2024 4th International Conference on Intelligent Technologies (CONIT)
Karnataka, India. Jun 21-23, 2024

A Client-Server Based Educational Chatbot for Academic Institutions

Rohan Paul Richard
Department of Computer Science and Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
rohanpaul@karunya.edu.in

Ebenezer Veemaraj
Department of Computer Science and Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
ebenezerv@karunya.edu

Juanith Mathew Thomas
Department of Computer Science and Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
juanithmathew@karunya.edu.in

Joel Mathew
Department of Computer Science and Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
joelmathew20@karunya.edu.in

Caleb Stephen
Department of Computer Science and Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
calebstephen@karunya.edu.in

Richie Suresh Koshy
Department of Computer Science and Engineering
Karunya Institute of Technology and Sciences
Coimbatore, India
richiesuresh@karunya.edu.in

2024 4th International Conference on Intelligent Technologies (CONIT) | 979-8-3503-4990-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/CONIT61985.2024.10627567

***Abstract*—The use of Generative AI applications in academia, such as ChatGPT, Bard and Perplexity among others is growing at a rapid pace. With its rise, certain ramifications are being felt in regards to the quality and correctness of knowledge and output of these apps. This is detrimental to the process of learning and understanding subject matter in relevant context. In order to stop this issue in its tracks, a solution must be identified, built, tested and deployed so that students will get reliable outputs from trusted sources. This research analyses and describes a sample architecture as well as implementation for such a solution. It incorporates the latest in AI research and development, such as the Large Language Model Mixture of Experts (MoE) architecture as well as a Retrieval Augmented Generation (RAG) with a vector store to use trusted documents such as presentation files, PDF handouts and more from instructors. This is used to add context to the request to the Large Language Model and enrich the understanding as well as response of the model. Apart from this, the application is packaged into a server that can be run on the intranet, as well as deployed for public access. A frontend client page is served to the user, and communicates with the server for all its functioning.**

Keywords—Generative AI, Large Language Model (LLM), Retrieval Augmented Generation (RAG), Mixture of Experts, LangChain

I. Introduction

AI tools like ChatGPT, DALL-E, and others are making a significant impact on various industries, revolutionizing the way people work. Programmers leverage tools like Github Copilot for collaborative coding, while marketing teams utilize AI such as Midjourney to generate content for their brands. There is a palpable excitement surrounding their potential applications, with researchers expressing eagerness to explore the boundaries of natural language processing (NLP) using the Transformers architecture. The utilization of chatbots in learning environments sparks curiosity, as educators and scholars envision innovative ways to enhance student engagement and provide personalized educational experiences. However, this optimism is tempered by a cautious awareness of ethical considerations, prompting a collective examination of issues such as bias, privacy, and responsible AI deployment within educational settings.

This conversation also engages in a conversation on the evolving dynamics of human-computer interaction (HCI). Researchers delve into the user experience of interacting with chatbots, contemplating how these AI-driven conversational agents influence communications in classrooms and shape the digital learning landscape. Simultaneously, there is a sense of interdisciplinary collaboration, with experts from diverse fields coming together to dissect the multifaceted implications of AI in this context.

Amidst the enthusiasm and scholarly exploration, there exists a nuanced sentiment. Some view these AI-driven tools as transformative catalysts, capable of revolutionizing accessibility, personalized learning, and administrative efficiency. Scepticism still remains as the academic community grapples with questions of reliability, accountability, and the potential over-reliance on AI in educational contexts. The transformative power of generative AI is evident, yet negative sentiments are emerging in academic circles regarding these issues. However, change is inevitable, and the use of chatbots in academia is a growing trend worldwide. There is a need to proactively address these changes and establish systems that promote the proper ethical, and beneficial use of such tools.

This research work aims to develop a technological solution to facilitate students' access to large language models, enabling them to use generative AI for knowledge acquisition in an efficient and useful manner, and ensure that the output of the AI is contextual to their education. Simultaneously, educators can employ this tool to disseminate knowledge effectively, providing students with trusted resources to optimize their learning.

II. Purpose and Related Works

A. Purpose and Objectives

The purpose of this work is to understand existing solutions, define the shortcomings of current technologies, and engineer a system with resolutions for these issues. To achieve this, clear objectives must be defined and met. Following are the objectives of the research:

- *Evaluate the current scope of solutions:* Study the current solutions for the identified problem, and understand the differences between these solutions. Identify the various shortcomings in the existing solutions and explore improvements that can be made. Explore advances and new research in relevant fields that can be used to improve this technology.
- *Build a solution to integrate with existing infrastructure and resources:* Design the system in a way that it leverages existing infrastructure, such as servers and GPUs that may already exist. Minimize the use of specialized hardware wherever possible to ensure widespread compatibility.
- *Design the entire system for scalability:* Work on ensuring robustness of the system and designing the architecture so that it is optimized for high user load. Ensure that the server system can handle spikes in usage when required and use proper mechanisms for reliability and recovery from failure.
- *Protect the system from attacks and errors:* Implement proper authentication standards to ensure that the privacy of data is not compromised. Store user information on the user interface securely and pass it to the server by HTTP. Secure the database in an optimized denormalized form.
- *Generate trustable and verifiable responses:* Use retrieval augmented generation to get trusted information into the model's response. Develop a customized prompt template to suit the model and generate helpful, useful responses.

B. Research and Related Works:

The applications of LLMs as chatbots are one of the hottest topics in research in the current academic environment. The following are the inferences gained from research literature to build the system.

The authors of Ref. [1] protray a system that integrates Retreival Augmented Generation into a. LLM system for educational contexts. They discuss the varied weaknesses of ChatGPT in this regard, particularly the inability to refer to specific books and materials as a severe drawback. There is also raised the issue that the best responses are from models that are pay-to-use. They present a basic framework for the implementation of a vector store database, Pinecone, in this instance. Their entire system is designed to be an operational framework for an educational based LLM.

The paper Ref. [2] by Guruswami Hiremath et al., discusses the development of an automated system for the education sector, aiming to provide responses to user queries. Unlike existing chatbots using local databases, the proposed system integrates both local and web databases for improved scalability, user-friendliness, and interactivity. The approach incorporates various techniques, including machine learning, natural language processing (NLP), pattern matching, and data processing algorithms.

Md. Abdullah Al Muid et al. introduces "EduBot," in Ref. [3], an unsupervised domain-specific chatbot designed for educational institutions. The research emphasizes the chatbot's role as a virtual representation catering to admission seekers, providing information on the university, its departments, admission fees, and frequently asked admission-related questions. The study utilizes unsupervised learning and natural language processing techniques, employing tokenization, stop words removal, and vectorization for training data preprocessing. User inputs undergo similar processing, with tf-idf-based cosine similarity used to retrieve optimal answers. The authors implemented a user-centric evaluation metric, indicating an approximately 80% accuracy for the model.

Babpu Debnath and Aparna Agarwal explore the implementation of an artificial intelligence (AI)-integrated chatbot framework in educational institutes in Ref. [4], specifically targeting schools and colleges. The study proposes a development plan for a multi-use chatbot, contrasting it with traditionally single-purpose chatbots designed for specific functions like answering admission queries. The authors emphasize the need for AI integration to enhance the chatbot's self-reliance, intelligence, and ability to handle diverse fields.

Ref. [5] explores the impact of ChatGPT, a generative AI chatbot powered by a large language model (LLM), on education and the role of teachers. The study involves eleven language teachers who used ChatGPT for instruction over a two-week period, followed by individual interviews and analysis of interaction logs. The research identifies four roles of ChatGPT and three teacher roles. The study emphasizes the complementary relationship between teachers and AI, highlighting the importance of teachers' pedagogical expertise in utilizing AI tools. The findings contribute to an in-depth discussion on teacher-AI collaboration and provide implications for the future use of LLM-powered chatbots in education.

Research by Mahyar Abedi, Ibrahem Alshybani, MRB Shahadat, and Michael Murillo, Ref. [6] explores the integration of large language models (LLMs) and chatbots into graduate engineering education. The core investigation involves applying an LLM-based chatbot in a graduate fluid mechanics course, evaluating its ability to provide accurate responses. Results indicate the chatbot's effectiveness in answering complex questions and highlight potential advantages, including promoting self-paced learning, providing instant feedback, and reducing instructor workload. The study examines the transformative impact of intelligent prompting on the chatbot's performance and explores the use of powerful plugins like Wolfram Alpha for mathematical problem-solving. The authors argue for the ethical and efficient use of LLMs in education.

III. Proposed Methodology

A. Technology Stack ang Architecture

The below image Fig. 1 contains a top-level Architecture of the system, as well as a basic flow of how the system will work in a regular use case. As it illustrates, there are 5 essential components to the system: the Client, Server, Database, vector store and Inference Engine. These are the building blocks on which the entire system runs. Users use the client UI to make requests to the server. Under the hood, the server does all the work in persisting to the Database and querying the vector store to get the inference engine to generate a successful result.

The diagram illustrates the essential flow of the system, and due to the monolithic architecture, it is infinitely horizontally scalable. This means that replicas of the system can be used in a production environment, and handle

additional user load. At the same time, it is vertically scalable, as well, for full flexibility in case-by-case deployments. Authentication is a key part of the system and JSON Web Tokens (JWT) are used for all operations to ensure authentication of the user and authorization to access data.

This ensures that the system can be stateless in nature, which means that there is no information retained such as session details and more. Being stateless means that concurrency is increased, there are fewer dependencies, and scaling becomes a lot easier.

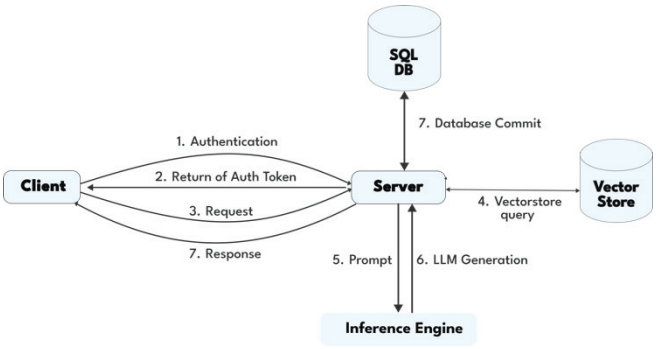

Fig. 1. Architecture Diagram of the proposed system and basic working flow

The technology stack in Fig. 2 contains all the important libraries and frameworks used to build and implement the system laid out in Fig. 1 accurately.

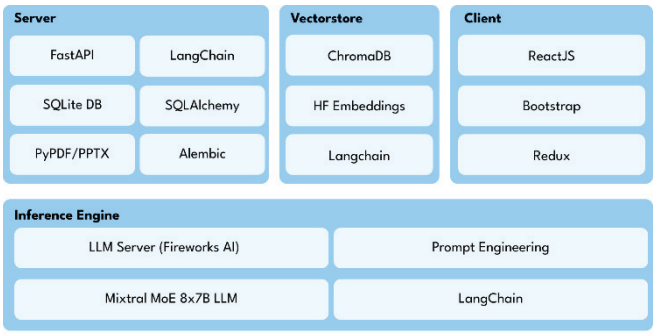

Fig. 2. Technology Stack of the proposed System

FastAPI [7] is the server-side framework being used**.** It is a modern, highly concurrent, and robust web framework to built APIs for Python. It is efficient, scalable, and easy-to-use. It uses type hints to enforce strict type checking. It is based on Starlette, which is a library to build lightweight asyncrhonous APIs in Python. It also uses Pydantic to create models for API request and responses, allowing for data validation and conversion.

It also uses Swagger UI to automatically generate all the documentation of each API route from the code, making it easy to document the application, including structures of inputs and outputs. It also includes features such as Dependency Injection, which allows to manage and inject dependencies at runtime based on authentication.

LangChain [8] is a cutting-edge orchestration framework designed for the seamless development of applications harnessing the power of large language models (LLMs). With libraries available in both Python and JavaScript, LangChain simplifies the creation of LLM-driven applications, including chatbots and virtual agents. It became the fastest-growing open-source project on GitHub, garnering significant traction in AI use cases.

LangChain is a versatile interface, providing a centralized development environment for building applications powered by various LLMs and providers. The modular environment facilitates the creation of applications capable of leveraging LLM capabilities**.** It also caters to a diverse range of LLM and NLP use cases, making it suitable for applications like chatbots, intelligent search, question-answering, summarization services, and virtual agents capable of robotic process automation. At its core, LangChain simplifies the programming of LLM applications. Providing a library of abstractions for both Python and JavaScript, LangChain offers modular components serving as building blocks. One example of this is the Chain. Chains form the heart of LangChain's workflows, seamlessly combining LLMs with other components to create applications through a sequence of functions. The framework offers various chain types, such as LLMChain, providing flexibility in designing applications with different prompts, tools, parameters, or models.

SQLAlchemy [9] is the server-side library that simplifies database interactions in Python. It's a modern and versatile tool for working with relational databases, offering a high-level and expressive interface. It enables efficient and scalable handling of databases, abstracting the underlying differences among various database engines.

Paired with PostgreSQL, the open-source and robust relational database management system used for this project, SQLAlchemy provides a seamless integration for high-load systems. PostgreSQL is known for its extensibility and advanced features like support for transactions. It also can have replicas to maximise availability and error recovery.

Using SQLAlchemy and PostgreSQL, developers benefit from a powerful combination, allowing for clean and object-oriented database operations. It's flexibility ensures smooth interactions with different databases, making it easy to switch or manage multiple databases within the same application. This means that with just a change in the connection URL, the database can be completely altered.

B. Retrieval Augmented Generation

Retrieval-Augmented Generation [10] is a natural language processing technique (NLP) that combines elements of retrieval-based and generative models to enhance the capabilities of language models. This methodology addresses the limitations of purely generative or purely retrieval-based systems by leveraging the strengths of each.

In a retrieval-augmented generation model, the system incorporates a retrieval mechanism to fetch relevant information or context from a predefined knowledge base or dataset. This retrieved information is then used alongside the model's generative capabilities to craft a more contextually aware and accurate response. The goal is to combine the fluency and creativity of generative models with the factual accuracy and specificity achievable through retrieval-based methods.

In this system, the use of Retrieval Augmented Generation is to add trusted documents to a vectorstore based on certain metadata. When content is added, data such as course code, and title which are used to uniquely identify the subject and filter responses to the correct subject during deployment. The advantages of this technique includes the ability to handle a

broader range of queries, improve response coherence, and enhance the model's overall performance by leveraging external knowledge. This means that the results of the model can be trusted along with proof for the response.

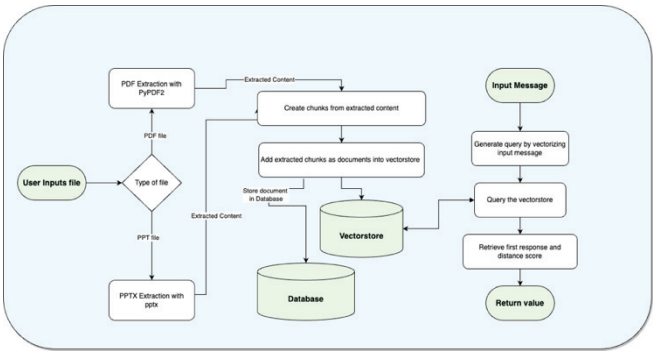

Fig. 3. Flow of Retrieval Augmented Generation

The addition of new documents as embeddings in the vector store, and the querying of the vectorstore for relevant content are the two main functions of this component of the system. Both these use a vectorstore database for these operations. To generate embeddings the MiniLM local LLM model is used for efficiency and accuracy. On querying, the vector store is searched, and relevant articles within a certain distance are added to the prompt for the system.

C. Database Persistence

Storing data in a high-level application is the most important part of a client-server system such as this. Databases are paramount to the functionality and success of chatbot applications for several compelling reasons. The ability to build a completely stateless server hinges on the most efficient use of the database reads and writes to maintain user state across sessions. They are instrumental in managing user authentication, supporting secure interactions, scalability, efficiency, and security of chatbot applications, serving as a foundational element for the seamless and dynamic operation.

Following in Fig. 4 is the database schema from the system. It contains five tables that are related to each other with a series of primary keys.

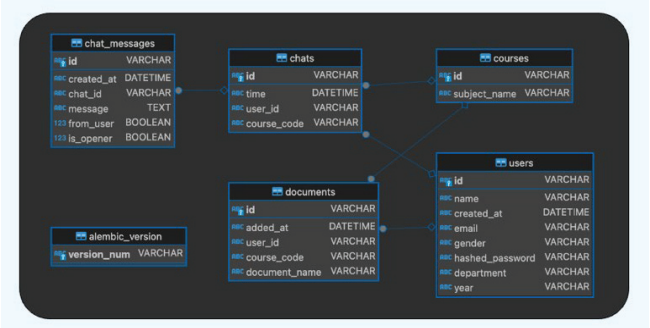

Fig. 4. Database Schema

Each table is denormalized but related to each other. All 'Users' have 'Chats' that group 'Chat Messages' associated with each 'Course' that they speak to the LLM model about. Each 'Course' has associated 'Documents' in the vectorstore, so that everything can be traced and easily identified. This leads to a balanced denormalized state of the database.

D. Mixture of Experts Architecture

The Mixture of Experts architecture [11] is a technique in machine learning where there are several expert models trained to each learn and act as an 'expert' a particular task. Then, for each input, n models are run to predict some output, which are combined for the final output.

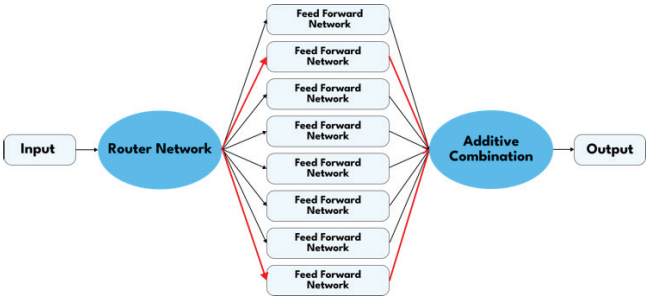

Fig. 5. Mixtral of Experts Architecture

This architecture differs from traditional Ensemble architectures since in an ensemble, all the different models are used, whereas in this, only one or a few specialized models are used. The division of the problem space is achieved by using a router which determines which tokens are sent to which network. This architecture is much faster than others in terms of quality to latency, as well as do better in a broad range of scenarios due to their sparse nature. In this system, the Mixtral 8x7B Mixture of Experts model is chosen due to the advantages of this architecture that have been listed out above.

IV. Results

The developed system can successfully be used concurrently by students to chat with the model and get robust answers with sources as described. The success of this system can be analyzed based on the responses recieved. Qualitatively, the system works quite well, as illustrated in Fig 6. to Fig. 8.

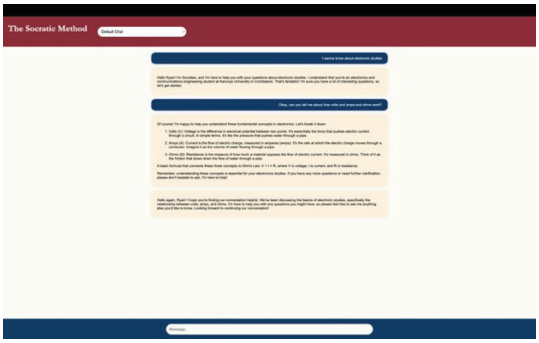

Fig. 6. Opening and basic chat on miscellaneous subjects

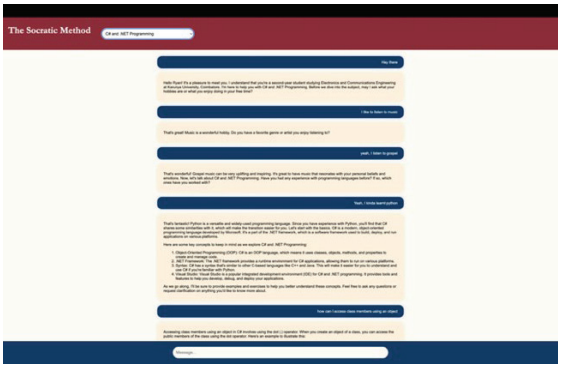

Fig. 7. Sample conversation on subject 'C# and .NET Programming'

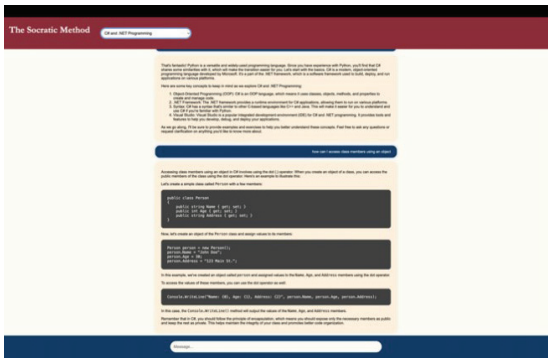

Fig. 8. Generation of C# code as an example during chat

V. Conclusion And Future Work

The undertaken research work endeavours to construct a sophisticated client-server model, improving the dynamics of educational interaction through the implementation of an advanced chatbot system. This system is meticulously designed to facilitate seamless communication between teacher's materials and students, helping in improving understanding with accurate information and proper context.

One future scope of this research is the introduction of multimodal means of communication between the student and LLM to improve the experience and learning of the student, as well as give better, more in-depth information to the model. This system will help improve the integration of Generative AI and Large Language models in educational settings.

References

[1] Matsuda, Koh & Frank, Ian. (2024). LangChain Unleashed: Advancing Education Beyond ChatGPT's Limits.

[2] Hiremath, Guruswami, et al. "Chatbot for education system." International Journal of Advance Research, Ideas and Innovations in Technology 4.3 (2018): 37-43.

[3] Al Muid, Md Abdullah, et al. "EduBot: An unsupervised domain-specific chatbot for educational institutions." Artificial Intelligence and Industrial Applications: Artificial Intelligence Techniques for Cyber-Physical, Digital Twin Systems and Engineering Applications. Springer International Publishing, 2021.

[4] Debnath, Babpu, and Aparna Agarwal. "A framework to implement AI-integrated chatbot in educational institutes." *Journal of Student Research* (2019).

[5] Jeon, Jaeho, and Seongyong Lee. "Large language models in education: A focus on the complementary relationship between human teachers and ChatGPT." *Education and Information Technologies* (2023): 1-20.

[6] Abedi, Mahyar, et al. "Beyond Traditional Teaching: The Potential of Large Language Models and Chatbots in Graduate Engineering Education." *Qeios* (2023).

[7] Ramírez, S. FastAPI [Computer software]. `https://github.com/tiangolo/fastapi`

[8] Chase, H. (2022). LangChain [Computer software]. `https://github.com/langchain-ai/langchain`

[9] Bayer, M. (2012). SQLAlchemy. In A. Brown & G. Wilson (Eds.), *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks* (pp. 217-234). Mountain View: aosabook.org.

[10] Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive NLP tasks." *Advances in Neural Information Processing Systems* 33 (2020): 9459-9474.\

[11] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation, vol. 3, no. 1*, pp. 79-87, March 1991, doi: 10.1162/neco.1991.3.1.79.