Project Statement for Milestone 1

CollabInsight

(Tony Cao, Julia Lee, Son Nguyen, Zirun Ye)

Project Report Topics:

1. Problem Statement and Project Goals:
   a. Give a formal description of the project. Include a description of the dataset and its links.

      Description of the project: An Academic Collaboration Analyzer based on NoSQL database designed to uncover meaningful collaboration patterns among scientists and researchers. By utilizing large scale bibliographic datasets, the system can store and manage publication records, author profiles, and other citation attributes within a scalable NoSQL data model.

      Description of the dataset: This is a citation dataset we found on Kaggle due to MAG shutdown, extracted from DBLP, ACM, MAG (Microsoft Academic Graph), and other sources. Each paper is associated with abstract, authors, year, venue, and title.
      Link: https://www.kaggle.com/datasets/mathurinache/citation-network-dataset

   b. Why is the problem you want to address important? What's its application?

      Gaining insight into academic collaboration is essential, given that scientific research is inherently collaborative. By observing co-authorship networks, we can also observe which communities of research are forming, and how these communities grow over time, and which researchers or institutions serve as the key intermediaries. This allows us to better understand the structure and dynamics of knowledge production.

   c. Specify the goal you want to achieve (an end-to-end application with a graphical user interface, and/or a research-based evaluation of existing and new algorithms).

      The goal of this project is to develop an end-to-end data science application that analyzes academic collaboration networks by integrating metadata into a Neo4j graph database, applying community detection and temporal analysis to track the evolution of research communities, using association rule mining to uncover correlations among authors, venues, and topics, and implementing a simple machine learning model for predicting future collaborations; the results will be delivered through an interactive Streamlit dashboard, providing both a graphical user interface for exploration and a research-based evaluation of the applied algorithms.

2. Team Description:

a. Who are the team members? What knowledge and skills do the team have from previous courses, projects, or internships?

Tony Cao:
- Skills: C/C++, C#, Git, GitHub

Julia Lee:
- Skills: Java, Python, MySQL, PostgreSQL, Databases, GenAI, Open API, Spring Framework, JavaScript, jQuery

Son Nguyen:
- Skills: Java (Spring), Python, C#, MySQL, basics DBMS designs, Git, GitHub

Zirun Ye:
- Skills: Java, Python, Linux, Git, GitHub, AI training, and models.

b. What will be each team member's roles and responsibilities in the current project?

- Son Nguyen (Data Engineer): Data collection and preprocessing (OpenAlex API, CSV parsing, JSON/graph schema design)
- Julia Lee (Graph Analyst): Neo4j DB construction and querying, community detection algorithms (Louvain, Label Propagation)
- Zirun Ye (ML Specialist): Association Rule Mining (ARM), link prediction modeling (scikit-learn)
- Tony Cao (Visualization & Cloud Specialist): Network visualization (NetworkX, PyVis, Gephi), Streamlit dashboard, Colab/AuraDB management

3. Dataset Type and Data Model:
   a. A dataset can be described with more than one data model. Evaluate the dataset against **each** of the following data models, reporting its statistics:
      i. Key-Value: Can the data be represented with a key-value data model? If so, how many key-value pairs represent the dataset? How many unique keys? What are the data types for keys and values? Are these basic data types or data structures?

Our team reached consensus that the dataset can be viewed through a key-value data model, albeit with varying emphases on structure and detail. Son suggested using a composite primary key like author ID and paper ID, where value could be stored as a JSON blob or key-value attributes like abstract, year, and references; as a result, keys are typically strings, and a value can be an array of attributes. Julia indicated that each paper is a dictionary, which has around 14–16 key-value pairs and approximately 20 unique keys (e.g., id, authors, title, year, n_citation, doc_type) across the dataset, where keys are strings, and a value can be integers, strings, arrays, and nested structures. Similar to Julia, Tony emphasized that there are roughly 16 unique top-level keys where the values are either basic types (strings, integers) or complex types (arrays, objects). In contrast, Zirun simplified the model to three main keys: author ID, paper ID, and collaboration ID, with values in the form of tuples or arrays of integers and strings. All of the perspectives combined suggest that the dataset can have

the capacity to be represented in a key-value data model, where keys are mostly string identifiers and values are basic data types as well.

ii. Graph: Can the data be represented with a graph data model? If so, how many nodes and edges represent the dataset? How many attributes are there for the nodes/edges? Is it labelled? Directed?

Our collective thinking aligns with the graph data model, and we can collectively wizard it from any perspective. Here's what each of us had to say: Son and Julia said they were envisioning a paper-centric view of the dataset, where each paper is a node and the citations are directed, labelled edges from the citing paper to the cited paper. In their view, you could also add attributes to the nodes such as ID, title, abstract, year, type, kind of field of study, and similarly for attributes on edges, such as citation type, context, and location of reference. Julia said that her estimate, in a sample size of the 56 items she had, the graph representation of the dataset would have 56 nodes, and as many edges as total references. Tony viewed this dataset in a less defined manner. He wanted to include authors, papers, venues, and fields of study as nodes. He envisioned edges and attributed relationships from each node type. For example, would have a relationship (or edge) from author wrote → paper, or paper published in → venue. He also described attributes on edges, such as author and published order, and citation counts. Zirun distinctly described having an author-centric graph with paper nodes connected to author nodes using edges. Author nodes had attributes (e.g., ID, name, and fields of study), and edges represented collaborations or citations: undirected collaboration edges and directed citation edges. All this in mind offered the representation may still avoid misinformation and form of inaccurate meaning. Collectively, we have learned that the graph model is more flexible and labelled than either of the researchers imagined, with both directed and non-directed edges. Also, it is capable of representing in the form of publication, authorship and collaboration networks embedded in the dataset.

iii. Document: Can the data be represented with a document data model? If so, how many documents represent your dataset? How many elements / sub-elements does each document has? What are the attributes?

As a group, we agreed that we could use a document data model for our dataset, with slight differences in how broadly we can define our granularity. Son, and Julia, and Tony envisioned modeling a single document for each paper, where the high-level attributes included identifiers like an ID, title, abstract, DOI, publication year, and publisher, while the sub-elements would model the nested structures like arrays of authors (e.g. with details like authors names, affiliations), references and citations. Julia noted that in a sample that is relatively small as 56 records, this would yield 56 documents, with each containing around 14–16 fields corresponding to bibliographic metadata, numerical indicators, textual content, and

categorical descriptors. In fact, this would suit a semi-structured system like MongoDB. Tony noted the scalability of the approach for very large datasets, estimating around 4.89 million documents in total, and with 16 top-level fields and numerous nested sub-elements for author, venues, and citations. Zirun, suggested an alternative view, conceptualizing authors as documents, where each author document would contain author attributes like ID, name, and field of study as attributes, and use sub-elements to represent the papers they were authors of. These views as a group show that the document model is quite flexible as it allows for either publishing centered, or author centered representations for the dataset we are working with.

iv. Other: Can the data be represented by any other non-relational data model? If so, describe the data model and the applicable statistics on your dataset.

Our team also investigated if the dataset could be represented through other non-relational data models other than document or graph models. Son and Julia suggested the column-family (or wide-column) data model where each paper ID functions as the row key and title, year, publisher, and citation would be in flexible columns, and repeated fields like authors could be represented as wide columns or collections. Compared to traditional normalized relational data models, the wide-column model allows for scalability, retrieval of attributes by ID, and is more suited to analytical queries involving publication metadata. Zirun also saw the possibilities of wide column representing authors as rows with attributes in columns. Tony similarly proposed a key-value data model, where the unique ID of the publication (including the DOI or URI) would be the key and the entire JSON object of the paper would be the value. This option is simple, efficient for fast lookups but does not delineate the internal nested structure of the documents. Regardless, these two competing alternatives demonstrate that the dataset could be represented in either a column-family or key-value model depending on considerations pertaining to analytics and scalability versus complex data processing and variety.

b. Describe the data model your team has chosen to represent the dataset? Justify why the data model is an appropriate one for the dataset. Note: You should be using a non-relational data model for this project.

Our group contemplated several approaches before finally arriving at a focus on non-relational data models that were appropriate for the structure of our dataset. Three of us (Son, Julia, and Tony) decided on the document data model, since the dataset was already in JSON format and naturally lent itself to the format of storing each of the papers as a self-contained document. Document databases have the capability to accommodate flexible schemas, support nested attributes such as authors and citations, and discourage complex joins, allowing faster queries to retrieve entire records from the database. Zirun, on the other hand, proposed a graph data model, emphasizing the graph model's superiority with representing

collaborations that demonstrate how authors, papers, and, sometimes, organizations can be nodes in the represented network and relationships (e.g., co-authorships, citations) are edges connecting the nodes. Both data models influence important aspects of the dataset: the document model is conducive for managing heterogeneous attributes at the publication level, and the graph model is superior for examining relational structures, such as collaborations and citations.

4. Tools:
   a. What database tools do you plan to use?
      We plan to use Neo4j, a graph-based NoSQL database, to store and query the relationships among authors, papers, venues, and institutions. Neo4j's Cypher query language makes it straightforward to explore co-authorship patterns and research communities. Optionally, we may use MongoDB (NoSQL) for raw metadata storage before loading the data into Neo4j, but the primary database for analysis will be Neo4j.
   b. What data processing tools do you plan to use?
      Our data processing will be primarily conducted in Python. We will use pandas for data cleaning and preprocessing, networkx for graph analysis on smaller subgraphs, mlxtend for association rule mining (Apriori and FP-growth), and scikit-learn for link prediction models such as Logistic Regression or Random Forest. Additionally, we will use Python drivers (py2neo or the official Neo4j driver) to integrate Neo4j with our analysis workflow.
   c. What cloud resources, if any, do you plan to use?
      We do not require heavy cloud infrastructure, since most computations can be performed locally. For collaboration, we plan to use GitHub for version control and code sharing, and Microsoft OneDrive for storing datasets, reports, and intermediate results. These resources will primarily support teamwork and accessibility, while the core analysis will still be run on local machines.