Data Engineering (LTAT.02.007)
Project
Designing and Implementing a Data Pipeline to Analyze Scientific Publications

This document contains the project to fulfill the grading requirements in Data Engineering (LTAT.02.007) for the academic year 2023/2024.

Dr. Feras Awaysheh
Fall 2023/2024

Introduction

This project is designed to provide you with hands-on experience in data engineering, modeling, and analytics while tackling real-world challenges in the realm of scientific publications data. The project's primary objective is to master the art of crafting multi-faceted analytical views and ensure the data pipeline's repeatability. As we delve into this endeavor, we will explore diverse analytical perspectives, from building data warehouses and data marts for Business Intelligence (BI) queries to harnessing the capabilities of graph databases for tasks like co-authorship prediction and community analysis. Moreover, we will delve into the intricacies of data transformations, data cleansing, and data augmentation, ultimately preparing you to address the complexities of real-world data engineering projects."

Objective

To practice the different concepts and modeling techniques and build multi-faceted analytical views on scientific publications data. One view is to build a data cube that can answer BI queries, i.e., building a data warehouse or a data mart. The queries can be in the form of ranking authors in a scientific discipline, computing authors, journals, and publishers' H-index. Another view is to utilize graph databases to query for, e.g., co-authorship prediction and analyzing communities. Other views, e.g., using document databases, could serve as an intermediate stage to systematically load and extract the data for the other analytical views. Another objective is to make the process repeatable. That is, you rerun the pipeline to ingest new batches of the data, simulating what happens in reality.

Dataset

You start from the ArXiv data set: https://www.kaggle.com/datasets/Cornell-University/arxiv?resource=download. You don't have to work with all the data. You can select, e.g., 200K records. Furthermore, split them into partitions e.g., 50K each and then feed them on iterations to the pipeline to simulate an incremental update to the target DW and Graph database.

DWH View

It is recommended to use Postgres or MySQL as an RDBMS.

You need to build a star/snowflake schema that stores the data about scientific publications. Example dimensions are authors, publication venues, scientific domain, year of publication, and authors' affiliation. The fact table should store the records of the combinations of the different dimension values. Queries could be to rank the authors of a given scientific discipline, compute the H-index, and Histograms the number of publications in a given topic over a given time period. You can come up with more queries. You also need to consider the attributes of the different dimension tables and

Graph View

It is recommended to use Neo4J as a graph database.

In this view, you need to establish what constitutes nodes and relationships (edges). For example, authors, papers, and journals, or other publication venues are good candidates to act as nodes in the graph. Edges can represent authorship between an author and a paper, co-authorship among authors, works-for between author and affiliation, cites relationship among papers, etc. You need to come up with your proposal. For graph analytics tasks, you can find influential papers. You can use Page rank for this using the citation relation. Detecting communities by finding strongly connected components in relationships like co-authorship or being a member of the same scientific domain. There are readily available libraries for graph analytics in Neo4J.

Data Transformations

Finding citations

The limitation of the data set is that it does not contain reference entries. We need this information and other information to enrich the data. One can use one of the following tools to enlarge the data set by first querying by the paper title. This can be repeated for several cycles, two or three. We are not expecting you to do that for all the papers you have, and we know that it is computation-intensive. Samples from the Kaggle data set are sufficient. It is better to seed the search for papers from different domains, where each domain is a cluster of papers. You can use the "categories" in the input data set to pick 10 to 20 papers from each cluster.

The following are tools that you can query for citations:

- scholar.py — A Parser for Google Scholar (icir.org)
- REST API - Crossref
- citations - Retrieving the references in a publication automatically - Academia Stack Exchange
- Publish or Perish (harzing.com)
- DBLP for computer science publications

Data cleansing, enrichment, and augmentation

The following are suggestions. You need to explore the data and come up with other transformation rules.

- You can drop publications with very short titles, e.g. one word, with empty authors,
- You can drop the abstract as it is not required in the scope of this project,
- You can use the DOI key to resolve and retrieve more information about the paper and the Crossref API's
- Defining the type of publication. The data set does not tell whether this is a conference, workshop, symposium, article, or book. Retrieved information from CorssRef, DBLP, or

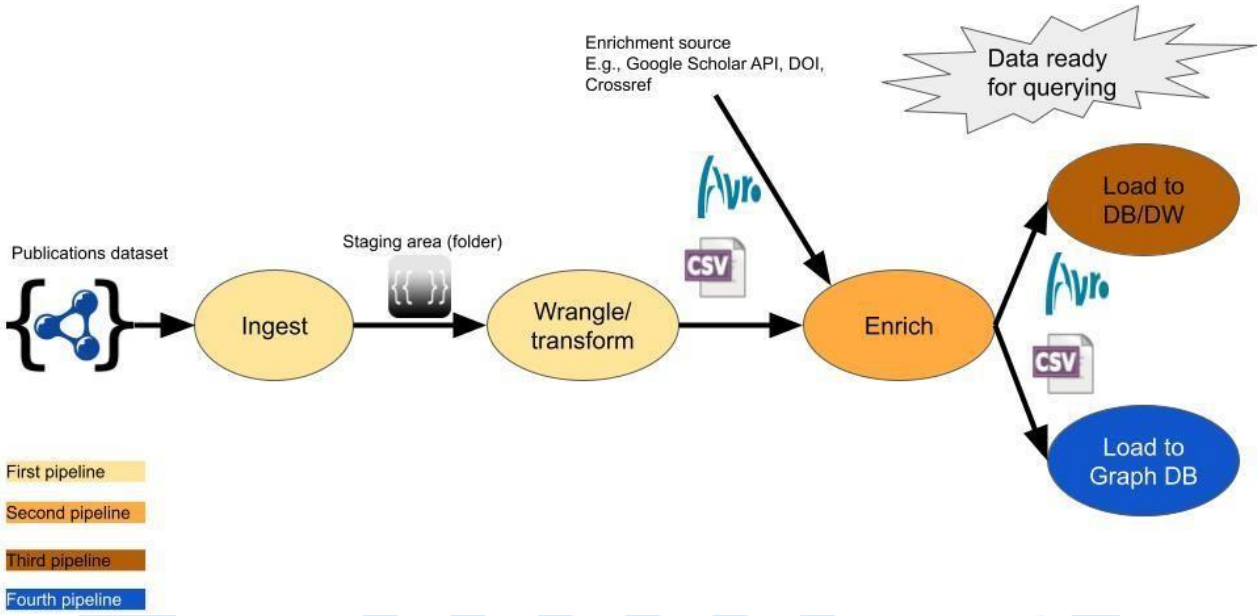
Google Scholar API, the BibTeX entry usually points to the publication type. For example:
    - @article means this is a journal article
    - @inproceedings, usually a conference or a workshop. You can check the booktitle entry and try to figure out whether it is a conference or a workshop
    - @book
    - Etc.
- Resolving ambiguous author names. You can use external systems like Google Scholar, DBLP, MSResearch
    - We have in this repo some Python scripts that we used in a slightly similar context to resolveauthors and refine their publications, https://github.com/DataSystemsGroupUT/Minaret. You can also read the paper of Minaret to help you search through the repository: https://openproceedings.org/2019/conf/edbt/EDBT19_paper_210.pdf
- Resolving ambiguous or abbreviated conference or journal names: You can use Google scholar orDBLP database
    - Google scholar APIs: https://serpapi.com/google-scholar-api
    - DBLP API: https://dblp.org/faq/How+to+use+the+dblp+search+API.html
    - You can also use this to resolve authors and publication type. Remember that this will be more computationally costly due to network data transfer. So, use only when noother means is helpful
- Refining venues
    - For computer science-related publications, you can use DBLP APIs
    - You can also use Google Scholar APIs for other scientific disciplines
- Author gender: You can infer the gender from an author's first name. You can use an API likehttps://gender-api.com/ or can have a lookup table.

Normalization:

- Field of study: The attribute "categories" is defined in many of the papers. But the values seem ad-hoc. So, one way to pick relevant and normalized values for the field of study is to have the following classification, https://confluence.egi.eu/display/EGIG/Scientific+Disciplines, as a lookuptable and match the paper's fos list to it and pick from the lookup tab

Pipelines

The following figure is a suggestion of how the pipeline should look like, you can improve it or suggestmore operations and pipes.



You can implement the pipeline DAG using Airflow/Data Vault.

Marking

The project amounts to 40 points out of 100 for the course. These 40 points are broken down as follows:

- Project design document (**5 points**) [Mandatory step]: This is where you need to define the DWH schema and the types of relationships for the Graph view. In addition, you must describe BI queries that your DWH schema will answer. Additionally, you must mention entities and relationships for your graph database and queries and graph analytics. You should briefly describethe data transformations and enrichment sources with examples of data that are retrieved and augmented with the original data set.
- Data cleansing, enrichment, and augmentation pipeline (**10 points**)
- Populating and querying DWH (**10 points**)
    - Successfully populates the data warehouse with the selected ArXiv dataset (2 points).
    - Implements a well-structured start/snowflake schema that includes essential dimensions and fact tables (3 points).
    - Demonstrates the ability to perform basic queries, such as ranking authors or computing H-index (2 points).

- Implements additional queries or features beyond the basic requirements, showcasing creativity and depth (2 points).
- Effectively handles changes in dimension values by utilizing slowly changing dimensions (1 point).

- Populating and querying Graph view (**10 points**)
  - Successfully populates the graph database (e.g., Neo4J) with relevant nodes and relationships (2 points).
  - Defines clear and logical nodes and relationships based on the dataset (3 points).
  - Implements basic graph queries, such as co-authorship prediction or community analysis (2 points).
  - Implements additional advanced queries or features, demonstrating a deeper understanding (2 points).
  - Shows proficiency in utilizing available graph analytics libraries for tasks like PageRank or community detection (1 point)
- Final report and presentation (**5 points**)
  - Observe page limit (7-10 pages, all-inclusive – without Appendix).
  - Language: no spelling/grammar errors, clarity of expression, appropriateness of expression (no slang!), correct usage of terminology.
- Creativity and work beyond the project objectives (**5 points**). This extra includes Scalability and Performance considerations in data engineering. For example, how would the project scale if the dataset size increased tenfold? What optimizations could be applied? Or introducing a new technology/framework to your solution design and implementation that has not been discussed in the course.

Report

The report shall comprise 7-10 pages without the Appendix.

Appendix

Suggested layout:

Abstract, Introduction, and Description (2 pages)
Analysis (4-7 pages, one page per data model, at least 2 for data model to be Implemented).
Conclusion (1 page)
Individual Contribution Statement
Appendix (no length limitation)

Best of luck,
Feras Awaysheh