Decision Support Systems (DSS) in Higher Education using Gen-AI

Aarav (2021302) Atharva (2020038) Kartikay (2021326) Aabhas (2021369)

Navnoor (2021401) Abhijay (2021439)

1. Problem Statement

Deciding on coursework in higher education, presents a significant challenge for students due to a multitude of factors that complicate the decision-making process. Firstly, the sheer volume and complexity of course options available can be overwhelming, making it difficult for students to identify which courses align best with their academic and career objectives. Additionally, the lack of centralized and easily accessible information about degree requirements, course prerequisites, and course outlines/guidelines further exacerbates the problem. Students often struggle to understand how specific courses fit into the broader context of their degree programs and future professional goals. This is compounded by the dynamic nature of the engineering and computer science fields, where emerging technologies and industry demands can quickly render certain skills more relevant than others. Moreover, the absence of a personalized guidance mechanism means that students frequently rely on fragmented advice from peers, faculty, or disparate online resources, which may not always be reliable or applicable to their unique circumstances.

Why is this problem Statement Important? A decision support system for higher education is crucial because it simplifies the complex process of course selection and academic planning by providing personalized recommendations aligned with students' career goals and interests. It addresses the challenge of navigating through various resources and documentation scattered across different sections of the institute's website, making vital information more accessible and understandable.

2. Background

The booming academic software industry offers significant potential for India's vast education system, with its 39,000+ colleges and 37 million+ students. While tools exist for course management, a gap remains in personalized recommendations considering academic requirements and career aspirations. Higher education's complexity makes navigating course selections challenging for students. Evidence suggests a growing demand for a system that not only tracks progress but also intelligently recommends courses aligned with both degree requirements and career interests.

3. Literature Review

[1] Decision support system (dss) in higher education system: In this research article, they emphasize the significance of Decision Support Systems (DSS) in higher education, highlighting their role in integrating data and intelligence to facilitate optimal decision-making. The paper reviews past contributions in representing DSS modules, examining current systems and databases in various institutions. The goal is to assist teaching, research staff, and students in making informed decisions across different scenarios. The article emphasizes that integrating a DSS with higher education ICT systems can significantly reduce costs and time associated with critical decision-making in diverse contexts. The proposed Decision Support System is tailored to align with the educational mission, research, data collection, and societal contributions of institutions. Ongoing development aims to extend its applicability to various aspects of modern higher education, including financial, public, and international relations, meeting the evolving needs of the workforce. Integration with Information and Communication Technology structures is expected to streamline decision-making processes for increased institutional effectiveness.

[3] Decision support system for course offering in online higher education institutes: This research focuses on optimizing course offerings by addressing the complexities faced by department administrators before each academic semester. The key contributions include identifying factors influencing online course selection, developing a neural network model to simulate student behavior, and implementing a decision support system (DSS) for administrators. The DSS, utilizing data from 298 online graduate courses, demonstrates high prediction accuracy, outperforming traditional regression techniques. It introduces a three-layer perceptron neural network, achieving acceptable prediction results despite uncontrollable variables like student preferences. The study highlights the DSS's potential for what-if analysis and goal-seeking behavior. Limitations are acknowledged, suggesting future enhancements such as automatic detection of optimal courses and exploring alternative machine learning methods for improved accuracy in different institutional contexts.

[4] Review on recommender systems for course selection in higher education: The paper explores various recommen-

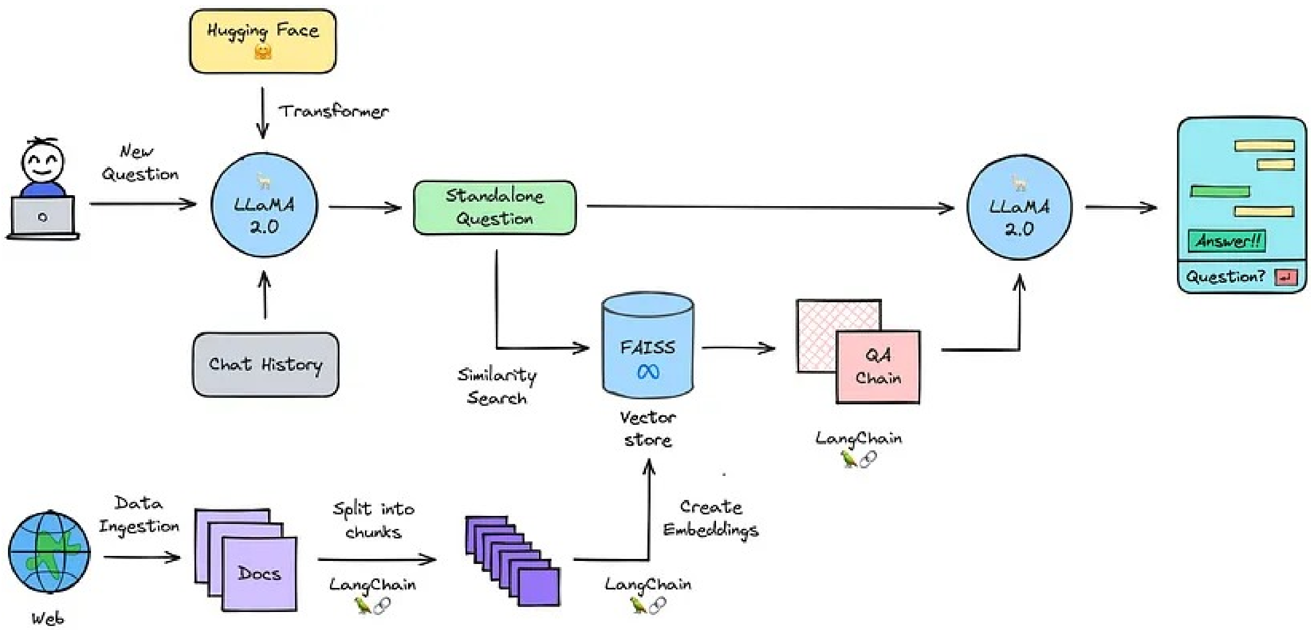

Figure 1. Outline of Decision based system

dation approaches, including content-based, collaborative filtering, and hybrid recommendation approaches. Among these, the Hybrid recommendation approach emerges as the most effective for course selection, combining the strengths of other approaches to provide excellent recommendations. This approach is deemed instrumental in helping students achieve their career goals successfully.
[5] Recommendation systems with complex constraints: This study addresses the challenge of making recommendations while considering constraints or requirements, with a specific focus on course recommendations within the CourseRank system at Stanford University. The primary objective is to recommend courses that not only align with requirements but are also desirable to students. The research introduces expressive models for course requirements and proposes schemes for both verifying requirements and providing recommendations while accounting for constraints. The study highlights that certain requirements are inherently expensive to check, leading to the presentation of exact and heuristic techniques for such cases. While the work centers around course requirements, it offers insights into the design of recommendation systems dealing with complex constraints applicable in various applications.

4. Methodology

4.1. Dataset

Upon an extensive review of the available literature, it became evident that no existing datasets precisely aligned with our study's specific requirements. Consequently, to bridge this gap, we undertook the initiative to generate a synthetic dataset tailored to our research needs [2]. This dataset was crafted by leveraging regulations and coursework documents that are readily accessible through the academic institution's official website guidelines. These documents served as a foundational basis for synthesising a comprehensive dataset. Our approach is to create a synthetic dataset using **gpt-4**, a strategic approach ensured that our synthetic dataset closely mirrored the real-world scenarios and complexities inherent in the academic domain, providing a robust foundation for our subsequent analyses and investigations. We needed to create synthetic queries to test our model so we created a set of **200** question-answer pairs for evaluating our model. Since the ERP system needs data for all programs and situations, we divided our query generation process into subtopics to create a balanced dataset and avoid skewness. We divided the sections into different streams, including CSE, CSD, CSAI, CSSS, CB, etc., and similarly for different programs to ensure a balanced model evaluation. In addition to these, we extracted FAQs from the institute's website to add to the dataset.
Apart from creating an evaluation dataset, we also needed institute documents to create an RAG system. In order to do so, we manually extracted relevant documents from the institute's website for various programs ranging from BTech to PhD. In total, we collected **38** documents and preprocessed each document to extract the text in each of them.

4.2. IR Retrieval

Initially, we add the documents to a folder and then use the LangChain framework to convert the documents into embeddings for faster access. We use the **sentence-transformers/all-mpnet-base-v2** model for converting the document words into embeddings of size 768. For storing the documents in sets of combination of words, we use the Recursive Text Splitter and create chunks of 1000 words with an overlap of 20 words in each chunk. Integrating Meta's LLaMA 2.0, LangChain framework, and the FAISS library optimizes how data is processed and utilized, particularly by leveraging Information Retrieval (IR) techniques with Generative AI. We use FAISS to index documents efficiently and perform faster similarity searches. Also, by harnessing LLaMA 2.0's advanced language comprehension and generation capabilities, decision-makers can efficiently extract and synthesize knowledge from vast educational content. The LangChain framework enables the modular assembly of AI-powered applications, facilitating the development of custom decision-support tools to analyze and interpret complex datasets. Meanwhile, FAISS enhances the system's ability to swiftly retrieve pertinent documents from extensive databases, using vector embeddings for efficient similarity searches. This combined approach not only streamlines the IR process but also ensures that decision-makers in higher education have access to timely, relevant, and comprehensive insights, thereby significantly enhancing the decision-making landscape.

4.3. Model Inference

The model inference involves using the Retrieval system with the LLM inference pipeline to get an accurate model response. We use the **LlaMA-2-7b instruct** model for our evaluation in order to ensure the evaluation is accurate. Due to computational limitations, we use the quantized version of the model using HuggingFace APIs for model inferencing. We use the BitsAndBytes library to quantize the model to its 4-bit parameter versions, and then we use the quantized model for inference. During inference, we use 0.1 as the model temperature for reducing hallucinations and also increase the repetition penalty to 1.1 and set the max new token count to 512. For inference we load the model on a T4 GPU in google collab with access to 12GB GPU RAM for loading model shards and inference.

4.4. Prototyping

Our initial prototyping stage involved crafting low-fidelity prototypes to outline the basic interaction flow of our decision support system. Following this, we gathered valuable insights and feedback through user evaluations conducted with a broad spectrum of students. This feedback was instrumental in developing a more refined and detailed high-fidelity prototype, which offered a clearer representation of the intended decision-based support system interaction model.

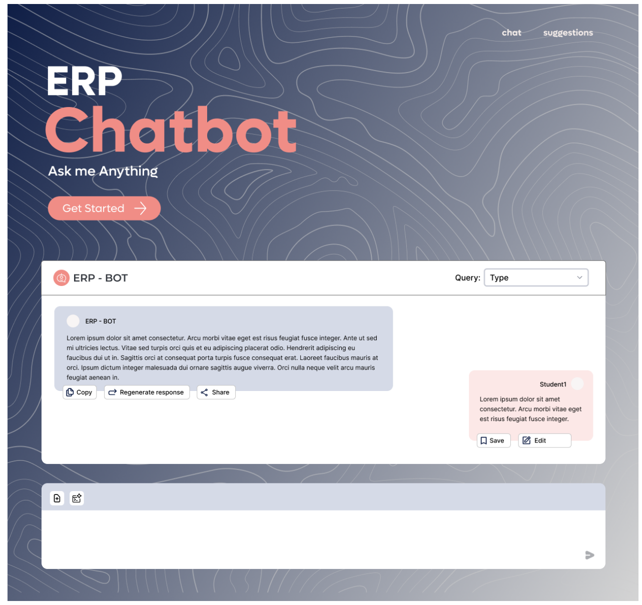

Figure 2. Prototype of ERP-Chatbot

Our project leverages an intricate methodology that integrates generative AI with Information Retreival to form a system tailored for educational decision support. Initially, the system employs rule-based filtering, utilising user-defined constraints to refine the selection of documents and data sources. This ensures that the information retrieval phase is deeply aligned with the user's specific academic requirements and constraints. Following this, the RAG component comes into play, where the system dynamically retrieves and augments the generation process with information relevant to the user's query, enriched by the user's academic profile and predefined constraints.

In the subsequent phase, this contextually enriched dataset is analysed by a Large Language Model (LLM), designed to generate informed responses that are closely aligned with the user's academic aspirations and the nuanced constraints of their queries. We can incorporate prompt engineering techniques like CoT(Chain of Thought) prompting and few shot prompting to improve the results given by the LLM.

5. Novelty

Our system incorporates several novel aspects that distinguish it from traditional approaches. Firstly, it leverages Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) prompting techniques with Information Retrieval (IR) [1] to gather relevant information and answer student queries interactively. This personalized approach caters to individual needs and goals at scale, surpassing the limitations of rule-based systems[3]. LLMs can consolidate detailed information on various topics without losing relevance, unlike standard ML techniques that require repeated feature extraction [1, 4]. Additionally, different document formats are automatically parsed and interpreted, enabling comprehensive analysis of various possibilities. Furthermore, unlike ontology-based systems like [5] that require redesign with every new course or guideline, our system exhibits superior scalability, readily adapting to evolving academic landscapes. Finally, by consolidating information and delivering interactive responses, our system offers a significantly enhanced user experience compared to traditional methods of browsing multiple documents. Leveraging an interactive chat format, the system fosters greater comprehension and engagement by presenting information in a more user-friendly and intuitive way.

6. Evaluation Metrics

In evaluating our Language Model (LLM) system, we employ a multi-faceted approach encompassing several key criteria. Firstly, we conduct a Fairness Evaluation to assess the LLM's responsiveness to diverse user groups, addressing potential biases and ensuring equitable responses. Secondly, we employ Perplexity Measurement to gauge the model's language understanding and fluency by calculating its ability to predict subsequent words in text sequences. Human Evaluation is also used as we have evaluators rank the relevance of documents retrieved by the LLM using a 0/1 labelling system, offering insights into its information retrieval capabilities. To evaluate the model from the perspective of user experience, we also calculate the inference time for assessment. We use human-based model response evaluation to better understand the performance of different LLMs for this task to establish baseline results.

Average Inference Time	13.991 sec
Mean Perplexity	3898
% Relevant Feedback	80.88%

For experimentation, we ran the inference pipeline and collected necessary metrics including inference time and perplexity and then we took the help of 4 human evaluators to evaluate the model response based on its relevance to the given query. The average inference time for any query was close to **14 seconds**, and the average perplexity of the model was **3898**, where the perplexity of the response ranged from **3200** to **5000**. Upon evaluation, the human evaluators found that close to **81%** of the model responses were relevant, with close to **5%** responses returning no response to the given query or responding that they do not know anything related to the given query. All the responses that were generated were only marked relevant if they were factually correct and relevant to the given query.

7. Contributions

Data collection, cleaning , management	Aarav, Kartikay, Atharva
Identification of Model & LLM inference	Navnoor, Aabhas, Abhijay
Prompt Engineering & Model parameter fine-tuning	Atharva, Kartikay, Abhijay, Aabhas
UI/UX Design	Aarav, Navnoor
Report Writing	All Members

8. Future Work

Building upon the initial prototyping phase, there is significant potential further to refine the user interface and overall system architecture. By leveraging advanced features and capabilities within React.js and CSS, coupled with the utilization of additional UI libraries, the design aesthetics and user experience can be greatly improved. We want to explore further and help students with course-related decisions by looking at their background and aspirations. We aim to increase our dataset size by creating a larger synthetic dataset with more datapoints. We also aim to expand our evaluation metrics for the LLM in order to get a thorough understanding of the model performance. We aim to fine-tune the model on this synthetically generated dataset to improve model perplexity and make it more confident while generating answers to certain queries. Additionally, we gauge User Satisfaction through surveys, obtaining feedback on user experience, ease of use, and overall system helpfulness.

References

[1] FAKEEH, K. Decision support system (dss) in higher education system. *International Journal of Applied Information System (IJAIS) 9*, 2 (2015).

[2] KADDOUR, J., AND LIU, Q. Synthetic data generation in low-resource settings via fine-tuning of large language models, 2024.

[3] KARDAN, A. A., AND SADEGHI, H. A decision support system for course offering in online higher education institutes. *International Journal of Computational Intelligence Systems 6*, 5 (2013), 928–942.

[4] LYNN, N., AND EMANUEL, A. A review on recommender systems for course selection in higher education. In *IOP Conference Series: Materials Science and Engineering* (2021), vol. 1098, IOP Publishing, p. 032039.

[5] PARAMESWARAN, A., VENETIS, P., AND GARCIA-MOLINA, H. Recommendation systems with complex constraints: A course recommendation perspec-

tive. *ACM Transactions on Information Systems (TOIS)* *29*, 4 (2011), 1–33.