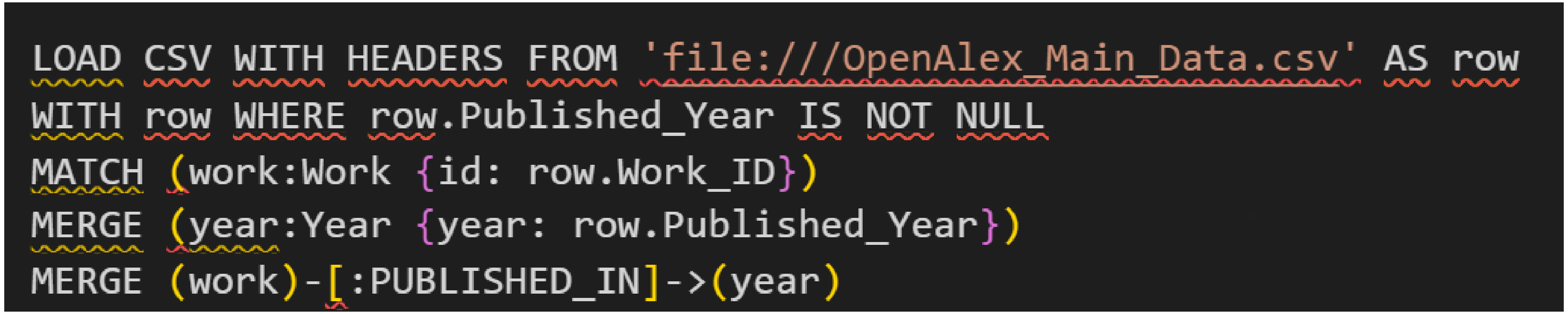


```
LOAD CSV WITH HEADERS FROM 'file:///OpenAlex_Main_Data.csv' AS row
WITH row WHERE row.Published_Year IS NOT NULL
MERGE (year:Year {year: row.Published_Year})
```

Technology
Arts Sciences
TH Köln

Steps to create knowledge graph in Neo4j.

Step 6: Creating Relationship of PUBLISHED_IN between Work and Year

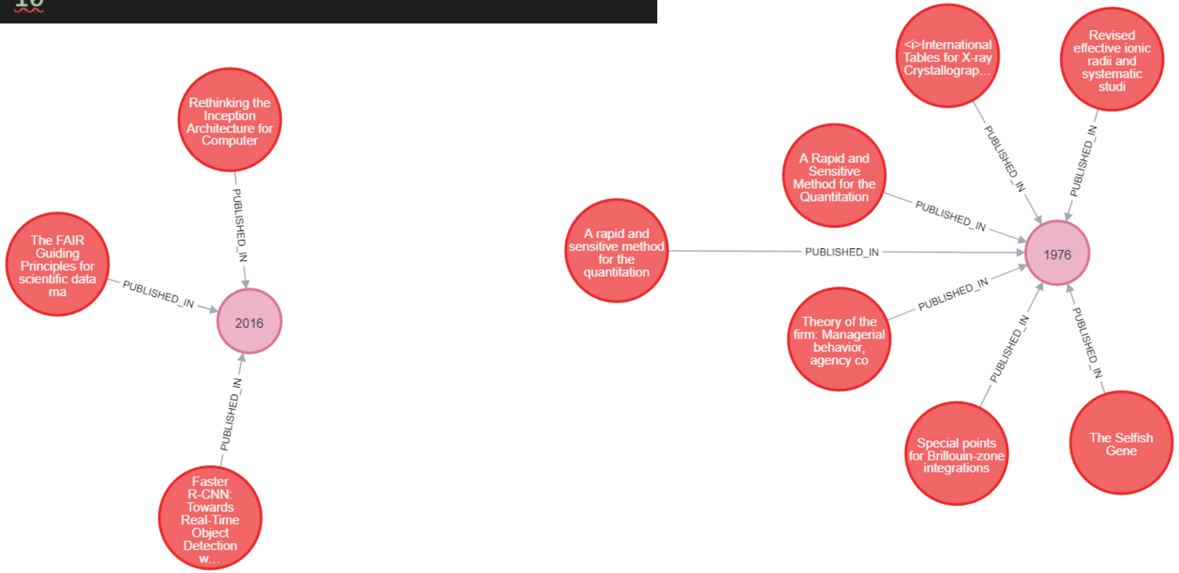


```
LOAD CSV WITH HEADERS FROM 'file:///OpenAlex_Main_Data.csv' AS row
WITH row WHERE row.Published_Year IS NOT NULL
MATCH (work:Work {id: row.Work_ID})
MERGE (year:Year {year: row.Published_Year})
MERGE (work)-[:PUBLISHED_IN]->(year)
```

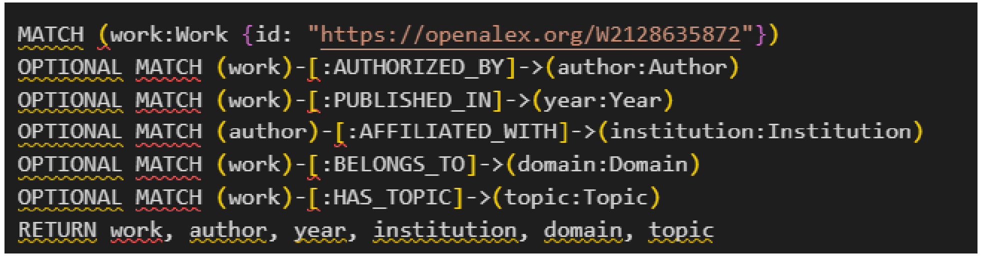
Steps to create knowledge graph in Neo4j.

Step 7: Visualizing Relationship of PUBLISHED_IN between Work and Year

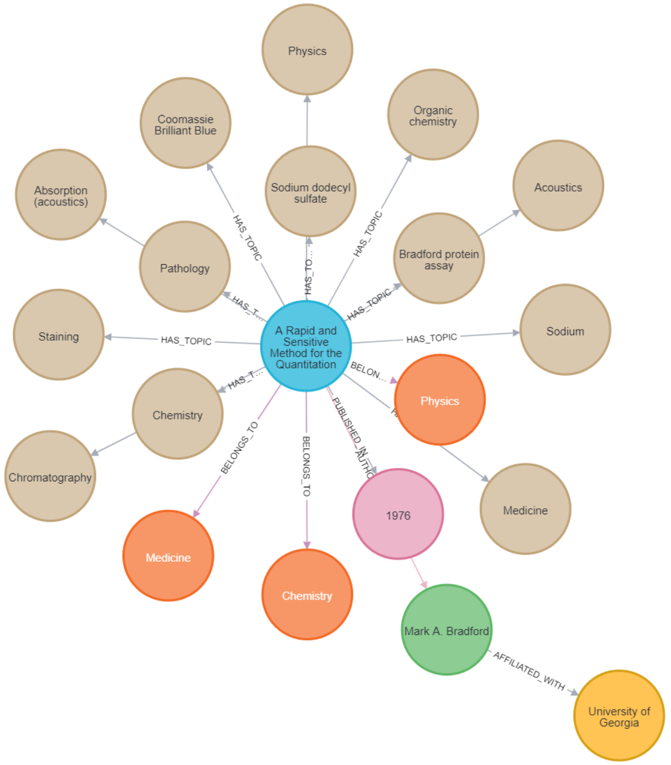
```
MATCH (work:Work)-[:PUBLISHED_IN]->(year:Year)
RETURN work, year
LIMIT 10
```


Rethinking the Inception Architecture for Computer
The FAIR Guiding Principles for scientific data ma
2016
Faster R-CNN: Towards Real-Time Object Detection w...
PUBLISHED_IN
<i>International Tables for X-ray Crystallograp...
Revised effective ionic radii and systematic studi
A Rapid and Sensitive Method for the Quantitation
A rapid and sensitive method for the quantitation
1976
Theory of the firm: Managerial behavior, agency co
Special points for Brillouin-zone integrations
The Selfish Gene

Example query with Knowledge graph



```
MATCH (work:Work {id: "https://openalex.org/W2128635872"})
OPTIONAL MATCH (work)-[:AUTHORIZED_BY]->(author:Author)
OPTIONAL MATCH (work)-[:PUBLISHED_IN]->(year:Year)
OPTIONAL MATCH (author)-[:AFFILIATED_WITH]->(institution:Institution)
OPTIONAL MATCH (work)-[:BELONGS_TO]->(domain:Domain)
OPTIONAL MATCH (work)-[:HAS_TOPIC]->(topic:Topic)
RETURN work, author, year, institution, domain, topic
```



A quick walk through the code, Knowledge graphs and relationships covered.

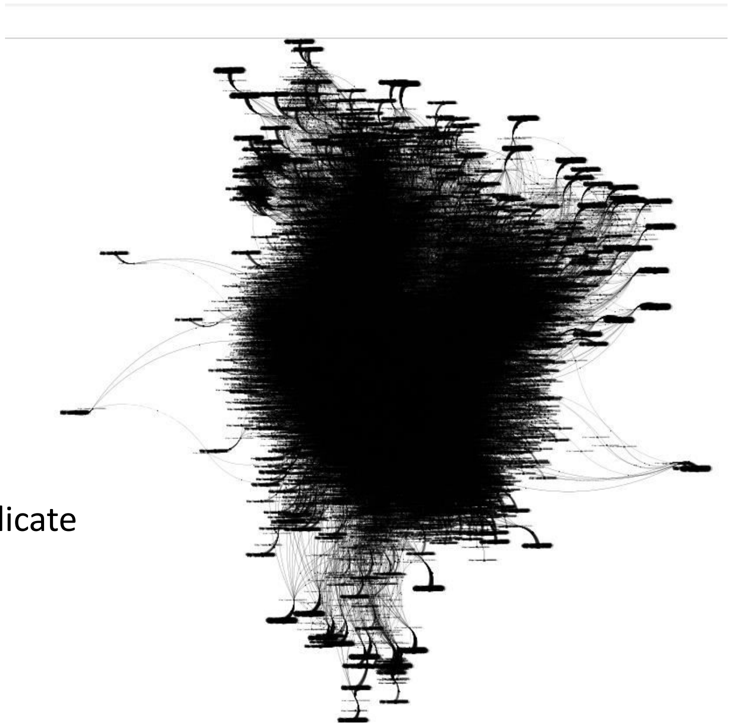
1. Work – AUTHORIZED_BY → Authors
2. Work – PUBLISHED_IN → Year
3. Authors – AFFILIATED_WITH → Institutions
4. Work – BELONGS_TO → Domains
5. Work – HAS_Topic → Topics

Issues faced with Neo4j

1. Project Saving Challenges: Unable to save projects directly within Neo4j, requiring external tools like Google Colab for code storage.
2. Code Snippet Loss: Initial Cypher code snippets disappear as more coding is done in the browser interface.
3. Integration Limitations: Difficulty connecting Neo4j projects to a GitHub repository for version control and collaboration.

Output in Gephi

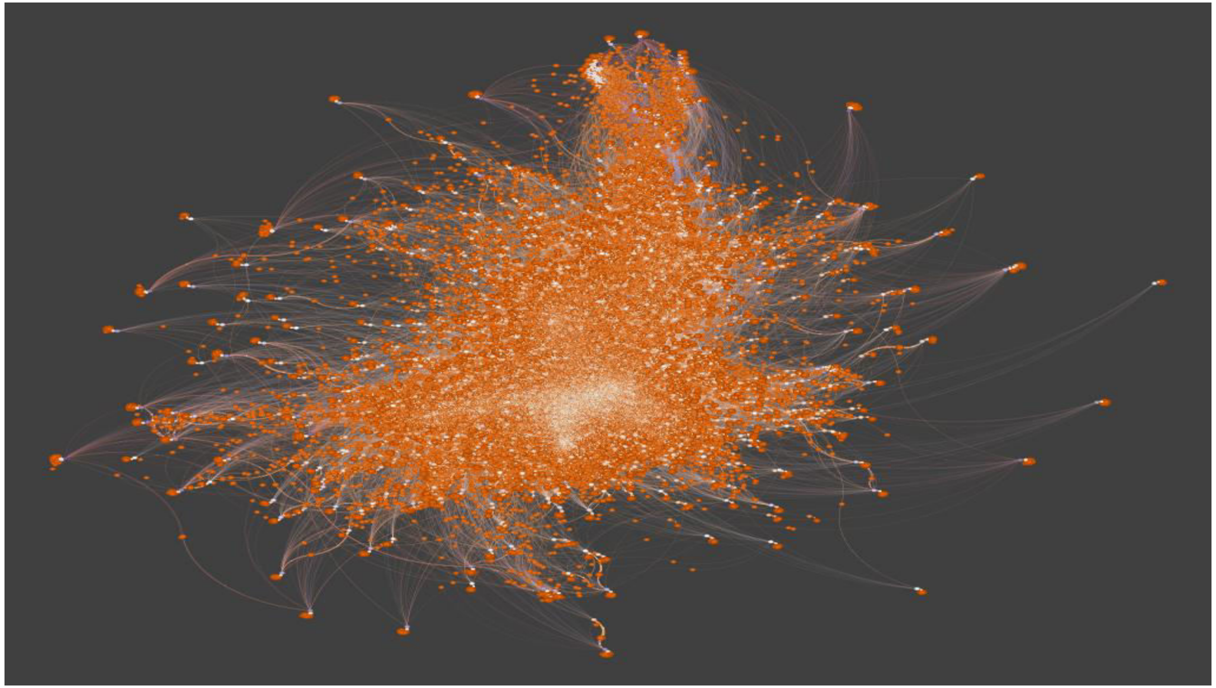
- Open-source tool for visualizing and analyzing networks.
- User-friendly interface for customizable graph layouts.
- Handles large datasets and calculates network metrics.
- Ideal for social network analysis and knowledge graphs.



Graph Shows:

- The graph's density (black cluster) suggests a highly connected network with many relationships.
- Specific nodes may vary in size or color (if styled) to indicate metrics such as centrality, importance, or degree of connectivity.

Better Output so far



Issues with Gephi

- Trying to model better the entity-relationship pattern
- There are some issues regarding data visualization in Gephi
- Lack of experience

Remaining work

- Will **try** to cover Cites and Cited by relationship.
- Work around more with Gephi to improve visualizations
- Complete the report.

Thank You ☺

References

- https://docs.openalex.org/
- https://docs.openalex.org/#why-openalex
- https://docs.openalex.org/quickstart-tutorial
- https://openalex.org/about
- https://docs.openalex.org/download-all-data/openalex-snapshot