A Recommender System Architecture for University Curriculum Advising

Zerui Ma[1], **Michael Hahsler**[1], **Peter Moore**[2]

[1]Department of Computer Science, Southern Methodist University
[2]Department of Mathematics, Southern Methodist University
jerryma@smu.edu, mhahsler@smu.edu, pmoore@smu.edu

Abstract

Effective academic advising plays a crucial role in student success, yet universities face challenges in optimizing advising processes and course enrollment. This task is complicated by the fact that several graduation requirements have to be met while also taking the students' interests into account. Academic advising has historically been performed by a skilled human adviser. Universities can optimize course planning and help students make informed decisions about their academic path with recommender systems. This case study develops a goal-based agent recommender system based on a large language model tailored to undergraduate students, depending on curriculum requirements, prerequisite dependencies, and student preferences. The developed recommendation system helps universities increase student advising efficiency and create more intuitive and student-centric curricula. We show how to structure and process complex curriculum data to create an algorithm-ready environment, simplifying the relationships between degree requirements and course offerings. This study evaluates multiple algorithms based on recommendation accuracy, computational efficiency, and their ability to meet degree requirements while fostering academic engagement. By streamlining course selection and exploring possible degree paths, the system may also help students graduate on time and navigate complex curricula. This system also collects important metrics to accurately predict student enrollment for classes, enabling college departments to plan their course offerings better. The system poses a significant benefit to university advising offices by reducing advisor workloads and encouraging student engagement, advancing the academic achievement of the entire student body.

1 Introduction

University students, especially students in the early stages of their academic journey, often have trouble accessing academic advising. In a student-based survey (Flaherty 2023) among more than 120 institutions, only 55 percent of students reported being advised on required coursework for graduation. Only 57 percent of students intending to graduate reported receiving guidance on required courses and course sequences needed for graduation via the advising process, with disparities among certain demographics. Specifically in an article by NACADA, researchers discovered that underprivileged demographics tend to lack academic advising resources during COVID lockdowns (Soria 2023). Lack of advising is not only persistent in universities but also disparages underprivileged students.

The growing availability of educational data and advancements in information systems have paved the way for new methodologies to enhance teaching, learning, and advising processes. Recommender systems have emerged as powerful tools to address these challenges, offering personalized and data-driven solutions. Prior research highlights the potential of recommender systems to revolutionize academic advising by leveraging machine learning (ML), graph analysis, and other computational techniques.

For example, a study conducted at the University of Dubai introduced a recommender system for academic advisors and students that analyzes student records to develop personalized study plans spanning multiple semesters (Atalla et al. 2023). This system incorporates graph theory, performance modeling, and explainable ML algorithms to ensure its recommendations align with curriculum rules and individual preferences. By leveraging network analysis, the system systematically evaluates the relevance of a student's study plan, offering insights into compliance with degree requirements. This study places precedence on the importance of explainability and computational efficiency in recommender system design. The integration of graph theory-based approaches allows for a systematic representation of curriculum requirements, simplifying the complexity of degree planning. Such innovations set a benchmark for developing advanced recommender systems tailored to academic advising.

Building on these findings, our research aims to extend the capabilities of recommender systems in academic advising by focusing on undergraduate course planning at Southern Methodist University. While the University of Dubai's model emphasizes single-program multi-semester planning and network analysis, our approach integrates multi-degree curriculum requirements, prerequisite dependencies, and student preferences into a unified system. Furthermore, our research evaluates multiple Large Language Models (LLMs) to identify optimal solutions for the precision and computational efficiency of the recommendation. By addressing the challenges of curriculum complexity and student engagement, our system seeks to improve advisor efficiency and empower students to make informed academic decisions, contributing to enhanced learning outcomes and timely graduation.

The current tool available to students at Southern Methodist University is called Degree Planner, an online program where students can manually select courses to build a degree plan. The tool has a simple user interface and lacks recommendations. When overwhelmed by information, it is difficult for students to make optimal decisions, making degree planning inherently difficult and intimidating. Our proposed system simplifies the user experience by incorporating an LLM that lets the student specify interests and preferences to generate a 4-year plan, setting an easy foundation for course planning. A recommender system is used to ensure that the plan meets curriculum criteria and that the student can refine the course selection to meet her specific preferences. This guided approach will reduce information overload and let the student explore course offerings and degree options with minimal effort.

The rest of the paper gives a short overview of related work, followed by the design of the system. Then we will present some preliminary results from a prototypical implementation of the system. The paper concludes with a discussion.

2 Related work

Before we discuss our methodology, we will review some important studies in the areas of effective academic advising and recommender systems.

2.1 Academic Advising and Student Success

Student success can be drastically improved by helping students with available and quality advising resources. Higgins, who emphasizes the importance of the advisor-advisee relationship in academic advising, asserts centrality to student success through personalized guidance and support. The article finds that the quality of the advising relationship impacts students' educational and personal development (Higgins 2017). In a similar study, the authors investigated advisor behaviors that improve communication with students (Barnett, Roach, and Smith 2006). They suggest "microskills" like active listening, nonverbal communication, and effective questioning increase student retention and satisfaction. Besides an interpersonal relationship, advisors can also facilitate a sense of belonging and validation for a student's success (DeRosa 2024). Regarding student retention, a recent study by EAB found that students who meet with advisors had higher persistence rates, with the greatest benefits seen among students with lower predicted likelihoods of persistence, suggesting that investing in advising can significantly improve student retention and equity gaps (Vernit 2021).

However, universities often face issues in advising availability and quality. A recent academic advisors' survey among educational institutions in the US suggests 40.8 percent of academic advisors feel burned out from their work at least once a week to every day, a rate that is higher than the 33 to 35 percent of college or university employees who reported experiencing burnout in 2021–2022 (Soria et al. 2023). In the study, many advisors reported being emotionally fatigued throughout the semester, especially during the advising peak seasons. A recent Tyton Partners survey found the higher the advising caseload, the shorter the meetings with students (Bharadwaj et al. 2023). Advisors who experience the highest caseload often only have 10 minutes to meet with each student. With excessive caseload, advising work efficiency decreases. This not only prevents students from necessary support but also creates more emotional distress and potential burnout for advisors. Advisor also faces a lack of student engagement, exacerbating the advisor workloads (Young-Jones et al. 2013).

To address the issues, a recommender system can recommend students possible courses to take, enabling students to gain a better understanding of their degree progress via engagement, decreasing advising workload, and improving advising quality. The Office of Undergraduate Education and Academic Success at Southern Methodist University developed a Degree Planner tool that helps students plan their academic paths and track their progress toward graduation. However, internal studies show that most students do not use the current Degree Planner tool. The Degree Planner offers the desired template for undergraduate students, laying out requirements and courses that fulfill the requirement, giving the user functionalities of planning what courses to take. This approach, while effective, can be very intimidating and time-consuming to first-year students who are not sure what classes to take in the next four years, discouraging student engagement. With a recommender system providing personalized recommendations, a student could conveniently plan and optimize course schedules, comparing possible 4-year plans tailored to their interests.

In general, recommender systems can help universities create an efficient, student-centric curriculum that helps students make informed decisions about their academic paths by improving the effectiveness and student participation in advising.

2.2 Recommender System

Recommender systems were introduced by Paul Resnick and Hal R. Varian as automated systems that could help filter information and provide personalized recommendations (Resnick and Varian 1997). Recommender systems are essential tools for various news, video, and social media websites to recommend content to users based on their preferences and behaviors.

In the age of big data and machine learning, recommender systems have been commonly utilized by big-tech companies like Google and Netflix due to their effectiveness and simplicity in recommending users various new products. However, academia and current literature are not keen on using recommender systems to solve intricate problems by designing the training parameters. Therefore, this research aims to show how recommender systems can be designed to handle complicated curriculum data in universities. Recent literature like recommender systems with generative retrieval (Rajput et al. 2023) has shown that recommender systems can be used in a complex schema like multimedia recommendation and generative retrieval. This study can expand on utilizing these data mining techniques to help train a recommender system for this complicated problem. Formalizing multimedia recommendation through multimodal deep learning (Malitesta et al. 2024) can be a prime example of how this particular recommender system model can be implemented.

Curriculum requirements can be checked using set-cover algorithms (Chvatal 1979). Set-cover algorithms are also not widely used in the field of recommender systems due to their unfavorable time complexity. Solving the graph burning problem for large graphs (Pereira et al. 2024) shows how this problem can be defined to implement logic-based AI models such as set-cover to achieve an optimal result. We aim to combine the two to create more efficient and student-centric recommender systems that help students make informed decisions about their academic paths and improve their overall learning experience.

In the context of universities, recommender systems can play a crucial role in helping students navigate the complex landscape of course offerings and requirements. By analyzing data on student enrollment, course schedules, curriculum structures, and other relevant factors, recommender systems can provide personalized recommendations to students on which courses to take next, how to fulfill their degree requirements, and how to optimize their academic paths. Especially using large language models' capabilities to interpret verbal descriptions of the curriculum and requirements in the form of a course catalog and degree plans will be very useful.

3 Methodology

3.1 Design

Figure 1 shows the design of the proposed recommender system which integrates curriculum data, student academic progress, and transformer-based large language models to provide personalized course recommendations, streamlining the advising process for students and academic advisors. The system is composed of multiple interconnected components, each fulfilling a specific role in enabling effective academic planning and course selection.

The user (student) plans their degree plan before their advising meeting. They will provide their academic goals (degrees pursuing), topics of interest, and current state (completed and in progress courses) to the system for relevant information retrieval. From the university degree catalog, the recommender system will retrieve all degree requirements for the degree(s) the student is pursuing. From the university courses vector database, the recommender system will retrieve all relevant courses a student can take to fulfill their academic interest, regardless of their interest in their degree(s). This information, alongside the student's current progress, will be compiled into a final prompt to query the LLM. The LLM will then create the degree plan tailored to the student's current academic progress, pursuing degrees, and academic interests based on the information given to the model.

Once the model has an initial draft of the degree plan proposal, the generated plan will be fed into the Requirements Checker, which will return which requirements have not yet been met in this proposed degree plan. This check will evaluate various graduation criteria. It will check if total credit hours exceed 120, if the courses taken fulfill target degrees, if all requirements of general education are met, etc. Since general education requirements can often be satisfied by the same course, the general education checker is based on the set-cover algorithm, which will identify requirements that have not been met and offer the LLM the minimal courses to take as potential options for course selections that meet those requirements. This information about unfulfilled requirements will be provided as feedback to the LLM to be corrected. This is followed again by checks and repeated until all requirements are met or a specified time limit has been reached, forming a Self-Correction Loop. Finally, the recommender system will propose this generated degree plan to the user, who can decide to accept, modify, or deny this proposed plan, creating a User in the Loop (UIL) decision-making and benchmarking implementation for a secure system design (Cranor 2008).

The curriculum data serves as the foundational knowledge base, encompassing course offerings, degree requirements, prerequisite structures, and other academic policies. This data is processed using chucking techniques vectorized by all-minilim-l6-v2 (Sentence Transformers Team 2023) for the course vector database, ensuring that the most relevant and accurate information is used in retrieving relevant courses. By appropriately structuring the curriculum data, the system ensures its recommendations align with student interests and degree requirements with fast information retrieval.

The user's current state captures the student's academic progress, including completed courses, courses in progress, GPA, and remaining degree requirements. This component acts as dynamic input that contextualizes the recommendations, ensuring they are tailored to each student's unique academic standing. The system continuously updates this state to reflect changes as students progress through their degree programs.

The recommender system acts as a goal-based intelligent agent, aiming to achieve a degree plan recommendation that fulfills each user's unique requirements. This ensures users will be able to graduate using the system to plan their degrees while maintaining a balanced and intriguing academic path by taking courses they are interested in. During the interaction with the system, the students can gain a broader perspective and identify their interests before meeting with their advisors, enhancing student engagement in their degree process planning.

User goals can include degrees pursuing, academic interests, or feedback directly given to the LLM, which can help identify the best courses to take in the upcoming semester, the best degrees to take based on the student's academic progress, or ways to expedite their graduation timeline. This input adds a layer of personalization to the recommendations and ensures that the system is responsive to the individual needs of its users. It enables the user to instruct the language

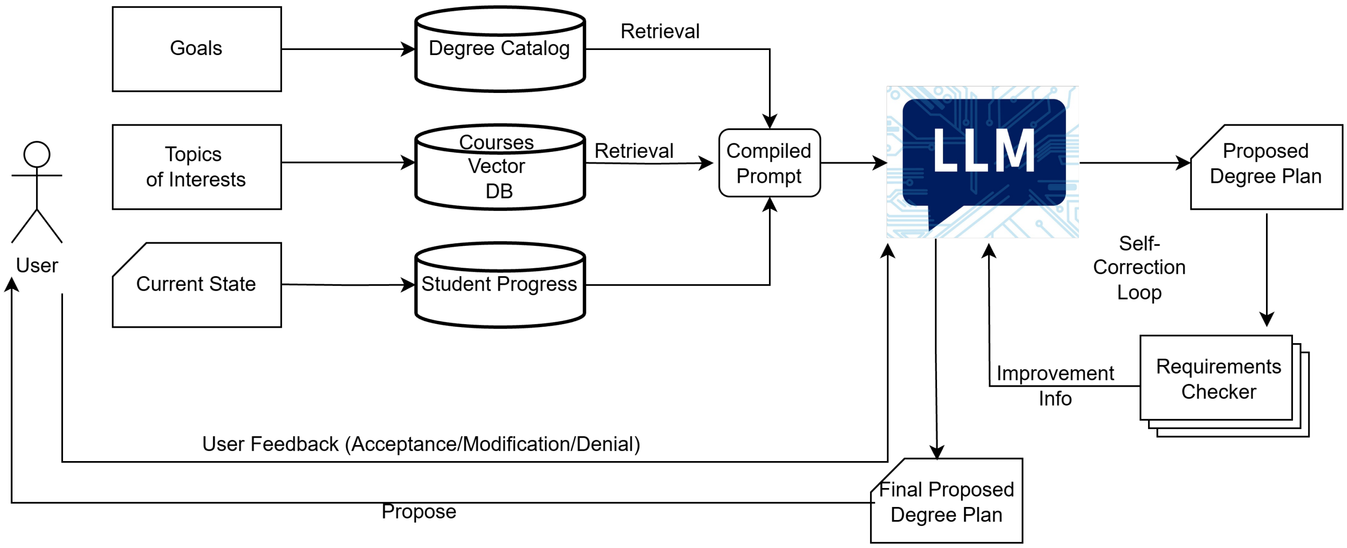


Figure 1: Architecture of the Recommender System Pipeline

models to recommend or offer useful data for a personalized user experience. It also makes the LLM a goal-based agent, ensuring a robust recommendation and personalized user experience.

To bridge the technical and user-facing aspects of the system, the LLM interprets user prompts and synthesizes outputs from the recommender system into a parsed and formulated interface. The LLM with the checker feedback loop ensures that the recommendations are clear, actionable, and tailored to the user's query. The LLM can also provide reasoning for its choices. It enhances the system's usability by presenting complex academic information in a simplified manner with explanations that students can easily understand, expanding reasoning and customization abilities beyond the traditional recommender system.

A critical aspect of the system's design is the degree progress checker, which validates all recommendations to ensure compliance with university policies, prerequisite structures, and degree requirements. This component safeguards against invalid or infeasible course selections, ensuring that students can confidently rely on the recommendations to meet their academic goals. Though the logic is often complicated, most universities already use degree progress-checking tools to ensure their students can graduate with an automated program. If such an automated check does not exist, then manual checking can be supported by an LLM agent who is asked to explain why a proposed degree plan meets all requirements or not.

Unlike traditional Retrieval-Augmentation Generation (RAG) techniques (Lewis et al. 2021), this recommender system does not chuck the entire university data into a vector database. It only retrieves relevant user data, degree requirement data, and courses of interest data with minimal vector database retrieval. This novel approach ensures data robustness for generation and guarantees only relevant data is being offered to the LLM in automation. Traditional RAG chatbot may be easier to implement, yet it can lose relevant data or offer unnecessary information to the LLM, causing the LLM to miss important details or to hallucinate.

3.2 Experiment

There are several metrics to test the performance of the recommender system.

1. Accuracy: How frequently does the system offer effective and correct recommendations where the proposed plan guarantees student graduation?
2. Speed: How fast does it take for the system to give an initial proposed plan to the users? How many iterations does it take on average for the user to accept the proposed plan?
3. Relevancy: How relevant are the recommendations to the student's degree requirements and academic interests?

Performance vs. Computational Cost Analysis To evaluate the performance and computational efficiency of various LLMs, we present a comparative analysis based on two key metrics: performance score (y-axis) and computational cost (x-axis).

100 simulated student user data were used to conduct this experiment, where each student pursued random degrees with random topics of interest while completing random courses during the first 1 or 2 semesters of their undergraduate career. These 100 user data were given to various LLMs to get an initial proposal, then sent to the Self-Correction Loop only once to test out the accuracy and efficiency of these models. Since the requirements checker has very complicated logic for each program, we implemented a requirement-checking agent (same model as the initial generating LLM) to prompt give the initial LLM requirements that need to be met. Another separate agent is then used to grade the degree plan proposal as pass or fail: pass being the LLM proposed a degree plan that reaches all requirements.

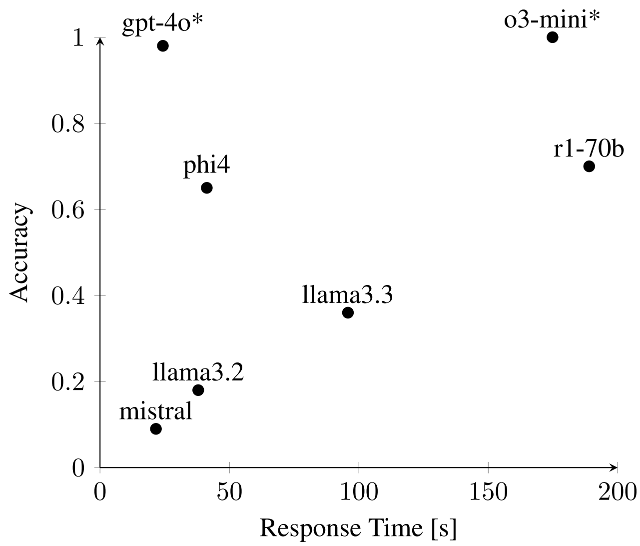


Figure 2: Scatter Plot of LLM Speed vs Accuracy

The x-axis of figure 2 represents the computational cost (average time in seconds used to generate a proposed plan), while the y-axis denotes the performance score (accuracy: what percentage of degree plans does the language model returns a degree plan that reaches all requirements.

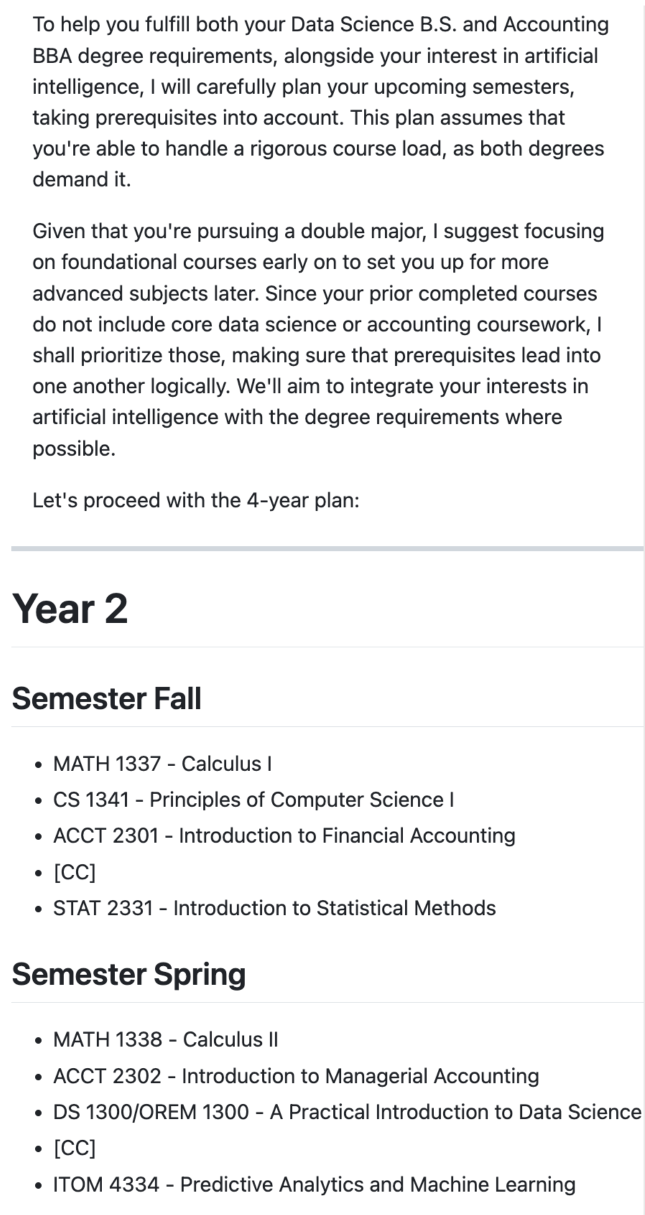
This experiment shows that using an appropriate LLM is crucial for good accuracy and speed outcomes. GPT-4o (OpenAI 2024a) performs better than any other models, reaching a 98% accuracy with just 24.29 seconds per recommendation on average. In comparison, the o3-mini (OpenAI 2024b) reaches a 100% accuracy, yet the Chain-of-Thought model runs more than 5 times slower for incremental improvement.

However, considering privacy concerns for student data and accumulating costs for OpenAI model services, open-source models are still attractive for their local execution and low cost. Phi4 (Abdin, M et al 2024) performs the best out of all open-source models experimented with, reaching 65% with 41.26 seconds per recommendation. DeepSeek R1 Distill Llama 70B (DeepSeek-AI 2025) performed better at 70% accuracy, yet took almost 5 times longer for each recommendation. Llama 3.3, llama 3.2 (Grattafiori 2024), and mistral (et al 2023) performed underwhelmingly, proving are not the best models for the task of course recommendation.

All open-source LLMs were run separately on one node of the NVIDIA DGX A100 GPU with 80GB of VRAM.

Though accuracy and speed can be easily measured, there are few metrics available for testing recommendation relevancy. The evaluation of this criterion will heavily rely on user feedback, such as how often the user abandons system suggestions and how many iterations it takes to reach a satisfactory recommendation.

Example Output After prompting the LLM with the user's academic goal, interest, and current state, the LLM will first output a proposed plan to be checked with the degree progress checker in the Self-Correction Loop. In Figure 3, the LLM was prompted by a user pursuing a Data Science B.S. and Accounting B.B.A. double major with interests in artificial intelligence.



To help you fulfill both your Data Science B.S. and Accounting BBA degree requirements, alongside your interest in artificial intelligence, I will carefully plan your upcoming semesters, taking prerequisites into account. This plan assumes that you're able to handle a rigorous course load, as both degrees demand it.

Given that you're pursuing a double major, I suggest focusing on foundational courses early on to set you up for more advanced subjects later. Since your prior completed courses do not include core data science or accounting coursework, I shall prioritize those, making sure that prerequisites lead into one another logically. We'll aim to integrate your interests in artificial intelligence with the degree requirements where possible.

Let's proceed with the 4-year plan:

Year 2

Semester Fall

- MATH 1337 - Calculus I
- CS 1341 - Principles of Computer Science I
- ACCT 2301 - Introduction to Financial Accounting
- [CC]
- STAT 2331 - Introduction to Statistical Methods

Semester Spring

- MATH 1338 - Calculus II
- ACCT 2302 - Introduction to Managerial Accounting
- DS 1300/OREM 1300 - A Practical Introduction to Data Science
- [CC]
- ITOM 4334 - Predictive Analytics and Machine Learning

Figure 3: Example System Output After Initial Prompt

This response was a snippet of the full response generated by GPT-4o. It started with a short analysis of the user's query demands, considering some nuances in the complicated task of double majoring, course orders, and interest integration. Then, the model generated the response with fundamental courses the user had not yet taken. It slowly increased the difficulty of the technical courses while keeping one course per semester as a general education requirement ([CC]). This showcases the transparency of the LLM's recommendation, giving the user a detailed strategy for planning the degree. This will help students understand the complexity of degree planning while keeping them informed about all actions the recommender system is taking to personalize their degree planning experience. After this step, the recommender system will check the LLM's response with the requirements checker to correct any mistakes or flaws in the Self-Correction Loop until the recommendation fulfills all degree and graduation requirements.

4 Discussion

In this study, a recommender system was made based on the Southern Methodist University Curriculum listed on the university catalog, which has been loaded into a relational database and vector database, contributing to the university advising efficiency to boost student success. The goal-based agent utilizes a self-correction loop and UIL for the customization of user inquiry in natural language, improving the personal-tailoring design and user interface, appealing to more students to explore courses offered and their 4-year degree plan.

4.1 Impact

Traditional methods of student degree planning require a complete familiarity with the university catalog, making the process intimidating and time-consuming for new students. Assuming a student will take only 90 seconds to determine each course to take during degree planning, it will cost the student an hour to do a 4-year plan. Within just a minute, this recommender system can recommend the student an entire degree plan, giving the user opportunities to adjust the degree plan by interacting with an intelligent agent, making the progress of degree planning streamlined and convenient.

This recommender system aims to increase student engagement in advising meetings by reducing the time spent filling out degree plans. This will enhance university advising efficiency and reduce advisor workloads. Improving advising quality will lead to overall student success, exacerbate student retention, and help persistent students with underprivileged backgrounds, reducing inequality in education resources.

Since almost all university stores their data electronically, it is easy to deploy this recommender system to any university or educational institution. With a university's degree catalog, student academic progress data, and degree progress checker, this system is universal to all sorts of educational institutions to benefit all enrolled students. This generalization demonstrates the scalability of the recommender system, as it can apply universally to various data environments.

4.2 Limitations

This system is not designed nor capable of replacing advising offices or student course planning in general. Any technological system should be designed to help those in need for the sake of goodwill and equality. LLMs are still facing issues of hallucination at unexpected times, making the system not entirely trustworthy due to the ambiguity of probabilistic models. The data pipeline is built from data that is meant to be read by humans, not to be automated by machines. This may cause a discrepancy between data and LLM understanding. Hence, this system still requires university workers to maintain and announce crucial details that will determine a student's academic career.

4.3 Future Works

While the database infrastructure for courses, degrees, and RAG is fully operational, additional data is needed to further train and refine the recommender system. Specifically, access to anonymized student data can help implement few-shot prompting and fine-tune a reinforced learning model for more robust recommendations. This data should include academic records such as completed courses, grades, declared majors and minors, and progression timelines. Gathering this data from the university's administration office is critical, as it will allow the system to analyze historical patterns and trends to produce accurate and personalized recommendations. Furthermore, such data will enable the system to learn how various factors—such as course difficulty, prerequisites, and individual student preferences—impact academic outcomes, thereby improving the overall efficacy of the recommendations.

Securing access to student data involves collaboration with the university administration to ensure compliance with privacy regulations and ethical standards. The data must be anonymized to protect student identities while retaining the key attributes required for training the recommender system. Once obtained, this data will be integrated into the system's workflow, enabling the development of predictive models that can anticipate student needs and provide tailored guidance. The combination of a well-structured course and degree database, a vector database for augmented retrieval, and robust student data will form the basis for a highly effective and scalable academic advising system. This methodology highlights the importance of comprehensive data infrastructure in building and training advanced recommender systems.

The recommender system needs to be integrated into University advising centers to gather feedback from student and advisor users. This will be the essential inference of whether the recommendations are effective and to improve user experiences.

The curriculum data and student interest are helpful for the department's course planning for the semester. The program predicts student enrollment based on its course recommendation to each student, helping the department administrators decide what courses and seating capacity of each course to open for the following semester.

The course data stored in the vector database can be generated into a high-dimensional graph visualization of all courses offered by the university. Graphing the relation of all courses gives important insights to administrators for understanding the knowledge coverage of each program and any interdisciplinary intersection of programs and courses.

5 Conclusion

In favor of student success and retention, this research proposes a goal-based intelligent agent recommender system to increase student engagement for advising quality and simplify student curriculum planning by recommending courses for each degree requirement based on student interest and academic pursuit. By leveraging data retrieval and a language model pipeline, the system aims to revolutionize academic advising in educational institutions by optimizing workloads for students and university faculty.

Acknowledgment

Zerui Ma is supported by a fellowship from the Robert Mayer Undergraduate Research Fellows Program of Dedman College Interdisciplinary Institute at SMU.

References

Abdin, M et al. 2024. Phi-4 Technical Report. arXiv:2412.08905.

Atalla, S.; Daradkeh, M.; Gawanmeh, A.; Khalil, H.; Mansoor, W.; Miniaoui, S.; and Himeur, Y. 2023. An Intelligent Recommendation System for Automating Academic Advising Based on Curriculum Analysis and Performance Modeling. *Mathematics*, 11(5).

Barnett, S.; Roach, S.; and Smith, M. 2006. Microskills: Advisor Behaviors that Improve Communication with Advisees. *NACADA Journal*, 26(1): 6–12.

Bharadwaj, P.; Shaw, C.; Condon, K.; Rich, J.; Janson, N.; and Bryant, G. 2023. Driving Toward a Degree 2023.

Chvatal, V. 1979. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3): 233–235.

Cranor, L. F. 2008. A framework for reasoning about the human in the loop. In *Proceedings of the 1st Conference on Usability, Psychology, and Security*, UPSEC'08. USA: USENIX Association.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

DeRosa, E. L. 2024. Academic Advising's Hidden Role in Fostering Validation/Belonging Leading to Improved Grades. *NACADA Journal*, 43(2): 121–135.

et al, J. A. Q. 2023. Mistral 7B. arXiv:2310.06825.

Flaherty, C. 2023. Survey: Students Cite Barriers to Success, Seek Flexibility. *Inside Higher Ed*.

Grattafiori, A. e. a. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Higgins, E. 2017. The advising relationship is at the core of academic advising. *Academic Advising Today*, 40(2).

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; tau Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

Malitesta, D.; Cornacchia, G.; Pomo, C.; Merra, F. A.; Di Noia, T.; and Di Sciascio, E. 2024. Formalizing Multimedia Recommendation through Multimodal Deep Learning. *ACM Trans. Recomm. Syst.* Just Accepted.

OpenAI. 2024a. GPT-4o System Card. arXiv:2410.21276.

OpenAI. 2024b. OpenAI o1 System Card. arXiv:2412.16720.

Pereira, F. d. C.; de Rezende, P. J.; Yunes, T.; and Morato, L. F. B. 2024. A Row Generation Algorithm for Finding Optimal Burning Sequences of Large Graphs. In *32nd Annual European Symposium on Algorithms (ESA 2024)*. Schloss Dagstuhl – Leiibniz-Zentrum für Informatik.

Rajput, S.; Mehta, N.; Singh, A.; Keshavan, R. H.; Vu, T.; Heldt, L.; Hong, L.; Tay, Y.; Tran, V. Q.; Samost, J.; Kula, M.; Chi, E. H.; and Sathiamoorthy, M. 2023. Recommender Systems with Generative Retrieval. arXiv:2305.05065.

Resnick, P.; and Varian, H. R. 1997. Recommender systems. *Commun. ACM*, 40(3): 56–58.

Sentence Transformers Team. 2023. all-MiniLM-L6-v2. Hugging Face Model Hub.

Soria, K. M. 2023. Disparities in College Students' Access to Academic Advising During the COVID-19 Pandemic. *NACADA Journal*, 43(1): 17–30.

Soria, K. M.; Kokenge, E.; Heath, C. A.; Standley, E. C.; Wilson, S. J. F.; Connley, J. R.; and Agramon, A. I. 2023. Factors Associated with Academic Advisors' Burnout. *NACADA Journal*, 43(2): 105–120.

Vernit, E. 2021. Who could benefit the most from more advising? EAB Blog.

Young-Jones, A.; Burt, T.; Dixon, S.; and Hawthorne, M. 2013. Academic advising: does it really impact student success? *Quality Assurance in Education*, 21(1): 7–19.