ECE 1786 - CampusCcompanion

Final Report

Word Count: 1982

Jijun Chi	1010253887
Joshua Abraham	1002514317

Permissions

Jijun Chi

- permission to post video: yes
- permission to post final report: yes
- permission to post source code: yes

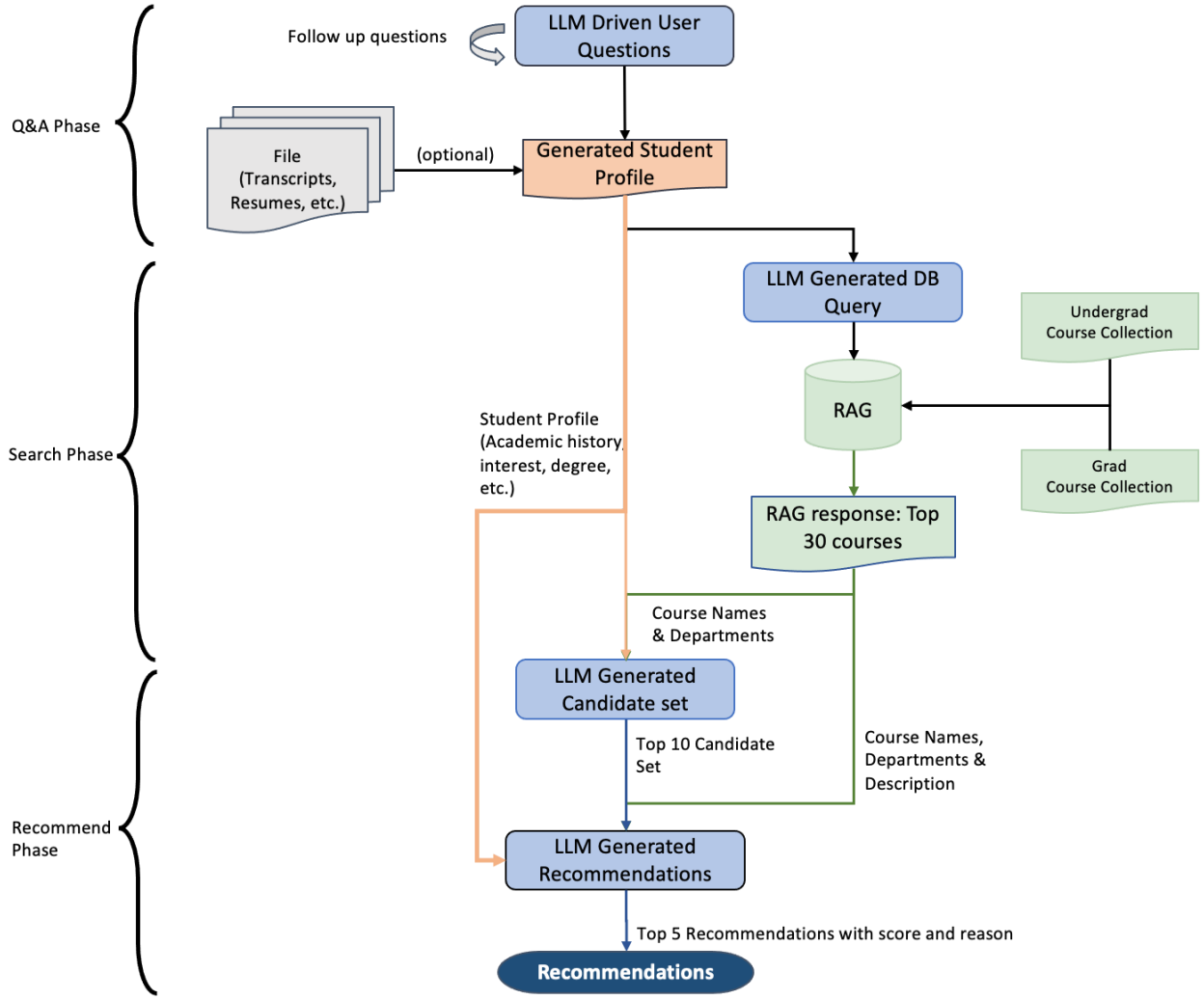
Joshua Abraham

- permission to post video: yes
- permission to post final report: yes
- permission to post source code: yes

Introduction

Navigating course selection at UofT presents challenges due to a system that lacks customization for individual interests, academic goals, and abilities. The abundance of options compounds stress, leading to suboptimal choices and setbacks. CampusCompanion addresses this by providing tailored course recommendations based on an in-depth understanding of UofT's unique data, including department descriptions, current courses, and prerequisites. The system considers students' academic history, achievements, interests, industry experience, research, and future goals. Machine learning enhances adaptability to diverse courses and student experiences. Unlike human advisors, the AI system is available 24/7, offering a comprehensive and adaptable approach to support students in their academic journey.

Illustration



Background & Related Work

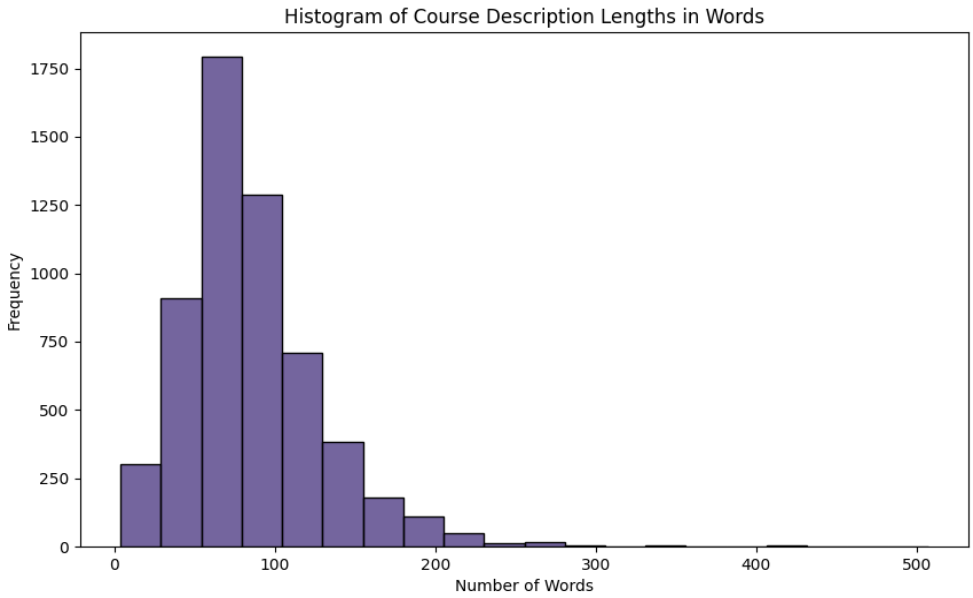
Recommender systems are evolving with the integration of large language models (LLMs) to address challenges like limited interactivity, poor explainability, and the 'cold start' problem. Research, exemplified by Wang's Zero-Shot Next-Item Recommendation (NIR) [1], demonstrates the effectiveness of LLMs in generating recommendations for subsequent movies, even surpassing models trained on full datasets. Another approach, GenRec [2] by Ge et al., employs a generative recommendation LLM that reformats user interaction sequences during training, allowing the model to predict the next item the user may engage with.

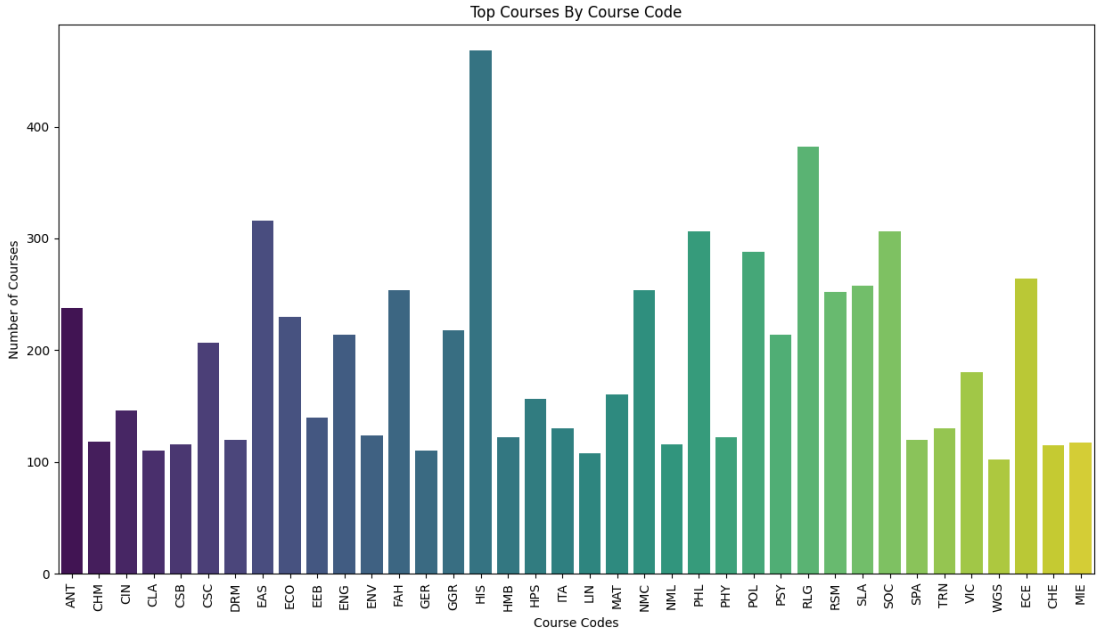
In course recommendation, challenges persist, including a scarcity of publicly accessible course selection datasets, complicating the extraction of student preferences. Moreover, the dynamic nature of course offerings requires a mechanism for LLMs to access the most recent course information. Our work on CampusCompanion addresses these challenges through retrieval augmented generation, in-context learning, and the Chain-of-thought process.

Data and Data Processing

There are two forms of data used: course description data within the vector database and a hold-out test set.

To collect course description data, we parsed information from three sources: the Undergraduate Engineering Academic Calendar, Undergraduate Art & Science Course Listing, and Graduate Engineering Course Listing. The process involved extracting course codes, names, descriptions, and prerequisites (if mentioned), with each course separated by a new line character. Additionally, course codes were mapped to their respective department, including a description of the department, and a year of study associated with each course. In total, we parsed 5768 course descriptions, enhancing our dataset with relevant context for the LLM.





Sample Course Description Parsing

```
RSM220H1 - Intermediate Financial Accounting I
Department Code: RSM
Department: Rotman School of Management, Commerce, and Business
Year of Study: 2
Foundations of financial reporting and analysis in Canada. Financial
accounting topics are covered at an intermediate level, including both
conceptual and technical aspects. Not eligible for CR/NCR option.
```

The second type of data was a hold-out test set, used for testing and validating system correctness. We crafted student responses based on student personas, aiming to cover a wide variety of students. Along with these handcrafted tests, synthetic student profiles were also generated for quantitative analysis, as further discussed in Section 7.

Architecture and Software

Prior to using the system, all the course data extracted in the previous section must be indexed into the RAG database, in our case ChromaDB.

The system architecture, depicted in the diagram in section 2, comprises three phases.

The Q&A Phase

LangChain-powered, our system facilitates LLM-driven conversations, adapting prompt context dynamically. It extracts information interactively and can also parse data from PDF files, like Transcripts and Resumes, all integrated into the student profile.

Summarized Q&A Prompt:

> *Act as an academic advisor at the UofT, engage in a non-directive assessment conversation with a student. Explore their degree program, department, interests, academic goals, research, volunteer, industry experience, and courses. If you gathered enough information for assessment format in the described output.*
> *Otherwise ask a follow up question to the student.*

Full Prompt

Summarized Transcript Extraction Prompt:

> *Extract the course name from the provided text. Output courses that are directly related to the {{field}}.*

Full Prompt

The Search Phase

This phase condenses the student profile into a search-optimized query, used to search ChromaDB for the top 30 results. The LLM prompt uses chain-of-thought reasoning to create a query that focuses on the key words, departments, and filters to either the undergraduate or graduate course collection. This prompt query will then be used as the search terms against the ChromaDB course collection.

Summarized Search Generation Prompt:

> *Create a RAG search query for courses with similar content, skills, interests, academic goals, and completed courses, incorporating relevant terms. Include the department code in the output.*
> *Input: I am in History …*
> *Output: HIS, Roman History, Ancient Civilizations, undergraduate …*

Full Prompt

The Recommendation Phase

This phase involves two key steps: candidate selection and final recommendation. First, from the initial 30 courses in the search result, the top 10 are chosen based on course names, department, and year of study, using the complete student profile. This approach helps narrow down options without requiring detailed descriptions for all 30, ensuring compatibility with the context window limitations of GPT-4. The final recommendation then takes these 10 candidates along with their full course description for final recommendation.

Summarized Candidate Prompt:

> *Advise students at the UofT by selecting {candid_size} courses from a provided list, considering alignment with personal interests, relevance to goals, and suitability to experience. Output in the described format.*

Full Prompt

Following the initial stage, the system offers a comprehensive recommendation. It considers the candidate set of courses and the entire student profile, providing complete descriptions for the selected 10 courses. The top 5 courses are presented in sorted order based on scores (0-100) reflecting relevance to the student profile, with justifications for their selection included.

Summarized Recommendation Prompt:

> *As a UofT advisor, choose five recommended courses based on student and course information. Score these courses from 0 to 100, considering alignment with the student's interests, academic goals, and experience. Output a sorted list of courses and scores. Exclude courses already taken or with similar names. If recommendations are possible, display "Success!" with the list; otherwise, show "Fail!" with additional information.*

Full Prompt

Baseline Model or Comparison

The report focuses on evaluating the Q&A and Recommendation Phases of CampusCompanion. Testing involved GPT-generated inputs simulating student profiles, including program details, interests, goals, and courses. The evaluation assessed summaries from the Q&A Phase and the course recommendations list in the Recommendation Phase.

Q&A Evaluation

In the Q&A Phase, we used a 'Consistency Score,' a systematic metric generated by GPT, to assess the relevance of CampusCompanion's summaries to the original student profiles. The following graph illustrates the manually-set criteria for this score.

Consistency Score criteria

- Score 1: The answer is completely unrelated to the reference.
- Score 3: The answer has minor relevance but does not align with the reference.
- Score 5: The answer has moderate relevance but contains inaccuracies.
- Score 7: The answer aligns with the reference but has minor errors or omissions.
- Score 10: The answer is completely accurate and aligns perfectly with the reference.

Recommendation Evaluation

In the Recommendation Phase, we first employed 'Recall' to reflect the proportion of courses taken that were predicted by the model:

$$Recall = \frac{\#of\ courses\ taken\ predicted\ by\ model}{\#of\ courses\ actually\ taken\ by\ the\ student}.$$

Considering students' varied influences and the potential inaccuracies in synthesized course data, relying solely on the Recall metric may not fully gauge our model's effectiveness. To address this, we introduced the 'Hit Ratio' to assess the suitability of recommended courses:

$$HitRatio = \frac{\#of\ recommended\ courses\ suitable\ for\ the\ student}{\#of\ recommended\ courses},$$

and we manually assess whether a course is suitable for a student.

The Hit Ratio helps us determine the suitability of the recommended courses for a student, without their ranking order. To address this, we calculate the Normalized Discounted Cumulative Gain (NDCG), which evaluates the ranking quality within the recommended course list. Using a graded relevance scale of courses in the recommendation set, Discounted Cumulative Gain(DCG) measures the usefulness, or gain, of a course based on its position in the result list:

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$$

Where rel_i is the graded relevance of the result at position i. To compute NDCG, the gain is accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks:

$$\mathrm{NDCG_p} = \frac{DCG_p}{IDCG_p},$$

where IDCG is ideal discounted cumulative gain,

$$\mathrm{IDCG_p} = \sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)},$$

and REL_p represents the list of relevant courses (ordered by their relevance) in the corpus up to position p.

The average NDCG value across all recommendations measures our model's ranking performance. Ideally, NDCG approaches 1.0, indicating perfect ranking.

Qualitative System Evaluation

We also assessed the recommendation quality through a manual rating system. We evaluated 30 different recommendation results with human input, uncovering both deficiencies and strengths in our model.

Quantitative Results

Q&A Results

We used 20 synthetic student profiles, for undergraduate to graduate students in a wide array of disciplines. Based on the predefined criteria, it's evident that the summaries generated by our model, incorporating GPT-4, closely align with the original student profiles.

	Consistency Score
gpt-3.5-turbo	6.94
gpt-4-1106-preview	8.33

Recommendation Results

For this, we evaluated the results of the Recommend Phase. Our model recommends 5 courses to 9 students who will take 3 courses in their synthetic profiles. Recall means the proportion of actual positive courses that the model correctly identified, Hit Ratio@5 means the number of suitable courses in the 5-course list and NDCG@5 is the ranking score for this list.
Our model, integrating GPT-4, achieves a Recall rate of 37.7%, indicating that, on average, it successfully predicts one course per student. The Hit Ratio suggests that a majority of the recommended courses are well-suited for students. Furthermore, an NDCG@5 greater than 0.9 demonstrates the model's strong capability in accurately ranking the relevance of courses.

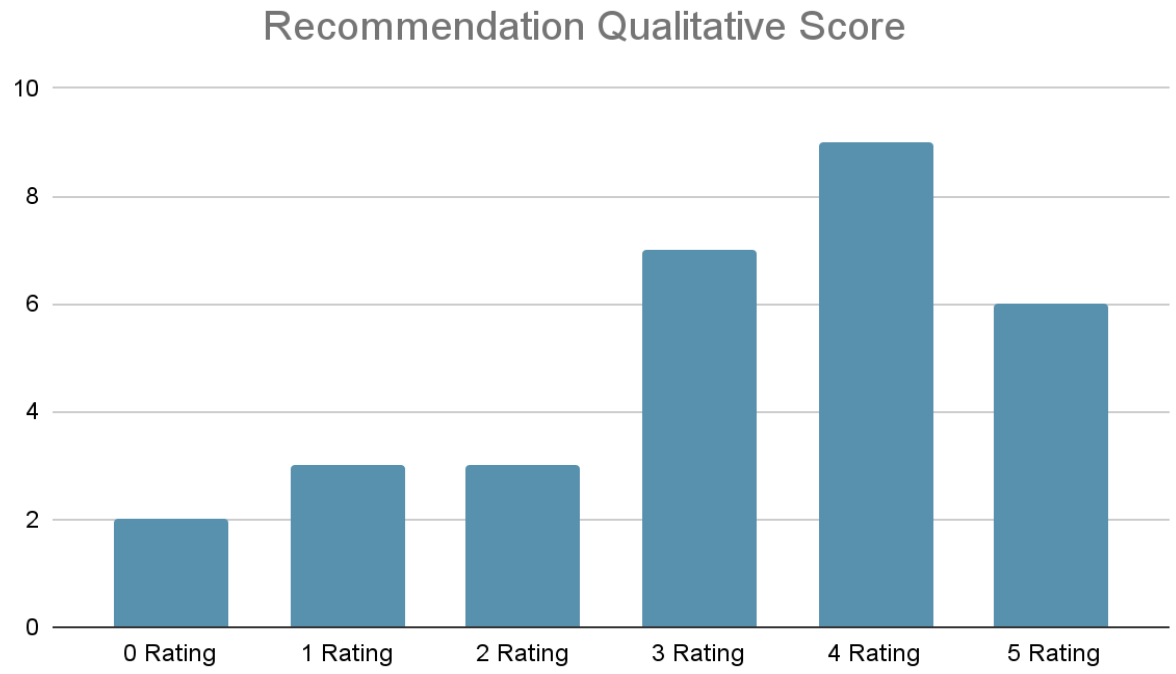
	Recall	Hit Ratio@5	NDCG@5
gpt-3.5-turbo	26.3%	67.5%	0.93
gpt-4-1106-preview	37.7%	87.5%	0.96

Qualitative Results

For qualitative analysis, we employed a manual rating system that gauged the recommendation lists' quality on a rubric from 0 to 5, with human input. Given the subjective nature of recommendations, this manual evaluation provides a nuanced approach to evaluating their quality. The scale was defined as:

	1	2	3	4	5
Recommendations are logical and based on the student's interests	For no courses	For few courses	For some courses	For most courses	For all courses
Year of study is taken into account	For no courses	For few courses	For some courses	For most courses	For all courses
Program of study is taken into account	For no courses	For few courses	For some courses	For most courses	For all courses
Scoring and reasoning given for a recommendation is logical	For no courses	For few courses	For some courses	For most courses	For all courses
Courses previously taken are not recommended	For no courses	For few courses	For some courses	For most courses	For all courses

* 0 means the system fails to give a recommendation



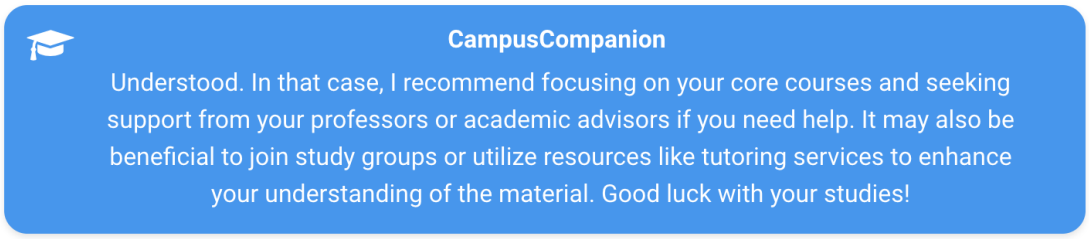
Below are some personas, the final recommendations, the qualitative score , and some analysis.

Persona 1:

The Impatient Student: Continually demands recommendations and does not provide any context to work from.

This is challenging because there is very little information to go off of.

Recommendation:



Qualitative Score: 0

Analysis: Though this response does sound reasonable for someone who provides no context, this does not go through the full pipeline, and is actually the Q&A section providing answers without going onto the Search or Recommendation prompts, which explicitly goes against the prompt's rule of not providing recommendations.

Persona 2:

The Aspiring Teacher: An upper year English student, looking to become a teacher, and wants to improve their introductory Math skills.

This is challenging because the system needs to not focus on degree of study, interests, year of study, or previous courses. It must instead focus on the more nuanced objective of the student.

Recommendation:

CampusCompanion

Code	Name	Score	Reason
EDS363H1	Teaching Math: The JUMP Approach	95	help students develop deeper levels of confidence and understanding in mathematics
EDS360Y1	Education Internship	85	gain practical experience in an educational environment
EDS360H1	Education Internship	80	practical experience in an educational environment
EDS265H1	Introduction to Teaching	75	builds understanding of teaching as professional practice
MAT296H1	Independent Reading in Mathematics	70	allows for independent study in mathematics

Qualitative Score: 3

Analysis: The system focuses on the Education stream instead of Math, so these might not be the most ideal courses. However, EDS363H1 and EDS265 are also great choices considering the educational focus and still having math teaching involved.

Persona 3:

The ECE Grad Student: A graduate ECE student who is interested in learning about advanced topics in NLP.

This is a standard use case.

Recommendation:

Code	Name	Score	Reason
ECE1786H	Creative Applications of Natural Language Processing	95	This course is directly aligned with your interest in NLP and large language models, and it will equip you with the skills to build NLP applications, which is your academic goal.
ECE1779H	Introduction to Cloud Computing	85	Given your experience in web application development and your interest in building applications, this course will provide you with necessary cloud computing skills that are increasingly important in deploying scalable NLP applications.
ECE1513H	Introduction to Machine Learning	75	Although you have taken deep learning and neural network courses, this course provides a strong foundation in machine learning that is beneficial for understanding the underpinnings of NLP models.
ECE1784H	Trustworthy Machine Learning	70	Trustworthy ML is becoming increasingly important with the rise of NLP models. Your experience with deep learning and transformers will be enhanced by understanding the security and privacy aspects of ML.
ECE1508H	Special Topics in Communications: Graphs, Codes, and Inference	65	Your research in graph neural networks suggests you would benefit from advanced topics in graphs and inference, which may provide new perspectives or techniques applicable to NLP.

Qualitative Score: 5

Analysis: This is a perfect score because it offers all courses within the correct department and area of interest, and correctly recognizes ECE1786 as the optimal choice for someone interested in learning about advanced NLP topics.

Discussion and Learnings

Based on the results, our system meets expectations by accurately offering recommendations in most conversations.

The following insights were made:

Initially, using the RAG queries to segment undergraduate and graduate courses resulted in cross-level recommendations, but additional data processing solved this by creating distinct sets for both groups.

Another lesson was refining results for courses from closely related departments (e.g., CSC and ECE) to avoid misselection of departments. To do this, the Q&A phase needed to pull out the department and that always needed to be a search parameter.

Understanding complex course codes, like RSM220H1, required context improvement by incorporating department names and study year when doing data processing. Without context, it would be difficult for an LLM to understand that this is a business course for second year students.

Orvall, we identified the pivotal role of RAG results in response quality, emphasizing the importance of preprocessing for search enhancement. In the future, employing multiple search layers could mitigate the impact of a poor search, allowing independent tuning and enhanced recommendation precision, especially in nuanced cases. For example, when a student wishes to take a course outside of their degree or department, being able to not search by department would be a huge benefit.

Individual Contributions

Overall, the work completed has been very equal. Both of us have taken on different aspects of the project, focusing on areas that were interesting to us, while still ensuring a good balance of work.

Work Item	**Completed By**
Undergraduate Course data collection and formatting	Joshua
Graduate Course data collection and formatting	Jijun
Vector Database Setup	Joshua
LLM guided question prompt	Jijun
LLM vector search query prompt	Joshua
LLM recommendation prompt	Jijun
Test Generator	Jijun
Quantative analysis(Q&A evaluation, Recommendation evaluation)	Jijun
Quantitative analysis(System Evaluation)	Joshua
Qualitative test	Joshua
Server with Django	Jijun
UI with Vue3	Joshua
Report Writing	Both

References

[1] L. Wang and E.-P. Lim, "Zero-Shot Next-Item Recommendation using Large Pretrained Language Models," Apr. 6, 2023, arXiv.org, https://doi.org/10.48550/arXiv.2304.03153.

[2] J. Ji, Z. Li, S. Xu, W. Hua, Y. Ge, J. Tan, and Y. Zhang, "GenRec: Large Language Model for Generative Recommendation," 2023, arXiv (Cornell University), https://doi.org/10.48550/arxiv.2307.00457.