Developing Knowledge Graph Based On OpenAire Database

Table of contents

Goal

To develop a knowledge graph based on the OpenAIRE dataset to extract information about the most important research papers.

What is the OpenAIRE Graph?

The OpenAIRE Graph is a free and open resource that brings together and interlinks hundreds of millions of metadata records from over 100k data sources trusted by researchers.

Dataset overview

The dataset selected contains metadata records of the OpenAIRE Graph for the research community “Technological University Network (TU-NET).”

TU-NET is a network for the Irish Technological Universities to share expertise, information and resources

Data structure

The dataset is stored in multiple archive files, where each line represents a metadata record. Below is an example of a single line from the dataset:


```
{"author":[{"fullName":"Hegarty, Nora","name":"Nora","surname":"Hegarty","rank":1,"pid":null},{"fullName":"Carbery,
    Alan","name":"Alan","surname":"Carbery","rank":2,"pid":null},{"fullName":"Hurley,
    Tina","name":"Tina","surname":"Hurley","rank":3,"pid":null}],"openAccessColor":null,"publiclyFunded":true,"type":"publication","language":{"code":"eng","label":"English"},"country":
    [{"code":"IE","label":"Ireland","provenance":{"provenance":"Inferred by OpenAIRE","trust":"0.85"}}],"subjects":[{"subject":{"scheme":"keyword","value":"*NONE OF
    THESE*"},"provenance":{"provenance":"Harvested","trust":"0.9"}}],"mainTitle":"Learning by Doing: Re-designing the First Year Information Literacy Programme at WIT
    Libraries","description":["The purpose of this paper is to describe the process involved in re-designing Waterford Institute of Technology (WIT) Libraries' information literacy programme
    for first year students. It is written by some of the members of the library learning support team, who deliver the programme. It describes the steps involved in the programme's development
    and design, discusses the pedagogical principles that influenced the initiative, and summarises the evaluations we have undertaken to date. These evaluations have yielded positive informal
    and formal feedback from the students and lecturers who participated in the programme. The value of a pedagogically sound, active learning approach to information literacy training is
    highlighted throughout the results. By providing constructive solutions for incorporating active learning into library user education programmes, this paper is expected to be a useful
    source of practical information for libraries in similar positions, of similar scale, faced with similar challenges. It is likely to be of particular interest to librarians involved in
    information literacy
    education."],"publicationDate":"2009-12-01","source":[],"format":["application/pdf"],"contributor":[],"coverage":
    [],"bestAccessRight":{"code":"c_abf2","label":"OPEN","scheme":"http://vocabularies.coar-repositories
    .org/documentation/access_rights/"},"id":"dedup_wf_002::12160c03473cd6c55729828ba2976da5","originalId":["oai:repository.wit.ie:1431","oai:generic.eprints.org:1431"],"pid":
    [],"dateOfCollection":"2024-07-05T19:14:43.524","lastUpdateTimeStamp":1720173410144,"indicators":{"citationImpact":{"citationCount":0.0,"influence":2.9837197E-9,"popularity":5
    .9487604E-10,"impulse":0.0,"citationClass":"C5","influenceClass":"C5","impulseClass":"C5","popularityClass":"C5"}},"context":[{"code":"eu-conexus","label":"European University for Smart
    Urban Coastal Sustainability","provenance":[{"provenance":"Inferred by OpenAIRE","trust":"0.8"}]},{"code":"tunet","label":"TU-NET","provenance":[{"provenance":"Inferred by
    OpenAIRE","trust":"0.8"}]}],"collectedFrom":[{"key":"opendoar____::4daa3db355ef2b0e64b472968cb70f0d","value":"SETU Open Access
    Repository"}],"instance":[{"pid":[],"alternateIdentifier":
    [],"accessRight":{"code":"c_abf2","label":"OPEN","scheme":"http://vocabularies.coar-repositories.org/documentation/access_rights/","openAccessRoute":null},"type":"Article","url":
    ["http://repository.wit.ie/1431/"],"publicationDate":"2009-12-01","refereed":"peerReviewed","hostedBy":{"key":"opendoar____::4daa3db355ef2b0e64b472968cb70f0d","value":"SETU Open Access
    Repository"},"collectedFrom":{"key":"opendoar____::4daa3db355ef2b0e64b472968cb70f0d","value":"SETU Open Access Repository"}}],"isGreen":true,"isInDiamondJournal":false}
```

Implementation steps

1. Read each line from all archive files and store it in a variable.

```
def read_json(input_folder):
    all_records = []

    for file_name in os.listdir(input_folder):
        if file_name.endswith(".json.gz"):
            input_path = os.path.join(input_folder, file_name)

            with gzip.open(input_path, 'rt', encoding='utf-8') as f:
                for line in f:
                    record = json.loads(line)
                    all_records.append(record)

    return all_records
```

Implementation steps

2. Convert the data into CSV format by selecting relevant fields. Each CSV file represents a node in the knowledge graph.

```python
def flatten_main(record):
    indicators = record.get('indicators', {}) or {}
    citation_impact = indicators.get('citationImpact', {}) or {}
    return {
        'id': record.get('id', ''),
        'title': record.get('mainTitle', ''),
        'description': ' '.join(record.get('description', [])),
        'type': record.get('type', ''),
        'citationCount': citation_impact.get('citationCount', 0),
        'influence': citation_impact.get('influence', 0),
        'popularity': citation_impact.get('popularity', 0),
        'url': record.get('url', [None])[0],
    }
```

Implementation steps

3. Preprocess and clean the data to ensure consistency and accuracy.

```
df = pd.read_csv("./keywords.csv")
# To lowercase
df = df.applymap(lambda x: x.lower() if isinstance(x, str) else x)
# Then remove duplicates
df = df.drop_duplicates()
df.to_csv("keywords.csv", index=False)
```

Implementation steps

4. Connect to Neo4j Aura (console) using the Neo4j library.

```
driver = GraphDatabase.driver(uri, auth=(username, password))
```

Implementation steps

5. Use Cypher queries to establish relationships between the nodes.

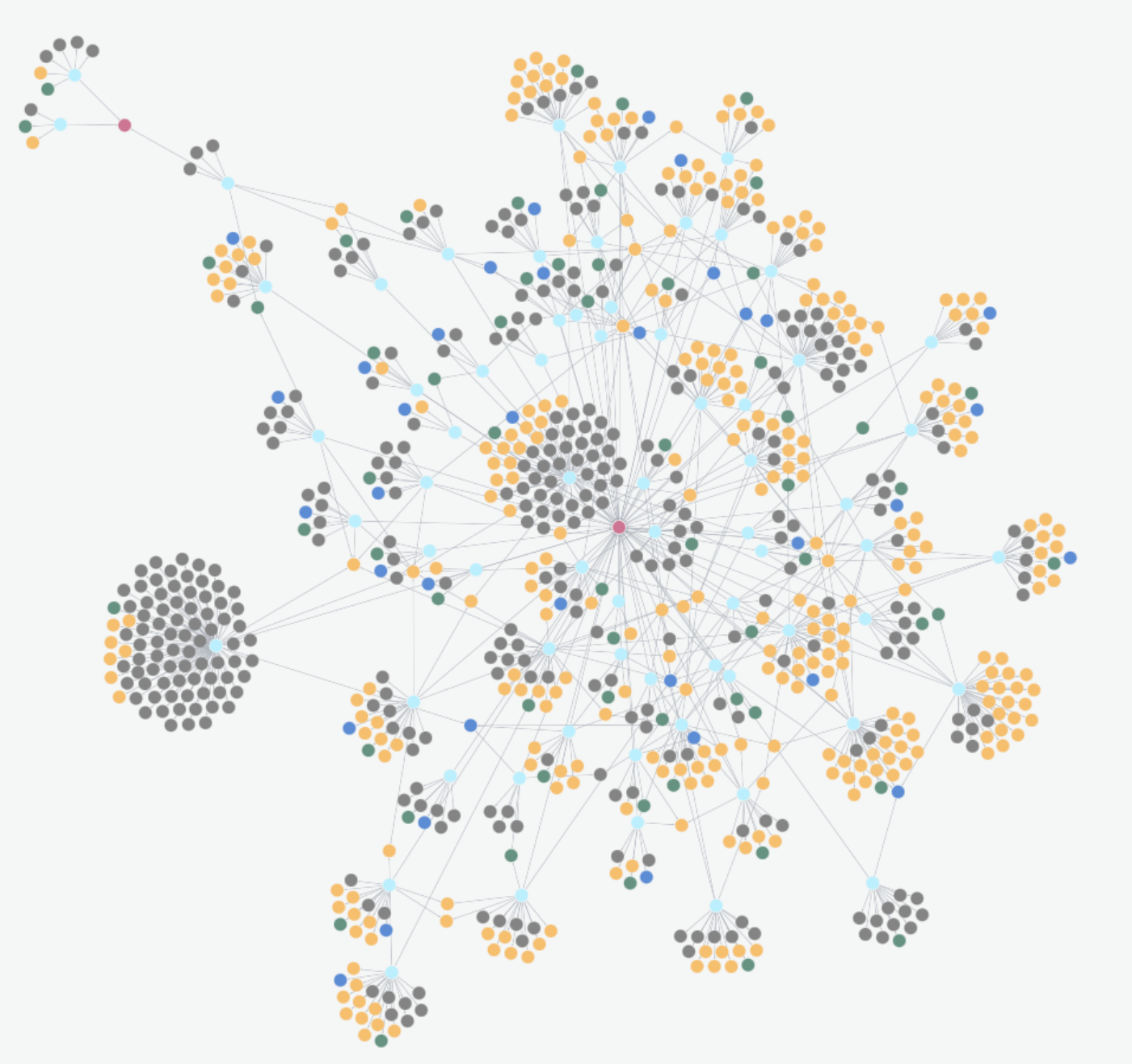
```
query_main = """
MERGE (r:Record {id: $id})
SET r.title = $title,
    r.description = $description,
    r.type = $type,
    r.citationCount = $citationCount,
    r.influence = $influence,
    r.popularity = $popularity,
    r.url = $url
"""
```

Implementation steps

6. Upload the processed data into the Neo4j database.

```
def upload_data_from_csv(file_path, query):
    with driver.session() as session:
        with open(file_path, 'r', encoding='utf-8') as file:
            reader = csv.DictReader(file)
            for row in reader:
                session.run(query, row)
```

Visualization of Knowledge Graph

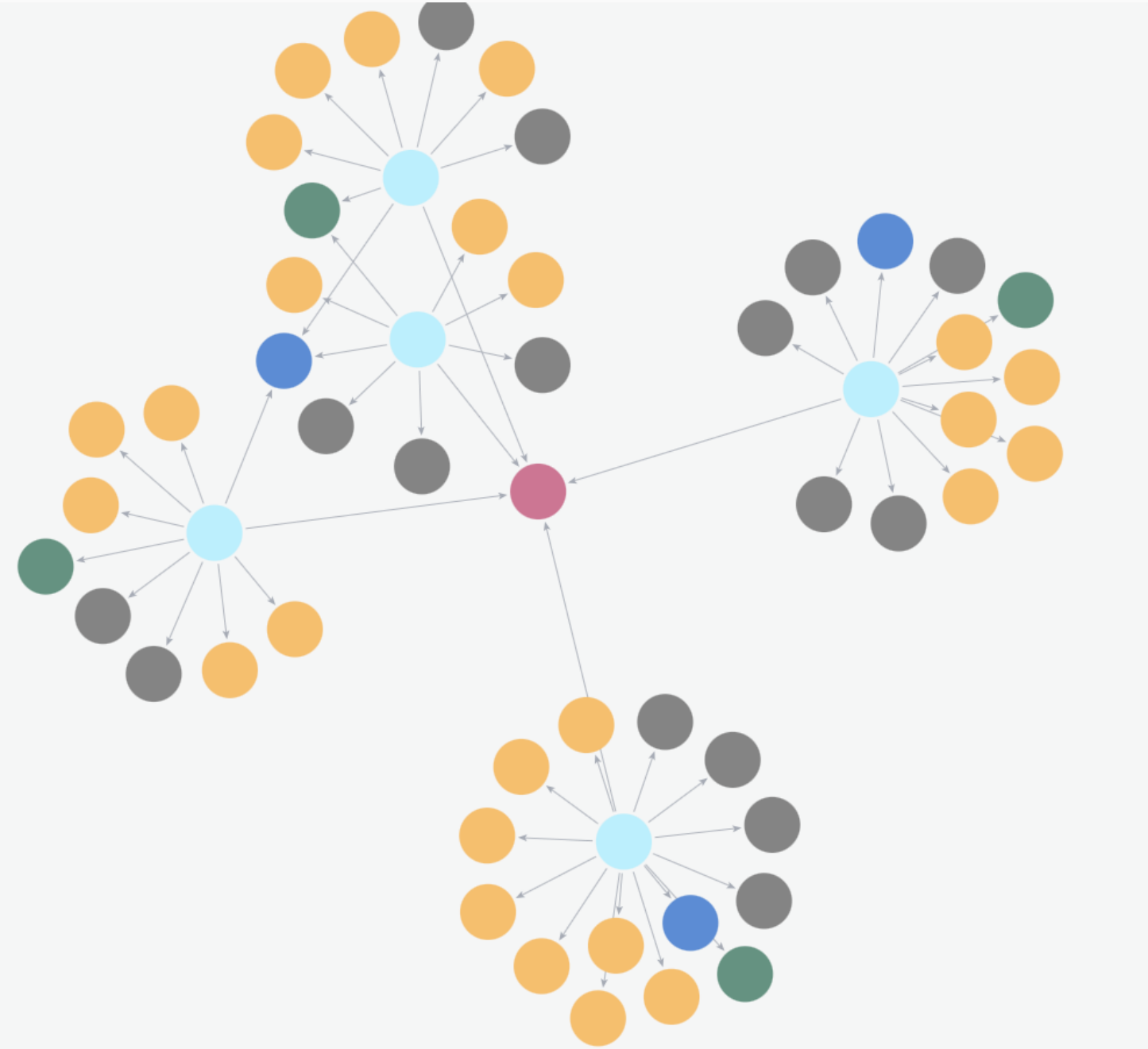


Usage examples

The following Cypher query retrieves the most cited records in English:

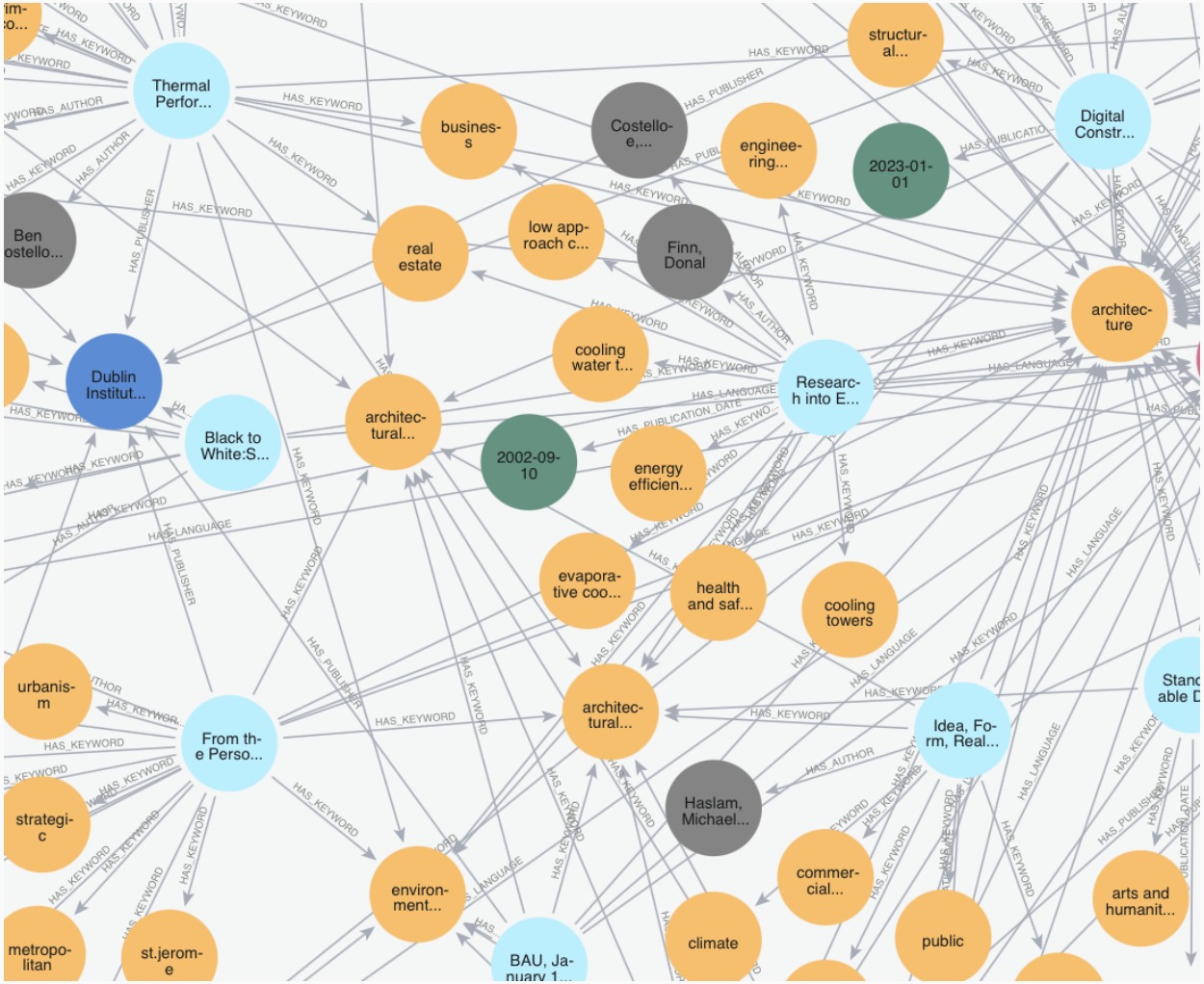
```
MATCH (r:Record)-[:HAS_LANGUAGE]->(l:Language)
WHERE r.citationCount IS NOT NULL AND l.language = 'English
WITH r
ORDER BY r.citationCount DESC
LIMIT 5

MATCH p=(r)-[rel]-(m)
RETURN p
```

Usage examples



Usage examples

All records associated with the keyword architecture



Sources

1. OpenAire Graph - https://graph.openaire.eu
2. Neo4j documentation - https://neo4j.com/docs/getting-started/languages-guides/neo4j-python/
3. Dataset source - https://zenodo.org/records/13135167