Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Engineering

Development and Comparative Evaluation of a Fine-Tuned Multimodal LLM Against Base Model and ChatGPT-4

Swetang Krishna

Supervisor: Prof. Vincent Wade

November 24, 2025

A dissertation submitted in partial fulfilment
of the requirements for the degree of
BAI (Computer Engineering)

Declaration

I hereby declare that this dissertation is entirely my work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open-access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signed: _____ Date: _____

Abstract

This dissertation delves into the transformative potential of chatbots in the educational sector, highlighting their role in personalizing learning experiences and providing administrative support. With an emphasis on student engagement and personalized support, it examines the integration of AI-driven interfaces into educational settings. The research explores how chatbots can evolve learning environments into more interactive and student-focused platforms. Furthermore, this study extends into the domain of artificial intelligence, evaluating the effectiveness of question-answering (QA) systems and developing a fine-tuned multimodal Large Language Model (LLM) specifically for context-based QA in probability and statistics. The investigation centres on three pivotal areas: a base model pre-trained on QA tasks, a fine-tuned multimodal LLM, and its comparison with the advanced ChatGPT-4 model. The development of the fine-tuned LLM encompasses the enhancement of a base model through Retrieval-Augmented Generation (RAG) techniques and specialized training on a dataset curated from probability and statistics queries, aiming to rival or surpass the proficiency of larger models like ChatGPT-4, while also addressing the computational and financial challenges associated with such models.

This project's comprehensive approach to development and evaluation illuminates the significant impact of fine-tuning, multimodal capabilities, and RAG on improving the model's ability to process and understand complex, domain-specific questions. The findings offer an in-depth analysis of the efficacy, accuracy, and efficiency of the developed model in comparison to existing models, underscoring the nuanced advantages and limitations of these methodologies in enhancing QA performance. By providing a detailed framework for the development and assessment of fine-tuned multimodal LLMs, this dissertation contributes to the broader AI and education fields. It paves the way for future AI-driven educational technologies, emphasizing the role of specialized, efficient, and accessible LLMs in tackling intricate QA challenges within educational contexts.

Lay Abstract

Imagine a classroom where every student gets immediate, personalized help on their studies, any time of the day - this is what chatbots in education can offer. This research project looks at how these smart, conversational computer programs, or chatbots, are changing the way we learn and manage educational tasks. They can tailor learning to each student's needs, answer questions instantly, and are always available, tackling common problems like keeping students interested and providing individual support.

Moreover, the study ventures further into the world of artificial intelligence by creating and testing a special kind of computer model designed to answer complex questions in a specific area: probability and statistics. This model is not just any model; it's built to be smarter and more efficient than some of the most advanced systems out there, like ChatGPT-4, but without needing as much computer power or money to run. This was achieved by starting with a basic model skilled in answering questions, enhancing it with advanced techniques, and training it on a special set of questions and answers.

The research aimed to find out how well this newly developed model can understand and answer tough questions compared to other big models. The results showed that by focusing on certain improvements, such as making the model work with different types of information and helping it retrieve relevant knowledge when needed, it could indeed handle complex, subject-specific questions very effectively.

This work doesn't just advance our understanding of artificial intelligence; it also opens up new possibilities for using smart technologies in education. It highlights how specialized, efficient computer models can be a powerful tool for overcoming challenges in learning and teaching, making high-quality education more accessible to everyone.

Acknowledgements

This project, culminating in May 2024, has been a journey of both challenge and discovery, none of which would have been possible without the unwavering support of several key individuals and groups. Foremost, I extend my deepest gratitude to Professor Vincent Wade, whose guidance and expertise were instrumental in navigating the complexities of this research. His patience, wisdom, and encouragement have left an indelible mark on my academic and personal growth.

To my family, whose belief in my capabilities never wavered, I owe a debt of gratitude. My parents, for their unconditional love and encouragement, and my sister, who has been my greatest cheerleader.

My heartfelt thanks go out to the friends I've been fortunate to make along this academic journey. Their camaraderie, support, and shared moments of stress and success have made this arduous journey a memorable one.

To my peers in the Class of 2023 and 2024, who have navigated the challenges of these times with resilience and solidarity, I am proud to have shared this journey with you. The friendships forged and the collective spirit of our cohort have enriched this experience beyond measure.

Lastly, I acknowledge the broader academic and research community, whose contributions have paved the way for this project. Their collective knowledge and innovations have been invaluable to my research.

Thank you all for your part in this journey, for the lessons learned, and for the memories cherished. Your support has been a beacon of light in the pursuit of knowledge and excellence.

Contents

List of Figures

1 Introduction

Artificial Intelligence (AI) has seen remarkable growth, especially within the realm of Natural Language Processing (NLP). This growth has facilitated the emergence of sophisticated applications that have revolutionized the way humans interact with technology and how machines interpret and process human language. Among these applications, chatbots represent a significant leap forward, evolving from simple text generators to complex entities capable of engaging in detailed conversations, answering questions, converting text to speech, and performing a multitude of tasks tailored to user needs.

This evolution of chatbots has been marked by the transition from basic text generation capabilities to the development of fully functional chatbots equipped with a wide array of features. These advanced chatbots serve various purposes across different sectors, showcasing their versatility and potential. Applications range from customer service bots that assist users by providing timely information and resolving queries to specialized tools like Programmer AI which supports coding and software development processes, and content generation bots that aid in creating diverse forms of written content. The breadth of these applications underscores the transformative impact of AI and NLP technologies across industries.

In the context of these advancements, our focus has shifted towards leveraging AI and NLP in the education sector, leading to the creation of an educational agent. This agent is envisioned as a digital study companion designed to enhance and facilitate the learning process. By integrating AI and machine learning algorithms, the educational agent assumes multiple roles - it can act as a tutor providing personalized instructions, a facilitator guiding learners through educational content, an advisor for academic and career planning, and a companion offering support and motivation throughout the learner's journey.

The educational agent is tailored to address the individual needs and learning preferences of users, providing customized recommendations. This personalized approach ensures a more engaging and effective learning experience, highlighting the potential of AI and NLP to transform educational methodologies and outcomes. Through the development of such educational agents, we aim to harness the power of AI to enrich the educational landscape, making learning more accessible, personalized, and efficient for students worldwide.

1.1 Motivation

The advent of Artificial Intelligence (AI) and Natural Language Processing (NLP) in education marks a pivotal shift towards interactive and personalized learning environments. The evolution from simple text-based interfaces to advanced educational agents encapsulates the technological strides made in AI, transforming how learners engage with educational content. Today, these agents serve not just as tools for information retrieval but as comprehensive educational companions, capable of providing tailored guidance and support. This leap forward is facilitated by significant advancements in AI's capacity to process and understand human language, making educational agents more intuitive and accessible.

The current technological ecosystem offers tools that simplify the creation of these agents, enabling educators and learners to develop bespoke educational platforms without extensive coding knowledge. This democratization of technology paves the way for widespread adoption in educational settings, promising a more personalized learning experience. However, this rapid integration also brings to the fore challenges like data privacy and the need for equitable access, underscoring the importance of navigating these issues as we harness AI to redefine education. Motivated by these developments, our project explores the potential of educational agents to enhance learning, aiming to make education more engaging, accessible, and effective for all.

1.2 Objective

The project's objective is to construct and assess an innovative educational agent, poised to redefine the standards of educational technology. Our vision encompasses the creation of a bespoke educational tool, meticulously designed to compare favorably against both a foundational model and the sophisticated capabilities of ChatGPT-4. This endeavor entails the integration of a finely tuned question-answering language model, enhanced by advanced text summarization and precise document querying features. Our goal is to forge an educational agent of unparalleled efficiency, capable of delivering queries from a curated document list on a compact scale. The incorporation of the Whoosh library for document retrieval, coupled with a custom-tailored language model on a Django framework, enables us to provide personalized, contextually aware responses aimed at augmenting understanding, bolstering knowledge retention, and customizing learning experiences to individual needs. This initiative seeks to bridge the educational divide, granting students seamless access to a vast repository of knowledge.

Embarking on this journey, we aim to craft a fully operational educational agent, harnessing minimal resources to rival the performance of extensive language models. Through the application of Retrieval-Augmented Generation (RAG) techniques and meticulous

fine-tuning, we have developed a mini-software that directs our model to the appropriate content within the document list.

The project is structured around seven key objectives:

1. Compilation of a custom dataset featuring context, question, and answer columns.

2. Identification and selection of an appropriate base model specialized in question answering.

3. Fine-tuning of the language model using a bespoke dataset focused on probability and statistics questions.

4. Development of a web interface to facilitate user interaction with the educational agent.

5. Establishment of a database to archive interactions between the user and the language model.

6. Creation of a mini-software tool leveraging the Whoosh library for efficient document indexing and retrieval.

7. Evaluation of the language model's performance through user experience metrics and a reward-based system.

The initial phase involved curating a specialized dataset for the base model. Despite the seemingly straightforward nature of this task, it presented significant challenges, including sourcing and converting data from the internet into a format digestible by our language model. A comprehensive review of existing educational agents and their training datasets was conducted to pinpoint gaps and opportunities for innovation. This process entailed gathering insights from various sources, including academic papers and user testimonials, to compile a robust dataset.

Subsequent phases focused on identifying a suitable language model with constraints on computational resources and the capacity for question answering. The selection process required experimentation with various models to find one that met our criteria for efficiency and ease of fine-tuning. Once selected, the base model underwent fine-tuning with the prepared dataset to enhance its ability to deliver descriptive responses.

Further steps included the creation of a web application to serve as the interface for our educational agent, facilitating user interactions and model evaluation. A comprehensive database schema was also developed to store user details and interaction histories, which was integrated into the web application.

The project's penultimate objective involved the development of a software tool for document context retrieval, optimized for relevance and efficiency. Finally, the project

culminates in the evaluation of the language model's performance, focusing on user experience and a novel reward system, rather than direct comparisons between the base and fine-tuned models. This comprehensive approach underscores our commitment to advancing educational technology through innovative solutions.

1.3 Technical Approach

The technologies, tools, environment and libraries used were either selected from the research done, some due to previous experience and few due to personal preference.

Our toolkit is diverse and comprehensive. We've utilized WampServer for our database needs, Anaconda for a virtual environment, Google-Colab for our computational work like fine-tuning and model testing, Django as our web framework, alongside other essential tools like Visual Studio Code, and libraries from the Hugging Face ecosystem.

Languages used were Python for back-end and HTML, CSS, and JavaScript for front-end development. The framework used was Django, a Python-based framework for building web applications.

1.4 Project Report Overview

The report is meticulously structured to navigate through the inception, development, and culmination of the project, summing up a comprehensive analysis and application of chatbot technology as an educational agent with question-answering abilities.

Chapter 1: Introduction - This initial segment offers a compact overview, laying the foundation with an introduction that outlines the project's motivation and the problem statement, objectives, and the technical approach adopted.

Chapter 2: State of the Art - In this chapter, an exhaustive review is conducted, surveying the landscape of current educational agents, advanced chatbots, and multimodal systems with question-answering capabilities. A comparative analysis is performed, particularly focusing on advancements and applications in educational agents. This serves to identify both the strengths and gaps within the current state of the art.

Chapter 3: Design - Reflecting on insights garnered from the comprehensive review in Chapter 2, this section delves into the specific design requirements of the project. It outlines the foundational requirements derived from the comparative analysis and discusses the logical framework for chatbot design. This chapter is pivotal in detailing the design process, including the rationale behind design decisions, the stages of development and setting the milestones is also discussed in this section, and the design was broken down into

milestones.

Chapter 4: Implementation - In this chapter, we dive into how we turned our design ideas from the previous chapter into a working chatbot. We walk through the technology choices we made, explaining why we picked them and how we integrated them to work together. We also take a closer look at how we designed the chatbot's user interface in Django and the database with WAMP, we will look at the process of data collection, fine-tuning the base model and the thought behind the chosen model. To bring it all to life, we share a real-life example that shows the steps we took to build the chatbot and tackle any challenges we faced along the way.

Chapter 5: Testing and Evaluation - In this chapter, our attention turns to the detailed testing and evaluation of the educational agent. Instead of merely comparing responses from the language models, we conducted a thorough examination of every phase of the project. Through statistical analysis and practical observations gathered during the project's execution, we assessed the design, architecture, and data used to train the chatbot. This evaluation aimed to critically analyze the chatbot's functionality and how well it meets the project goals. Based on our findings, we identified areas for future improvements. The evaluation methodology adopted a unique approach by assigning ChatGPT-4 a baseline score of 10/10. This allowed us to gauge the educational agent's responses in comparison, offering a relative measure of performance.

Chapter 6: Conclusions - This final chapter brings together the essential insights, results, and achievements of the project, alongside considerations for enhancing the design's scalability. It outlines the significant contributions made to the domain and explores possibilities for further research and development. The chapter also reflects on the obstacles faced during the project and proposes strategies for addressing such challenges in the future. Concluding remarks provide a comprehensive summary of the project's journey, shedding light on its impact on the evolution of educational chatbot technologies. This includes the prospect of imbuing educational agents with emotional and social intelligence, tailoring them to meet the specific needs of users.

2 Literature Review

In this Chapter, we will discuss the Overview of Current Technologies or Theories, Recent Research and Developments, Comparative Analysis of the leading approaches, limitations, gaps, and challenges that current state-of-the-art technologies, Trends and Evolutions, and References. The platforms used for research and development in the domain were Medium, Google Scholar, and LinkedIn.

2.1 Background

Before delving deeper into our discussion, it's crucial to define what chatbots are, particularly for those who might be unfamiliar with the concept within the realm of Artificial Intelligence (AI). At first, glance, interacting with chatbots may seem like exchanging messages with a robotic entity trying to mimic human behaviour. However, at their core, chatbots are software applications designed to predict and assemble words in a sequence that aims to replicate the flow of humans.

Earlier when chatbots were first introduced they were known as 'Dialogue Systems'. Chatbots have been trained on large corpses of data that contain texts from different subjects, They learn all the texts and then try to predict what word from their knowledge base would fit to make sense in the situation.

There are a variety of chatbots, Chatbots are categorized based on their functionalities, the technology they use, and their interaction methods. Each category of chatbots serves different needs and purposes, from simple task automation to providing complex, personalized user experiences. The development and implementation of chatbots depend on the specific goals of an organization and the needs of its users. Here in Fig 1, we can see a mind map representing different categories of chatbots as given in Fig 2.1.

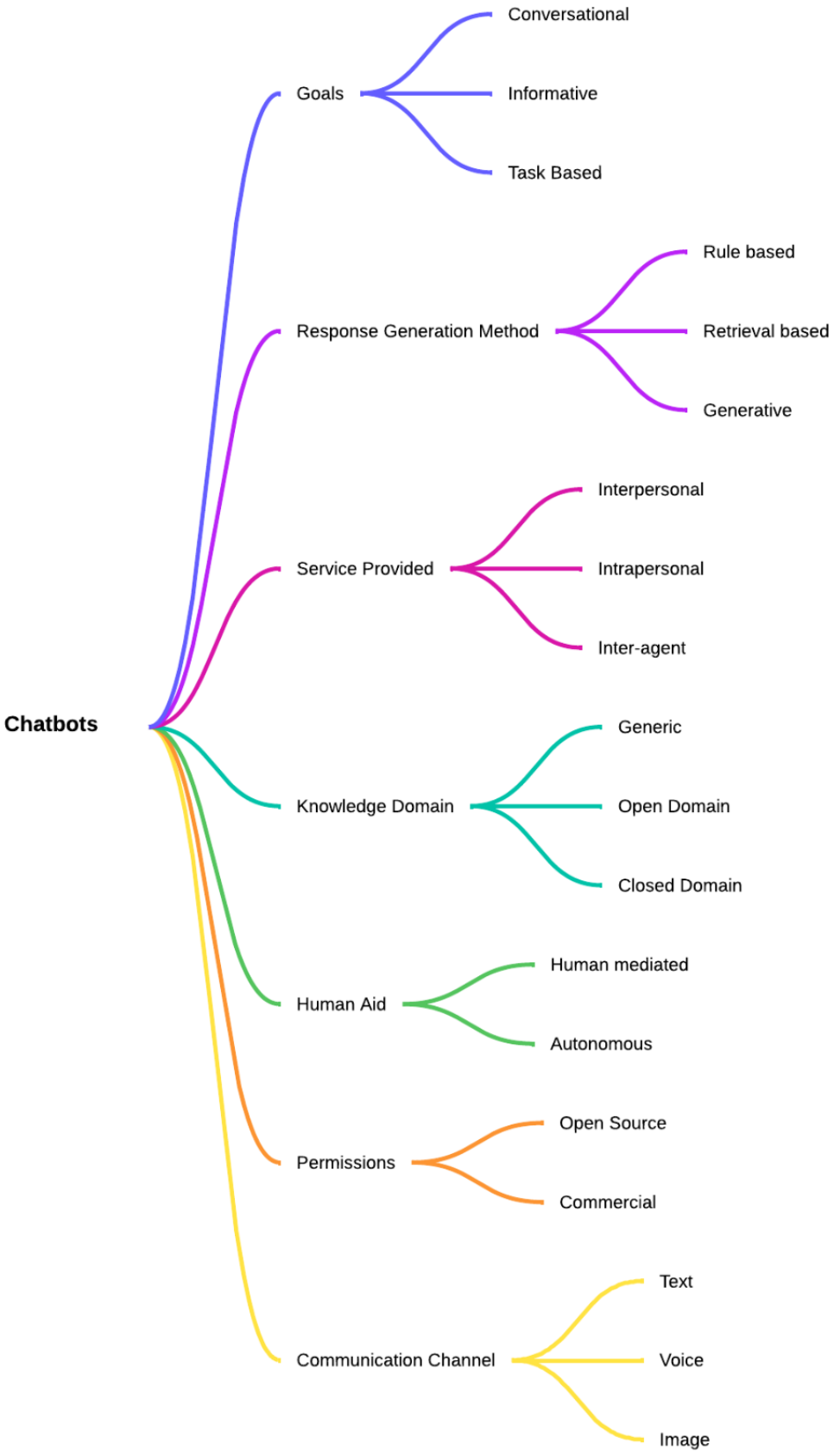

Figure 2.1: Categories of Chatbots [2]

Let us discuss a few examples to understand the different categories of chatbots.

Conversational

A chatbot on a social media platform that can chit-chat with users about daily topics like the weather or news.

Informative

A chatbot is used by a public library to inform users about book availability and library hours.

Task-Based

A chatbot on a banking website that helps users pay bills, transfer money, or check account balances.

Rule-Based Chatbots

A restaurant's chatbot helps customers place orders by asking them to choose from a menu, specify their order, and provide delivery details, all through a structured set of options.

Retrieval-Based

A FAQ chatbot for a software product that retrieves answers from a set of predefined responses to help users troubleshoot common problems.

Generative

A chatbot that helps users write creative stories by generating original content based on the user's input and plot choices.

Interpersonal

A mental wellness chatbot that provides daily affirmations and engages users in conversations about their feelings and well-being.

Intrapersonal

A personal finance chatbot that helps users set and track financial goals and spending habits.

Inter-agent

A logistics chatbot that coordinates with various warehouse bots to update inventory levels and track shipments.

Generic

A virtual assistant chatbot that can perform a variety of tasks like setting reminders, answering general knowledge questions, or giving directions.

Open Domain

A trivia chatbot that can converse on a wide array of general knowledge topics without specific expertise.

Closed Domain

A medical chatbot that answers health-related questions and provides advice within the confines of medical knowledge.

Human Mediated

A technical support chatbot that initially attempts to solve user issues can escalate the session to a human agent if the problem is too complex.

Autonomous

A hotel booking chatbot that can handle the entire booking process from inquiry to confirmation without human intervention.

Open Source

A chatbot platform that is available on GitHub, allowing developers to contribute to its code and customize it for various uses.

Commercial

A proprietary restaurant reservation chatbot service that restaurants subscribe to for handling their table bookings.

Text

A chat widget on various websites that allows users to type their questions and receive text responses.

Voice

A voice-activated assistant on smartphones that users can talk to for making calls, setting alarms, or searching the internet.

Image

A chatbot in a mobile app that analyzes photos of clothing and suggests similar items for purchase from an online store.

2.1.1 History

The history of chatbots dates back to 1950 (Turing A.M. Computing Machinery and Intelligence). Alan Turing wondered if a computer program could think and talk to a group of people. His ideas laid the groundwork for what would become chatbot technology.

Turing aimed to sidestep the philosophical debate about the nature of mind and consciousness and provide a clear, operational test for machine intelligence. He did this by reframing the question of whether machines can think to whether machines can imitate human behaviour indistinguishably. This rephrasing led to the formulation of what he called the "Imitation Game," which we now know as the Turing Test. Turing test is considered by many to be the generative idea of chatbots. Turing test is a game of imitation, often referred to as the "Imitation Game," which involves three participants: a computer, a human, and an interrogator. The interrogator stays in a separate room, away from the computer and the human. The goal of the interrogator is to determine which of the other two participants is human and which is a machine. (Weizenbaum J. ELIZA—A computer program for the study of natural language communication between man and machine)

In 1966, Joseph Weizenbaum at MIT created ELIZA, the first chatbot ever developed. ELIZA used pattern matching and substitution methodology to simulate conversation and could create the illusion of understanding, although it had no built-in framework for contextualizing events.

ELIZA was a simple program that mimicked conversation by using pattern matching to generate responses, whereas today's chatbots leverage sophisticated artificial intelligence, including natural language processing and machine learning, to deliver personalized, context-aware interactions. Modern chatbots can maintain the context of a conversation over time, learn from past interactions to improve their performance, and seamlessly integrate with various communication platforms, providing services ranging from customer support to personal assistance. This evolution reflects significant advancements in computational linguistics and AI technology, allowing modern chatbots to offer a user experience that is remarkably more dynamic and interactive compared to the basic

capabilities of their predecessors.

From simple scripts to advanced AI, chatbots have significantly evolved, becoming more integrated into the fabric of digital interaction and continuing to advance with improvements in Natural Language Processing.

2.1.2 Developing a Chatbot

Developing a chatbot involves various approaches, each with its unique methodologies, tools, and technologies. These approaches can be broadly categorized based on the complexity of tasks the chatbot is designed to perform. There are two primary types based on the underlying algorithms and techniques used, "Pattern Matching" and "Machine Learning".

Pattern Matching

Encompasses both rule-based and retrieval-based methods under a broader umbrella. This category highlights the reliance on identifiable patterns in user input to guide the bot's responses, emphasizing the deterministic nature of the interaction.

Machine Learning

Aligns with the more advanced capabilities provided by machine learning and generative models, focusing on the bot's ability to understand natural language, learn from interactions, and generate novel responses based on learned patterns rather than pre-defined rules.

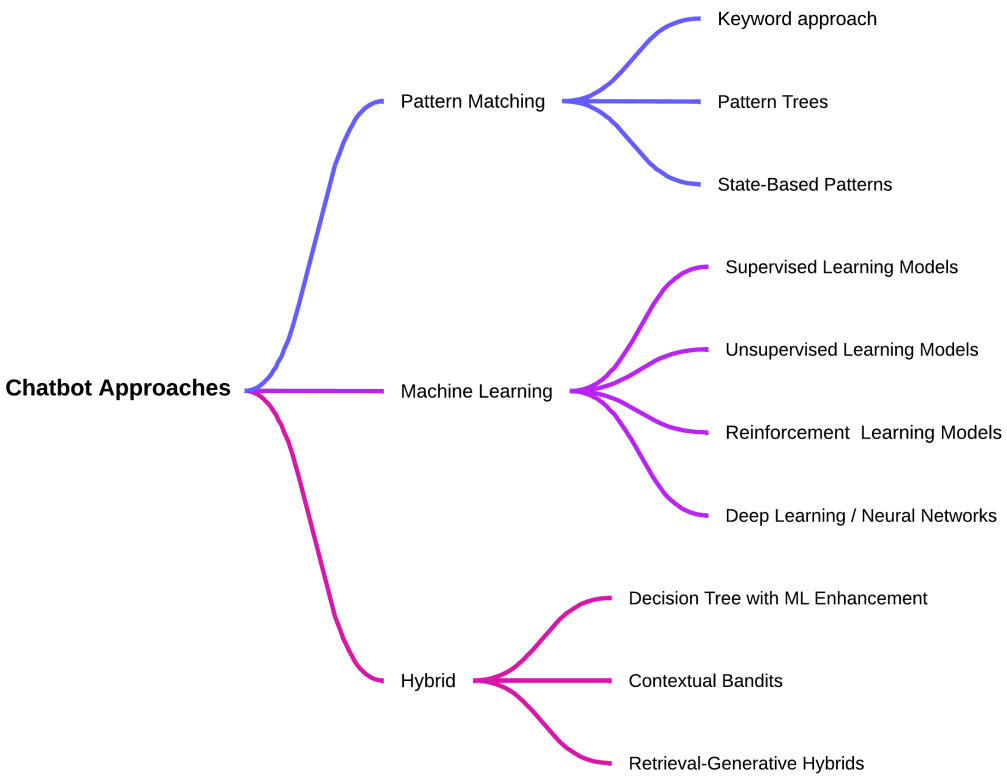

Figure 2.2: Approaches to develop a Chatbot [2]

2.1.3 Chatbot Applications Across Sectors

Customer Service and Support

In the realm of customer service, chatbots have emerged as pivotal tools, offering immediate response capabilities to user inquiries, efficiently addressing routine issues, and directing more complex matters to human agents. This technological advancement significantly boosts customer satisfaction by guaranteeing swift and precise answers around the clock.

E-commerce and Retail

Within the e-commerce industry, chatbots play a vital role in guiding customers through extensive product catalogues, providing customized recommendations, and streamlining the purchasing journey. Their involvement extends to order tracking and return facilitations, thereby elevating the consumer shopping experience.

Banking and Financial Services

The banking sector witnessed a transformative impact through chatbot integration, which delivers personalized financial advice, facilitates transactions, and offers instant account information. This innovation fosters a frictionless banking experience, empowering customers to manage their finances effortlessly.

Healthcare

In healthcare, chatbots serve as instrumental gateways to medical guidance, appointment scheduling, medication reminders, and preliminary diagnostic support. Their deployment aims at enhancing healthcare accessibility and optimizing patient care management.

Education and Learning

Chatbots in the educational sphere function as interactive tutors, delivering personalized support across various subjects and enriching the educational journey with tailored content. This represents a leap forward in fostering engaging and customized learning experiences.

Travel and Hospitality

Travel chatbots assist with the reservation of flights and accommodations, offer insights into destinations, and issue timely travel updates, thereby smoothing out the travel planning process and enriching the hospitality experience.

Human Resources and Recruitment

In human resources and recruitment, chatbots streamline candidate screening, swiftly address employment inquiries, and organize interview schedules, rendering the recruitment workflow more efficient for employers and candidates alike.

Entertainment and Media

Chatbots within the entertainment and media sector tailor recommendations for movies, music, or games, and facilitate bookings and reservations, thus amplifying user engagement with content and events.

Personal Assistants

Personal assistant chatbots play a crucial role in daily schedule management, reminder settings, information retrieval, and task execution, contributing significantly to personal organization and productivity enhancement.

Social Media Management

In social media management, chatbots automate interactions, handling everything from frequently asked questions to audience engagement and content management, aiding brands in sustaining an active and captivating online presence.

Real Estate

Chatbots in real estate aid prospective buyers by offering property listings, arranging viewings, and addressing inquiries, thereby streamlining the property search and acquisition journey.

Feedback Collection and Surveys

Chatbots revolutionize feedback gathering and survey administration by automating these processes, furnishing businesses with critical insights into customer satisfaction and potential areas for enhancement.

Event Planning and Management

Event-focused chatbots provide essential event information, manage registrations, and outline scheduling, simplifying event logistics and improving the attendee experience.

2.1.4 Full-text search principles [1]

Full-text search refers to a comprehensive and sophisticated method for searching within the complete content of documents. It allows users to input text queries and retrieve documents that contain those queries, handling vast amounts of text quickly and efficiently. Full-text search is a fundamental feature of various applications, such as search engines, digital libraries, and document management systems, enabling them to provide relevant results based on the actual content of the documents.

Components of full-text search

Indexing

Indexing is the process of analyzing the text of documents and organizing the information in a way that makes it fast to search. This usually involves creating an inverted index, which is a data structure that maps significant words or tokens found in the documents to their locations in the text. Indexing makes it possible to quickly find all occurrences of a word or phrase within the entire dataset.

Tokenization

During indexing, the text is broken down into individual units called tokens. Tokenization involves parsing the text into words, phrases, or other meaningful elements that can be used in search queries. This step is crucial for understanding the structure and content of the text.

Stemming and Lemmatization

These processes are applied to reduce words to their base or root form. Stemming chops off the ends of words in a heuristic manner, while lemmatization involves using vocabulary and morphological analysis of words to remove inflectional endings. Both are used to improve the chances that different forms of the same word (e.g., "run", "running") are matched during a search.

Stop Words Removal

Common words such as "the", "is", and "at", which appear frequently but don't contribute much to the meaning of the text, are often removed from the index. This step helps in focusing on the more meaningful words that are likely to be used in queries.

Ranking and Relevance

Full-text search systems often incorporate algorithms to rank the results based on their relevance to the search query. This can involve complex calculations that take into account factors such as the frequency of query terms in the document (term frequency), the importance of the terms across all documents (inverse document frequency), and the proximity of query terms within documents.

Search Query Processing

When a search query is received, the system processes this query like how the documents were indexed. This includes tokenization, and possibly stemming or lemmatization, to ensure that the query terms match the indexed terms as closely as possible.

Boolean Queries, Phrase Searches, and Proximity Searches

Full-text search supports complex queries that can include Boolean operators (AND, OR, NOT), exact phrases (" "), and proximity searches that look for terms appearing close to each other within the text. This allows users to formulate precise queries to find exactly what they're looking for.

2.2 AI in Education

From the era of chalk dust to the precision of digital styluses, technology has continuously reshaped the landscape of education.

Artificial Intelligence (AI) in education encompasses a broad spectrum of applications designed to enhance both teaching and learning experiences. The integration of AI technologies, such as machine learning (ML) and natural language processing (NLP), adaptive learning systems, and AI-based grading and test assessments, revolutionizes educational methodologies by offering personalized, engaging, and efficient learning pathways. [3] These technologies enable the analysis of data to identify patterns and make predictions, facilitating a personalized learning experience for each student and aiding educators in customizing learning material to suit individual learning styles [4].

AI also supports the administrative side of education, automating tasks such as grading and attendance tracking, thereby saving time and enhancing efficiency. Intelligent tutoring systems, chatbots, and automated assessment tools offer consistent and accurate feedback, further enriching the educational ecosystem. [5]

However, the implementation of AI in education faces several challenges, including privacy and security concerns, the cost of AI technologies, potential biases in AI algorithms, and ethical considerations around accessibility and fairness. Addressing these challenges is crucial for realizing the full potential of AI in enhancing educational outcomes. [6]

In conclusion, AI in education promises a transformative impact on how education is delivered and received, offering personalized, efficient, and interactive learning experiences. The potential benefits of AI in education are significant, ranging from improved student outcomes to enhanced administrative efficiency. Nevertheless, it is essential to navigate the challenges associated with AI's integration into educational settings to ensure a beneficial and equitable impact on the educational landscape.

2.3 The Role of Chatbots in Education

In the contemporary educational landscape, the emergence of chatbot technology represents a paradigm shift towards more interactive and personalized learning experiences. This dissertation chapter examines the multifaceted role of chatbots within the educational sector, highlighting their capacity to redefine traditional learning environments and support mechanisms. Central to this discussion is the premise that chatbots, powered by advancements in artificial intelligence (AI), offer unprecedented opportunities for enhancing student engagement, facilitating personalized learning, and streamlining administrative tasks.

2.3.1 Enhancing Teaching and Learning Through Personalized Pathways

Artificial Intelligence (AI) in education heralds a new era of customized teaching and learning, with technologies like machine learning and natural language processing paving the way for adaptive learning systems, personalized learning experiences, and AI-driven assessments. These innovations enable educators to tailor learning materials to individual students' needs, optimizing the educational process for effectiveness and engagement.

2.3.2 Personalized Learning Experiences

The essence of AI in education lies in its capacity to create highly personalized learning experiences that adapt to the pace and style of each student. AI technologies can analyze vast amounts of data to identify patterns and predict learning outcomes, offering students tailored learning pathways that enhance their understanding and retention of knowledge. Studies highlight AI's potential to revolutionize e-learning through virtual tutors and adaptive learning platforms, significantly improving student engagement and outcomes.

One good example of a personalized tool being used in the education sector is. It is a web-based tool tailored for humanities students to effectively summarize their lecture transcripts and to personalize the summaries to their specific needs. [7]

2.3.3 Administrative Efficiency

AI significantly enhances the efficiency of educational administration by automating routine tasks such as grading, attendance tracking, and personalized feedback. This not only saves time but also ensures accuracy and consistency in assessments and feedback, creating a more supportive learning environment for students. [8]

2.3.4 Challenges and Ethical Considerations

While the benefits of AI in education are substantial, several challenges must be navigated to realize its full potential. Privacy and security, cost, algorithmic bias, and ethical concerns about accessibility and fairness must be addressed. Ensuring equitable access and responsible use of AI technologies in education is critical for harnessing their benefits without exacerbating existing inequalities. [9]

2.4 Natural Language Processing Techniques in Education

Natural Language Processing, or NLP, is a part of artificial intelligence that helps computers understand, interpret, and respond to human language. This technology is a game-changer in education, making it possible for computers to read, analyze, and even generate text like a human would. It's helping to create smarter, more personalized learning experiences for students everywhere.

Text Classification

Text classification is a process where an NLP model assigns a category or label to a given text. In the educational domain, text classification can automate the sorting of academic papers into fields and specialities, facilitating the organization and retrieval of literature for students and researchers. For instance, Huang discussed how NLP technology, through processes like word segmentation and synonym analysis, enhances the retrieval accuracy of educational resources, allowing for a more efficient search based on user requirements. [10]

Token Classification

Token classification, such as Named Entity Recognition (NER), involves labelling specific words or phrases within a larger text corpus. An educational application is the identification of key concepts and terms within learning materials, thus aiding content analysis and helping educators highlight important information for students. Swapnil Raj and Mrinal Paliwal examine how NLP facilitates understanding in educational settings, particularly through its ability to address language barriers between educators and learners. [11]

Table Question Answering

In education, table question answering can significantly aid data literacy, enabling students to ask and retrieve specific information from structured data sets, such as statistical tables or research data. NLP models specialized in this area can help students extract and comprehend complex data without extensive training in data analysis techniques.

Question Answering

Question-answering systems can assist in educational environments by providing students with immediate, accurate answers to their queries. These systems can draw from a vast pool of knowledge, ranging from textbook databases to online educational resources, offering a level of interaction and responsiveness akin to a personal tutor.

A research paper titled "Reasoning with Language Models and Knowledge Graphs for Question Answering" talks about QA-GNN, an innovative model blending Graph Neural Networks with Large Language Models and Knowledge Graphs, aimed at enhancing educational chatbots. Through encoding QA contexts and integrating a relevance scoring mechanism, QA-GNN achieves a nuanced understanding and processing of educational content. Evaluated across various domains, it outperforms existing models, showcasing its potential to revolutionize educational assistance by offering precise, context-aware answers. This advancement underscores the symbiotic potential of AI technologies in redefining educational paradigms, highlighting the importance of tailored, efficient learning experiences. [12]

Zero-Shot Classification

Zero-shot classification is particularly useful in educational contexts when students or researchers encounter subjects with sparse training data, such as niche fields of study. This technique can classify content or questions that the model has never explicitly learned, thereby facilitating the organization of new and emerging topics in academia.

Meta-tuning emerges as a groundbreaking approach to refine the zero-shot learning capabilities of large language models like GPT-3, by fine-tuning them on a vast meta-dataset. This method not only surpasses existing benchmarks but also unveils the critical role of model size in performance enhancement. It underscores the untapped potential in zero-shot learning, advocating for unified dataset formats and community collaboration to optimize language models further. [13]

Translation

Translation services, powered by NLP, are crucial in multilingual education settings, allowing for cross-language understanding and access to a broader range of academic materials. Students and educators can access resources in their native languages, which enhances comprehension and facilitates a more inclusive learning environment.

Exploring the untapped potential of large language models (LLMs) in translation, this study introduces the MAPS framework, embodying Multi-Aspect Prompting and Selection, to closely mimic human translation strategies. By guiding LLMs to analyze source texts and extract key information on keywords, topics, and demonstrations, MAPS significantly refines translation quality, outperforming conventional models and previous state-of-the-art systems in various languages. Comprehensive evaluations, both automatic and human, underscore MAPS's ability to enhance accuracy and reduce errors, spotlighting the value of human-like preparatory steps in machine translation. This work paves the way for future advancements, suggesting that further exploration into LLMs' translation processes could yield even more

sophisticated and nuanced translation capabilities. [14]

Summarization

Automatic summarization tools help students quickly grasp the key points of lengthy academic texts, research papers, or book chapters. This can be particularly beneficial for learners who need to review extensive materials within limited time frames, as highlighted by a comprehensive survey which explores the advancements in abstractive text summarization facilitated by pre-trained language models (PLMs), emphasizing the transition from traditional methods to deep learning techniques that mimic human summarization processes. Highlighting the superiority of PLMs in handling various summarization tasks, the study analyzes these models quantitatively and qualitatively, identifying performance-boosting strategies such as domain adaptation, model augmentation, stable finetuning, and data augmentation. The research underscores the challenges in fine-tuning PLMs and proposes solutions to enhance abstractive summarization systems, offering insights for future innovations in the field. [15]

Feature Extraction

Feature extraction in NLP is used to identify key linguistic patterns within educational texts. It can, for example, detect the complexity of language used, or identify stylistic features that differentiate academic writing from informal text. Such analysis can assist in the automated grading of student essays or the development of language learning tools.

Text Generation

Text generation capabilities can support creative writing exercises or generate study materials. For example, an NLP system could automatically produce essay prompts or generate example sentences to illustrate grammar rules, thereby enhancing language learning experiences. Now, we can also control the attributes of the generated texts like politeness, formality, sentiment, etc.; demographic attributes of the person writing the text such as gender, age, etc.; content such as information, keywords, entities, etc.; ordering of information, events, like plot summaries etc. Controlling various attributes of text generation has manifold applications. [16]

Text2Text Generation

Text2Text generation might be employed to paraphrase complex texts into simpler language, aiding comprehension for younger students or those with learning difficulties. It can also be used to adapt educational content to various reading levels or learning styles.

Fill-Mask

The fill-mask task is particularly relevant in language learning applications where students are prompted to complete sentences with the correct word, testing their vocabulary and grammar skills. NLP models can generate such exercises dynamically, providing personalized practice that adapts to a student's learning progress.

Sentence Similarity

Sentence similarity tools can support peer review systems by comparing student submissions to evaluate originality or by matching student questions to existing answers in discussion forums, enhancing the efficiency of online learning platforms.

2.4.1 Existing RAG Chatbots

Retrieval-augmented generation (RAG) chatbots represent a groundbreaking fusion of cutting-edge machine learning and vast information repositories. These intelligent systems redefine the frontiers of digital interaction by drawing from a pool of real-time, up-to-date knowledge to deliver responses that are not only contextually rich and nuanced but also verifiable and informed.

RAG is significant because it addresses challenges related to the static nature of LLMs' training data, which can become outdated and lead to responses that are either incorrect or based on non-authoritative sources. By incorporating an information retrieval step, RAG equips LLMs with the ability to access the most current facts, thereby enhancing user trust through improved response quality and verifiability [17] [18] [19].

In practice, RAG chatbots work by first processing a user query and then searching an external knowledge base to retrieve information relevant to that query. This information is combined with the LLM's internal knowledge to generate a response. By doing so, RAG chatbots are not only providing the most current information but are also capable of citing their sources, which increases their reliability and the users' confidence in the generated answers [17].

Enterprises can benefit from RAG in various ways. For example, RAG can power chatbots that provide specific product information, enhance customer service with precise and current information, and improve internal enterprise search functions for technical documentation and policies. Additionally, RAG helps maintain data privacy and reduces the occurrences of "hallucinations" — where LLMs generate convincing but incorrect responses — by grounding the model's responses in factual information. [19]

NVIDIA's RAG pipeline is an example of an architecture that can be deployed, showcasing how RAG can be integrated into AI applications. It consists of phases such as document

ingestion, pre-processing, embedding generation, and storing these embeddings in vector databases for quick retrieval during user interactions. [19]

These developments represent the pinnacle of current RAG chatbot technology, providing a framework for chatbots that can engage in natural interactions with users by leveraging real-time data access, maintaining privacy, and delivering verifiable information. For further information and in-depth technical explanations on RAG and its applications.

2.4.2 Fine-tuned LLMs

Fine-tuned Large Language Models (LLMs) stand at the pinnacle of AI's interpretive and generative power, pushing the boundaries of what machines can understand and express. With precision sharpened by domain-specific refinement, these sophisticated AI constructs offer a tailored intelligence, capable of nuanced interaction, complex problem-solving, and generating insightful content that resonates with human inquiry and creativity.

Models like GPT-4 by OpenAI and Claude v1 by Anthropic are leading the way with their remarkable abilities in complex reasoning, advanced coding capability, and high performance in benchmark tests. Claude v1, for example, is noted for its high scores in tests like MT-Bench and MMLU, surpassing some other models in specific scenarios. [20]

Another noteworthy development is the Zephyr-7B model created by Hugging Face. It leverages fine-tuning processes such as Distilled Supervised Fine-Tuning (dSFT) and AI Feedback through Preferences (AIF) to optimize its performance and reliability. [21]

Additionally, Google's PaLM 2 and Meta's LLaMA 2 models are demonstrating the power of fine-tuning with improvements in speed, parameter efficiency, and support for numerous languages. PaLM 2, while not multimodal like GPT-4, has added capabilities in specific domains like medicine with Med-PaLM 2. [22]

In the open-source domain, models such as Falcon, Guanaco, and Vicuna have become key players. Falcon is notable for being released under a permissive Apache 2.0 license, allowing commercial use, and has models trained on up to 40 billion parameters. Guanaco has introduced innovative fine-tuning techniques like QLoRA, allowing for efficient memory usage while preserving performance, and it has performed impressively on various benchmarks. [20]

2.4.3 Limitations

Large Language Models (LLMs), has made significant progress, there are inherent limitations that need to be acknowledged:

Generalizability vs. Specialization Trade-off

Fine-tuned LLMs, despite their precision in specific domains, may lack the broad applicability of more generalized models. The specialization that allows for accuracy in certain contexts can also limit their use in others where the fine-tuning may not align with the required knowledge base. [20] [21]

Data Privacy and Security

As with any AI system that handles real-time data, RAG chatbots must navigate complex privacy and security landscapes. The integration of external databases raises concerns about the handling of sensitive information and the potential for data breaches [17] [19].

Ethical and Societal Implications

Fine-tuning processes such as distilled supervised fine-tuning (dSFT) and AI feedback can sometimes lead to biases in model responses. These biases may reflect the data on which the models are trained, which can perpetuate stereotypes or unfair representations. [21]

Maintenance and Upkeep

Both RAG chatbots and fine-tuned LLMs require continuous updates to their knowledge bases and fine-tuning datasets to stay relevant and accurate. This ongoing maintenance is resource-intensive and may not be sustainable for all organizations. [17] [18]

Complexity and Resource Intensity

Developing and deploying RAG systems or fine-tuned LLMs often demands significant computational power and expertise, which can be barriers for smaller enterprises or independent developers. [18] [19]

Model Interpretability and Explainability

The internal workings of these advanced AI models can be opaque, making it challenging to interpret or explain their decision-making processes. This "black box" issue can complicate efforts to audit or validate the models' responses. [18]

Dependence on Large Datasets

The efficacy of LLMs relies heavily on the availability of large, high-quality datasets. Obtaining such datasets is often difficult, especially for less-represented languages and specialized domains. [22]

Potential for Misuse

The advanced capabilities of these models also bring the risk of misuse, such as generating misinformation or being employed in manipulative ways. Ensuring responsible use is an ongoing challenge for the AI community. [22]

2.4.4 Terminologies

Retrieval-Augmented Generation (RAG)

A process that enhances the output of large language models by referencing external, authoritative knowledge bases before generating responses. [17]

Fine-Tuned Large Language Models (LLMs)

These are advanced AI models that have been specifically adjusted or "fine-tuned" on a smaller, specialized dataset to perform well on specific tasks. [20]

Distilled Supervised Fine-Tuning (dSFT)

A method that involves training a model on high-quality instructions and responses generated by a teacher language model, allowing for more efficient learning. [21]

AI Feedback through Preferences (AIF)

A technique where human feedback is collected to assess and enhance the quality of model responses. In the context of AI, it often involves using preferences from a teacher model. [21]

Distilled Direct Preference Optimization (dDPO)

A strategy aimed at refining models by maximizing the likelihood of ranking preferred responses higher, often utilizing feedback data for optimization. [21]

Transformer-based Models

A type of neural network architecture that has become the foundation for most recent advancements in NLP and generative AI. It's renowned for its efficiency in handling sequences of data, like text . [22]

Generative AI

Refers to the use of machine learning models, especially LLMs, to generate new content, including text, images, or music, that mimics human-like creativity. [22]

Context Window Size

The maximum length of the input that a model can handle at one time, is measured in tokens. Larger context windows allow models to consider more information when generating responses. [22]

Embeddings

High-dimensional vectors that represent text in numerical form, allowing for the efficient processing and comparison of textual data. [19]

Model Hallucinations

Occurrences where a model generates convincing but factually incorrect or nonsensical responses. [19]

Open-Source LLMs

Models that are made publicly available for anyone to use, modify, and distribute. They are essential for democratizing access to advanced AI technologies. [20]

Reinforcement Learning from Human Feedback (RLHF)

A training method where models are fine-tuned based on human preferences or corrections, improving their alignment with human values or expectations. [22]

3 Design

The design part of the educational agent will be described in this chapter of our dissertation. It is broken down into six sections. Starting with receiving the query up to generating a response.

3.1 System Architecture

Let us take a look at the overview of the design. It's a meticulous process that begins with a user query. This is the starting point, where the interplay between the user and our system initiates. From here, our Whoosh library takes charge, sifting through a vast database to find the most relevant documents.

Once identified, the selected document becomes the 'context' for our query. This context, alongside the query, is then distilled by a text summarizer, which essentially prepares the input for our fine-tuned LLM.

This is where the magic happens – our fine-tuned LLM specifically sharpened for probability and statistics, processes the input. And finally, a precise and informed response is generated, which marks the culmination of a sophisticated query-answering journey.

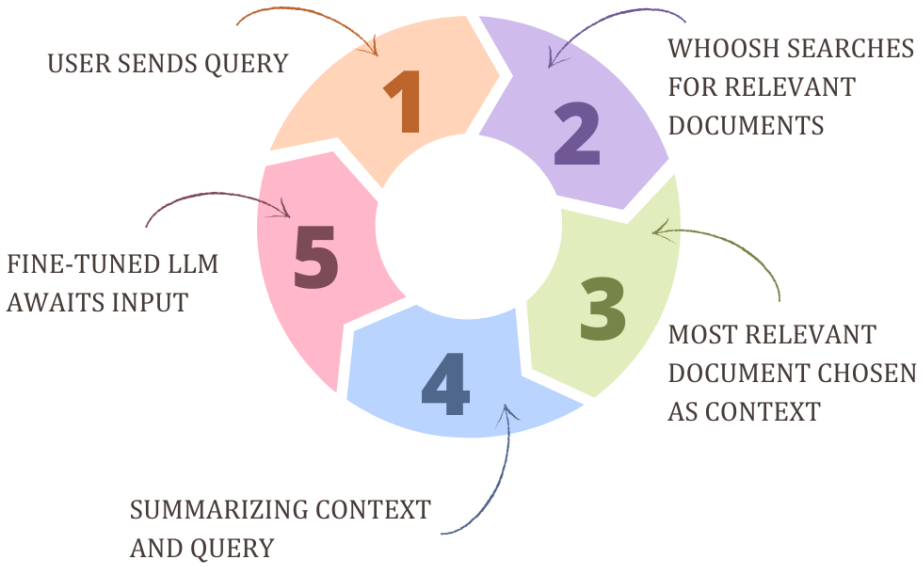

Figure 3.1: Design overview

3.1.1 Analysis of the State of the art

We're developing an educational bot designed to tackle user inquiries through document analysis. By merging full-text search techniques with a Question-answer model, we're enhancing its context-aware response capability. Our strategy includes using fine-tuning along with RAG for superior performance.

Our educational agent's design is crafted to offer an interactive and intelligent platform, enabling users to find answers to complex queries with ease and accuracy. At the heart of this system lies a meticulously engineered process that bridges the gap between vast databases of information and the specific needs of the user. Through a harmonious fusion of advanced search technologies and language processing models, we've created a workflow that not only understands the essence of user queries but also delivers concise and relevant responses. This design narrative begins with the user's curiosity and unfolds through a series of sophisticated technological layers, each contributing to the final goal of providing an enlightening and satisfying user experience. Let's delve into the details of this journey, starting from the initial user query to the delivery of a tailored response.

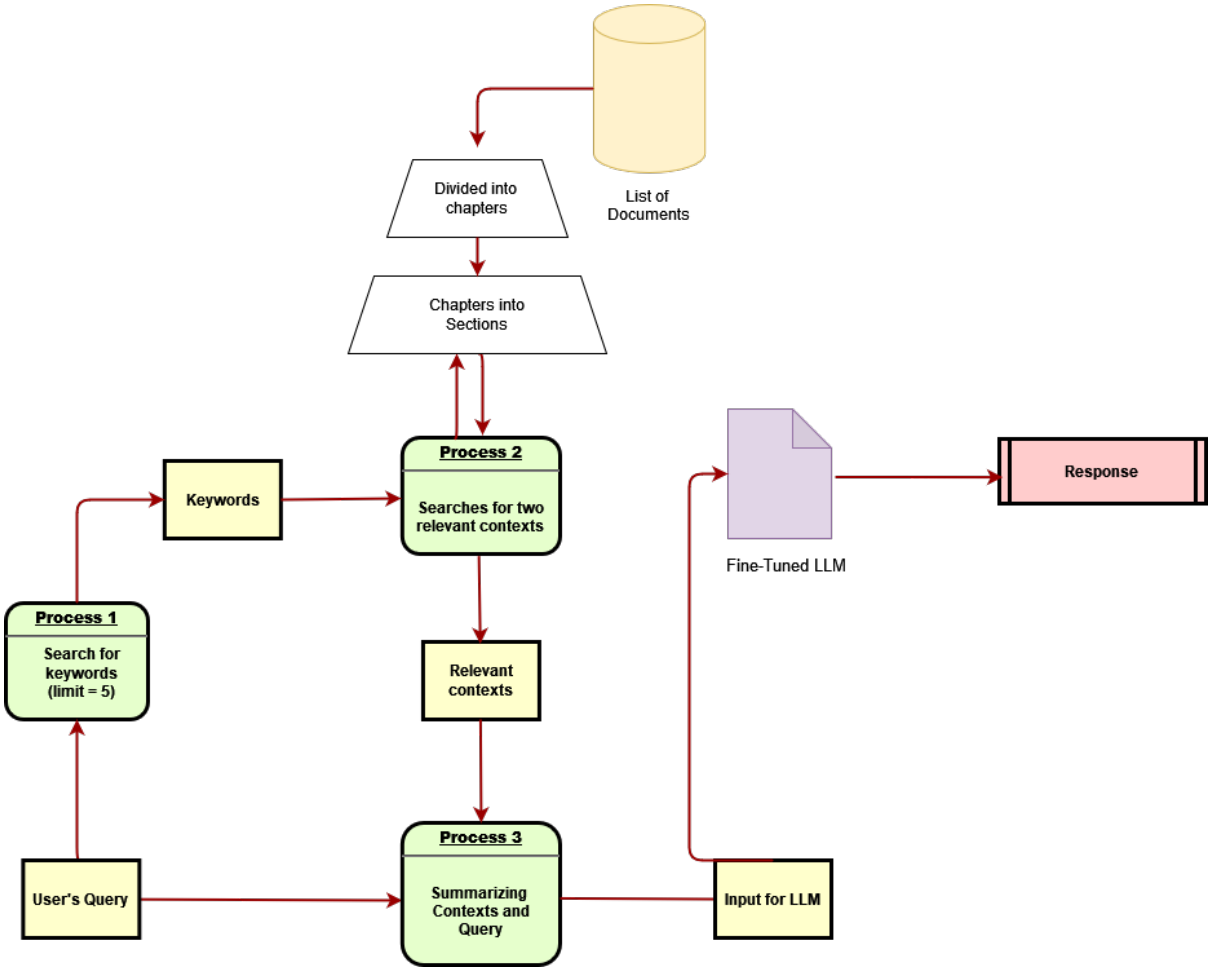

Figure 3.2: Query-Response Flow

3.2 Document Retrieval System

In a full-text search, the query interacts with the inverted indexes through a process designed to efficiently find all instances of a search term within a database or a collection of documents.

This part of the chapter discusses the design principles underpinning full-text search mechanisms, emphasizing the role of inverted indexes. Transforming unstructured text into a structured form that can be queried efficiently, inverted indexes serve as the backbone of modern search engines.

3.2.1 Inverted Index Creation

An inverted index is a data structure used to store a mapping from content keywords to their locations in a database of documents. It lists each word appearing in the documents and records the list of documents or specific locations within documents where each word occurs. It is the first step in enabling efficient full-text searches across large data sets.

3.2.2 Query Processing Workflow

Query Analysis

Upon receiving a search query, the system parses and analyzes the query to identify the keywords or phrases to be searched. This phase may involve several preprocessing steps, including:

Normalization: Converting all terms to a standard format, typically lowercase, to ensure consistency in matching.

Stemming: Reducing words to their root form, thereby grouping different forms of a word.

Stopword Removal: Eliminating common words that are unlikely to be useful in finding relevant documents.

Index Lookup

The search engine consults the inverted index to locate the documents containing the query terms. This lookup is efficient, as the index directly maps terms to document locations without necessitating a full scan of the document corpus. For queries comprising multiple terms or incorporating boolean logic, the system retrieves and combines lists from the index as dictated by the query's structure.

Ranking and Relevance

Given that searches may yield a multitude of documents, it is crucial to rank these documents by relevance to the query. The ranking algorithm assesses factors such as the frequency of query terms in each document (term frequency), the significance of the terms across the document corpus (inverse document frequency), and the proximity of query terms within documents. The outcome is a score that signifies each document's relevance, guiding the order in which search results are presented.

3.2.3 Results Presentation

The final step involves presenting the ranked list of documents to the user, typically highlighting query terms within the context of the document excerpts. This presentation enables users to quickly assess the relevance of each result and select the most pertinent documents for their needs.

Imagine searching a vast digital library with ease, thanks to the magic of inverted indexes. This smart system breaks down the user's query, consults a detailed map of words, and

ranks results by relevance, guiding you straight to the information you seek. It's like having a librarian who knows exactly where everything is.

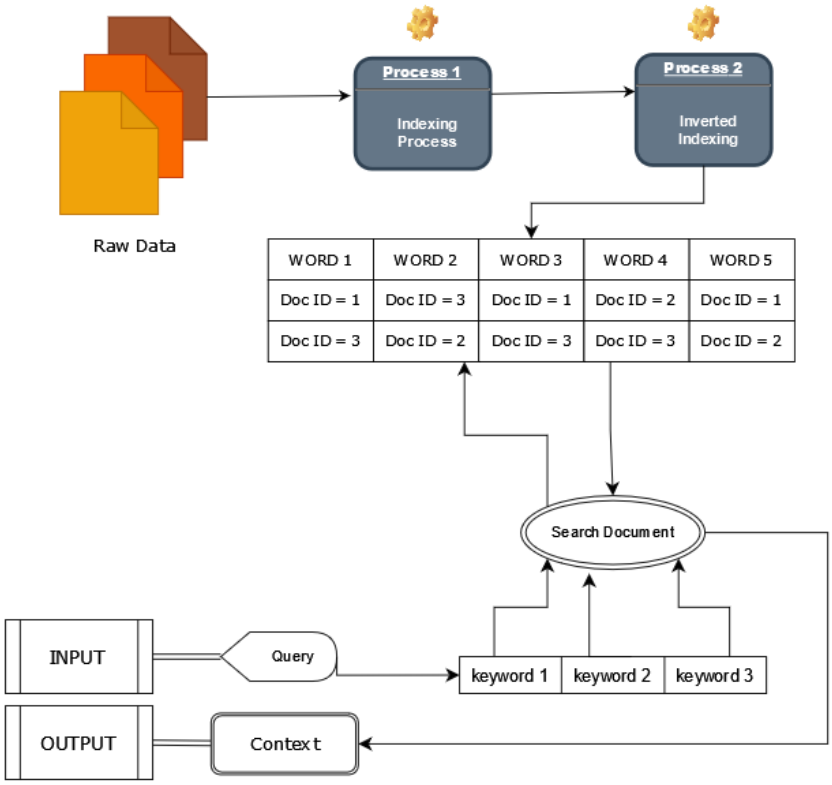

Figure 3.3: Full-text Search

3.3 Language Model Integration

The language model is seamlessly integrated with our platform's educational content, enabling it to draw upon a vast repository of knowledge. This integration supports the model's capability to offer contextually relevant information, answer questions accurately, and provide personalized learning experiences.

3.3.1 Fine Tuning

To tailor the language model to our specific educational needs, we embarked on a fine-tuning process, leveraging a custom dataset rich in educational content, ranging from basic concepts to advanced topics in various subjects. This dataset included structured

question-answer pairs and explanatory text, enabling the model to learn the nuances of educational discourse and improve its ability to deliver precise, informative responses.

Our fine-tuning approach was iterative, allowing us to refine the model's performance based on continuous feedback loops. We incorporated adjustments to ensure the language model could interpret complex queries, engage users in meaningful educational dialogues, and provide explanations that are accessible.

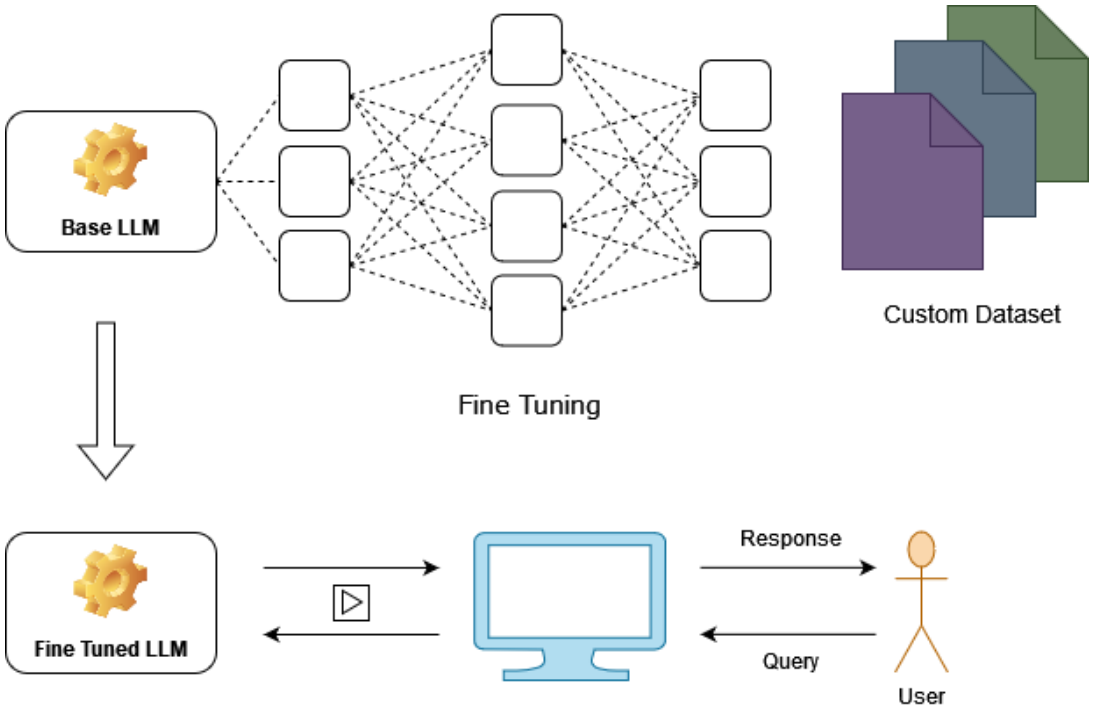

Figure 3.4: Model Integration

3.4 Current challenges and Future Directions

With innovation comes the challenge. We're tasked with the development of a sophisticated prompt generator to facilitate dynamic engagement, the adoption of Elasticsearch in future for scalability, personalized education, and the enhancement of problem-solving abilities within our AI agents. This can be done by bringing a custom tokenization method to the picture.

The need for advanced keyword search designs remains paramount for efficient information retrieval, allowing us to optimize keyword search functionality through advanced prioritization and relevance assessment.

We address the crux of our technical challenge: enhancing keyword search functionality. We aim to refine our system to not just search, but to understand the intent and context of queries, ensuring relevance and precision in the information retrieved.

Our proposed solution involves a priority system based on uniqueness and relevance, a dynamic model that adapts and learns continuously to ensure that the educational content provided is not just accurate, but truly beneficial for the learner's journey.

4 Implementation

In this chapter, we are going to delve into the detailed implementation phase of the project. We will begin by outlining the crucial milestones that were pivotal for the project's fruition. The project was systematically segmented into several key stages to ensure a structured and efficient approach to development. Initially, our efforts were concentrated on collecting training data, which laid the foundational groundwork for our project. Subsequently, we shifted our focus to selecting a suitable base model that aligns with our project's objectives and requirements. Following this, the next step involved fine-tuning our chosen base model to enhance its performance and adaptability to our specific needs.

Further advancing in our project timeline, we dedicated resources to creating a web application utilizing Django, a decision motivated by Django's robustness and scalability for web development. Parallelly, we embarked on the development of a mini software designed to meticulously search for keywords within queries, a functionality essential for the interactive aspect of our project. In addition to this, we also developed a program powered by Whoosh, aiming to efficiently locate relevant contexts within a document, thereby augmenting the project's capability to provide contextual information.

The culmination of these individual milestones was the integration of all these components, which facilitated the development of a functional educational agent. This agent is envisioned to serve as an innovative tool in the educational domain, offering a new dimension of interactive learning and information retrieval.

4.1 Tools, Technologies and Libraries

let us explore the diverse and comprehensive toolkit that was instrumental in our project's completion. We initiated our setup by employing WampServer, which served as the backbone for our database requirements, ensuring a stable and efficient data management system. For our computational tasks, we leveraged the capabilities of Google Colab, and Anaconda provided the necessary environment for running our intensive data processing and machine learning models.

Moving forward, Django was chosen as our web framework due to its versatility and support for rapid development, allowing us to construct a robust web application that could seamlessly integrate with the other components of our project. Additionally, we utilized a range of other essential tools, including Visual Studio Code, which offered a conducive development environment for writing and testing our code.

Furthermore, our project benefited significantly from the integration of libraries from the Hugging Face ecosystem. These libraries furnished us with advanced functionalities for natural language processing, enhancing the intelligence and responsiveness of our educational agent. Collectively, these tools and libraries formed the cornerstone of our project, enabling us to achieve our objectives and deliver a sophisticated educational tool.

4.1.1 ChatGPT-4

ChatGPT-4 was utilized for the generation of question-answer pairs from a curated list of documents. We commenced this phase by uploading more than 30 documents, setting the stage for ChatGPT-4 to perform its task. The directive given to ChatGPT-4 was to generate questions based on the content of each document and to provide answers to these questions separately and descriptively. Through this method, we successfully generated an estimated 100-200 questions per document, amassing a substantial dataset for our purposes.

The subject matter chosen for the focus of our question-answering model was "Probability and Statistics." This decision was guided by the relevance and complexity of the topic, which offered a rich domain for our educational agent to explore and interact with. This initiative not only contributed to the enhancement of our model's capabilities in understanding and processing statistical information but also enriched the pool of educational content available for users interacting with the agent.

4.1.2 Django

We used Django, it is a high-level Python web framework that encourages rapid development and clean, pragmatic design. [23]

It can be used for almost any type of website, from social networks to news sites, and from content management systems to scientific computing platforms. Its versatility makes it suitable for both simple and complex projects.

Django follows the "Don't Repeat Yourself" (DRY) principle, which promotes the reusability of components, reducing the time and effort required to develop new applications. Its object-oriented design and a wealth of pre-built modules allow for quick development from concept to completion.

4.2 Data Gathering

We are going to detail the rigorous approach adopted for the collection of training data, a foundational step critical for the success of our project. The process began with the extraction of information from a myriad of sources, with Wikipedia standing out as a primary reservoir of knowledge. This extracted data, initially in diverse formats, was systematically converted into a uniform text format to facilitate ease of processing.

Further into the development phase, ChatGPT emerged as an instrumental tool, assuming a pivotal role in the augmentation of our training dataset. By generating questions and answers based on the information gleaned from our sources, ChatGPT significantly enriched our training set, providing a deeper, more nuanced layer of data for our project. This enhancement was crucial, setting the stage for a more robust and intelligent system capable of understanding and interacting with the complexities of human language.

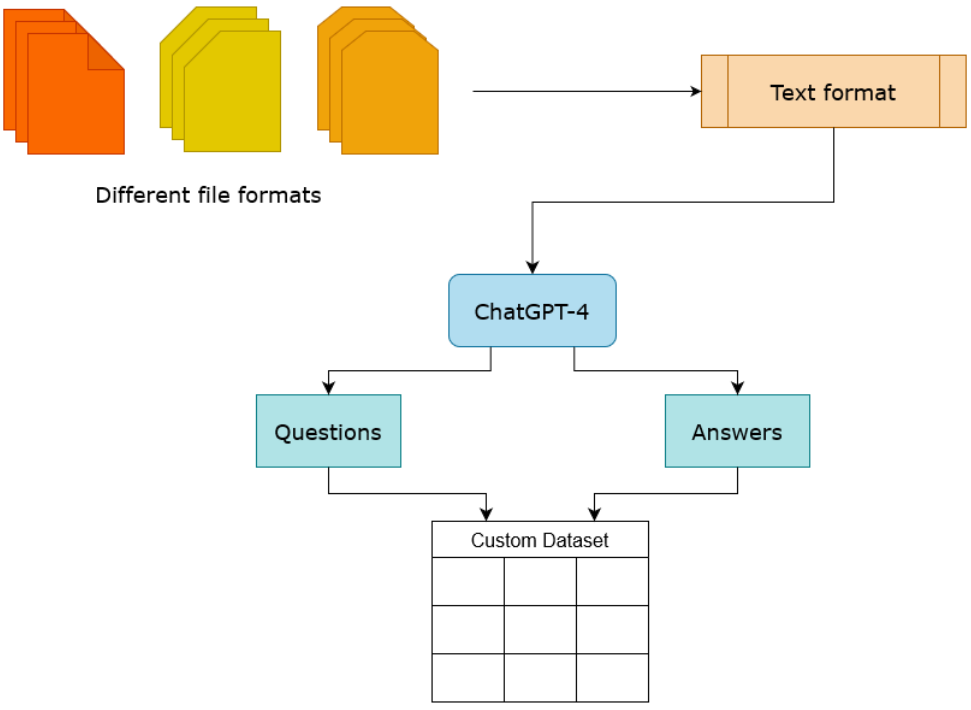

Figure 4.1: Custom Dataset Creation

4.3 Selecting a Base Model

We will delve into the critical process of model selection, a phase that requires meticulous attention to align with the specific needs of our project. It was imperative not just to choose any model, but to identify one that was precisely attuned to our project's objectives. The task at hand was to develop a system capable of answering user queries effectively, necessitating expertise in natural language processing (NLP).

After careful consideration, we selected the 'Roberta-base-squad2' model from the Hugging Face platform. This model is renowned for its NLP capabilities, particularly in question-answering within given contexts. The choice of 'Roberta-base-squad2' was motivated by its proven proficiency in understanding and processing complex queries, making it an ideal candidate for our requirements. This decision was pivotal, ensuring that our system was equipped with a robust and capable NLP foundation, essential for addressing the sophisticated demands of our project.

This model represents an advanced iteration of the 'Roberta-base' model, which has been fine-tuned using the SQuAD 2.0 dataset—a challenging dataset that includes both answerable and unanswerable questions. This fine-tuning process has prepared the model exceptionally well for the task of extractive Question Answering (QA).

Model Specifications:

Language Model: Roberta-base

Language: English

Downstream Task: Extractive QA

Training and Evaluation Data: Both the training and evaluation were conducted using the SQuAD 2.0 dataset.

Technical Details:

Infrastructure: Utilized 4x Tesla V100 GPUs for training, ensuring robust computational power.

Hyperparameters:

Batch Size: 96

Number of Epochs: 2

Base Language Model: "Roberta-base"

Maximum Sequence Length: 386

Learning Rate: 3e-5

Learning Rate Schedule: LinearWarmup

Warmup Proportion: 0.2

Document Stride: 128

Maximum Query Length: 64

Performance Metrics:

Exact Match: 79.87

F1 Score: 82.91

Detailed performance for answerable questions ('HasAns exact': 77.94, 'HasAns f1': 84.03) and unanswerable questions ('NoAns exact': 81.80, 'NoAns f1': 81.80).

Usage:

This model can be integrated into NLP frameworks like Haystack for scaled question-answering across multiple documents. In the Haystack environment, it can be loaded using either FARMReader or TransformersReader configurations, supporting diverse operational needs. Additionally, using the Transformers library, the model and tokenizer can be directly employed to process QA inputs, offering streamlined access to robust QA capabilities.

By leveraging the 'Roberta-base-squad2' model, our project harnesses top-tier NLP technology to create a dynamic and responsive educational agent, capable of effectively processing and responding to complex query contexts, thereby enhancing the learning experience.

4.4 Fine-Tuning LLM

In this detailed exploration, we delve into the nuanced process of fine-tuning our foundational Large Language Model (LLM), a pivotal step in realizing the ambitious objectives set forth by our project. The odyssey commenced with the meticulous preparation of our dataset, a phase where precision and foresight played crucial roles. Initially, the dataset was uploaded onto our secure drive, a manoeuvre designed to ensure easy access and manipulation. After this, we transitioned the dataset into the Colab environment, a move executed with the utmost care to guarantee seamless integration. This initial phase was not merely procedural but essential in ensuring the dataset's compatibility with the complex requirements of the

Hugging Face ecosystem. By achieving this, we laid down a robust foundation for the intricate steps that followed, setting the stage for a smooth fine-tuning process.

Venturing further into our fine-tuning journey, we encountered the critical task of tokenization. This process is fundamental to the model's understanding and interaction with the dataset. By converting the raw text into a structured array of vectors, we facilitated a form of communication that the LLM could comprehend, effectively narrowing the chasm between human linguistic nuances and machine interpretation. This transformation was not just a technical requirement but a bridge facilitating the LLM's ability to learn and adapt from the dataset presented, a cornerstone in the path towards achieving a model that can mimic human language patterns with high fidelity.

With the dataset now perfectly aligned and tokenized for the LLM's consumption, we advanced to the crucial stage of setting the training arguments. This step was approached with a strategic mindset, aligning our parameters meticulously with the guidelines and recommendations outlined in the model's official documentation. Such alignment was paramount to ensure that our training regimen resonated well with the model's inherent design and parameters, fostering an environment conducive to effective learning.

Following the meticulous configuration of our training parameters, we initiated the trainer, integrating our carefully chosen model and the primed dataset into this framework. This initiation signified the beginning of the fine-tuning phase, a critical period of adaptation and learning for the LLM. The fine-tuning process was both rigorous and enlightening, culminating in the achievement of a model characterized by negligible loss—a testament to the efficacy of our training strategy. This remarkable outcome was not just a milestone but a clear indicator of a highly successful training endeavour, compelling us to safeguard the refined model by saving it onto our drive and readying it for future deployment and the myriad of possibilities that lie ahead.

A Step by Step explanation:

First, we had to get our data ready. This meant uploading it to our drive and then moving it to the Colab environment. It was important to do this carefully to make sure the data would work well with the Hugging Face tools we were using.

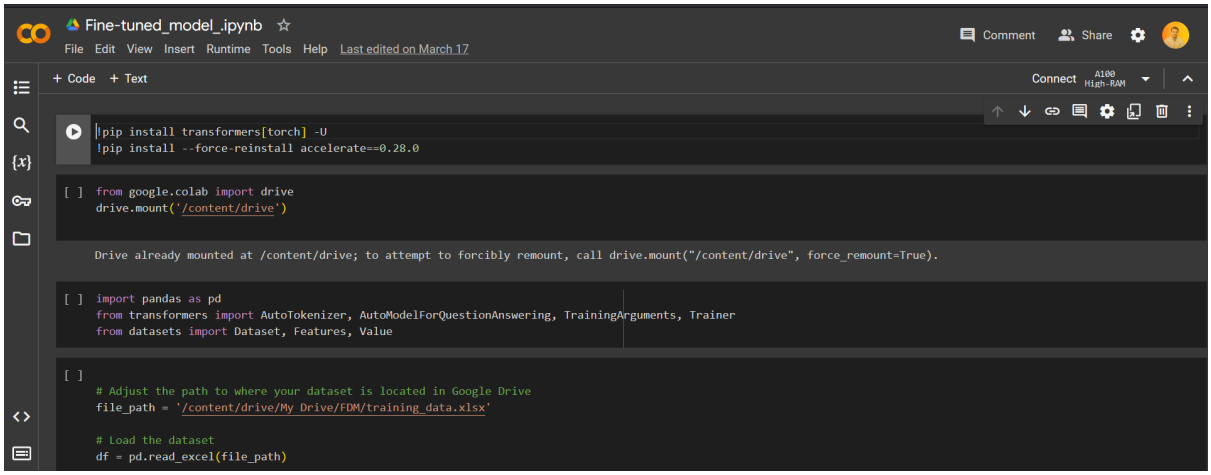

Figure 4.2: Preparing Our Dataset

Next, we turned our dataset into a format the LLM could understand. We did this by converting the text into vectors, which are like a language that both humans and machines can understand. This step is crucial because it helps the model learn from our data.

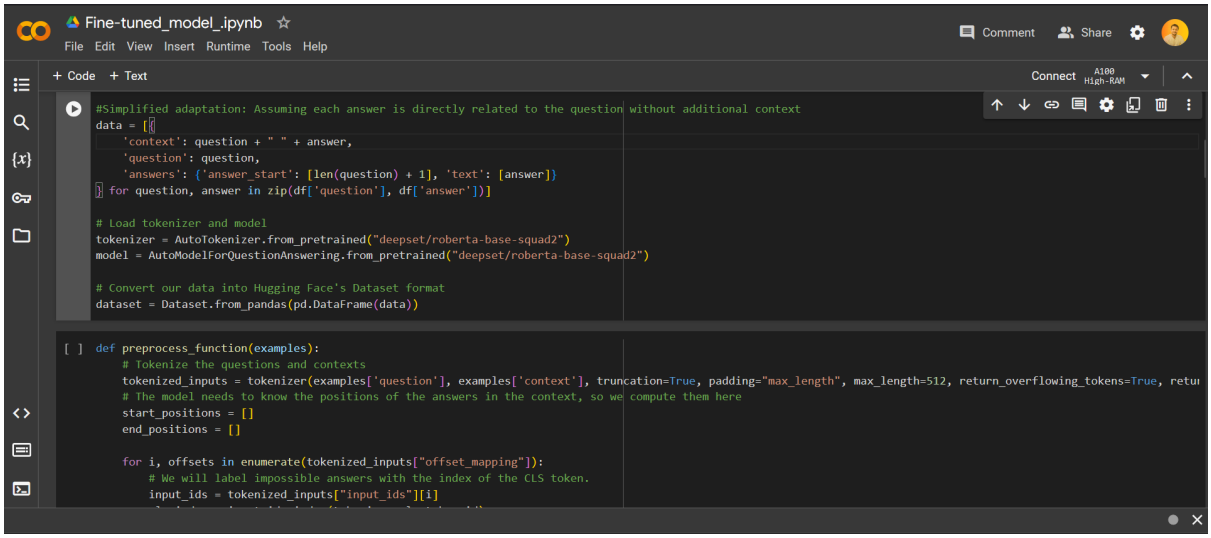

Figure 4.3: Tokenizing the Data

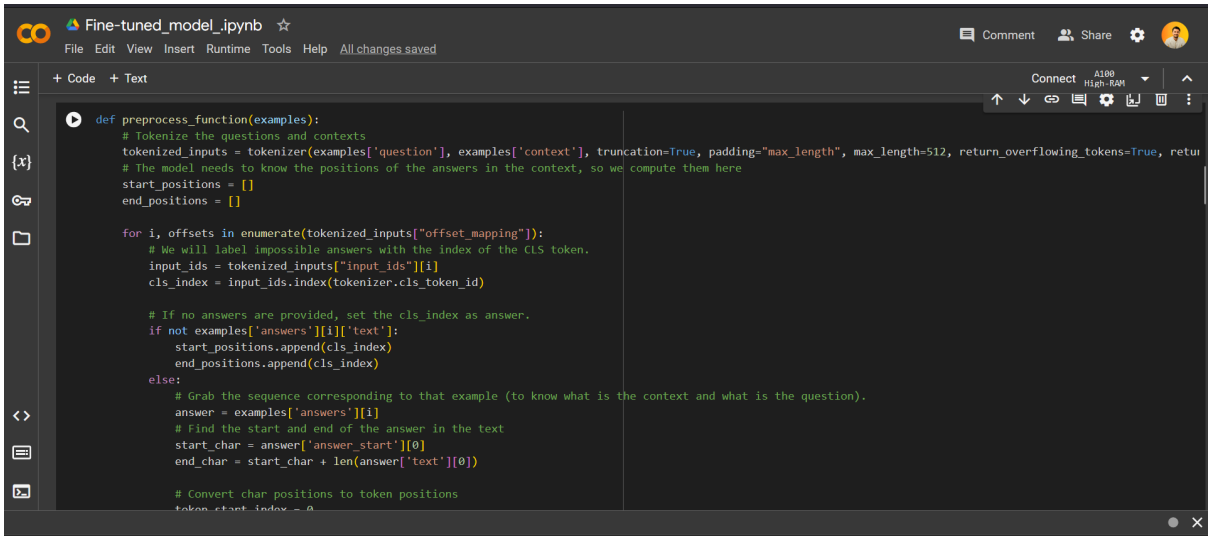

Figure 4.4: Preprocessing Function

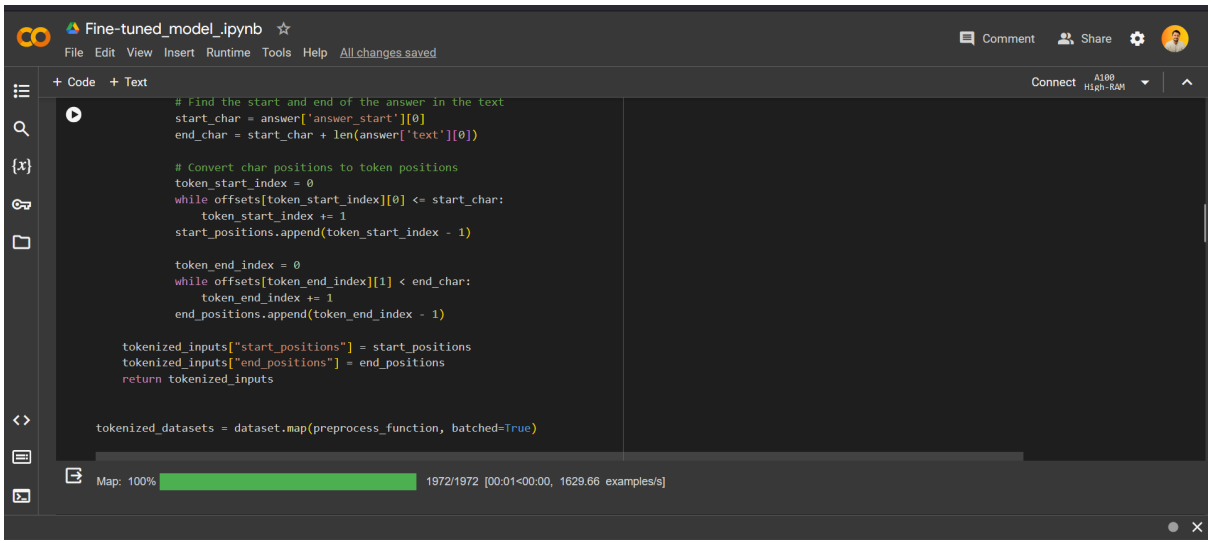

Figure 4.5: Tokenization Complete

After our data was ready, we needed to decide how to train our model. We looked at the model's instructions to figure out the best settings. Matching our training to these recommendations was important to make sure everything worked smoothly.

With our model and dataset ready, and the training settings in place, we began the actual fine-tuning. This process taught our model to get better at its tasks, aiming for very little error or loss by the end.

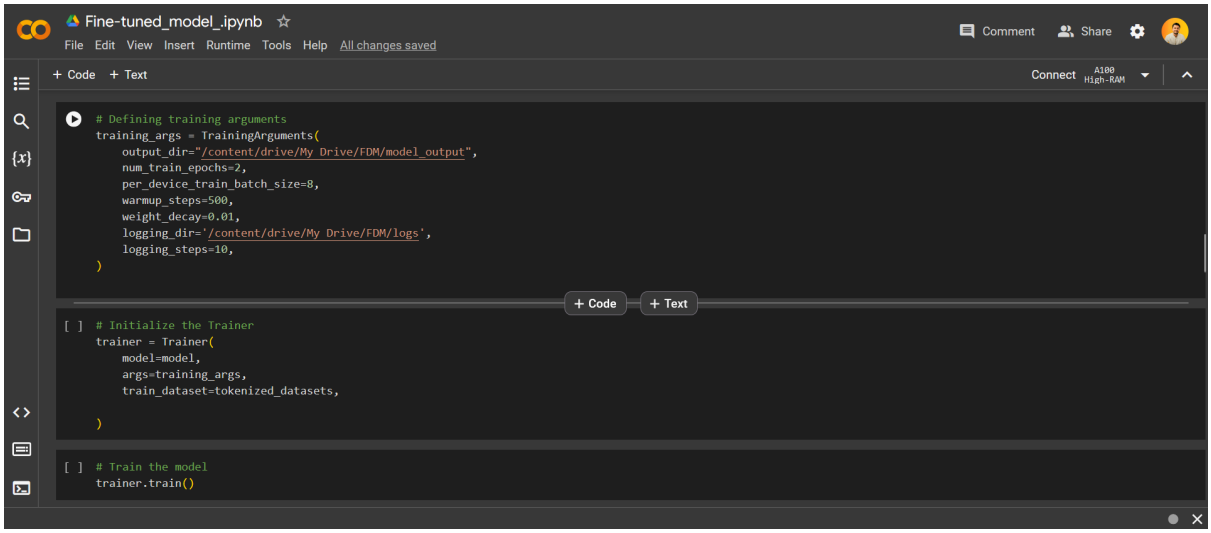

Figure 4.6: Setting Training Arguments and starting the Trainer

Once the training was done and we were happy with how little error there was, we saved this updated model. Now, it's ready to be used for further tasks.

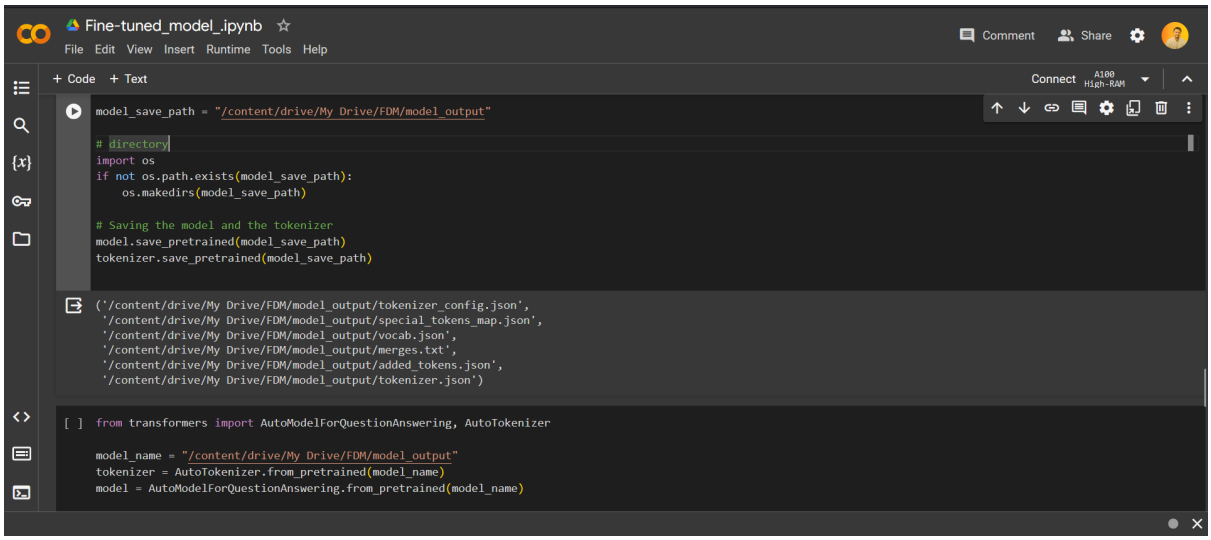

Figure 4.7: Saving the Improved Model

4.5 User Interface

In this section, we will delve into the development of a web application, a pivotal component designed to bridge our Large Language Model (LLM) with its intended users. Utilizing Django, a choice informed by its robustness and seamless integration with the Python framework, we embarked on creating an interactive platform. This platform is not merely the interface of our application but the conduit through which users engage with our LLM. From the moment users navigate to the login page to their dynamic interactions within the user-agent space, our platform ensures a fluid and intuitive experience, facilitating

meaningful exchanges and fostering an environment where our LLM can truly come to life.

Upon first visit, users are greeted by the home screen of Edu Score, which presents a clean, professional look with a welcoming message overlaying a serene background of a modern educational setup. The transparency of the overlay ensures the interface feels light and unobtrusive, inviting users to explore the capabilities of the platform. Here, Django's templating language weaves dynamic content into HTML, setting the stage for a tailored user journey.

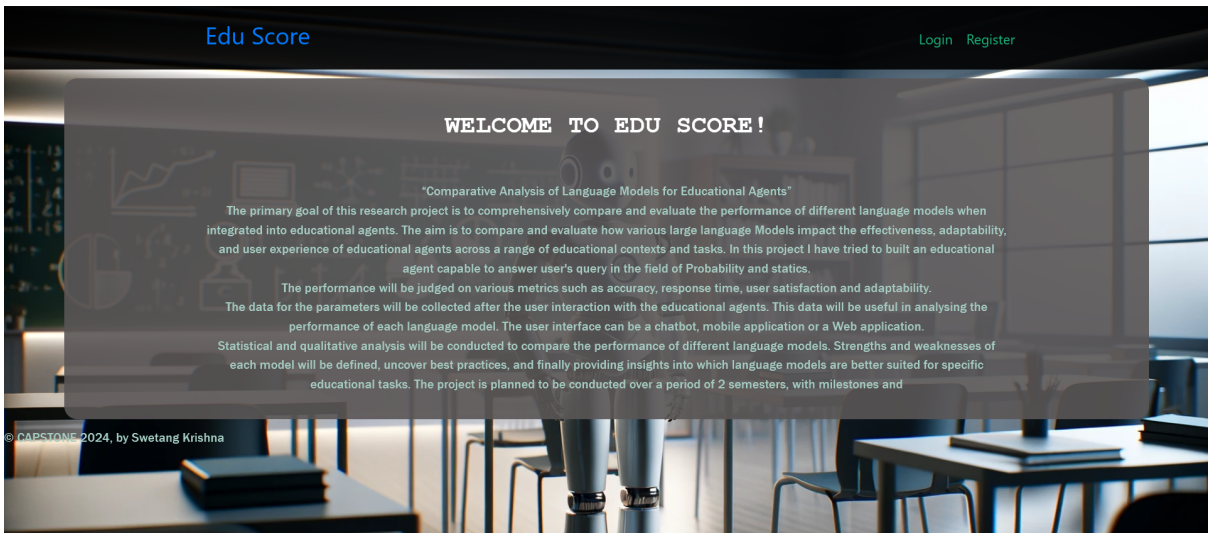

Figure 4.8: Home Page (Firefox)

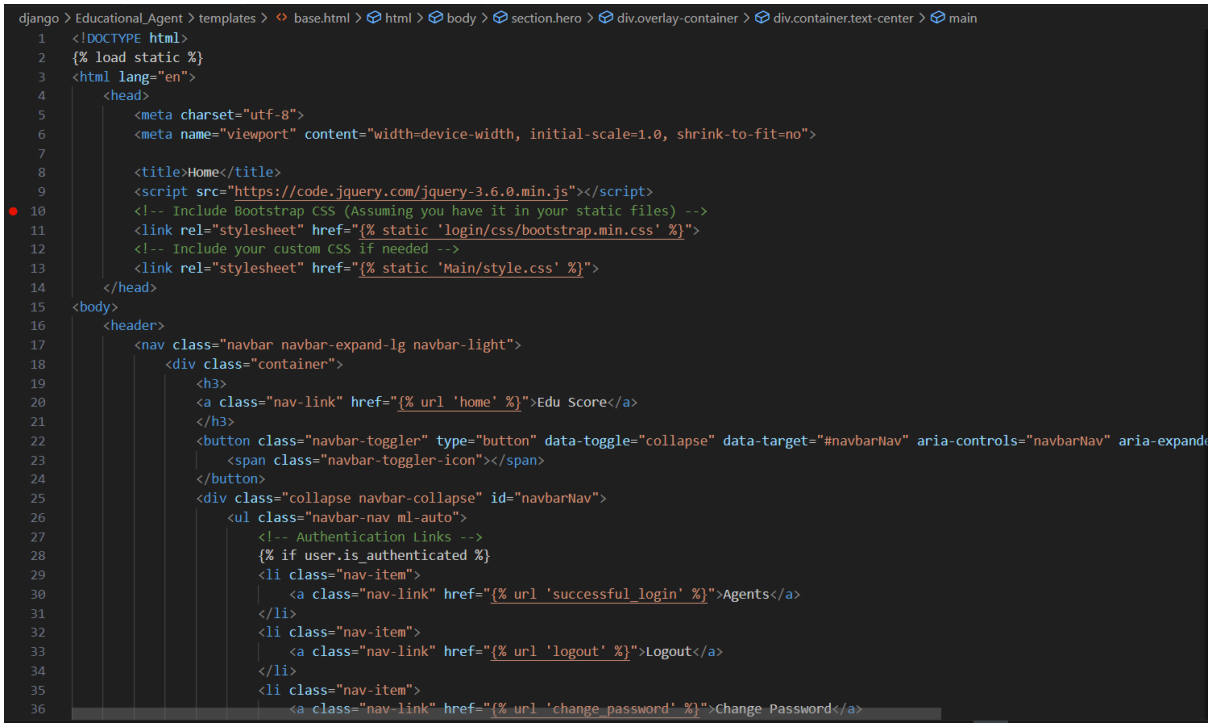

Figure 4.9: Base HTML file

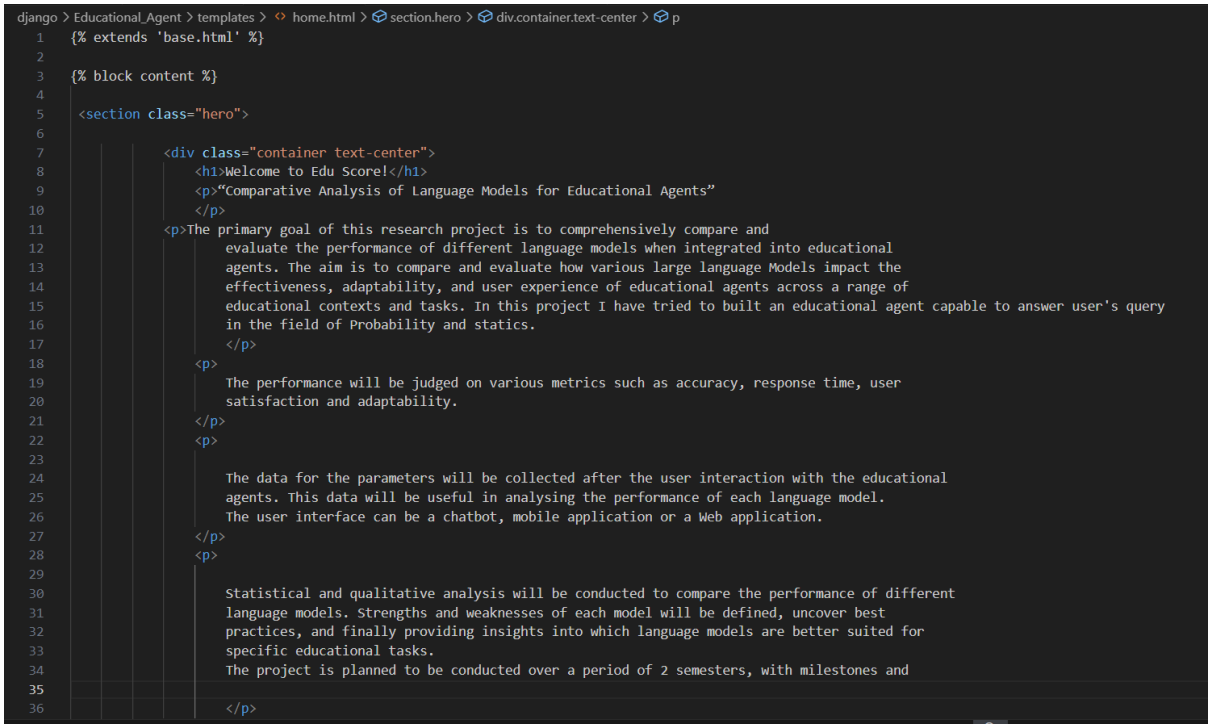

Figure 4.10: Home HTML file

Figure 4.11: Views python file

The registration page serves as the initiation point for user engagement, presenting a welcoming and secure entryway into the Edu Score environment. Users are met with a minimalist and straightforward form that requests essential details such as username, email, and password. The design choices here are deliberate—transparent fields overlay the serene

backdrop of a scholarly setting, reinforcing the academic focus of the platform while maintaining an inviting ambiance.

Django's authentication system, highly regarded for its security and ease of use, manages the registration process. It ensures that all user credentials are handled with the utmost care, encrypting passwords and safeguarding user data. The simplicity of the form masks the complexity of Django's validation mechanisms, which operate behind the scenes to ensure the integrity of user data and the overall system.

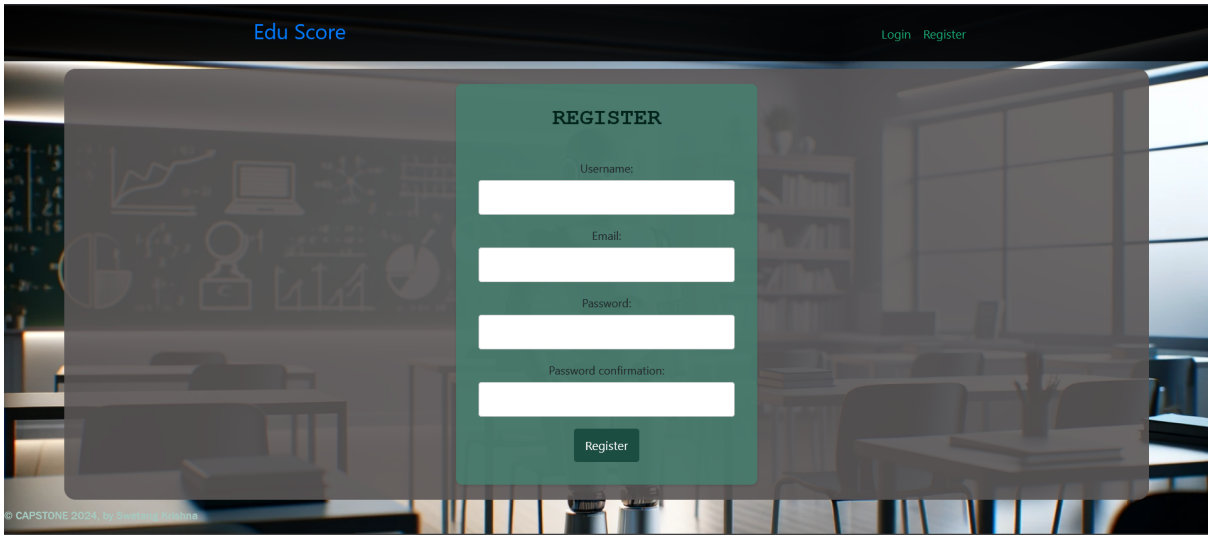

Figure 4.12: Register Page (Firefox)

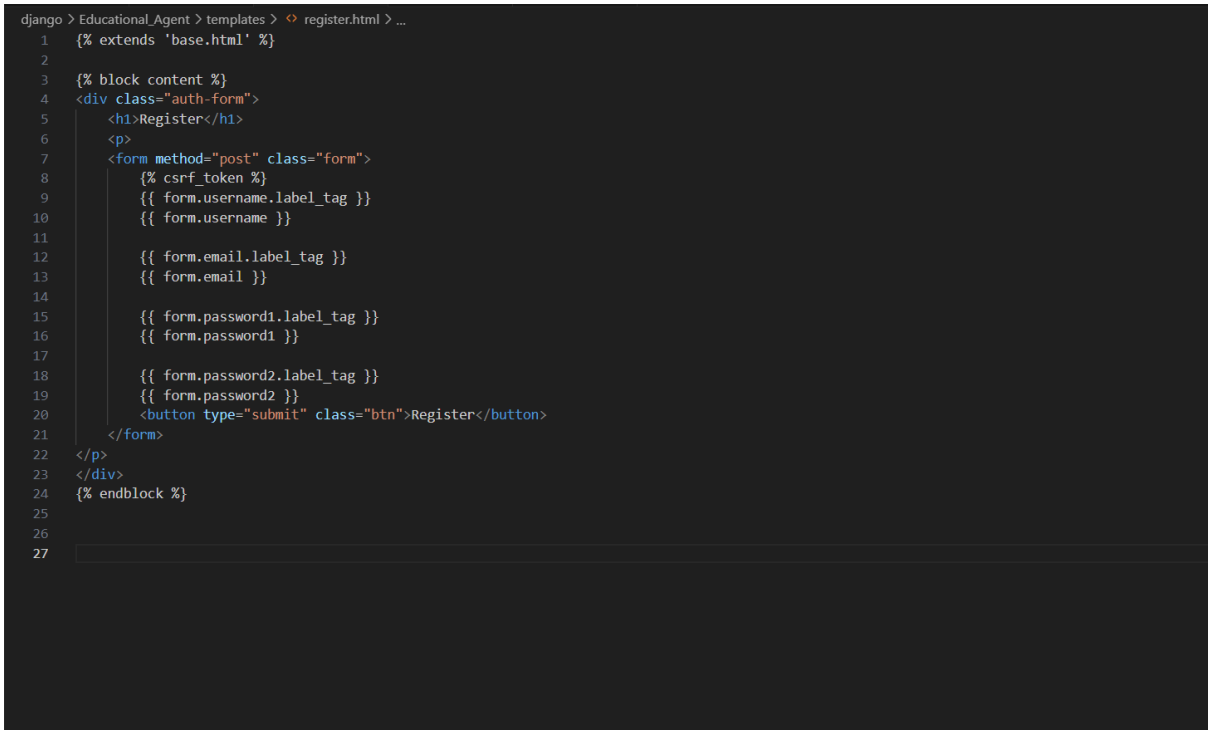

Figure 4.13: Register HTML file

```
django > Educational_Agent > Main > views.py > ...
49
50  def register(request):
51      if request.method == 'POST':
52          form = CustomUserCreationForm(request.POST)
53          if form.is_valid():
54              user = form.save()
55              auth_login(request, user)
56              return redirect('login')
57      else:
58          form = CustomUserCreationForm()
59      return render(request, 'register.html', {'form': form})
60
61
62
63
64  def login(request):
65      if request.method == 'POST':
66          form = AuthenticationForm(request, data=request.POST)
67          if form.is_valid():
68              username = form.cleaned_data.get('username')
69              password = form.cleaned_data.get('password')
70              user = authenticate(username=username, password=password)
71              if user is not None:
72                  auth_login(request, user)
73                  return redirect('successful_login.html')
74              else:
75                  messages.error(request, 'Invalid username or password.')
76          else:
77              messages.error(request, 'Invalid username or password.')
78      else:
79          form = AuthenticationForm()
80      return render(request, 'login.html', {'form': form})
81
82
83
```

Figure 4.14: Views python file

Transitioning to the login page, users encounter a secure and straightforward form, designed to facilitate quick access while maintaining user privacy. Django's authentication system comes into play here, managing user credentials with robust security measures.

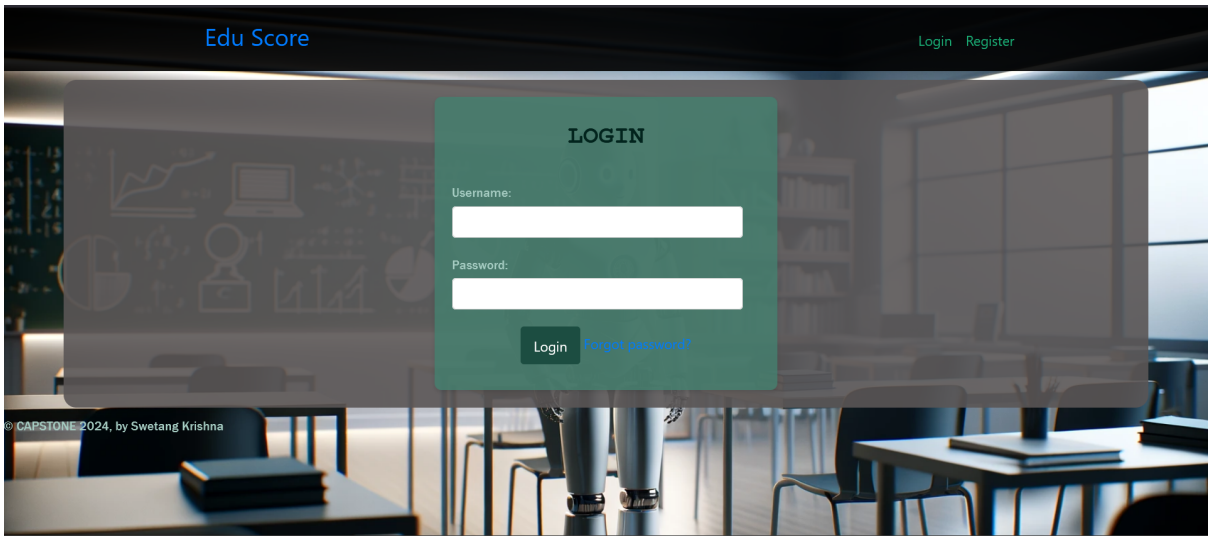

Figure 4.15: Login Page (Firefox)

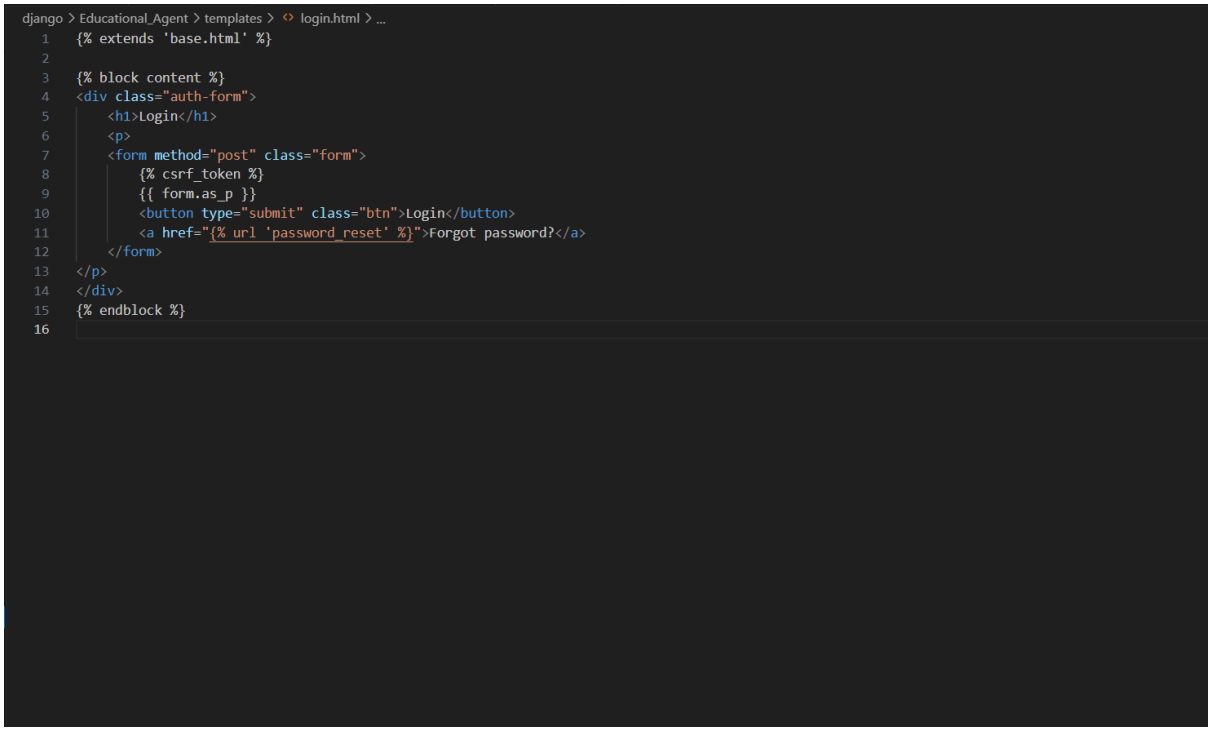

Figure 4.16: Login HTML file

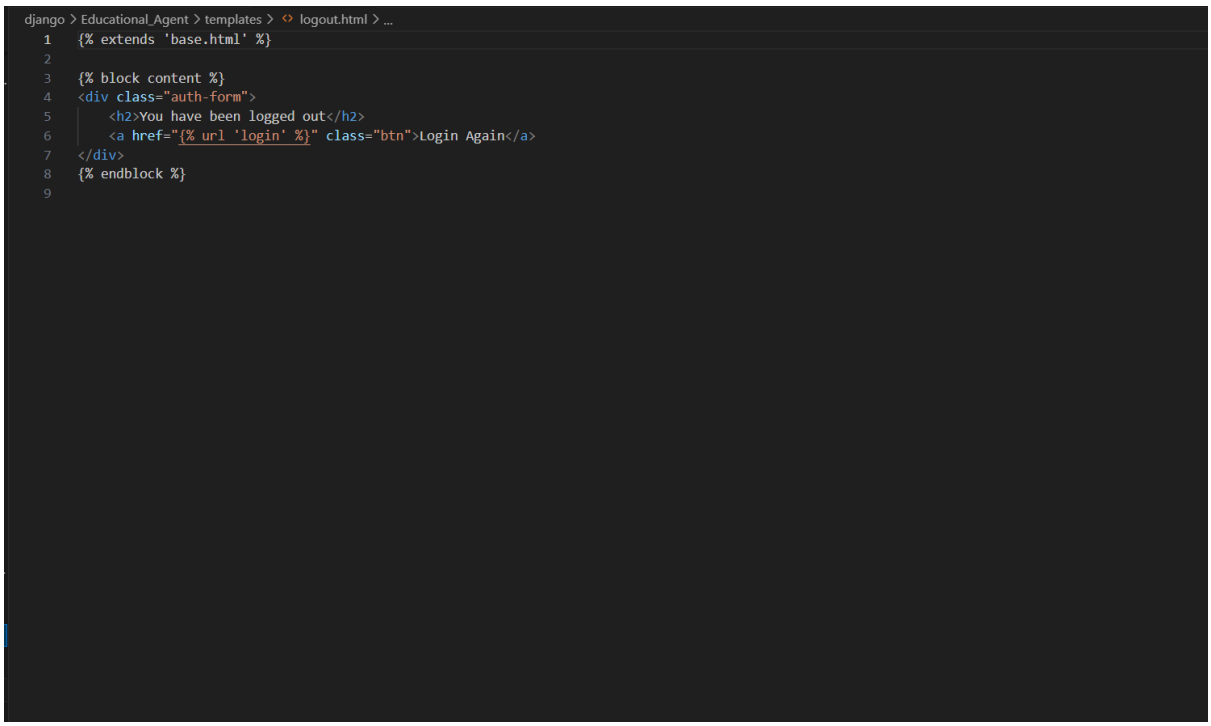

Figure 4.17: Logout HTML file

A dedicated change password page underscores our commitment to security and user control, allowing users to update their credentials through a simple, secure form. Django's built-in authentication system simplifies the process of password management, enforcing strong password policies to protect user accounts.

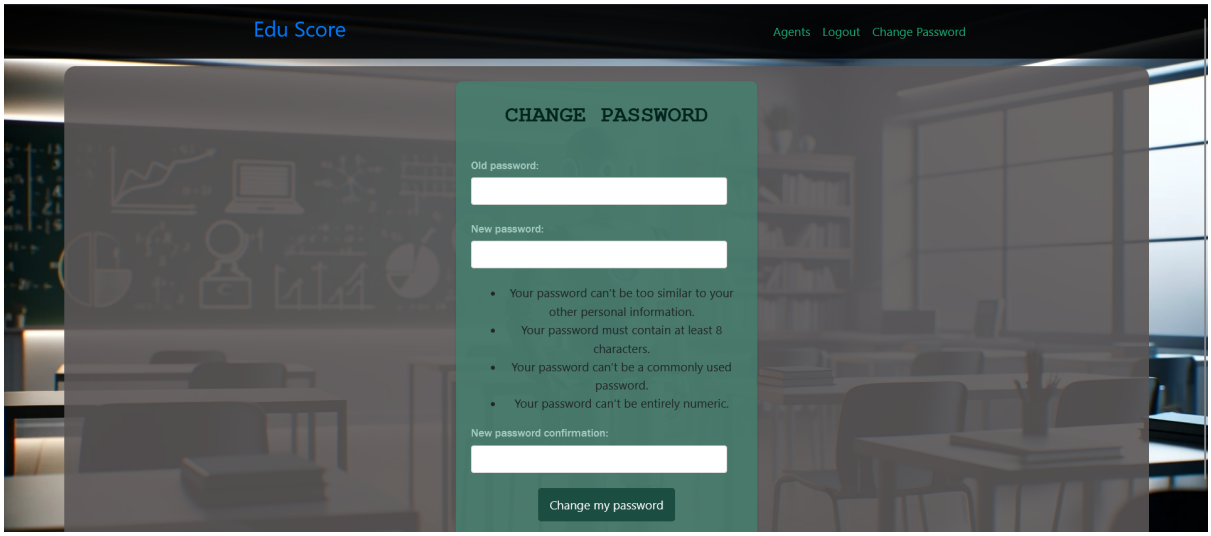

Figure 4.18: Change Password Page (Firefox)

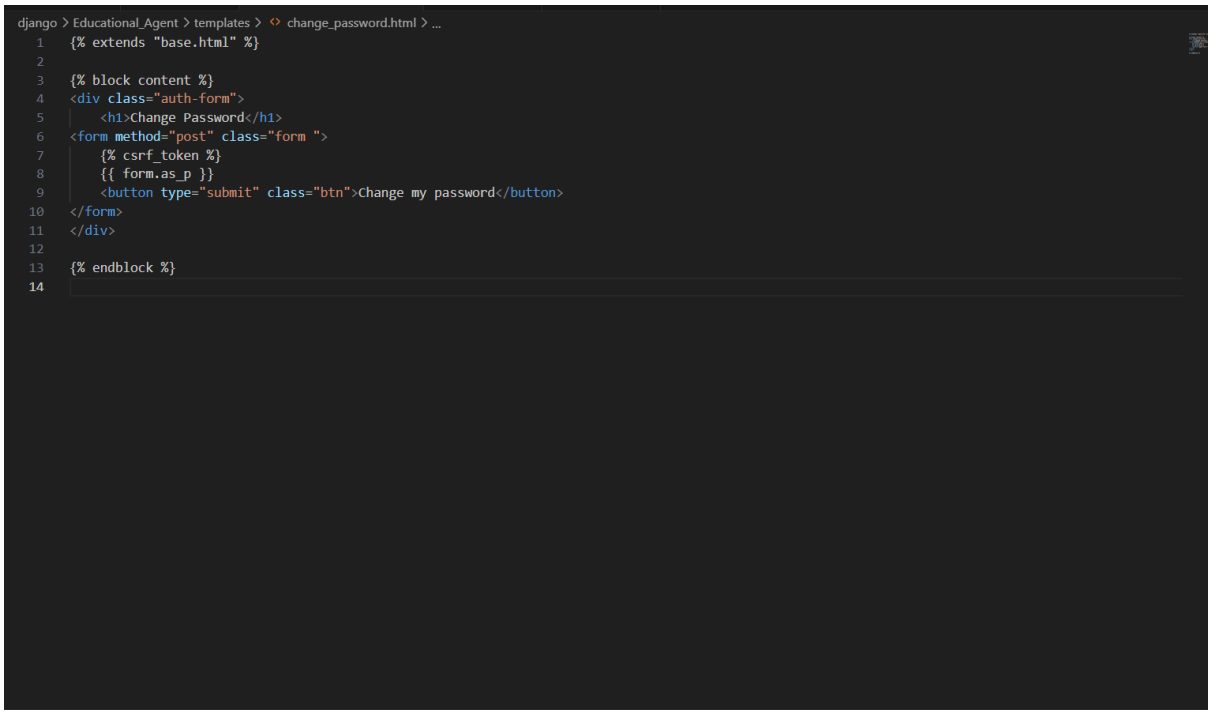

Figure 4.19: Change Password HTML file

The agents' page is where the core functionality lives. Here, users can interact with the educational agents—each a product of distinct LLMs, ready to field questions on Probability and Statistics. The page is designed with clarity in mind, hosting interactive elements that allow users to engage with either the base model or its fine-tuned counterpart. This functionality is supported by Django's capability to handle complex back-end logic and serve it up in a user-friendly front-end.

An innovative addition is the 'White Noise' button, which when activated, plays a soothing

background sound to enhance concentration while users interact with the application. This feature is a testament to Django's flexibility in integrating multimedia and improving user experience.

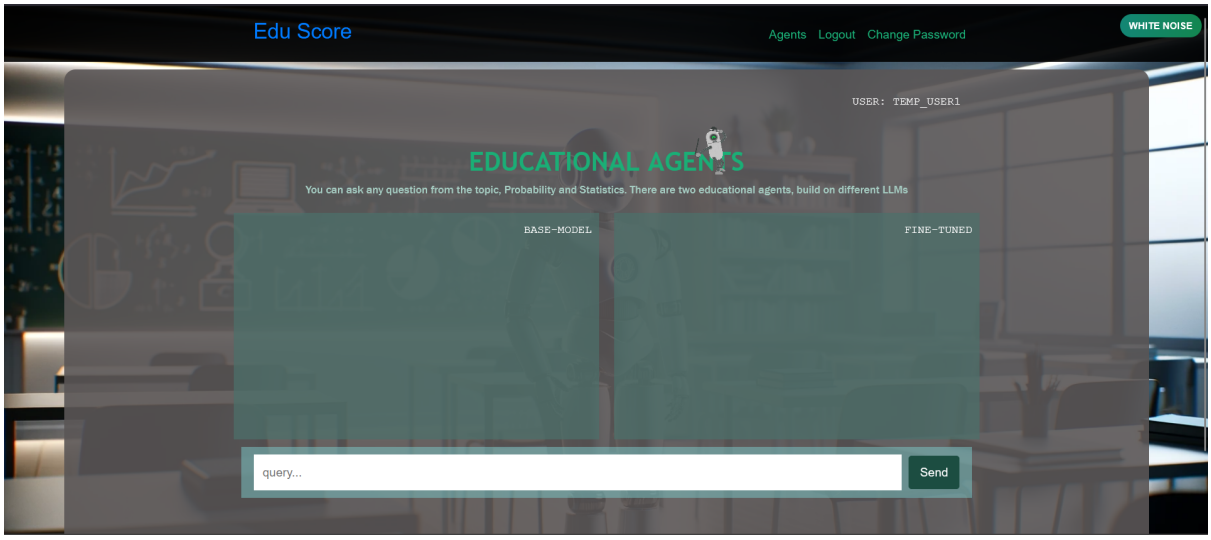

Figure 4.20: Agents' Page (Firefox)

```
{% extends 'base.html' %}

{% block content %}
{% load static %}

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Educational Agent</title>
    <style>
    </style>
</head>
<body>
    <section class="hero">

        <div class="container text-center">

            {% if user.is_authenticated %}
                <h4>USER: {{ user.username }}</h4>
                <button id="music-btn" onclick="toggleMusic()">White Noise</button>
                <div class='p1'> Educational Agents</div>
                <p> You can ask any question from the topic, Probability and Statistics. There are two educational agents, build on different LLMs
                <img src="{% static 'Main/images/walking-robot.gif' %}" alt="Moving Image" id="moving-gif">
                <audio id="background-music" loop>
                    <source src="{% static 'Main/audio/videoplayback.webm' %}" type="audio/mpeg">
                    Your browser does not support the audio element.
                </audio>
            {% endif %}

    </div>
</section>
<div class="chat-container">
    <div class="chat-panels-container">
    <div class="chat-panel" id="chatGPT-panel">
        <h4>Base-Model</h4>
    </div>
```

Figure 4.21: Agents' HTML file

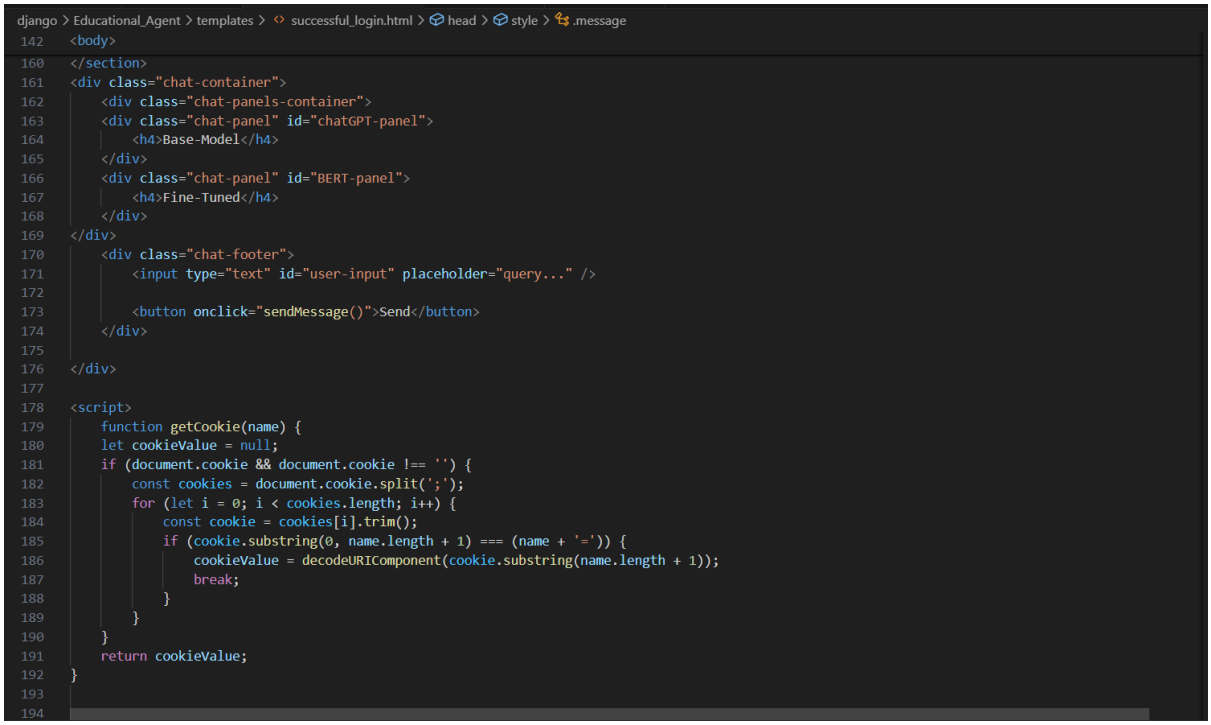

Figure 4.22: Agents' HTML file

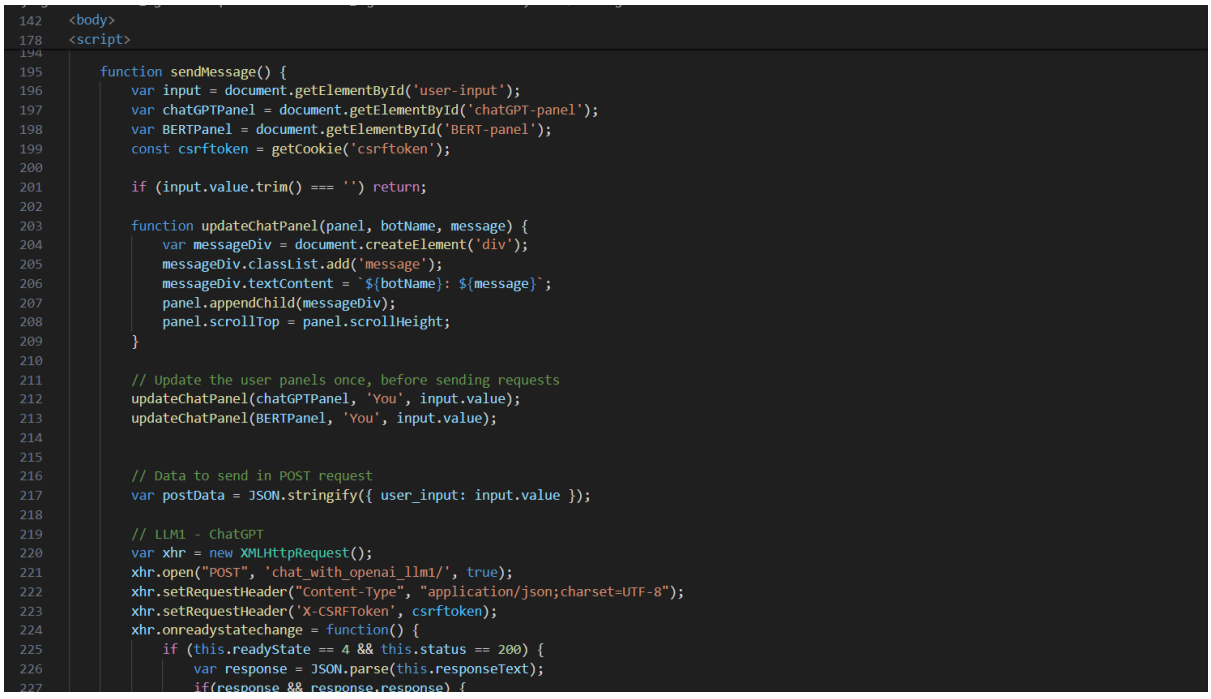

Figure 4.23: Agents' HTML file

4.6 Micro Program

In this chapter, we shine a spotlight on our micro software, a crucial yet often underappreciated component of our platform. Crafted with precision, this software adeptly

segments documents into digestible sections. These sections are subsequently analyzed by Whoosh, a tool we employed for its adeptness at identifying sections most relevant to user queries. Through this process, we are empowered to amalgamate the most pertinent documents into a unified, comprehensive context, thereby ensuring the delivery of highly relevant content to our users. This functionality not only enhances the efficiency of our platform but also significantly elevates the user experience by providing targeted, meaningful information.

This software is a quintessential element within our platform, meticulously designed to parse and dissect documents into manageable sections with surgical precision. Each document is skillfully split into chapters and further divided into smaller sections to ensure that the content is easily navigable and comprehensible.

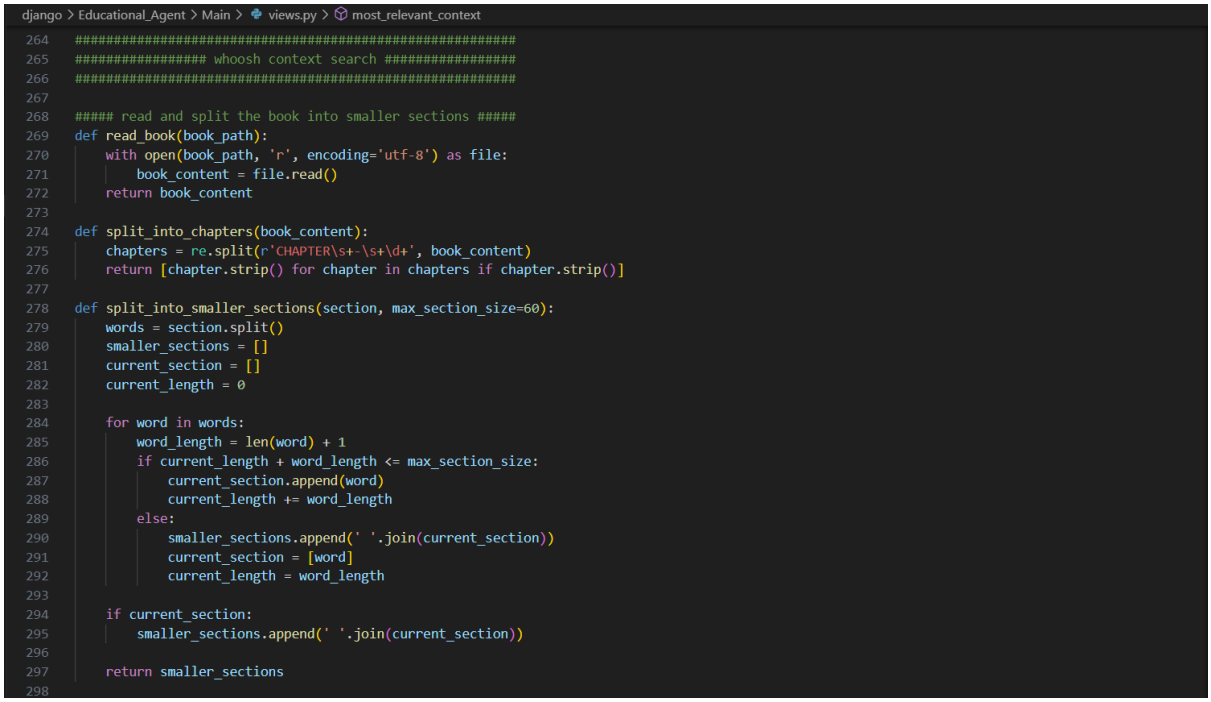

Figure 4.24: Context splitting

Our software includes a keyword search functionality, elegantly extracting the essence of user queries and using TF-IDF (Term Frequency-Inverse Document Frequency) to determine the weight and relevance of terms within the corpus. This ensures that the content presented to the user is not only relevant but also of the highest pertinence and quality.

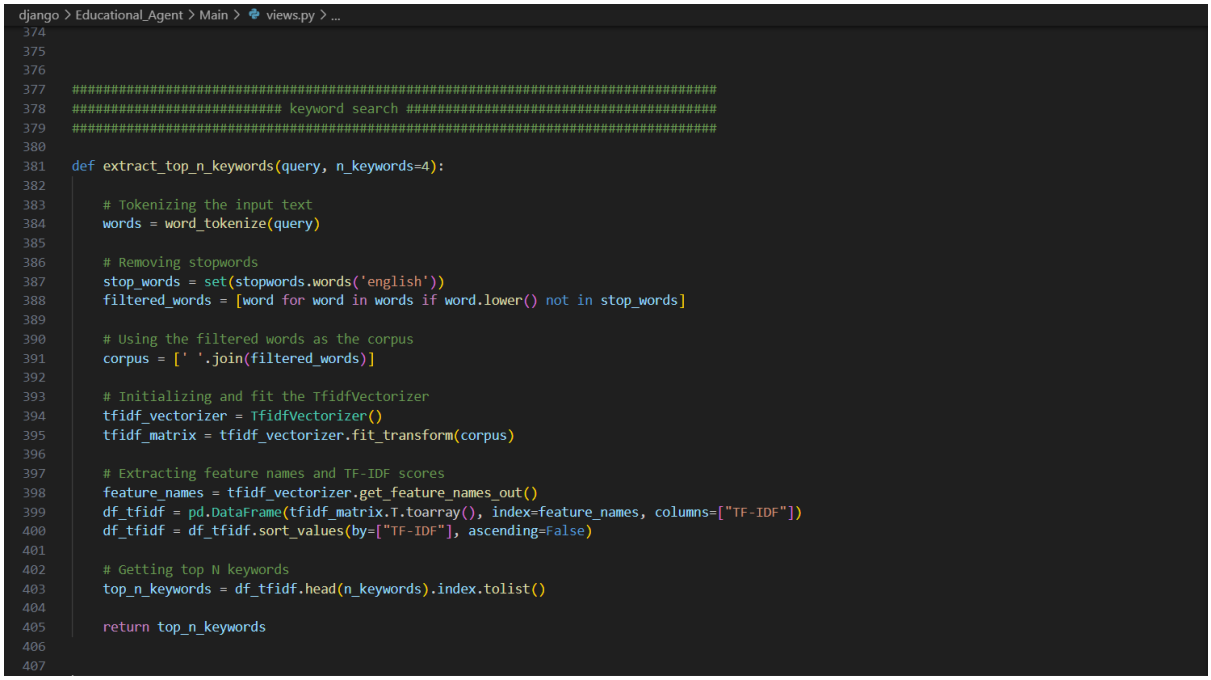

Figure 4.25: Keyword search

The role of cosine similarity calculations in this ecosystem cannot be overstated. By calculating the cosine similarity between the user's query and the array of document sections, we can rank the content in order of relevance, thereby providing a targeted and tailored response to the user's inquiry.

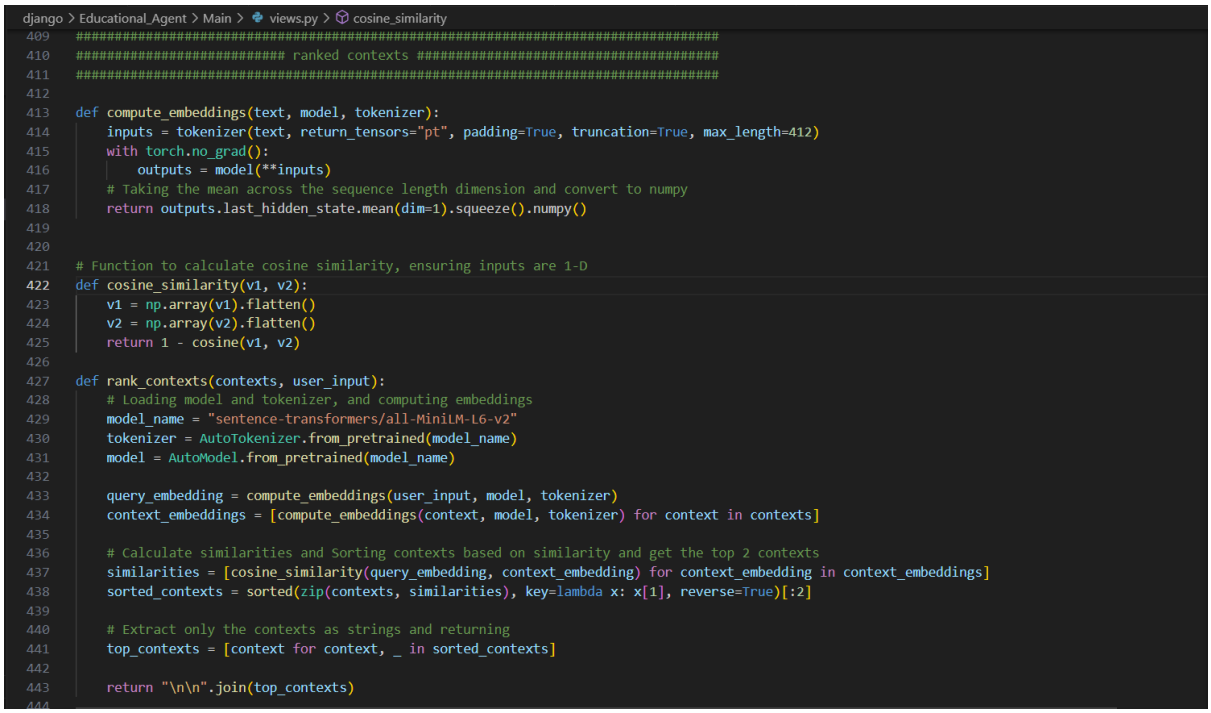

Figure 4.26: Context search

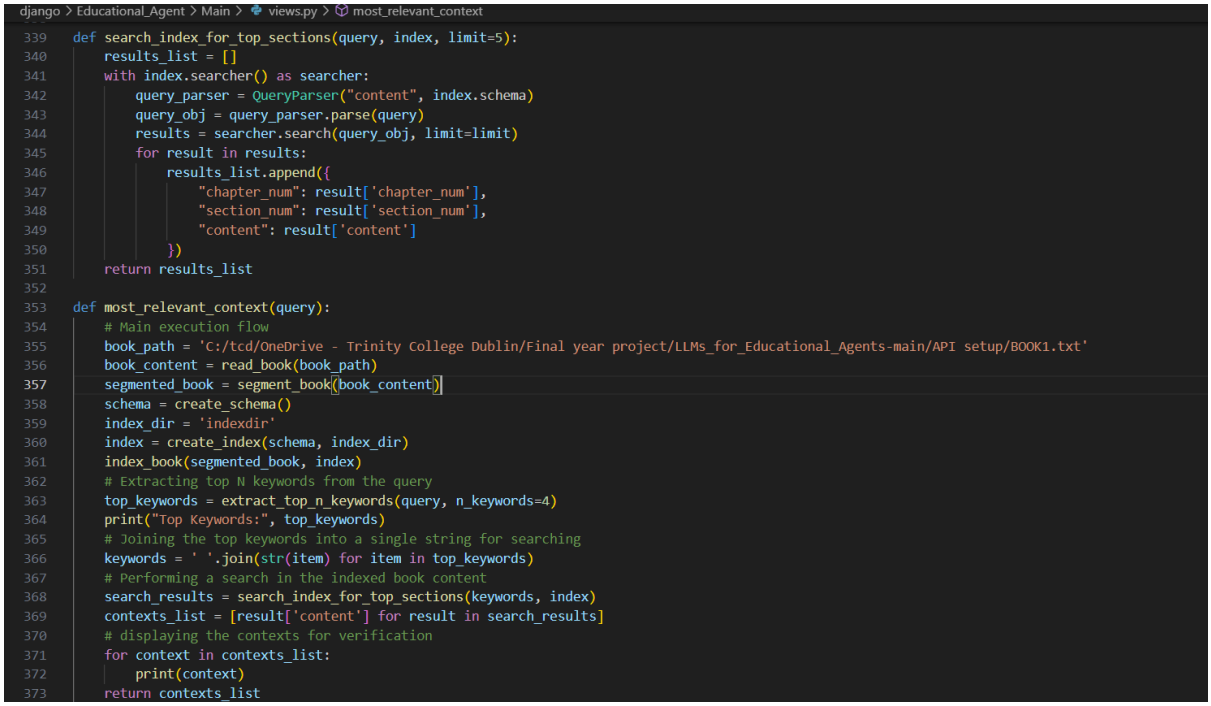

Figure 4.27: Ranking contexts

4.7 Database Management

In this section, we turn our focus to the robust database that serves as the backbone of our educational platform. Utilizing the renowned phpMyAdmin interface to interact with MySQL, establishing a strong, structured foundation that effectively supports the complex functionalities of our system.

Within the database, there are tables designed that cater to various aspects of the platform, including user authentication, session management, and storage of conversational histories, all facilitated by Django's ORM (Object-Relational Mapping) capabilities. This system allows us to manage the relational database through Django models, which simplifies database operations and ensures that our data remains consistent and easily accessible.

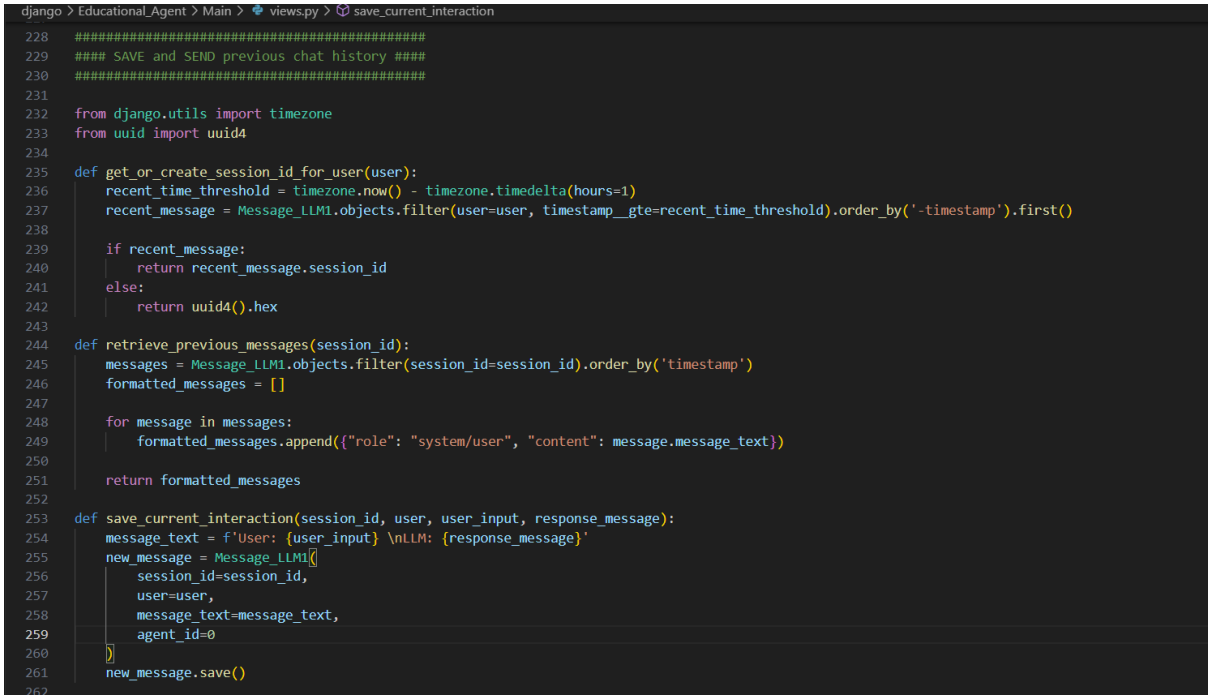

Figure 4.28: Views python code for saving chats to database

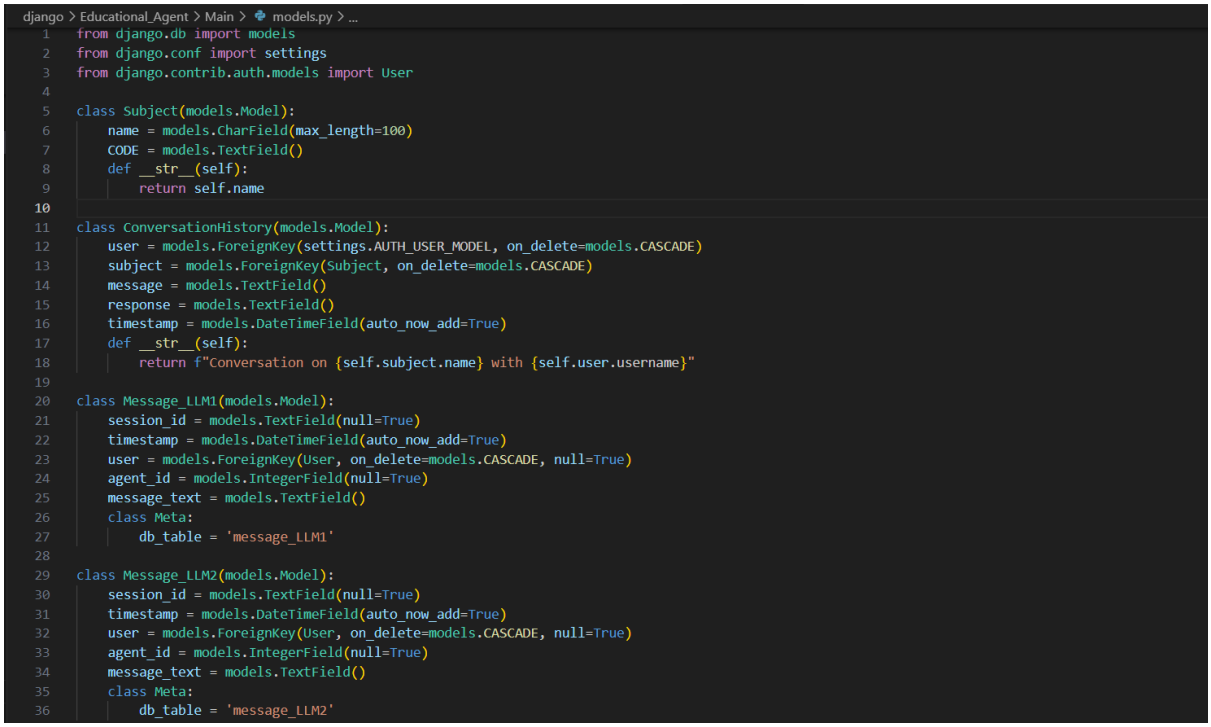

Figure 4.29: Object Relational Mapping

The tables within our database, such as agents, auth user, main conversation history, message llm1*, and message llm2* are meticulously crafted to store and organize the data necessary for our platform to function efficiently. For instance, the auth user table manages user information and credentials, while the message llm* tables are designed to store the

logs of interactions between users and the various language models employed by our educational agents.

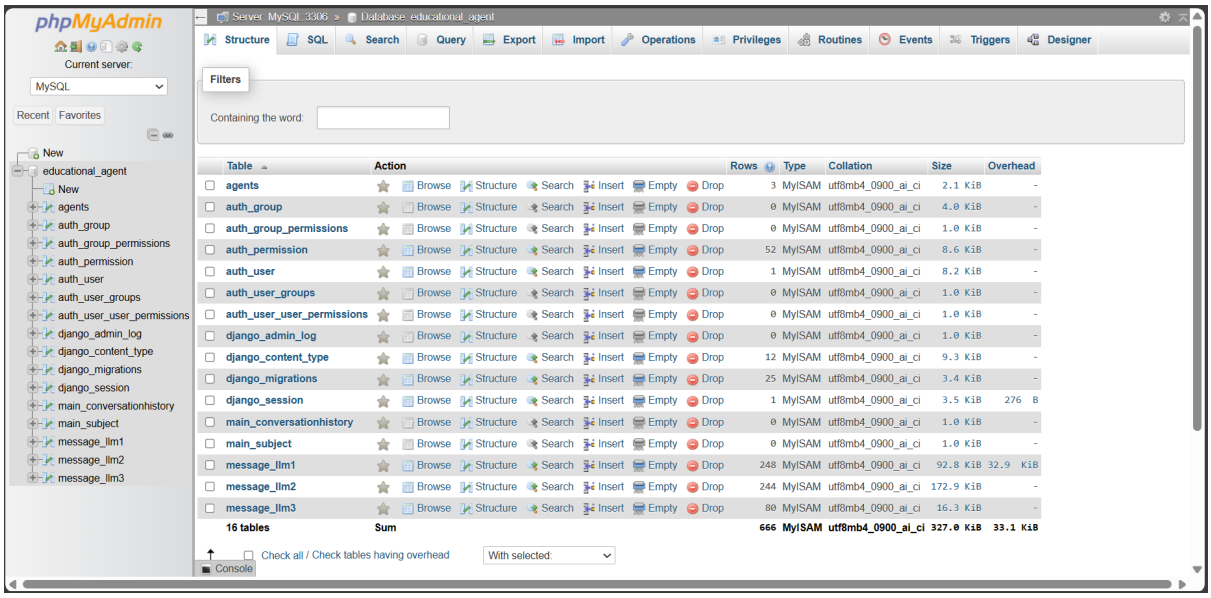

Figure 4.30: Object Relational Mapping

The message llm1* table within our database is a pivotal structure for recording and analyzing the interactions between users and the educational agents powered by our LLMs. A similar approach has been taken for message llm2*. Let's delve into the design and purpose of each column in this table:

id:

A unique identifier for each entry in the table, this field is typically set to auto-increment, ensuring that each new record receives a unique ID automatically.

session id:

This column records the ID of the user session, linking messages to the specific session in which they were created. It's essential for tracking interactions within a single session, enabling us to provide continuity and context in conversations.

timestamp:

It stores the exact date and time when the message was logged. This temporal data is critical for time-series analyses, such as understanding peak usage times or measuring response latencies.

agent id:

This field identifies which of the educational agents has processed the message or provided a response, allowing for granular performance analysis of individual agents.

user id:

By recording the ID of the user who sent the message, this column allows for personalized user tracking, analysis, and the ability to offer tailored educational experiences.

message text:

The core content of the user's message or the agent's response is stored here. This text data is crucial for NLP tasks, user interaction analysis, and further machine learning training.

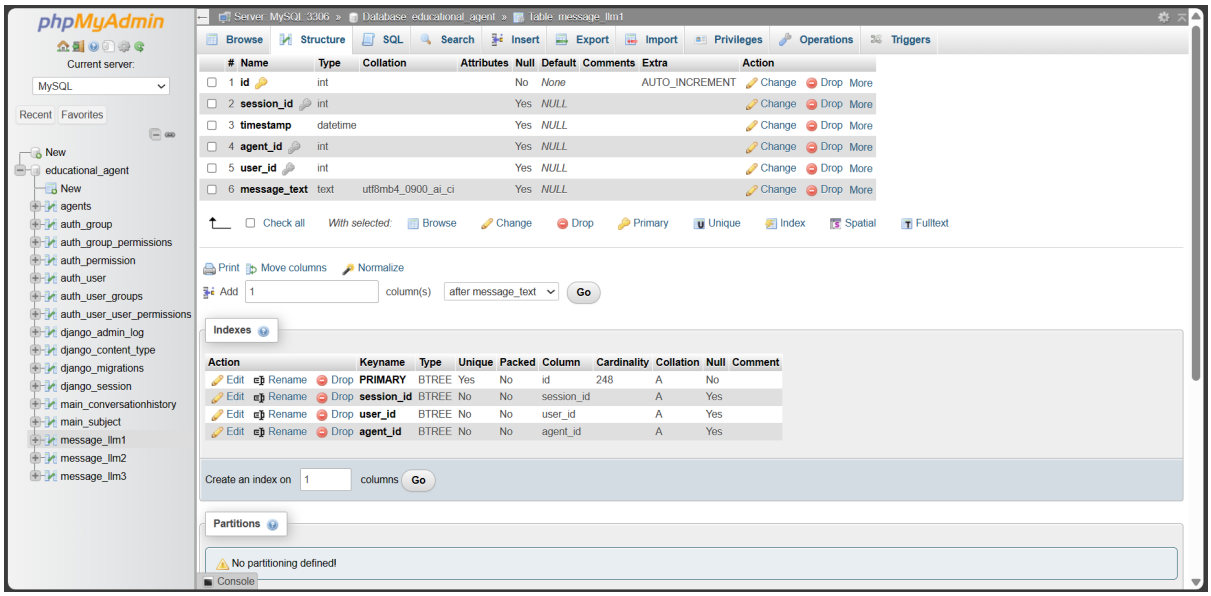

Figure 4.31: Columns in table "message llm1*"

To facilitate smooth communication between Django and MySQL, we've configured the database settings in Django's settings.py file, specifying the database engine, name, user, password, host, and port. This configuration not only ensures that Django can communicate seamlessly with the database but also provides the flexibility to adapt to different environments and requirements.

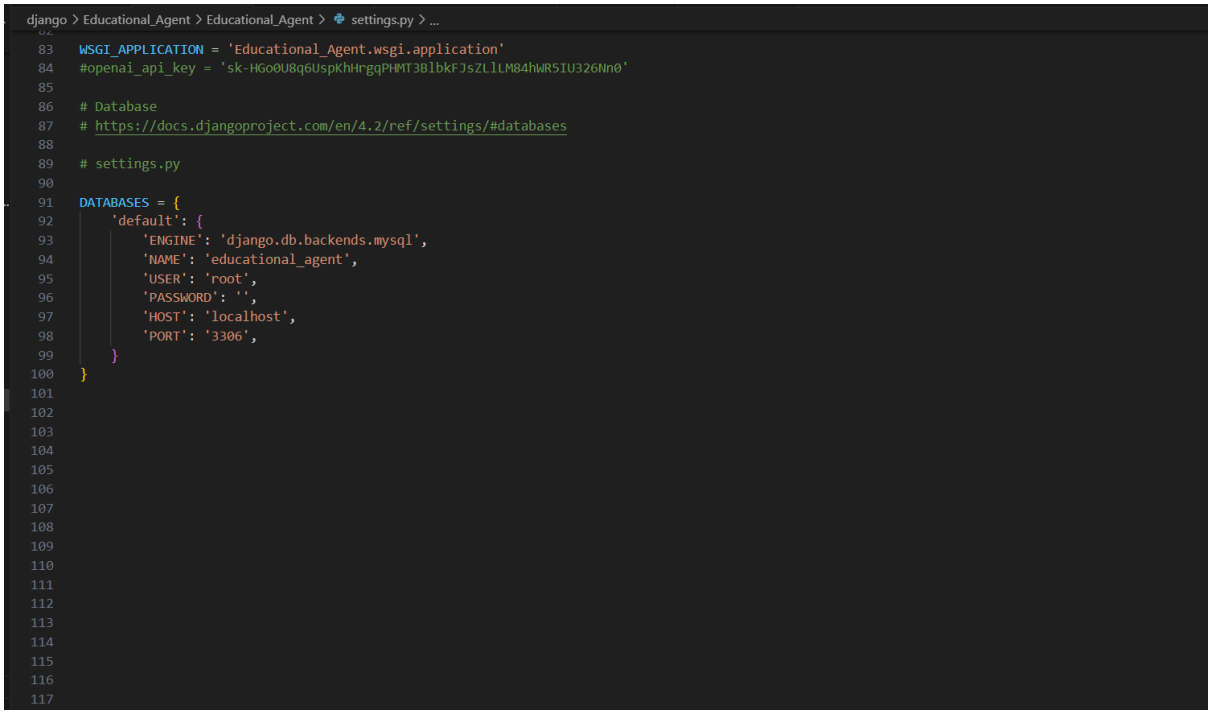

Figure 4.32: Connecting Database

5 Testing and Evaluation

This chapter details the evaluation methodology and testing processes used to assess the efficacy of the newly developed multi-modal system, which integrates a fine-tuned large language model (LLM) with keyword and context search functionalities. The goal is to compare the performance of the multi-modal fine-tuned LLM against the baseline LLM and ChatGPT-4 across various metrics to ascertain improvements in search relevance, user satisfaction, and system response times. Unit testing is also done to test each component individually.

5.1 LLM Evaluation:

The evaluation utilized a corpus of documents and real-world queries collected from the internet, categorized by relevance and from the subject "Probability and statistics". Data preprocessing involved normalization of text and removal of identifiable metadata to maintain privacy and consistency.

General queries were sent to the LLM with contexts, and the responses generated were descriptive, indicating a successful test result. Two models were evaluated, A base LLM model, presumably pre-trained on a diverse corpus of texts and a fine-tuned LLM, specifically optimized for improved performance on domain-specific queries.

5.1.1 Base Model

The base model provided a concise response: "mathematically" This answer, isn't what was expected, lacks depth fails to fully utilize the context provided and eventually fails the test.

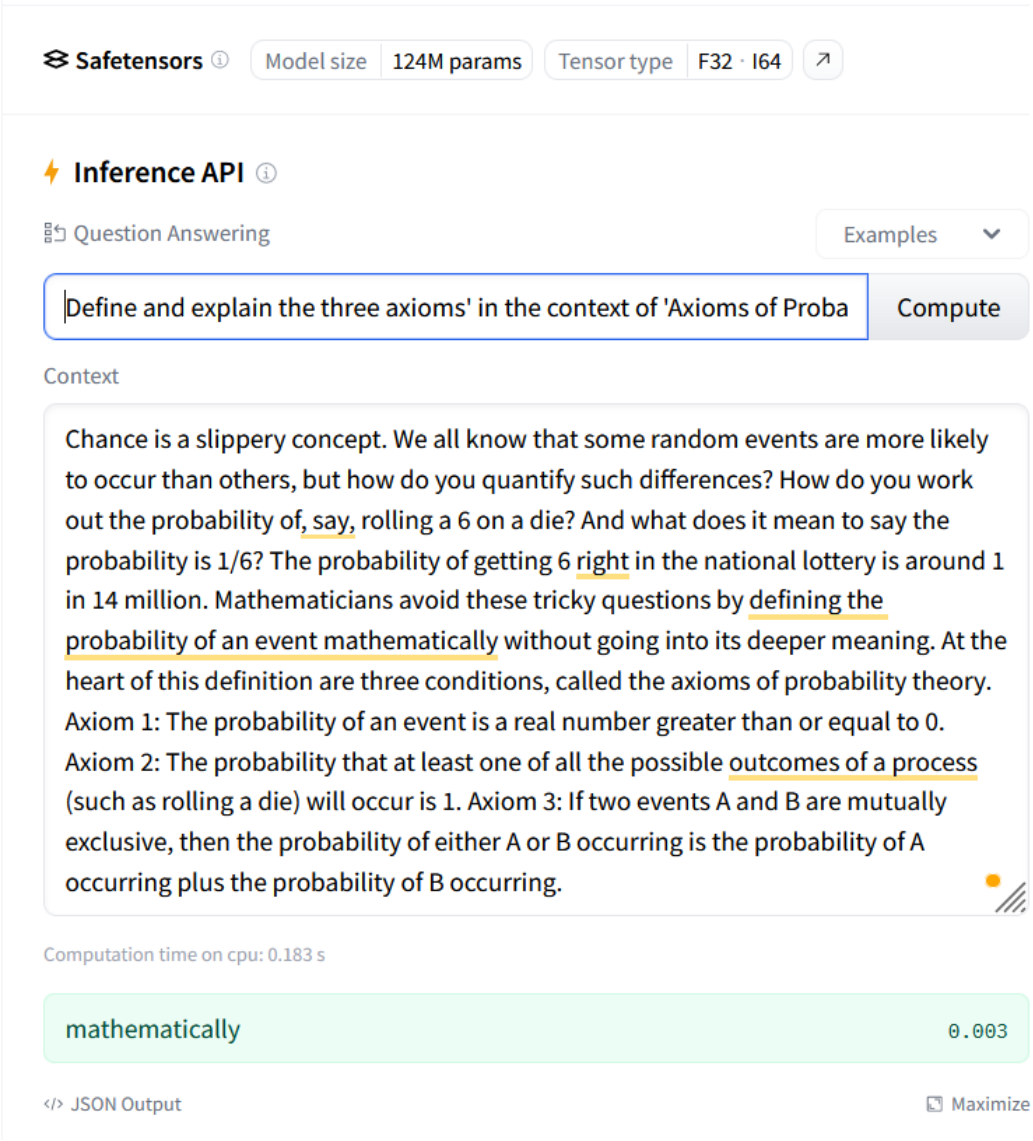

Figure 5.1: Responses generated Base model

5.1.2 Fine-Tuned Model

The fine-tuned model's response was correct and descriptive:

" Axiom 1: The probability of an event is a real number greater than or equal to 0. Axiom 2: The probability that at least one of all the possible outcomes of a process (such as rolling a die) will occur is 1. Axiom 3: If two events A and B are mutually exclusive, then the probability of either A or B occurring is the probability of A occurring plus the probability of B". This answer not only addressed the query directly but also enriched the response with relevant details from the context, demonstrating an improved understanding and application of the concept.

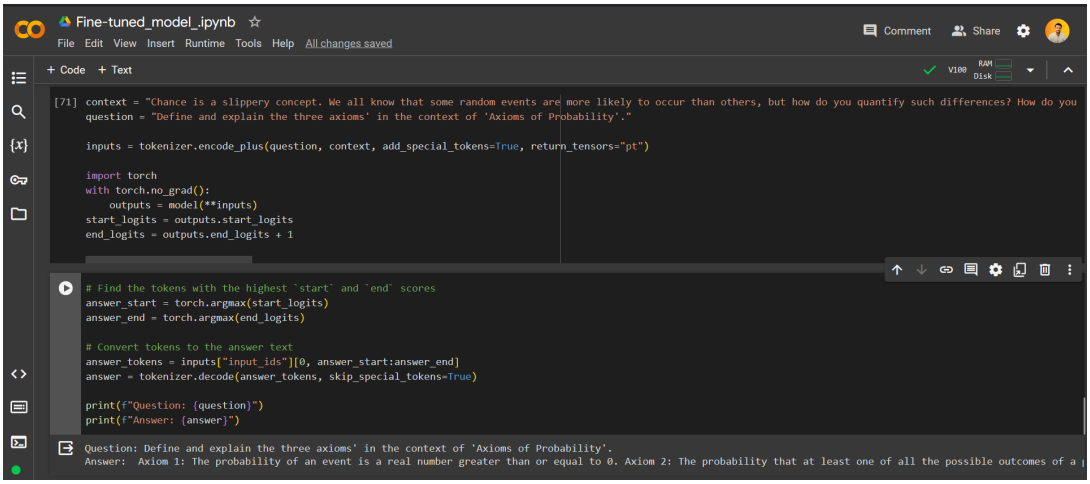

Figure 5.2: Descriptive Responses generated by our fine-tuned model

5.2 Keyword Search and Context Retrieval

This section of the report discusses the evaluation of the keyword search and Context Retrieval functionality within a large language model (LLM) designed for educational agents. The objective was to assess the model's ability to identify and extract key terms from natural language queries and return relevant contexts.

Keyword Search

A Python script was executed which utilized the Natural Language Toolkit (NLTK) for text processing and the Scikit-learn library for extracting keywords using the Term Frequency-Inverse Document Frequency (TF-IDF) method. The query given to the system was: "What is central tendency? What are its measures?"

The keyword search process successfully identified the following top four keywords from the input query: ['central', 'measures', 'tendency']

These keywords accurately reflect the essential components of the query related to the concept of central tendency in statistics.

```
(Capstone) C:\tcd\OneDrive - Trinity College Dublin\Final year project\LLMs_for_Educational_Agents-main\API setup>python keyword_sear
ch.py
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\sweta\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\sweta\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\sweta\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Top 4 Keywords: ['central', 'measures', 'tendency']
```

Figure 5.3: Keyword Search Results

5.2.1 Context Retrieval

In continuation of our keyword search evaluation, we progressed to examine the context retrieval capabilities of our Large Language Model (LLM). Using the keywords identified in the previous stage, we tested the model's ability to extract relevant sections from a text, aiming to provide informative content based on the keywords.

We applied a Whoosh-based indexing and search system to a segmented text corpus. The corpus was structured into chapters and sections for granular retrieval. The input for the context retrieval was the set of keywords previously extracted: "central" "measures" and "tendency".

The context retrieval system effectively located several relevant sections within the corpus:

Chapter 3, Section 5: Detailed measures of central tendency and their application in statistical analysis.

Chapter 3, Section 42: Discussed the normal distribution, its properties, and its relation to measures of central tendency.

Chapter 5, Section 1: Outlined the types of descriptive statistics, emphasizing central tendency and its measures.

```
(Capstone) C:\tcd\OneDrive - Trinity College Dublin\Final year project\LLMs_for_Educational_Agents-main\API setup>python context_sear
ch.py
Number of Chapters: 8
Chapter 1: 2 smaller sections
Chapter 2: 122 smaller sections
Chapter 3: 168 smaller sections
Chapter 4: 118 smaller sections
Chapter 5: 114 smaller sections
Chapter 6: 196 smaller sections
Chapter 7: 167 smaller sections
Chapter 8: 117 smaller sections
Chapter: 3, Section: 5
Content: ◦ Center of the Data uses measures of central tendency to locate the middle or center of a distribution where most of the da
ta tends to be concentrated. The three best known measures of central

Chapter: 3, Section: 42
Content: variable follows a normal distribution, the histogram is bell-shaped and symmetric, and the best measures of central tendenc
y and dispersion are the mean and the standard deviation.

Chapter: 5, Section: 1
Content: types of descriptive statistics: frequency distribution, central tendency, and variability/dispersion. With descriptive stat
istics there is no uncertainty as you are summarizing the characteristics

◦ Center of the Data uses measures of central tendency to locate the middle or center of a distribution where most of the data tends
to be concentrated. The three best known measures of central

variable follows a normal distribution, the histogram is bell-shaped and symmetric, and the best measures of central tendency and dis
persion are the mean and the standard deviation.

types of descriptive statistics: frequency distribution, central tendency, and variability/dispersion. With descriptive statistics th
ere is no uncertainty as you are summarizing the characteristics
```

Figure 5.4: Context Retrieval Results

5.3 Comparative Analysis of Responses from Base Model, Fine-Tuned Model, and ChatGPT-4

The models were provided with a document containing specific information related to statistics and public health. The evaluation involved querying each model with identical questions to assess their ability to extract and present information correctly and contextually.

Each model was queried with the same set of questions, and the responses were compared to evaluate their precision, adherence to the provided document, and the depth of information provided.

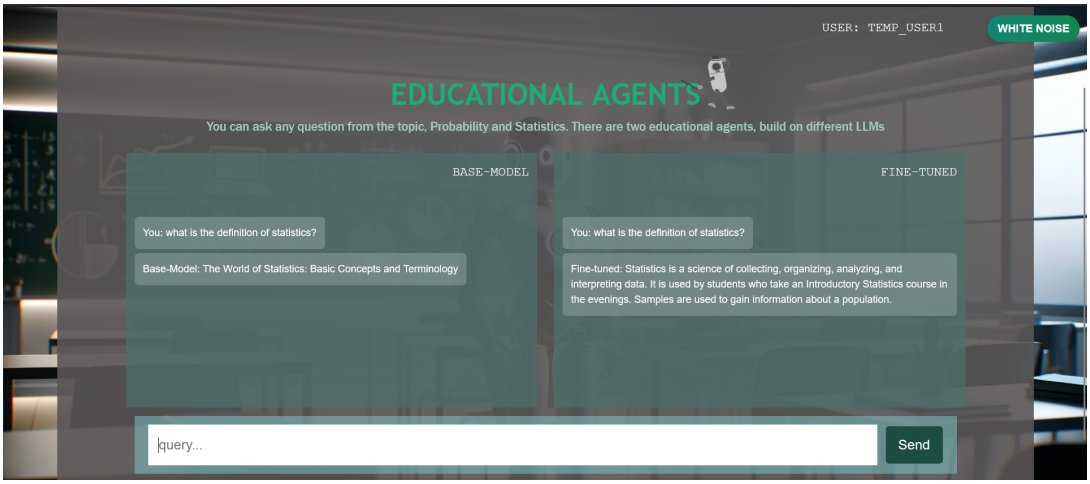

Figure 5.5: Simple Query Response (Base and Fine-tuned Model)

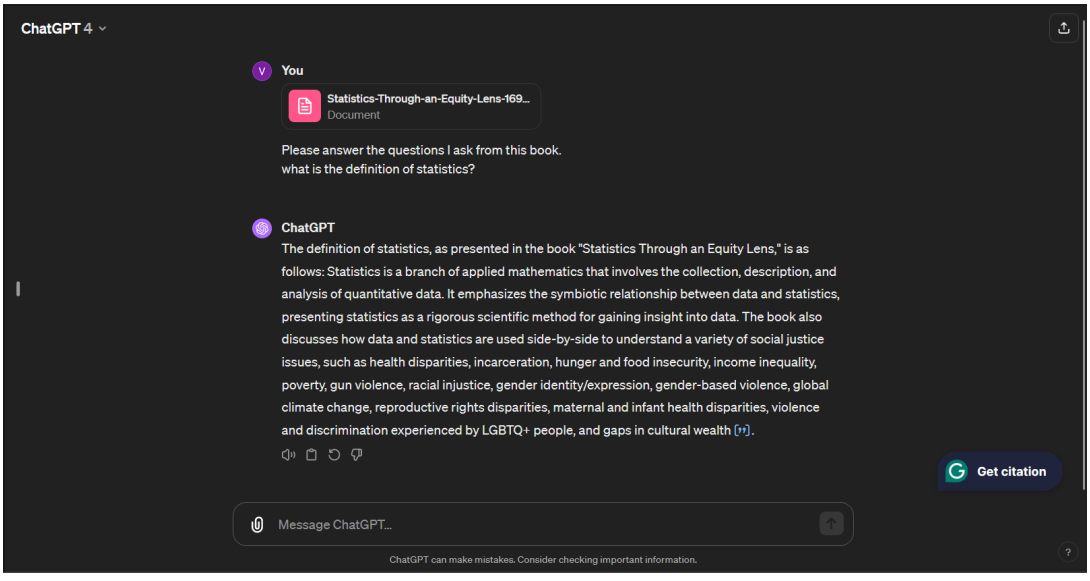

Figure 5.6: Simple Query Response (ChatGPT-4)

Model	QUERY	RESPONSE
ChatGPT	What is the definition of statistics?	Provided a general definition of statistics but lacked context-specific detail.
Fine-Tuned LLM	What is the definition of statistics?	Offered a more detailed explanation, incorporating the context of an Introductory Statistics course and referencing the use of samples to understand a population.
Base LLM	What is the definition of statistics?	Gave a comprehensive definition with a focus on the application of statistics through an equity lens, describing its role in addressing social justice issues.

Figure 5.7: Simple Query-Response Analysis

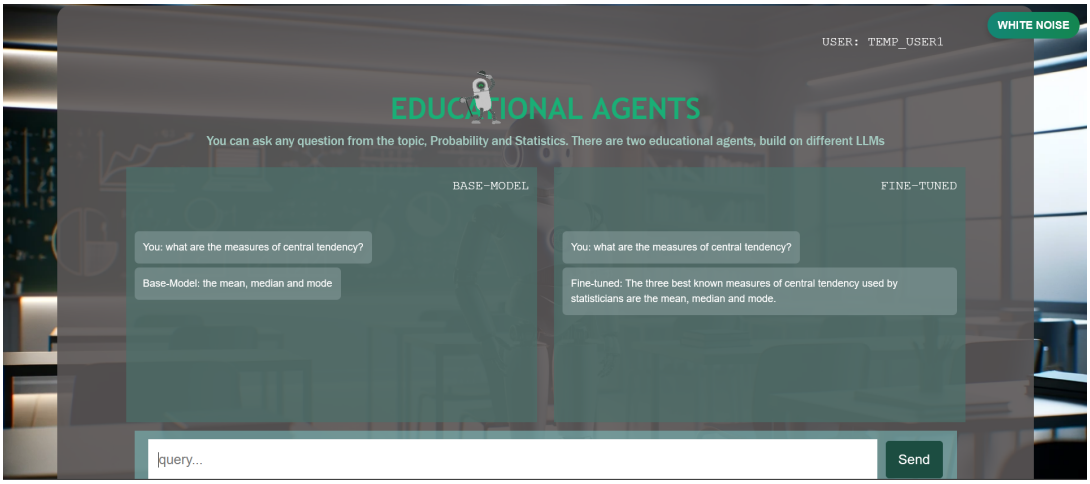

Figure 5.8: Multiple Keyword Query-Response (Base and Fine-tuned Model)

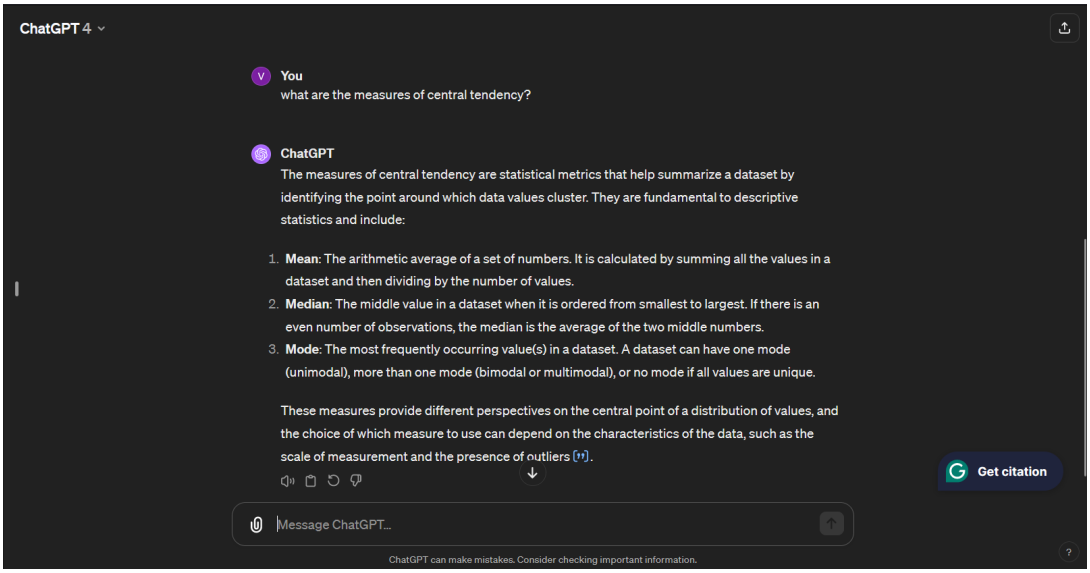

Figure 5.9: Multiple Keyword Query-Response (ChatGPT-4)

Model	QUERY	RESPONSE
ChatGPT	What are the measures of Central Tendency?	Correctly listed the mean, median, and mode.
Fine-Tuned LLM	What are the measures of Central Tendency?	Offered a more detailed explanation, incorporating the context of an Introductory Statistics course and referencing the use of samples to understand a population.
Base LLM	What are the measures of Central Tendency?	Gave a comprehensive definition with a focus on the application of statistics through an equity lens, describing its role in addressing social justice issues.

Figure 5.10: Multiple Keyword Query-Response Analysis

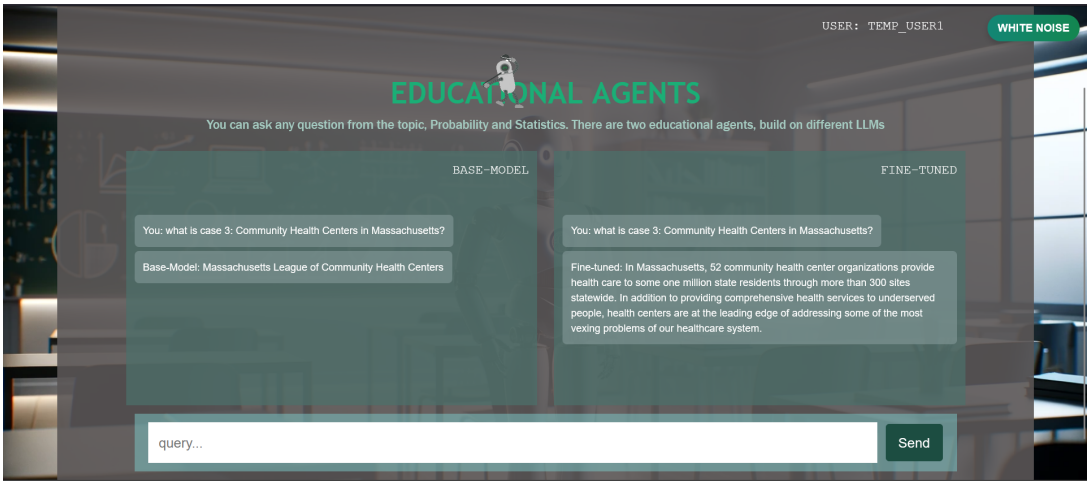

Figure 5.11: Document Based Query-Response (Base and Fine-tuned Model)

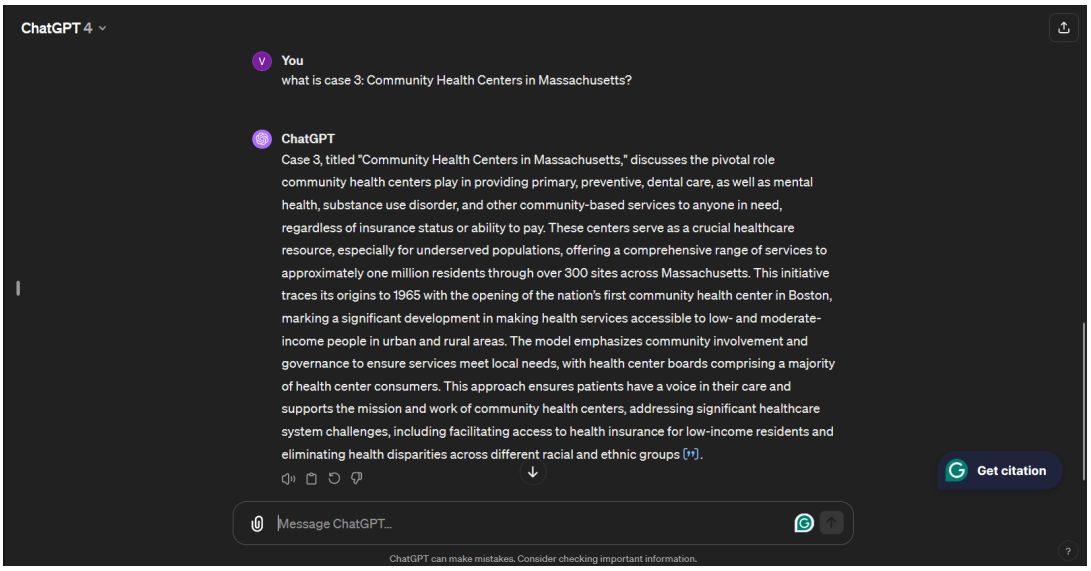

Figure 5.12: Document Based Query-Response (ChatGPT-4)

Model	QUERY	RESPONSE
ChatGPT	Community Health Centers in Massachusetts	Correctly listed the mean, median, and mode.
Fine-Tuned LLM	Community Health Centers in Massachusetts	Also correctly identified the measures and provided contextually relevant detail.
Base LLM	Community Health Centers in Massachusetts	Provided a detailed explanation of each measure and discussed how the choice of measure can depend on data characteristics.

Figure 5.13: Document Based Query-Response Analysis

The fine-tuned model and ChatGPT-4 generally provided more detailed and contextually relevant responses compared to the base model. This suggests that the fine-tuning process and advanced capabilities of ChatGPT-4 allow for better utilization of provided documents, resulting in more informative and precise answers.

6 Conclusion

6.1 Summary of Findings

This dissertation has thoroughly explored the successfulness of advanced natural language processing (NLP) techniques within educational technology, focusing specifically on the development and application of an AI-enhanced educational agent. The project's core objective was to harness AI and NLP to create a highly personalized, interactive, and efficient Educational agent. Through innovation, rigorous testing and iterative development, the educational agent demonstrated its ability to significantly enhance the learning experience by offering engaging content delivery, and adaptive learning support based on a list of documents provided.

Throughout the project, the educational agent was evaluated across multiple dimensions. It was found that by integrating sophisticated machine learning algorithms and NLP techniques, the agent could effectively interpret and respond to student inquiries with high relevance and personalization. The agent's performance in real-world educational settings highlighted its potential to not only support but also transform traditional learning environments by making them more interactive and responsive to individual learner needs.

6.2 Significance of the Research

The research conducted provides substantial contributions to the field of educational technology. It illustrates the transformative potential of AI and NLP in enhancing educational practices through the development of digital tools that support both teachers and students. Some tools were based on customization and personalisation of Educational content. The educational agent developed in this project exemplifies how AI can be tailored to facilitate personalized education, reflecting significant advancements in the application of intelligent technologies in learning environments with a multimodal system.

6.3 Implications for Practice

The practical applications of this research are manifold. The educational agent can serve as a foundation for further development of intelligent tutoring systems and educational platforms that leverage AI to provide dynamic learning experiences.

Schools, universities, and other educational institutions could adopt similar technologies to support personalized learning paths, thereby enhancing student engagement and improving educational outcomes. LLMs can tailor educational content to meet individual student needs, adapting to their learning pace, style, and preferences. This personalized approach can improve engagement and outcomes. LLMs can assist both students and educators in navigating vast amounts of academic literature, summarizing research papers, and finding relevant information quickly.

6.4 Limitations and Challenges

While the results of this project are promising, there are limitations and challenges to consider. The effectiveness of the educational agent is largely dependent on the quality and extent of the training data, which may limit its ability to accurately respond to less common or out-of-scope queries.

There were a few challenges faced during the testing process which can be overcome by adding a few more components like text classification, prompt generator, and sentence similarity models to enhance its performance.

Furthermore, the integration of such AI systems into existing educational infrastructures requires careful consideration of ethical and privacy concerns, as well as substantial customization to meet diverse educational needs.

6.5 Directions for Future Research

Future research should aim to expand the capabilities of AI in education by exploring the integration of multimodal data (e.g., video, images, and audio) to further enhance the interactivity and effectiveness of educational agents. for example, integrating a text-to-speech model to talk to the agents, a model to read emotions from the text, and a summarization model to summarize users' past chats and queries into a single prompt for the LLM's reference.

Investigating the scalability of such systems across different subjects and educational levels will also be crucial for broader application. Additionally, longitudinal studies could be conducted to assess the long-term impact of AI-enhanced learning tools on student

performance and retention. LLMs can customize educational content to suit each student's needs. They adapt to individual learning speeds, styles, and preferences. This makes learning more personal and effective.

6.6 Final Thoughts

In conclusion, this dissertation has effectively demonstrated the substantial benefits of integrating AI and NLP technologies in the educational sector. The creation and implementation of an educational agent represent a major step forward, highlighting the capabilities of applications that are not only more adaptable but also more efficient. As AI technology continues to evolve and advance, its incorporation into educational frameworks is expected to bring forth even more groundbreaking solutions. These innovations could greatly enhance teaching methods and learning experiences. We are just starting to blend AI into educational practices, and the future looks very promising. With continuous progress, we expect to witness transformative changes that could fundamentally reshape the educational scene, making learning more interactive and customized to individual needs. This venture into AI-enhanced education is set to open up new opportunities for both teachers and students, leading to a more dynamic and impactful educational environment.

Bibliography

[1] J. Beall, "The weaknesses of full-text searching," *The Journal of Academic Librarianship*, vol. 34, no. 5, pp. 438–444, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0099133308001067

[2] E. Adamopoulou and L. Moussiades, "Chatbots: History, technology, and applications," *Machine Learning with Applications*, vol. 2, p. 100006, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827020300062

[3] J. Shrungare, "Ai in education," *XRDS*, vol. 29, no. 3, p. 63–65, apr 2023. [Online]. Available: https://doi.org/10.1145/3589657

[4] A. Harry, "Role of ai in education," *Interdiciplinary Journal and Hummanity (INJURITY)*, vol. 2, no. 3, pp. 260–268, 2023.

[5] L. Chen, P. Chen, and Z. Lin, "Artificial intelligence in education: A review," *IEEE Access*, vol. 8, pp. 75 264–75 278, 2020.

[6] H. Jaakkola, J. Henno, and J. Mäkelä, "Computers in education," in *2023 46th MIPRO ICT and Electronics Convention (MIPRO)*, 2023, pp. 833–839.

[7] Z. Kolagar and A. Zarcone, "HumSum: A personalized lecture summarization tool for humanities students using LLMs," in *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, A. Deshpande, E. Hwang, V. Murahari, J. S. Park, D. Yang, A. Sabharwal, K. Narasimhan, and A. Kalyan, Eds. St. Julians, Malta: Association for Computational Linguistics, Mar. 2024, pp. 36–70. [Online]. Available: https://aclanthology.org/2024.personalize-1.4

[8] M. Murtaza, Y. Ahmed, J. A. Shamsi, F. Sherwani, and M. Usman, "Ai-based personalized e-learning systems: Issues, challenges, and solutions," *IEEE Access*, vol. 10, pp. 81 323–81 342, 2022.

[9] A. Anuyahong, C. Rattanapong, and I. Patcha, "Analyzing the impact of artificial intelligence in personalized learning and adaptive assessment in higher education," *International Journal of Research and Scientific Innovation*, vol. X, pp. 88–93, 05 2023.

[10] Y. Huang, "Application of natural language processing technology in educational resources retrieval," *Journal of Physics: Conference Series*, vol. 1881, no. 3, p. 032020, apr 2021. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1881/3/032020

[11] *International Journal of Innovative Research in Computer Science amp; Technology*, vol. 9, no. 6, p. 184–187, Nov. 2021. [Online]. Available: https://acspublisher.com/journals/index.php/ijircst/article/view/11000

[12] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," 2022.

[13] R. Zhong, K. Lee, Z. Zhang, and D. Klein, "Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections," 2021.

[14] Z. He, T. Liang, W. Jiao, Z. Zhang, Y. Yang, R. Wang, Z. Tu, S. Shi, and X. Wang, "Exploring Human-Like Translation Strategy with Large Language Models," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 229–246, 03 2024. [Online]. Available: https://doi.org/10.1162/tacl_a_00642

[15] A. A. Syed, F. L. Gaol, A. Boediman, T. Matsuo, and W. Budiharto, "A survey of abstractive text summarization utilising pretrained language models," in *Intelligent Information and Database Systems*, N. T. Nguyen, T. K. Tran, U. Tukayev, T.-P. Hong, B. Trawiński, and E. Szczerbicki, Eds. Cham: Springer International Publishing, 2022, pp. 532–544.

[16] S. Prabhumoye, A. W. Black, and R. Salakhutdinov, "Exploring controllable text generation techniques," 2020.

[17] "What is Retrieval-Augmented Generation? -amazon-aws," https://aws.amazon.com/what-is/retrieval-augmented-generation/.

[18] "What is retrieval-augmented generation? -ibm," https://research.ibm.com/blog/retrieval-augmented-generation-RAG.

[19] "RAG 101: Demystifying Retrieval-Augmented Generation Pipelines -nvidia," https://developer.nvidia.com/blog/rag-101-demystifying-retrieval-augmented-generation-pipelines/.

[20] "12 Best Large Language Models (LLMs) in 2024 -beebom," https://beebom.com/best-large-language-models-llms/.

[21] "Inside Zephyr-7B: HuggingFace's Hyper-Optimized LLM that Continues to Outperform Larger Models," https://towardsai.net/p/machine-learning/inside-zephyr-7b-huggingfaces-hyper-optimized-llm-that-continues-to-outperform-larger-models.

[22] "Exploring the World of Generative AI, Foundation Models, and Large Language Models: Concepts, Tools, and Trends," https://towardsai.net/p/machine-learning/exploring-the-world-of-generative-ai-foundation-models-and-large-language-models-concepts-tools-and

[23] "Django makes it easier to build better web apps more quickly and with less code." https://www.djangoproject.com/.

A1 Appendix

A1.1 Abbreviations

A comprehensive list of all the abbreviations used throughout the document have been presented here.

AI - Artificial Intelligence

API - Application Programming Interface

CSS - Cascading Style Sheets

HTML - HyperText Markup Language

HTTP - Hypertext Transfer Protocol

JS - JavaScript

LLM - Large Language Model

NLP - Natural Language Processing

NLTK - Natural Language Toolkit

ORM - Object-Relational Mapping

QA - Question Answering

RAG - Retrieval-Augmented Generation

SQL - Structured Query Language

SQuAD - Stanford Question Answering Dataset

TF-IDF - Term Frequency-Inverse Document Frequency