MINOR PROJECT-1

SYNOPSIS

B.Tech (CSE) Semester V

Knowledge Graph based Multimodal E-learning Recommender System, Online Courses Data Analysis and Visualisation

MADE BY:

AARUSH GUPTA - 22103030

SAKSHAM PUROHIT - 22103017

SHIVAPRASAD ARUNKUMAR FARALE - 22103012

Minor Project Supervisor: Dr. Ankita Verma

Panel members: Dr. Meenal Jain, Dr. Mradula Sharma

Multimodal E-learning Recommender System

1. Introduction

As online education platforms continue to grow, users are presented with a vast array of courses across diverse domains. However, identifying the right course based on skills, interests, and career goals can be challenging. Traditional recommendation systems often fall short by not capturing the intricate relationships between courses, instructors, and skills.

This project, developed by a team of three, aims to build a Multimodal E-learning Recommender System using graph-based insights from Neo4j and machine learning algorithms to provide dynamic and personalized course suggestions. The system will efficiently represent and analyze the relationships between entities such as courses, students, instructors, and learning materials.

Apart from building the recommender system, we will also use Python data analysis and visualization techniques to better understand course distribution, student engagement, and various key attributes (e.g., ratings, categories). These insights will further improve recommendation accuracy.

2. Motivation

With the sheer number of online courses available, learners often struggle to find the ones that best align with their needs. From our own experience, we have frequently encountered challenges in selecting courses that truly fit our learning goals and skills. Existing recommendation systems tend to rely on simple methods like collaborative filtering or content-based approaches, which overlook the complexity of relationships between different course entities (e.g., prerequisites, instructors, and skills).

This project aims to leverage the power of multimodal graphs to better capture these relationships, providing more accurate, relevant, and dynamic course recommendations that adapt to the user's learning history, preferences, and goals. By addressing these limitations, the system will offer a more personalized and insightful recommendation experience.

3. Project Objectives

The key objectives of this project include:

1. Flexible Schema Design: The system will be flexible enough to work with varied data sources, such as course content, user interactions, and instructor profiles.
2. Graph-Based Database Design: Implement a flexible Neo4j graph database schema that captures relationships between students, courses, skills, instructors, categories, and other relevant attributes.
3. Hybrid Recommendation Model: Develop a hybrid recommendation system combining collaborative filtering with graph-based insights, allowing for a more personalized recommendation process.
4. Personalized Recommendations: Provide course suggestions by considering user preferences like skill level, language, and category preferences, as well as insights derived from the graph database.
5. Interactive User Interface: Build a user-friendly platform where learners can interactively receive course suggestions based on real-time data from the graph database and machine learning model (Future Aspect).
6. Incorporating Data Analysis and Visualization: We will use Python-based data analysis tools to explore key trends, such as most popular courses and skills gained.

4. Problem Statement

Choosing appropriate courses from the vast pool of available content is a major challenge for online learners. Existing systems do not account for deeper relationships between course attributes, such as skills taught, instructors, and course categories, leading to limited personalization and relevance in recommendations. Additionally, a flexible system that can adapt to different datasets and schema structures is essential to accommodate the evolving nature of e-learning platforms.

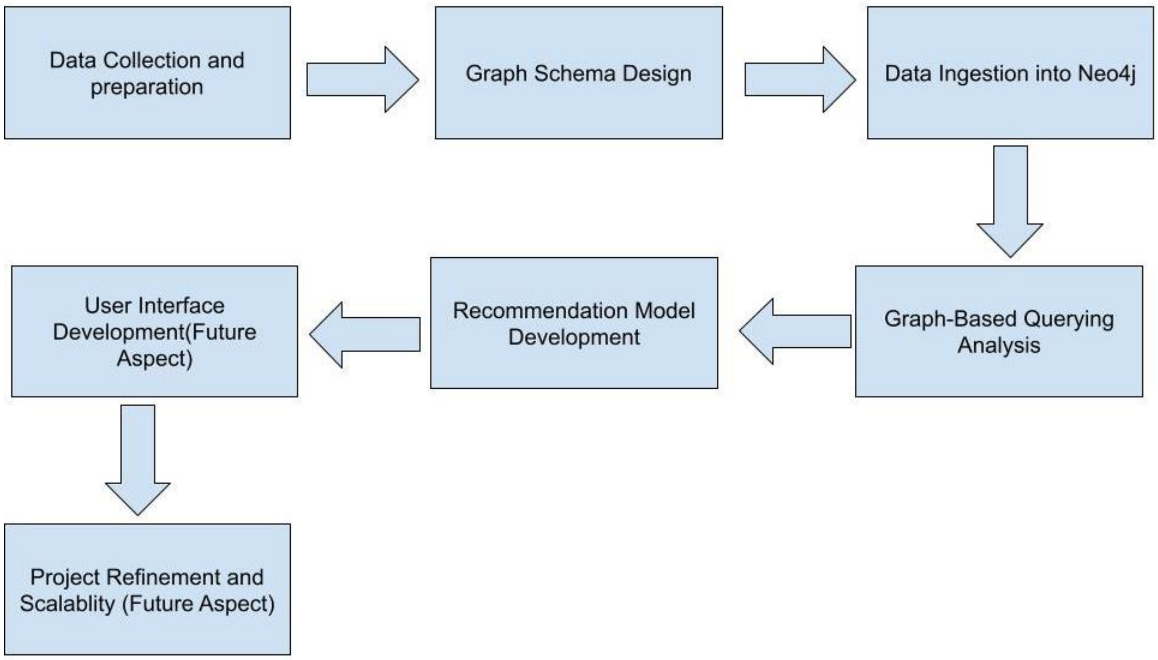
5. Proposed Solution

The Multimodal E-learning Recommender System will address the shortcomings of traditional methods by utilizing the following key features:

1. Flexible Schema Design: The system will be adaptable, allowing for integration with different datasets and schema structures once they are finalized. This flexibility will enable the system to work with various data sources, including course content, user preferences, and instructor profiles.
2. Graph-Based Insights with Neo4j: The graph database will model relationships between entities such as courses, students, skills, and instructors, allowing the system to capture and analyze complex interactions. For instance, the system could prioritize courses with popular instructors or those essential for certain skills.
3. Hybrid Recommendation Model: The recommendation engine will combine traditional algorithms (collaborative filtering) with graph-based relationships. This

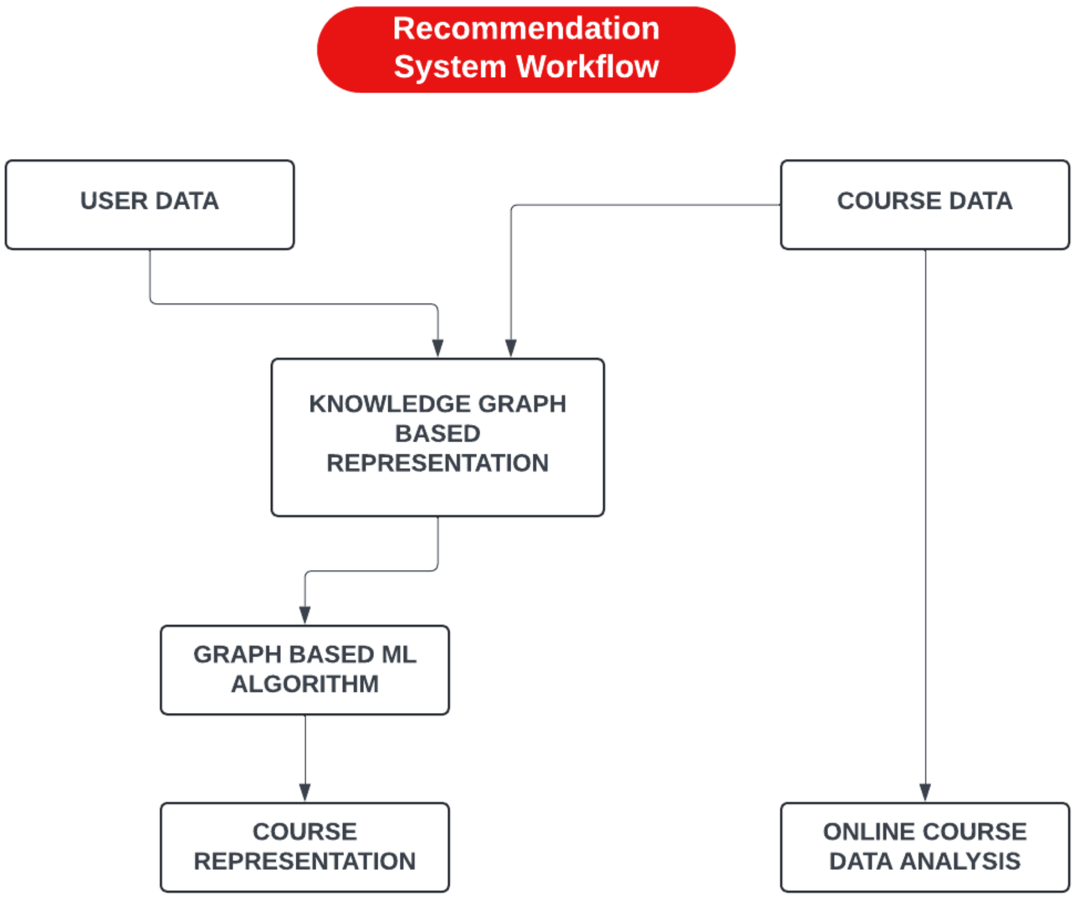
hybrid model will enable the system to suggest courses based on both user-item interactions and the contextual relationships between courses, skills, and instructors.
4. Real-Time, Dynamic Recommendations: The system will offer personalized course suggestions in real-time, using graph-based queries and machine learning algorithms to adapt to the learner's evolving preferences and skills.

5. Workflow



1. Data Collection & Preparation: Collect course data, including content, instructor profiles, user interactions, and skill requirements. Clean and structure the data for efficient use in graph modeling and machine learning.
2. Graph Schema Design: Define a schema in Neo4j with nodes (e.g., Course, Instructor, Skill) and relationships (e.g., TAUGHT_BY, REQUIRES_SKILL). Ensure its flexible for future data expansions.
3. Data Ingestion: Import the cleaned data into Neo4j, maintaining the defined schema and relationships to enable effective querying and visualization.
4. Querying & Analysis: Use Cypher queries to extract insights like popular courses or skills, setting the foundation for the recommendation model.
5. Recommendation Model: Build a hybrid recommendation engine combining collaborative filtering with graph-based insights for personalized course suggestions. Optimize the model for accuracy.
6. User Interface: Develop an interactive web UI that dynamically displays personalized course recommendations and allows users to filter by skills, categories, or instructors.

7. Testing & Evaluation: Assess the recommendation engine using metrics like precision, recall, and user engagement. Gather user feedback to refine the system.
8. Refinement & Scalability: Improve the model based on feedback, focusing on scalability and future-proofing. Plan for real-time feedback loops and additional data integration to enhance recommendations.

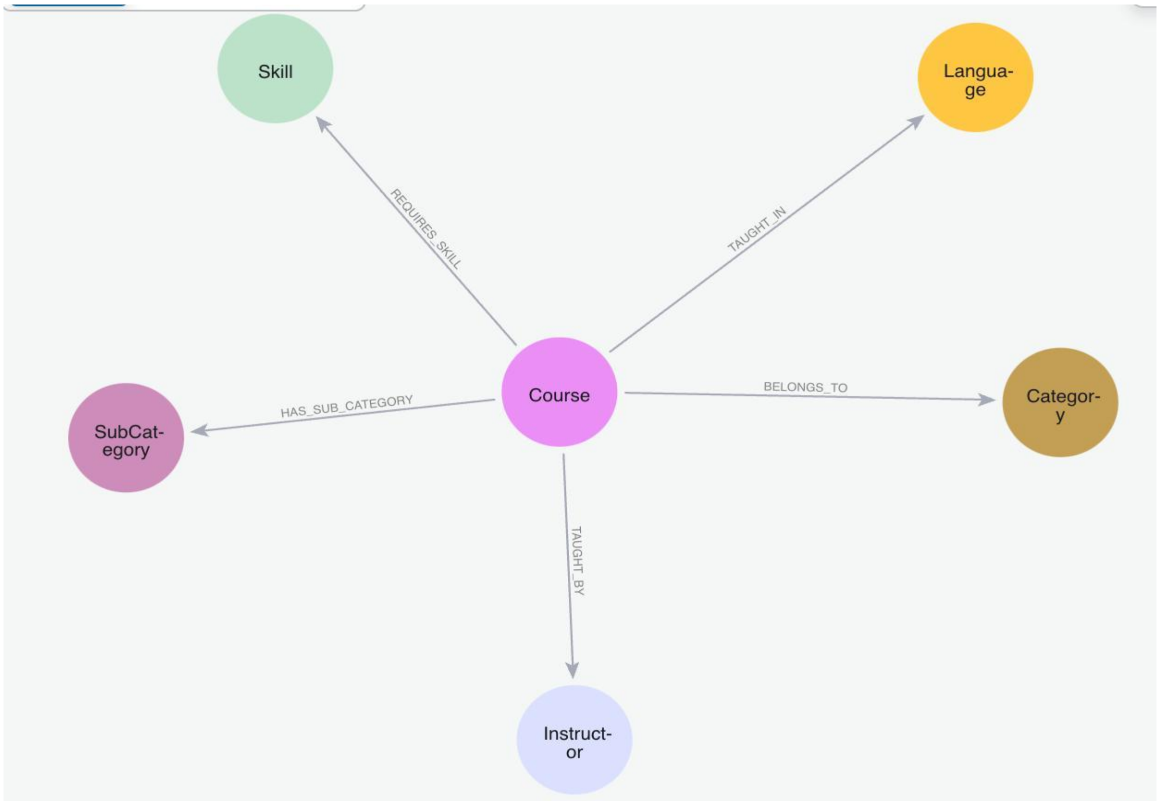


6. Evaluation and Future Refinement

The system will be evaluated using metrics such as user engagement, course completion rates, and overall satisfaction. Feedback from users will be used to continuously refine the recommendation engine, ensuring that the platform remains effective, scalable, and responsive to the learning needs of users.

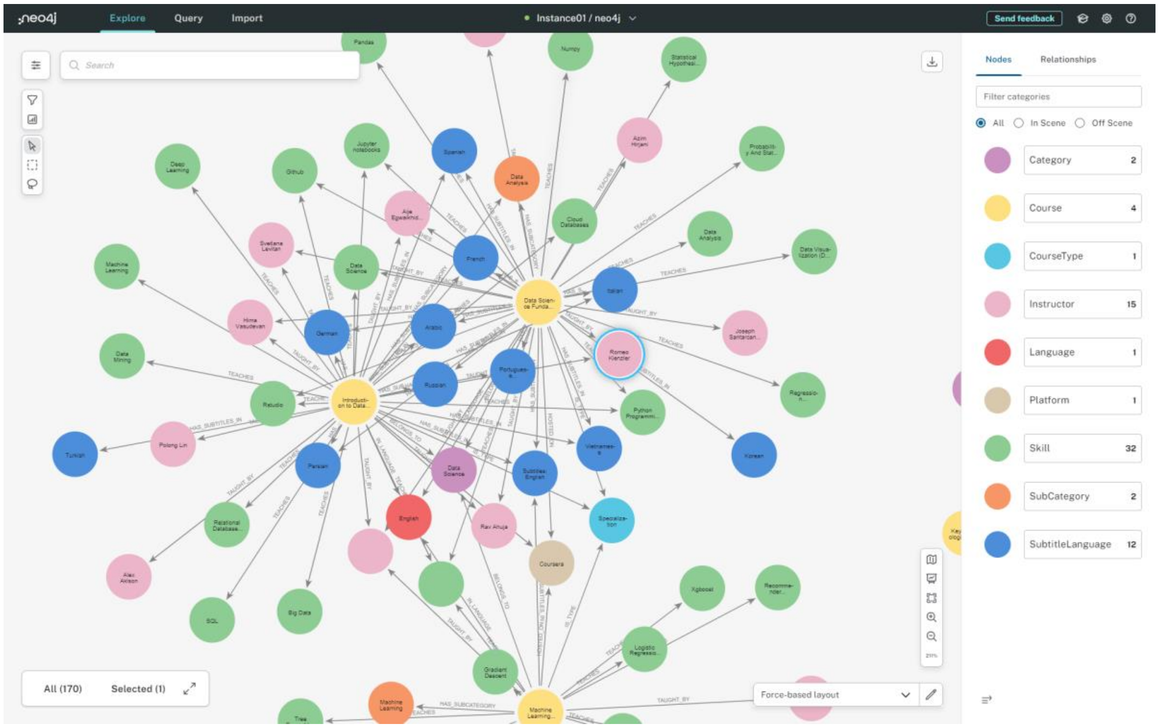
By incorporating both graph-based data insights and traditional recommendation techniques, this multimodal system could provide users with a more holistic and engaging course recommendation experience, enhancing the way online learners discover and interact with educational content.

Tentative Data Schema of dataset for Multimodal E-learning Recommender System

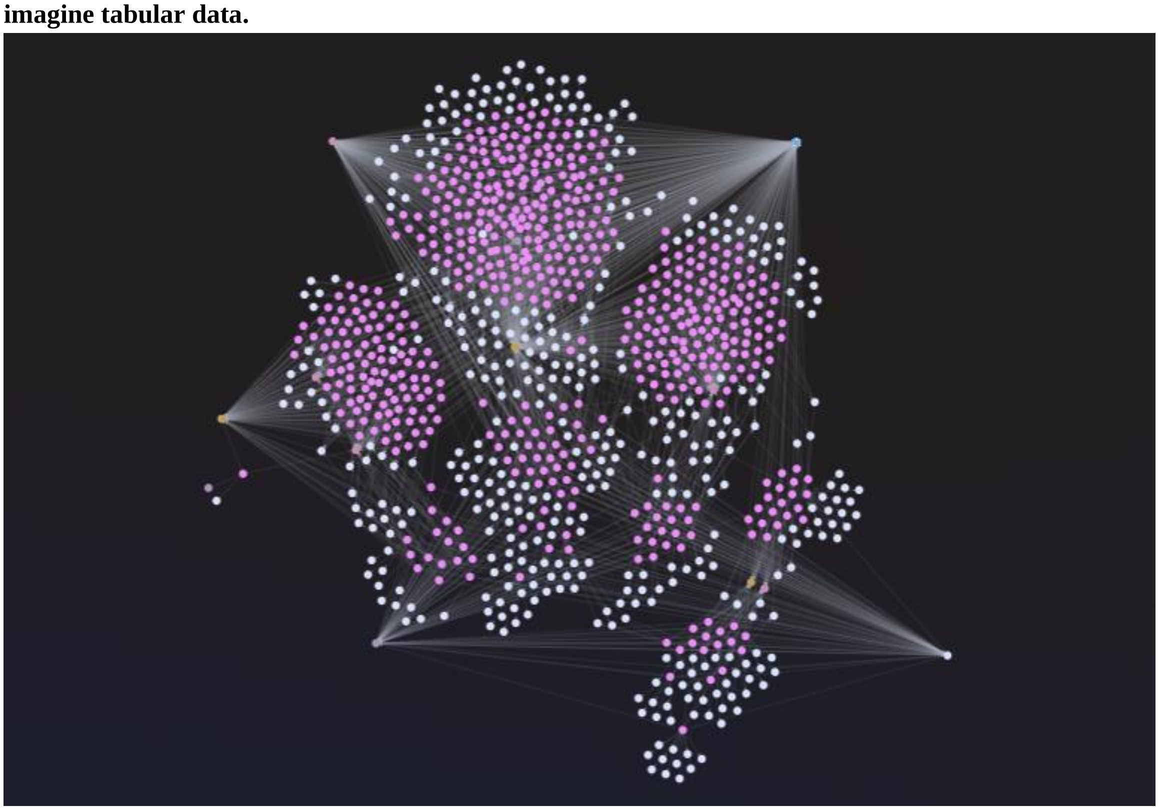


Nodes
[
(:Skill {name: "Skill", indexes: [], constraints: []}),
(:Category {name: "Category", indexes: [], constraints: []}),
(:Language {name: "Language", indexes: [], constraints: []}),
(:Instructor {name: "Instructor", indexes: [], constraints: []}),
(:SubCategory {name: "SubCategory", indexes: [], constraints: []}),
(:Course {name: "Course", indexes: [], constraints: []})
]

Relationships
[
[:TAUGHT_IN {name: "TAUGHT_IN"}],
[:TAUGHT_BY {name: "TAUGHT_BY"}],
[:REQUIRES_SKILL {name: "REQUIRES_SKILL"}],
[:BELONGS_TO {name: "BELONGS_TO"}],
[:HAS_SUB_CATEGORY {name: "HAS_SUB_CATEGORY"}]
]



This is the realisation of the schema showcased above. It helps understand the vastness of data and visualise how nodes within the schema are connected and provides life to a mundane and difficult to imagine tabular data.



7. References

https://dl.acm.org/doi/10.1145/3535101
https://neo4j.com/docs/
https://www.oaepublish.com/articles/jsegc.2020.06