An Open-Source AI Chatbot for Course Advising: Leveraging Retrieval Augmented Generation

[redacted]

ABSTRACT

The selection of a university course is a crucial decision that profoundly influences a student's academic journey and future career opportunities. Given the vast array of options and the often complex navigation of university websites, this process can be particularly daunting, especially for international students. This thesis addresses this challenge by developing an AI-powered chatbot aimed at helping students easily access course information from various Technological Universities across Ireland. The chatbot is built on a Retrieval Augmented Generation (RAG) framework, which combines a text-generation model with a vector database to deliver contextually relevant and accurate responses to student inquiries.The research also includes a comparative analysis of different open-source Large Language Models (LLMs), specifically Llama3, Llama2, Mistral, Gemma, and Phi, to determine the most suitable model for the RAG system in this context. The evaluation is conducted using the Retrieval Augmented Generation Assessment Suite (RAGAS), which assesses the models based on metrics such as Answer Relevancy, Answer Correctness, Faithfulness, Context Precision, and Context Recall. The findings of the study underscore the strengths and limitations of each LLM, with Llama3 emerging as the most suitable model for the RAG system due to its superior performance across multiple metrics. The developed chatbot, implemented using Streamlit, simplifies the process of finding relevant course details, eliminating the need for students to manually navigate through numerous university websites. This research contributes to the expanding body of knowledge on AI chatbots in education and the evaluation of open-source LLMs, providing valuable insights for future research and development in this field.

Keywords: AI Chatbot, Course Selection, Open-Source LLMs, Retrieval Augmented Generation (RAG), Technological Universities in Ireland, Data Scraping, Text Generation, Model Evaluation, Answer Relevancy, Answer Correctness, Faithfulness, Context

Precision, Context Recall, Llama 2, Llama 3, Mistral, Phi, Gemma, ChromaDB, Streamlit, RAGAS

Contents

List of Figures

1 INTRODUCTION

The selection of a university course is one of the most pivotal decisions a student can make, as it profoundly influences their academic journey and professional trajectory. This process involves navigating a complex landscape of options, where students must balance personal aspirations with academic realities while making informed choices that align with their goals and capabilities (C. Brown, Varley, and Pal 2009). University websites serve as a critical resource in this decision-making process, particularly for prospective international students. These websites often represent the primary source of information, offering valuable insights into academic programs, campus life, and the overall image of the institution. The way universities present themselves online significantly impacts student perceptions and can ultimately sway their enrollment decisions. A well-designed, informative website enhances a university's attractiveness and contributes to a positive image, which is crucial in influencing student choices (Polat and Çelik 2022). Conversely, if a website has usability issues, such as poor navigation or unclear presentation of information, students may struggle to find essential details about course requirements, schedules, or prerequisites. This can lead to confusion and hinder their ability to make well-informed decisions (Yerlikaya and Durdu 2017).

When selecting a course in a different country, the decision-making process becomes even more challenging. Factors such as career opportunities, the reputation of the college, and the facilities provided can play a crucial role in the decision-making process (Hussin, Muhamad, and Sukor 2019). In this context, the significance of advancements in artificial intelligence (AI) in attracting students is increasingly evident. AI is transforming various aspects of the global education experience, including how students interact with universities, navigate the admissions process, receive academic support, and adapt to cross-cultural environments (Bansal et al. 2024).

1.1 Motivation

The role of AI in education is becoming increasingly important, revolutionizing the way students interact with universities around the world. AI assists in various tasks, including course selection, understanding university resources and facilities, and even helping students adapt to new cultural environments (Galstyan et al. 2024). One prominent example of AI's impact is the evolution of chatbots, which have progressed from basic question-answering tools to sophisticated conversational agents capable of providing human-like responses to student inquiries (Amin et al. 2024).

Generative AI, a specialized subset of AI focused on content creation, is driving significant advancements in Natural Language Processing (NLP). This technology enables the generation of text that is not only grammatically correct but also contextually relevant and convincingly human-like (Bai 2024). While chatbots are increasingly common in educational settings to enhance learning experiences, the use of advisory chatbots with distinct personalities remains less frequent. However, incorporating personality into these chatbots has the potential to significantly boost student trust and engagement, making them more effective in guiding students through their academic journey (Kuhail et al. 2022).

1.2 Research Aim and Objective

This thesis focuses on the development of an advisory chatbot specifically designed to guide students in their academic journey. The chatbot aims to provide essential information about the wide range of courses offered across Technological Universities in Ireland, thereby empowering students to make informed decisions about their educational paths.

The research involves a comparative analysis of various open-source Large Language Models (LLMs) to evaluate their text generation accuracy within the specific context of academic advising. This evaluation will be conducted through the implementation of a Retrieval Augmented Generation (RAG) pipeline, which utilizes a shared vector database to examine how this pipeline influences the performance of different LLM models.

The ultimate goal is to create a practical and effective chatbot tool that can genuinely assist students in navigating their academic choices, thereby optimizing their overall educational experience. By leveraging advanced AI technologies, this chatbot aims to simplify the course selection process, making it more accessible and less overwhelming for students.

2 LITERATURE REVIEW

2.1 Introduction

The rapid advancements in artificial intelligence (AI) and natural language processing (NLP) have led to the emergence of chatbots as transformative tools across various sectors, including education. Within educational institutions, chatbots are increasingly deployed to enhance student engagement, streamline administrative processes, and personalize learning experiences. These AI-driven systems offer greater accessibility, convenience, and precision compared to traditional human advisors, making them invaluable assets in academic environments.

The literature presents several notable developments in chatbot technology, particularly in the context of large language models (LLMs) and Retrieval-Augmented Generation (RAG) frameworks. These innovations highlight the growing importance of chatbots in enhancing the accessibility and effectiveness of educational resources.

Beyond chatbots, the evolution of LLMs has been a key area of focus, with significant breakthroughs in model architectures, training methodologies, and real-world applications. The public release of ChatGPT in November 2022 marked a pivotal moment, bringing LLM technology into mainstream use and sparking widespread interest in its capabilities. The ongoing evolution of LLMs continues to drive innovation across multiple domains, from natural language processing and content creation to code generation and scientific problem-solving. The dynamic nature of LLM research, as highlighted by Akhtar (2024) and others, promises continued advancements in AI technology, with significant implications for education and beyond. However, as LLMs become increasingly integrated into real-world applications, it is crucial to address the associated challenges, including issues of scalability, ethical considerations, and the need for responsible deployment.

This literature review explores these developments in detail, providing a comprehensive overview of the current state of chatbot technology, LLMs, RAG frameworks, prompt engineering, and the ongoing evolution of AI models.

2.2 The Role of Chatbots in Educational Institutions

Chatbots, powered by artificial intelligence (AI) and natural language processing (NLP), have emerged as transformative tools across various sectors, including education (Kumar 2024). Chatbots are increasingly employed within educational institutions to enhance student engagement, streamline administrative tasks, and personalize learning experiences (S. Yang, Dong, and Z. Yu 2024). In the educational sector, chatbots offering administrative or consulting services often provide greater accessibility, convenience, and precision compared to human advisors (Fitria, Simbolon, and Afdaleni 2023).

Odede and Frommholz (2024) presents JayBot, an advanced LLM-based chatbot system developed to enhance the user experience for prospective and current students, faculty, and staff at a UK university. The primary purpose of JayBot is to provide accurate and timely information on various topics, including course modules, fees, entry requirements, and career paths. JayBot is powered by OpenAI's GPT-3.5 turbo model, utilizing an embedding transformer model alongside a vector database and vector search to improve response accuracy and relevance, while also addressing issues such as hallucination. The development process involved sophisticated prompt engineering techniques to enhance JayBot's ability to effectively respond to user inquiries. Preliminary user studies have shown that JayBot is both effective and efficient in delivering necessary information, highlighting its potential utility in educational settings. Additionally, the paper discusses a demo that focuses on JayBot's application in university admissions and explores other potential use cases.

Galstyan et al. (2024) presents SmartAdvisor, an AI-driven chatbot designed to provide comprehensive course information and academic guidance to university students. The chatbot leverages a Retrieval-Augmented Generation (RAG) framework, combining a text-generation model with a vector database to ensure contextually relevant and data-driven responses. The system utilizes web scraping techniques to gather course data from the university's platform, which is then stored in a MySQL database. The chatbot's user-friendly interface, built with ChainLit, simplifies interaction and navigation, enabling students to access course descriptions, prerequisites, and semester-related details. The authors explore the use of both OpenAI's GPT models and Meta's Llama 3 model for text generation, highlighting the trade-offs between performance and cost-efficiency. The evaluation of SmartAdvisor focuses on the quality and relevance of generated responses, demonstrating the chatbot's effectiveness in providing accurate and informative academic counseling.

Wei et al. (2022) propose an intelligent course recommender system that assists students in selecting suitable courses based on their preferences and strengths. The system employs NLP techniques, including a Convolutional Neural Network (CNN) for Part-of-Speech tagging, to process student input and identify keywords as shown in Figure 1. These keywords are then matched with course keywords using TF-IDF and cosine similarity to rank and recommend relevant courses. The chatbot interface facilitates interactive communication, capturing student preferences through natural language conversations. The system's evaluation, conducted with students at Universiti Malaysia Sabah, demonstrates its acceptability and potential to aid students in making informed course selections.

Neupane et al. (2024) introduce BARKPLUG V.2, an LLM-based chatbot designed to provide information about various campus resources at Mississippi State University. The system utilizes a RAG pipeline, incorporating university data as an external knowledge base for domain-specific question-answering tasks. The chatbot's architecture comprises two phases: context retrieval and completion. The retrieval phase utilizes an embedding

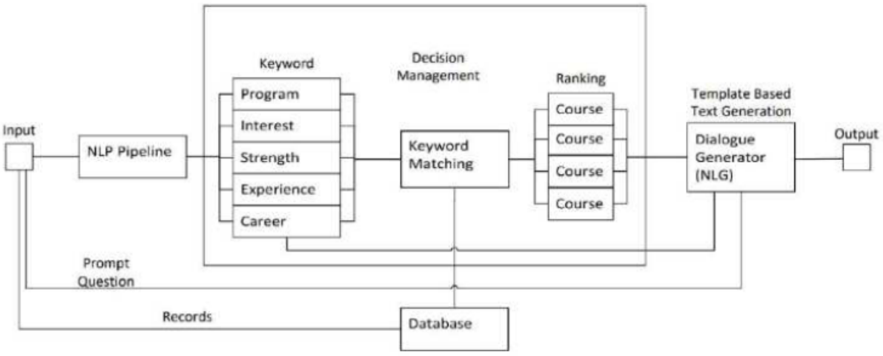


Figure 1: The architecture of the chatbot proposed in intelligent course recommender system (Wei et al. 2022).

model and a vector database to identify relevant documents based on user prompts. The completion phase leverages a GPT-based LLM to generate contextually appropriate responses using the retrieved documents. The system's evaluation employs both quantitative measures, such as the RAGAS framework, and subjective satisfaction surveys using the SUS. The results demonstrate BARKPLUG V.2's effectiveness in providing accurate and relevant information, highlighting its potential to enhance user engagement and access to university resources.

The RAG concept, though fairly new and coined by Lewis et al. (2021) in April 2021, was not implemented in the work of Wei et al. (2022), While the paper was relatively new when the RAG concept was coined and further utilized in the studies by Neupane et al. (2024) and Galstyan et al. (2024). The specifics of RAG are further elaborated upon in section 2.3.

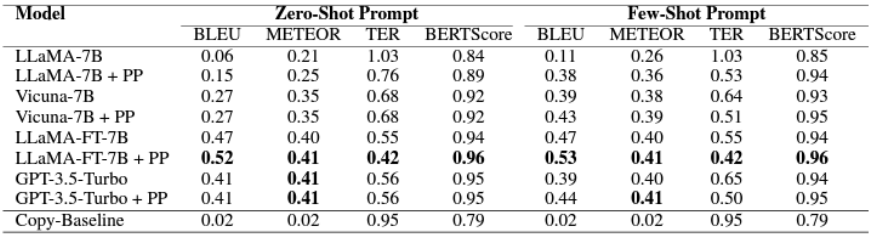
2.3 A Comparative Analysis of Open-Source Large Language Models

Large Language Models (LLMs) are advanced artificial intelligence systems designed to understand and generate human language. They are characterized by their significant size, often containing tens or hundreds of billions of parameters, which allows them to perform various natural language processing tasks effectively. They exhibit special abilities, such as in-context learning, which are not present in smaller models (Chang et al. 2024). This scaling effect enhances their performance across a range of tasks. Scaling up the size of LLMs, including increasing the number of parameters and the amount of training data, generally leads to enhanced model capacity. This means that larger models can better understand and generate language, resulting in improved performance on various tasks (Zhao et al. 2023).

ChatGPT's public release on November 30, 2022, marked a pivotal moment in making LLM technology accessible to a broader audience, moving beyond expert users to the general public. It serves as a ”killer application” that effectively demonstrates the capabilities of LLMs, allowing users to engage in dynamic conversations and understand the technology's potential.ChatGPT quickly became a cultural sensation, sparking widespread interest and discussions about the implications of LLMs in various fields, including education and business (Teubner et al. 2023). ChatGPT, while versatile, can produce incorrect or nonsensical responses. This inconsistency highlights the need for alternative models that may offer improved accuracy and reliability in generating responses. Different applications may require specific functionalities that ChatGPT does not fully support. Alternatives can be tailored to meet unique needs across various domains, such as specialized industries or specific user groups (Alipour, Pendar, and Roy 2024).

Chang et al. (2024) discusses Natural Language Generation (NLG) Evaluation, NLG assesses how well large language models (LLMs) can create texts, including tasks like summarization, dialogue, translation, and question answering. TNLG v2 performed best in summarization, while ChatGPT showed decent but not top performance. Evaluating LLMs in dialogue tasks is important for improving human-computer interactions. Claude and ChatGPT generally perform well, with Claude slightly better in some areas, and fine-tuned models often outperform ChatGPT in specific dialogue tasks.

Schneider et al. (2024) presents a comparative analysis of the performance of Large Language Models (LLMs) on the task of generating natural language text from knowledge graph triples. Figure 2 shows the models evaluated include GPT-3.5-Turbo, LLaMA, LLaMA-FT, and Vicuna. The evaluation is based on the WebNLG+ 2020 dataset, and the models are assessed using metrics such as BLEU, METEOR, TER, and BERTScore. The study explores both zero-shot and few-shot settings, with and without post-processing of the generated output. The results highlight the superior performance of LLaMA-FT, followed by GPT-3.5-Turbo and Vicuna. They also discuss common generation issues and the impact of few-shot prompting and fine-tuning on conversational data.



Model	**Zero-Shot Prompt**				**Few-Shot Prompt**			
	BLEU	METEOR	TER	BERTScore	BLEU	METEOR	TER	BERTScore
LLaMA-7B	0.06	0.21	1.03	0.84	0.11	0.26	1.03	0.85
LLaMA-7B + PP	0.15	0.25	0.76	0.89	0.38	0.36	0.53	0.94
Vicuna-7B	0.27	0.35	0.68	0.92	0.39	0.38	0.64	0.93
Vicuna-7B + PP	0.27	0.35	0.68	0.92	0.43	0.39	0.51	0.95
LLaMA-FT-7B	0.47	0.40	0.55	0.94	0.47	0.40	0.55	0.94
LLaMA-FT-7B + PP	**0.52**	**0.41**	**0.42**	**0.96**	**0.53**	**0.41**	**0.42**	**0.96**
GPT-3.5-Turbo	0.41	**0.41**	0.56	0.95	0.39	0.40	0.65	0.94
GPT-3.5-Turbo + PP	0.41	**0.41**	0.56	0.95	0.44	**0.41**	0.50	0.95
Copy-Baseline	0.02	0.02	0.95	0.79	0.02	0.02	0.95	0.79

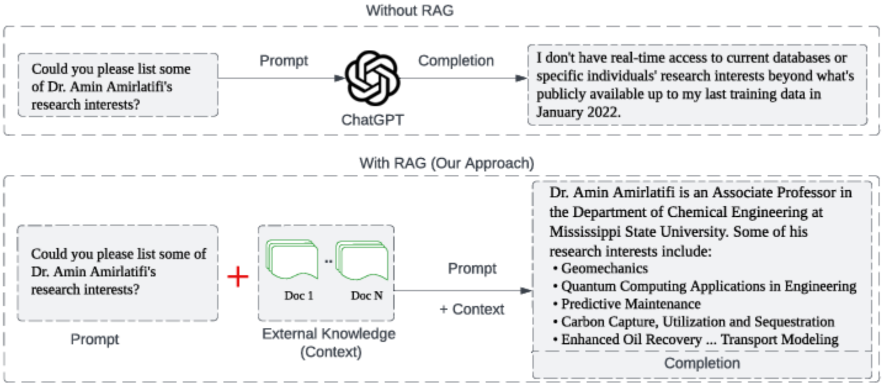
Figure 2: Performance metrics on WebNLG testset evaluated by BLEU, METEOR, TER, and BERTScore-F1 (Schneider et al. 2024).

Mahapatra and Garain (2024) investigates the impact of model size on the performance of fine-tuned Large Language Models (LLMs) in data-to-text (D2T) generation tasks. The study evaluates 12 LLMs from 5 families (BART, T5, OPT, BLOOM, and Llama 2) across 5 datasets (E2E, ViGGO, WikiTableText, DART, and WebNLG) using 6 metrics (BLEU, METEOR, BERTScore, MoverScore, Parent, and BARTScore) to assess three key qualities: readability, informativeness, and faithfulness. The findings reveal that increasing model size generally improves readability and informativeness but may negatively impact faithfulness. The study also highlights that larger LLM families do not consistently outperform smaller ones, and smaller models show more resilience in handling source-reference divergence.

In addition to the findings of Chang et al. (2024) and Schneider et al. (2024) regarding how open-source models like LLaMA and Vicuna are holding their ground and even outperforming ChatGPT in some aspects Peng et al. (2024) demonstrate that further fine-tuning models can yield consistent results that surpass those of closed-source models (GPT-3.5-Turbo, GPT-4o). The paper compares the performance of its proposed model, Review-LLM, which is built on Llama-3 using Supervised Fine-Tuning (SFT), against several reference models and variations on the task of personalized review generation. The evaluation is conducted on two types of datasets: a simple evaluation dataset and a hard evaluation dataset designed to test the generation of negative reviews. The key metrics used for evaluation are ROUGE-1/L and BERT similarity score (BertScore). The results demonstrate that Review-LLM consistently outperforms all other models across both datasets and all metrics, highlighting the effectiveness of its approach in leveraging user historical behavior and ratings for personalized review generation.

2.4 Benefits of Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) addresses the limitations of LLMs in handling extensive contextual information by leveraging external knowledge bases. This integration enables LLMs to access a broader range of information, leading to the generation of more contextually relevant and accurate outputs, particularly in tasks that demand a deep understanding of specific domains or topics (Jiang, Fan, and Y. Yu 2024). RAG (Retrieval Augmented Generation) allows chatbots to access a broader range of information by retrieving relevant data from external sources. By combining retrieval with generation, RAG can provide more accurate and contextually relevant answers. This is achieved through the use of embeddings and local Large Language Models (LLMs), which help in understanding user queries better (Burgan, Kowalski, and Liao 2024). The effectiveness of the RAG method can be assessed using various metrics such as Recall, Precision, BLEU, and ROUGE scores. This systematic evaluation helps in refining the chatbot's performance over time (Maryamah et al. 2024). Figure 3 illustrates how RAG influences response generation specific to an individual at Mississippi State University. By providing sufficient data to the RAG model, it can generate more coherent and contextually relevant responses, as demonstrated in the work by Neupane et al. (2024).



Could you please list some of Dr. Amin Amirlatifi's research interests?

I don't have real-time access to current databases or specific individuals' research interests beyond what's publicly available up to my last training data in January 2022.

Could you please list some of Dr. Amin Amirlatifi's research interests?

Dr. Amin Amirlatifi is an Associate Professor in the Department of Chemical Engineering at Mississippi State University. Some of his research interests include:

- Geomechanics
- Quantum Computing Applications in Engineering
- Predictive Maintenance
- Carbon Capture, Utilization and Sequestration
- Enhanced Oil Recovery ... Transport Modeling

Figure 3: Comparative example of completion without using the RAG approach versus using the RAG approach for a given user prompt related to a specific individual at Mississippi State University (Neupane et al. 2024).

Feldman, Foulds, and Pan (2024) empirically evaluates RAG against standard LLMs using prompts specifically designed to induce hallucinations. The results indicate that RAG can improve accuracy in certain scenarios, demonstrating its effectiveness in countering hallucinations. Despite its advantages, RAG is not foolproof. The model can still be misled when prompts contradict its pre-trained understanding, highlighting the complexity of hallucinations and the need for further improvements. The context effectiveness analysis conducted by Feldman, Foulds, and Pan (2024) reveals substantial differences in accuracy across both scenarios, highlighting the crucial interplay between context and prompting in language models.

Z. Wang et al. (2024) introduces M-RAG, a novel method designed to enhance Large Language Models (LLMs) by incorporating Retrieval-Augmented Generation (RAG) with a focus on memory retrieval from an external database. Traditional RAG methods typically organize all memories within a single database, which can lead to a dilution of focus on critical memories and the introduction of irrelevant information. In contrast, M-RAG employs a multiple partition approach, where each partition serves as a distinct unit for RAG execution, enabling more targeted memory retrieval. The framework further integrates Multi-Agent Reinforcement Learning to optimize various language generation tasks. Through experiments conducted on seven datasets, M-RAG demonstrated its effectiveness, achieving significant improvements over baseline methods, including an 11 percent increase in text summarization accuracy, an 8 percent improvement in machine translation, and a 12 percent enhancement in dialogue generation. This approach highlights the innovation of M-RAG in refining memory retrieval processes to boost the performance of language generation tasks.

2.5 Prompt Engineering

Poorly constructed questions can significantly impact the performance of GPT models, particularly in their ability to understand context. When questions lack clarity or specificity, GPT models often struggle to generate appropriate responses, as these models rely heavily on well-defined inputs to interpret context accurately. Additionally, vague or ambiguous questions can undermine the coherence of the generated responses. While GPT models are generally proficient at producing coherent text, unclear prompts can lead to less relevant or even nonsensical answers. Moreover, the diversity of responses generated by GPT can be limited by poorly framed questions. Instead of offering varied and contextually rich answers, the models may default to generic or common responses, which diminishes the effectiveness of communication. Handling rare or out-of-domain inputs poses another challenge for GPT models. When a question deviates from common language patterns due to poor construction, the model's performance can suffer, leading to suboptimal outcomes in tasks such as question-answering. Overall, the quality of the questions posed plays a critical role in determining the performance of GPT models in natural language processing tasks. Clear and well-structured questions are essential for maximizing the effectiveness of these models, enabling them to generate useful and accurate responses (Thakkar 2023) (Yenduri et al. 2024).

Prompt engineering is the process of designing and optimizing prompts to effectively communicate with Large Language Models (LLMs) like ChatGPT. It involves crafting instructions that guide the model to produce desired outputs. As LLMs become more prevalent in conversational AI, prompt engineering has emerged as a crucial skill. It helps to ensure high-quality and relevant responses from these models (Marvin et al. 2024). The two main aspects of prompt engineering according to Liu et al. (2021) are Prompt Shape and Prompt Creation. Prompt shape focuses on the structural format of the prompt, distinguishing between cloze prompts (filling in blanks) that suit masked language models and prefix prompts (continuing a string) that are ideal for generative tasks. Prompt Creation deals with the methods used to generate the prompt, which can be either manual, relying on human expertise, or automated, utilizing algorithms to discover optimal templates.

One of the most popular techniques under Prompt engineering is Chain-of-Thought (CoT), used to enhance the reasoning capabilities of Large Language Models (LLMs). It involves prompting the LLM to generate intermediate reasoning steps before arriving at a final answer, similar to how humans break down complex problems. This technique is closely tied to prompt engineering, as the specific wording and structure of the prompt can significantly impact the model's ability to generate coherent and accurate chains of thought (Z. Zhang et al. 2022).

Cao et al. (2024) performed tests to assess how certain LLM models are highly sensitive to the phrasing of prompts. This evaluation included six open-source LLMs from the Llama, Mistral, and Gemma families, as well as ChatGPT. The results demonstrated that although larger models generally exhibit superior average performance, they do not necessarily show improved robustness to variations in prompts.

2.6 Evolution of Large Language Models

The evolution of LLMs, as highlighted by Akhtar (2024) has been marked by significant breakthroughs in model architectures, training methodologies, and real-world applications. The transformative power of LLMs is evident in their ability to generate coherent and contextually relevant text, translate languages, write different kinds of creative content, and provide informative answers to a wide array of questions.

The advancements of LLMs have created an interest in research and development, leading to the implementation of many open-source models with diverse use cases. S. Gao and A. Gao (2023) presents a comprehensive analysis of the evolutionary relationships among a vast collection of LLMs, offering insights into the prominent model families and their underlying structures. At the time when this study was done almost 15,821 Large Language Models were in use. which hasn't stopped growing since. Chen et al. (2024) explores the transformative impact of LLMs on recommender systems, highlighting their potential to redefine personalization technologies and interfaces. The evolution of recommender systems, from traditional list-wise recommendation to conversational recommendation powered by LLMs, and discusses the challenges and opportunities in this evolving field. Hemberg, Moskal, and O'Reilly (2024) delves into the novel concept of using LLMs to evolve code, presenting LLM GP, a formalized LLM-based evolutionary algorithm designed for code evolution. The study explores the design and implementation of LLM-based operators, prompt functions, and preparatory steps involved in utilizing LLMs for genetic programming, offering insights into the potential of this approach. Sohail and L. Zhang (2024) provides a comprehensive summary of recent literature on the capabilities of Large Language Models (LLMs) in academic research and teaching. It identifies three key areas where LLMs can be especially beneficial: education and assessment, academic writing, and simulating human behavior. This structured approach helps clarify the potential uses of LLMs in these fields. Additionally, the current challenges of integrating LLMs into academic work, including ethical and practical concerns, and propose good practices to guide researchers and educators in their use. Future directions for research and the application of LLMs, aiming to enhance awareness and proper usage within academic settings. By highlighting both the benefits and barriers to using LLMs.

Along with numerous others in the field, collectively illustrate the dynamic and rapidly evolving nature of LLM research. The ongoing advancements in LLMs promises a transformation in various domains, from natural language processing and content creation to code generation and scientific problem-solving. As LLM technology continues to mature, it is crucial to address the associated challenges and ensure responsible and ethical development and deployment of these powerful models.

2.7 Evaluation of RAG models

Evaluating Retrieval-Augmented Generation (RAG) models is crucial for understanding their effectiveness in various applications, particularly in domain-specific contexts. Recent studies highlight the strengths and weaknesses of RAG systems, emphasizing the need for comprehensive benchmarks and evaluation frameworks.

S. Wang et al. (2024) developed a comprehensive dataset called DomainRAG to evaluate critical abilities of RAG models in domain-specific scenarios, specifically in college enrollment. They crawled relevant webpages and created two types of corpora—HTML and pure text—to construct sub-datasets for assessing various RAG capabilities, including conversational RAG, structural information analysis, faithfulness to external knowledge, denoising, handling time-sensitive problems, and understanding multi-document interactions. Their experiments demonstrated the effectiveness of RAG models in tackling domain-specific challenges that LLMs alone struggle with, particularly in answering expert-level questions. However, the study also highlighted areas where RAG models need further improvement, such as better comprehension of conversational history, structural knowledge analysis, denoising, managing multi-document interactions, and maintaining fidelity to expert knowledge. The authors suggest that future research should focus on advancing these aspects to enhance RAG model performance.

The Comprehensive RAG Benchmark (CRAG) introduced by X. Yang et al. (2024) advances research in retrieval-augmented generation (RAG) for question answering (QA) tasks. CRAG includes 4,409 question-answer pairs and mock APIs that simulate web and Knowledge Graph (KG) searches, covering a wide range of questions across five domains and eight categories. The evaluation of current advanced large language models (LLMs) on CRAG revealed that they achieve only up to 34 percent accuracy, highlighting significant room for improvement in QA systems. Incorporating RAG methods increased accuracy to 44 percent, but even the best industry solutions could only correctly answer 63 percent of the questions without hallucinations, underscoring the challenges of creating reliable QA systems. The benchmark also showed that accuracy is particularly low for questions with higher dynamism, lower popularity, or greater complexity, indicating key areas for future research.

BERTSCORE is an automatic evaluation metric for text generation that leverages BERT contextual embeddings. It calculates the similarity between a candidate sentence and a reference sentence by summing the cosine similarities between their token embeddings. The metric addresses the limitations of traditional n-gram based metrics like BLEU by effectively handling paraphrases and capturing distant dependencies. BERTSCORE offers flexibility through importance weighting (e.g., using IDF) and supports multiple languages.(T. Zhang et al. 2020)

Shahul et al. (2023) primarily focuses on the evaluation of RAG systems, not their direct practical application. The main practical implication is that RAGAS, the proposed evaluation framework, can help developers build and improve RAG systems more efficiently. By providing reference-free metrics for faithfulness, answer relevance, and context relevance, RAGAS allows for faster evaluation cycles, which is crucial given the rapid adoption of LLMs. This can lead to more effective and reliable RAG systems in various real-world applications, such as question-answering systems, chatbots, and information retrieval tools. this method of evaluation can be seen in works of Neupane et al. (2024) emphasizing that the RAGAS framework is specifically designed to assess RAG pipelines. It contrasts RAGAS with popular evaluation metrics such as ROUGE and BLEU, highlighting that ROUGE is primarily used for evaluating summarization tasks, while BLEU is designed for evaluating language translation tasks.(Papineni et al. 2002)(Lin 2004)

2.8 Research Gap

In this section, the research gaps identified in the literature review are discussed. The concept of RAG, although relatively new and first introduced by Lewis et al. (2021). in April 2021, was not implemented in the work of Wei et al. (2022). Despite the ease of learning and implementing of the RAG concept at that time, it was later utilized in studies by Neupane et al. (2024) and Galstyan et al. (2024). The findings of Chang et al. (2024) and Schneider et al. (2024) reveal that open-source models like LLaMA and Vicuna are not only holding their ground but, in some cases, outperforming proprietary models like ChatGPT. Peng et al. (2024) further demonstrate that fine-tuning these open-source models can lead to consistent results that surpass those of closed-source models such as GPT-3.5 Turbo and GPT-4.

However, the lack of open-source LLM models in the chatbots developed by Odede and Frommholz (2024), Neupane et al. (2024), and Galstyan et al. (2024) raises questions that will be explored in this work. Additionally, while Feldman, Foulds, and Pan (2024) accepts that RAG models are not foolproof and can still produce hallucinations, Marvin et al. (2024) suggests that this issue could be mitigated by imposing sufficient rules on the model to set boundaries for creativity. This research will also investigate how open-source models handle hallucinations and explore strategies to reduce this phenomenon.

3 RESEARCH METHODOLOGY

3.1 Introduction

This research is dedicated to the development of a chatbot utilizing open-source Large Language Models (LLMs) designed to assist students in efficiently accessing course details from various Technological Universities across Ireland. The primary goal of this chatbot is to simplify the process of locating relevant course information, thereby reducing the need for students to manually navigate through multiple course pages on different university websites.

Previous studies have demonstrated the creation of chatbots that engage in dialogue pertaining to a single institution or focus on a specific use case within that institution. However, this chatbot distinguishes itself by aggregating and consolidating course information from a range of educational institutions within a specific geographic region. Furthermore, unlike many earlier works that predominantly relied on proprietary, closed-source models for generating text, this study emphasizes the use of open-source models such as Llama, Mistral, and Gemma. The text generation capabilities of these models are rigorously evaluated based on criteria including accuracy, relevance, and coherence.

3.2 Data Collection and storage

Course details were extracted from the websites of Technological Universities located in Ireland, including Technological University of the Shannon, Technological University Dublin, South East Technological University, Munster Technological University, and Atlantic Technological University (Sligo campus). The BeautifulSoup and Langchain packages were utilized for this scraping process.

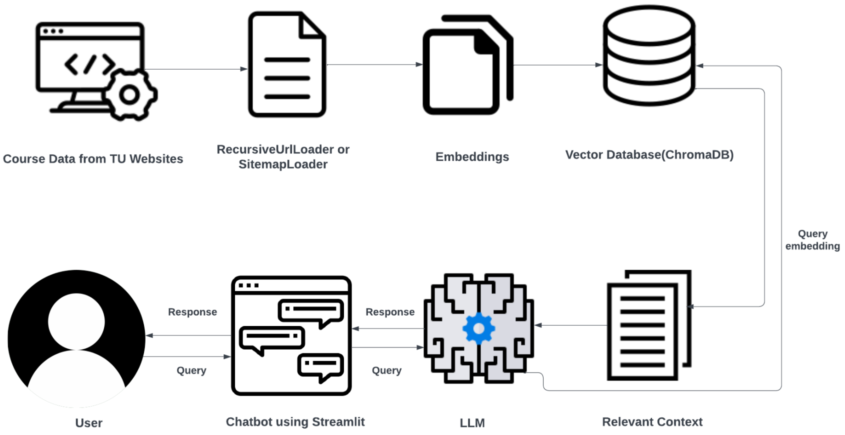


Figure 4: Methodology

3.2.1 Challenges faced while Scraping

Initially, the data scraping process began with the use of the RecursiveURLLoader to gather information from the Technological University of the Shannon (TUS). However, as the project progressed, the approach evolved, ultimately leading to the inclusion of the SitemapLoader in conjunction with the RecursiveURLLoader.

RecursiveURLLoader is a dynamic document loader provided by the LangChain package, which scrapes data recursively by following all child links from a root URL and parsing them into documents. This loader also supports the use of extractors, allowing for the customization of the scraping process by enabling users to extract specific data elements using tools like BeautifulSoup. However, the recursive scraping approach proved to be time-consuming and often resulted in connection timeouts, which hindered the efficiency of data collection. This challenge led to the exploration of sitemaps, which are structured lists containing all the webpage links within a domain, used by web browsers to facilitate easier navigation of websites.

Some universities maintained well-categorized sitemaps, which proved instrumental in streamlining the data scraping process through the use of the WebBaseLoader and SitemapLoader. Initially, the WebBaseLoader was employed, where all the links from the sitemap were compiled into a list and then fed into the WebBaseLoader, which subsequently collected data from those links. However, a significant limitation of the WebBaseLoader was its lack of support for extractors that were as versatile as those available in the RecursiveURLLoader.

To address this limitation, the SitemapLoader was introduced, as it not only utilized sitemaps but also offered the capability to customize the extraction process. As a result, two distinct approaches were ultimately employed for data collection. For websites with an accessible sitemap, data scraping was conducted by targeting the links within the sitemap and iterating over each one to load the data using LangChain's SitemapLoader. This method significantly reduced unnecessary website crawls compared to the RecursiveURLLoader.

However, for institutions that lacked an accessible sitemap, the RecursiveURLLoader was still necessary. Specifically, data from Technological University of the Shannon (TUS), Technological University Dublin (TUD), South East Technological University (SETU), and Atlantic Technological University (ATU) Sligo campus were collected using the SitemapLoader. Meanwhile, data from Munster Technological University (MTU) and ATU's St. Angela's College were gathered using the RecursiveURLLoader.

The decision to scrape only the Sligo campus details from the Atlantic Technological University was driven by the fact that the main ATU website redirects to outdated links associated with individual institutions that existed prior to the formation of the technological university. Additionally, the websites for the Letterkenny and Galway campuses of ATU did not maintain accessible sitemaps like the Sligo campus, which posed a significant challenge for data scraping. Addressing this challenge will be an important focus in future work.

3.2.2 Splitting text into chunks using LangChain

Chunking refers to the technique of dividing large texts into smaller, manageable segments or chunks. This process is particularly important in the context of Retrieval-Augmented Generation (RAG). RAG is often used to enable Large Language Models (LLMs) to answer questions based on extensive datasets, such as technical manuals, onboarding guides, and other substantial documents. Providing large documents as input to an LLM forces the model to crawl through vast amounts of information to find the most relevant details to respond to user queries. This task can be difficult because LLMs process input sequentially, token by token. Although these models can capture long-range dependencies and context to some degree, their effectiveness decreases as the length of the input sequence grows. This is where chunking becomes valuable. By chunking, you break down large documents into smaller segments, allowing you to embed these chunks into the database rather than the entire document. When a user query is made, only the most relevant chunks need to be retrieved, reducing the number of input tokens and providing a more focused context for the LLM to work with, (Joshi 2024).

After loading the data using either of the loaders, it is divided into chunks of several characters, depending on the specific Technological University (TU) website from which it was scraped. An overlap is maintained between each chunk to preserve contextual continuity. The Langchain library offers various text splitters for this purpose:

RecursiveCharacterTextSplitter: This is the recommended starting point for text splitting. It recursively divides the text while striving to keep related sections together, making it suitable for general-purpose text splitting.

HTMLHeaderTextSplitter, HTMLSectionSplitter: These splitters are designed specifically for HTML content, dividing the text based on certain HTML tags. They also include metadata about the chunk's origin within the HTML structure.

MarkdownHeaderTextSplitter: Similar to the HTML splitters, this splitter is tailored for Markdown content, dividing the text based on Markdown-specific elements and adding relevant metadata.

CharacterTextSplitter: A straightforward method that splits text based on a user-defined character, providing basic segmentation.

Semantic Chunker: This splitter initially divides text into sentences and then combines adjacent sentences if they are semantically similar. Although potentially powerful, this method is still experimental.

In this work, the RecursiveCharacterTextSplitter was chosen. The main reasons for selecting this splitter are its speed compared to some other options and the fact that a Markdown-based splitter seemed unnecessary, as the extractors used during scraping already handle the extraction of text data. Additionally, the RecursiveCharacterTextSplitter is more efficient than the Semantic Chunker, particularly when managing large volumes of scraped data, where the Semantic Chunker, although effective, is extremely slow. Therefore, the RecursiveCharacterTextSplitter offers an optimal balance between maintaining context, ensuring efficiency, and aligning with the data format, making it the best choice for this project.

3.2.3 Importance of Embeddings

The chunks created by the text splitter are then transformed into embeddings. Embeddings are numerical representations of text that capture semantic meaning and relationships, essentially functioning as mathematical codes that Large Language Models (LLMs) use to understand and compare the meaning of different text segments. The process of creating embeddings involves tokenization, where the text is broken down into smaller units—such as words, subwords, or characters—each of which is assigned a unique numerical vector that represents its meaning within a multi-dimensional space. In this space, tokens with similar meanings have vectors that are positioned closer together. LLMs analyze these vector relationships to determine similarity, relatedness, or relevance, forming the foundation for various natural language processing tasks. Embeddings are crucial for LLMs because they enable the models to comprehend meaning beyond literal definitions, allowing them to perform complex tasks like text classification and question answering. Additionally, embeddings facilitate the efficient processing of large volumes of data due to their compressed and structured representation.

The process of converting text chunks into embeddings in this project is carried out using OllamaEmbeddings, a feature supported by LangChain, which leverages the nomic-embed-text model to generate embeddings. The nomic-embed-text model is a high-performance, open-source embedding model that is part of Ollama's collection of models. By utilizing the OllamaEmbedding class from LangChain, it is possible to generate embeddings using various LLM models like Llama2 or Llama3. However, nomic-embed-text is specifically designed for creating high-quality embeddings.

NomicEmbeddings can be directly accessed from the extensive library of embeddings available within LangChain, but the installation and use of nomic are simplified because Ollama is already integrated into the development of the Retrieval-Augmented Generation (RAG) framework. The embeddings generated by the nomic-embed-text model are ideally suited for storage in a specialized database known as a vector database, which is designed to efficiently store and manage embeddings.

In this project, the vector database used is called ChromaDB. Chroma is an open-source vector database specifically built for handling LLM-generated embeddings, facilitating seamless data access, and streamlining application development. It efficiently stores both the embeddings and their associated metadata, ensuring that the data is easily retrievable and can be effectively used in various downstream applications (Huber and Troynikov n.d.).

3.3 Large Language Models

Large Language Models (LLMs) are trained on massive datasets, enabling them to generate contextually relevant text across various domains. Their applications span chatbots, content creation, code generation, and scientific research, with industries like healthcare and finance also leveraging their capabilities for tasks like summarization and translation (T. B. Brown et al. 2020). However, challenges such as bias in training data, high computational costs, and the potential for generating misleading information persist (Bender et al. 2021). Ollama is an open-source project that simplifies the use of LLM models on local machines. It streamlines the often tedious process of individually downloading and installing LLM models and boasts a vast collection of open-source models. In this work, Llama2, Llama3, Mistral, Phi, and Gemma are utilized for a comparative analysis of text generation capabilities among open-source LLMs.

3.3.1 Llama

Llama is a large language model developed by Meta AI. Llama is a versatile LLM designed for tasks like text generation, summarization, and translation. Trained on a massive dataset of 1.4 trillion tokens, it comes in various sizes, with the largest boasting 70 billion parameters. Meta has released multiple versions of Llama each new version that improves upon its predecessor, with Llama 3 being the most recent, offering advanced architecture and training techniques for better context understanding and accuracy. In this work, both Llama 2 and Llama 3 are considered.

3.3.2 Mistral

Mistral is developed by Mistral AI, a French company specializing in artificial intelligence products. It comes with 7 billion parameters. Despite having fewer parameters than some other models, Mistral is designed to be efficient in terms of both computation and memory usage, making it suitable for deployment in environments with limited resources. There are also other models like CodeMistral and MathMistral, which focus on AI code generation and STEM subjects respectively.

3.3.3 Phi

Phi is a small language model with 2.7B parameters developed by Microsoft, designed to understand language and generate optimal results even with fewer parameters. The goal of using a small language model is to explore the capabilities of text generation in models with limited parameter counts. Phi is particularly useful in situations where deploying larger models may not be feasible due to resource constraints. It can be used in edge devices, mobile applications, and other scenarios requiring efficient text processing.

3.3.4 Gemma

Gemma is a lightweight open model developed by Google and its DeepMind team. It is inspired by the Gemini models, which are known for their high performance in various natural language processing tasks. Gemma is available in 2 billion and 7 billion parameter versions. The training process for Gemma involved exposure to a wide range of text data, allowing it to handle different linguistic styles, topics, and vocabularies effectively. This makes it versatile and adaptable to various applications. Gemma 7B is used in this analysis.

3.4 How Model files affect the responses from the LLM

Model files significantly impact an LLM's behavior in several ways. They serve as the knowledge repository, encompassing facts, language understanding, and even potential biases from the training data. The training process and data influence the style and tone of responses, allowing for customization to suit specific needs. Model files dictate task performance; some excel at code generation, others at creative writing, highlighting the need to choose a model that aligns with the desired application. Additionally, model files can inadvertently encode biases, necessitating careful selection to ensure responsible and safe AI behavior. Finally, larger model files require more computational resources, emphasizing the trade-off between capabilities and efficiency. In summary, the choice of model file plays a pivotal role in determining an LLM's knowledge base, response style, task capabilities, and ethical implications, making it a crucial factor in harnessing the full potential of these powerful tools.

When creating model files for each LLM being compared in this analysis, while the fundamental principles and guidelines will remain consistent, the specific syntax for each LLM will differ. The following model instructions aim to achieve the desired responses from each LLM model:

```
Core Instructions
Data Source and Scope:
You have access to course data from Technological Universities in
    Ireland, specifically:
Technological University of the Shannon (TUS)
Technological University Dublin (TUD)
South East Technological University (SETU)
Munster Technological University (MTU)
Atlantic Technological University (ATU)
Utilize the metadata from ChromaDB to accurately identify which
   course data belongs to which university.
User Queries:
Your primary function is to answer student queries about these
   courses.
Queries may include questions about course details, entry
   requirements, fees, etc.
```

Response Guidelines:

Accuracy is paramount: Only provide information that is explicitly present in the available course data. Do not hallucinate or invent details.

Conciseness: Keep your responses clear and to the point. Avoid unnecessary verbosity.

Neutrality in Comparisons: When asked to compare courses, present the factual details of each course side-by-side. Refrain from offering personal opinions or recommendations. Leave the final decision to the user.

Admit Ignorance: If you cannot find the answer in the provided data, state that you don't have the information. Do not attempt to guess or provide inaccurate information.

University Abbreviations:

Remember and use the following abbreviations consistently:

Technological University of the Shannon: TUS

Technological University Dublin: TUD

South East Technological University: SETU

Munster Technological University: MTU

Atlantic Technological University: ATU

Do not confuse these abbreviations or use any others.

Example Interactions

User: "What are the entry requirements for the BSc in Computer Science at TUS?"

Chatbot: (Retrieves and presents the relevant entry requirements from the TUS course data)

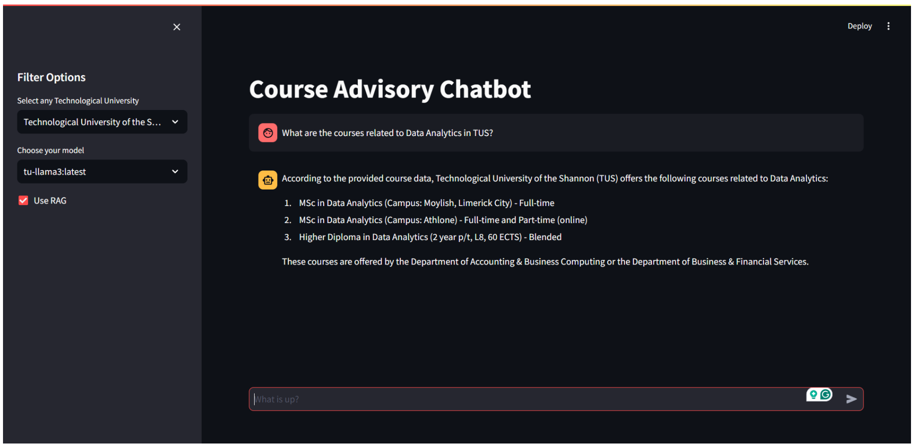
```
User: "Does ATU offer any part-time PhD programs in Engineering?"
Chatbot: (If the information is not available in the ATU course
   data) "I'm sorry, I don't have information about part-time PhD
    programs in Engineering at ATU."
```

These instructions outline the operation of the Large Language Models designed to answer student queries about courses at Irish Technological Universities. The chatbot relies on a database (ChromaDB) of course information, using prompts to accurately associate data with the correct university. The instructions emphasize the importance of accuracy and conciseness in responses, and the chatbot should maintain neutrality when comparing courses, leaving the final decision to the user. If information is unavailable, the chatbot should admit its lack of knowledge rather than provide potentially inaccurate information. The instructions also include a list of abbreviations for the universities that the chatbot should use consistently.

3.5 Creating a chatting interface using Streamlit package

Streamlit is an open-source Python library specifically designed for creating interactive web applications from your Python scripts. It simplifies the process of building custom data visualization tools, dashboards, machine learning demos, and other types of web apps without needing extensive knowledge of front-end development. In this project, Streamlit is used to create a webpage with a chatting interface similar to some popular chatbots in the market. The main reason for choosing Streamlit is its simplicity; it uses Python syntax, making development and integration into the application easier.

The Application sets up a Streamlit-powered chatbot application that utilizes Retrieval Augmented Generation (RAG) for context-aware responses, and it begins by importing essential libraries like Streamlit, Ollama, and components from the langchain-community library for vectorstore management and embeddings, then it defines a function to create a



Filter Options

Select any Technological University

Technological University of the S...

Choose your model

tu-llama3:latest

☑ Use RAG

Course Advisory Chatbot

What are the courses related to Data Analytics in TUS?

According to the provided course data, Technological University of the Shannon (TUS) offers the following courses related to Data Analytics:

1. MSc in Data Analytics (Campus: Moylish, Limerick City) - Full-time
2. MSc in Data Analytics (Campus: Athlone) - Full-time and Part-time (online)
3. Higher Diploma in Data Analytics (2 year p/t, L8, 60 ECTS) - Blended

These courses are offered by the Department of Accounting & Business Computing or the Department of Business & Financial Services.

Figure 5: Chatbot Interface developed using Streamlit

Chroma vectorstore, specifically designed for storing information related to ”COURSES” and using Ollama embeddings to represent the textual data within this vectorstore, and another crucial function, rag-chain, handles the core RAG functionality by retrieving relevant documents from the vectorstore based on user queries, formatting the context, and constructing prompts for the Ollama language model to generate informed responses. The Streamlit app itself is then built, starting with setting the title and initializing session state variables to manage chat history, model selection, and the vectorstore, and it allows users to choose their preferred Ollama model and the university from a dropdown list and efficiently loads or creates the vectorstore using Streamlit caching. The chat history is displayed interactively, showing both user and assistant messages in distinct chat bubbles, and when a user provides input, the prompt is added to the chat history, and if RAG is enabled, the rag-chain function is called to retrieve context and generate a response, otherwise, the response is obtained directly from the language model without RAG, and in both cases, the assistant’s response is streamed back and incorporated into the chat history, and finally, a checkbox is provided to allow users to toggle the use of RAG, enhancing the chatbot’s ability to provide contextually relevant and informative answers.

3.6 Evaluation of RAG pipeline using Ragas

Evaluating the RAG pipeline is a critical aspect of this research. Initially, the plan was to manually evaluate the responses generated by the model. However, as the research progressed, it became clear that there are many existing evaluation methods that were either resource-intensive or expensive, especially those requiring the use of OpenAI API services to compare the model's output against GPT models provided by OpenAI.

One such framework is Giskard, which uses GPT models to create a comprehensive test set by generating a series of questions and ground truths to evaluate the model across various metrics. Although it is possible to use Giskard with Ollama models, testing was unsuccessful due to dependency issues, prompting the search for alternative frameworks.

RAGAS (Retrieval Augmented Generation Assessment Suite) emerged as a promising evaluation framework for assessing the performance of RAG models. In contrast to Giskard, RAGAS offers the ability to manually frame the necessary data so that the framework evaluates the LLM responses without using any LLM or GPT models. Ragas uses metrics such as answer relevancy, answer correctness, faithfulness, context precision, and context recall to evaluate the RAG pipeline.

3.6.1 The key elements required for evaluation

Question: The query posed by the user to the model.

Answer: The response generated by the LLM model in reply to the user's query.

Context: The information or contexts provided to the LLM to help formulate an accurate response to the question.

Ground Truth: The correct or expected answer to the question, used as a benchmark for evaluation.

3.6.2 The Metrics used in the evaluation of the RAG pipeline

Answer Relevancy: Answer Relevancy measures how closely the generated answer aligns with the given prompt. Answers that are incomplete or include unnecessary information receive lower scores, while those with better relevance are rated higher. This metric is determined based on the question, the context, and the answer.

Answer Correctness: Answer Correctness evaluates the accuracy of the generated answer by comparing it to the ground truth. This evaluation is based on both the ground truth and the generated answer, with scores ranging from 0 to 1. A higher score reflects a closer match between the generated answer and the ground truth, indicating greater correctness. Answer correctness includes two key dimensions: semantic similarity between the generated answer and the ground truth, and factual accuracy. These dimensions are combined using a weighted approach to calculate the overall answer correctness score.

Faithfulness: Faithfulness evaluates the factual consistency of the generated answer in relation to the provided context. It is determined based on the answer and the retrieved context, with scores scaled between 0 and 1, where a higher score indicates better consistency. An answer is considered faithful if all the claims made within it can be inferred from the given context. To assess this, a set of claims from the generated answer is first identified, and each claim is then cross-checked against the context to verify if it can be inferred from it.

Context Precision: Context Precision is a metric that assesses whether all ground-truth relevant items within the contexts are ranked appropriately high. Ideally, all relevant chunks should appear among the top-ranked items. This metric is calculated using the question, ground truth, and contexts, with scores ranging from 0 to 1, where higher scores indicate greater precision.

Context Recall: Context Recall measures how well the retrieved context aligns with the ground truth, which is based on the annotated answer. This metric is calculated using the question, ground truth, and retrieved context, with scores ranging from 0 to 1, higher values indicate better recall. To estimate context recall, each claim in the ground truth answer is examined to determine whether it can be supported by the retrieved context. Ideally, all claims in the ground truth answer should be supported by the retrieved context.

3.7 Summary

The research methodology employed in this thesis involves several key steps aimed at developing and evaluating an AI chatbot for course information retrieval. Course details are systematically scraped from the websites of various Technological Universities in Ireland using BeautifulSoup and Langchain. The challenges encountered during scraping, such as outdated links and inaccessible sitemaps, are addressed by utilizing a combination of RecursiveURLLoader and SitemapLoader. The scraped data is then meticulously cleaned and structured. The cleaned course data is divided into smaller, manageable chunks using the RecursiveCharacterTextSplitter from the Langchain library. These chunks are then transformed into numerical representations called embeddings using the OllamaEmbeddings, which leverages the nomic-embed-text model. These embeddings are subsequently stored in a ChromaDB vector database, enabling efficient retrieval and comparison of semantically similar text segments.

The study employs five open-source Large Language Models (LLMs) These models are evaluated and modeled to follow some instructions which reflect on their text generation capabilities, including accuracy, relevance, and coherence. These customized LLMs are then integrated into a Retrieval Augmented Generation (RAG) pipeline, which combines the LLM's language generation abilities with the contextual information retrieved from the ChromaDB. A user-friendly chatting interface is built using the Streamlit package, providing a seamless interaction platform for students to query the chatbot about course information.

The performance of the RAG pipeline is rigorously evaluated using the RAGAS framework. This framework employs metrics such as answer relevancy, answer correctness, faithfulness, context precision, and context recall to assess the quality and effectiveness of the chatbot's responses. The evaluation results inform further refinement and optimization of the system, ensuring its accuracy and reliability in providing course-related information to students.

4 ANALYSIS OF FINDINGS

4.1 Introduction

This section offers a comprehensive analysis of the text generation capabilities of five different open-source LLM models Llama3, Gemma, Mistral, Phi, and Llama2 using the RAGAS framework. As outlined in the previous chapter, the evaluation criteria include Answer Relevancy, Answer Correctness, Faithfulness, Context Precision, and Context Recall.

Based on this analysis, the goal is to identify the most suitable model for a Retrieval Augmented Generation(RAG) system designed to assist students in accessing course information from Technological Universities across Ireland.

4.2 Evaluation of RAG pipeline using Ragas

To assess the performance of the models, A structured series of questions alongside their corresponding ground truths is generated manually. These questions were then posed to the chatbot, and the answers generated by the LLM model were recorded. Additionally, the contexts retrieved by ChromaDB in response to each question were documented. All this information was systematically stored in a CSV file, with columns labeled as follows: Question, Answer (from the LLM model), Context (fetched by ChromaDB), and Ground Truth.

The structured set of questions (eight in total) was designed to encompass a range of universities available in the database, ensuring broad coverage, and followed best practices in prompt engineering to ensure that each question was clear and concise. The ground truths were manually sourced from the respective university websites to ensure accuracy. Below are a few examples:

Question: *"Can you tell me the entry requirements for the Bachelor of Business*

(Honours) in Accounting at MTU?"

Ground Truth: *"The entry requirements for the Bachelor of Business (Honours) in Accounting at MTU are 293 CAO Points in 2023 and Leaving Certificate in six subjects...."*

Question: *"Are there any postgraduate courses related to Marketing at ATU St. Angela's?"*

Ground Truth: *"There aren't any courses related to Marketing at ATU St. Angela's."*

Question: *"Which Process Validation courses are taught at TUS Moylish Campus?"*

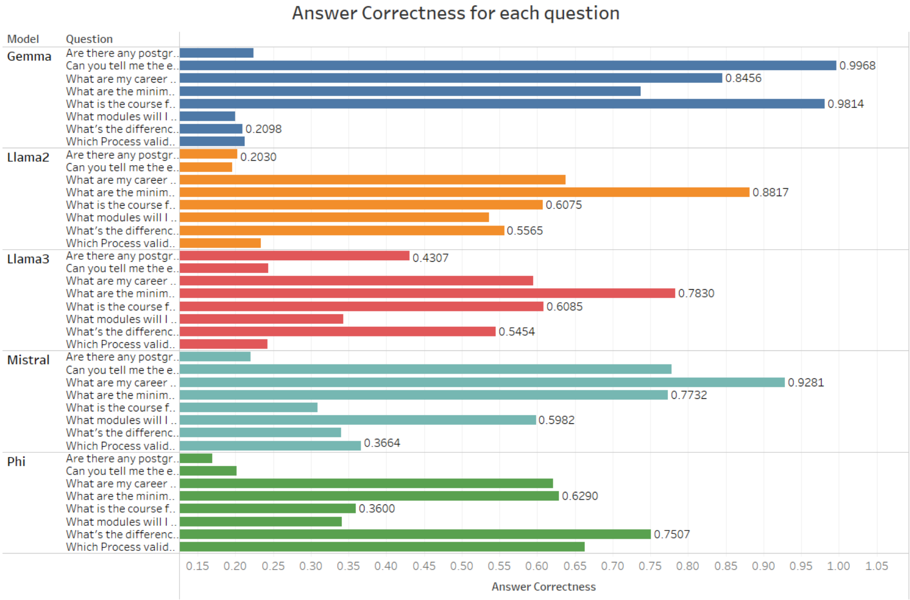
Ground Truth: *"There are two courses related to Process Validation and Regulatory Affairs offered at TUS Moylish Campus: 1. Process Validation and Regulatory Affairs (Medical Technology) – MSc, and 2. Process Validation and Regulatory Affairs (Pharmaceutical) – MSc."*

This approach ensures a thorough evaluation of the models' performance across various queries, representing different Technological Universities in Ireland.

4.2.1 Answer Correctness

This section delves into the Answer Correctness metric for each model in relation to the corresponding questions. The primary objective is to determine which model produces the most accurate answers when compared to the ground truth. This evaluation is based on a direct comparison between the generated answers and the ground truths. The chart below illustrates the scores for each model, reflecting their performance in terms of answer accuracy.

Figure 6 reveals several key insights. Notably, Gemma consistently provided some of the highest-scoring answers in terms of answer correctness. One of the standout qualities of Gemma is its straightforwardness—its responses were precise, with no embellishments or unnecessary details. When Gemma didn't know the answer, it simply acknowledged its inability to assist with the query, rather than attempting to generate a response.



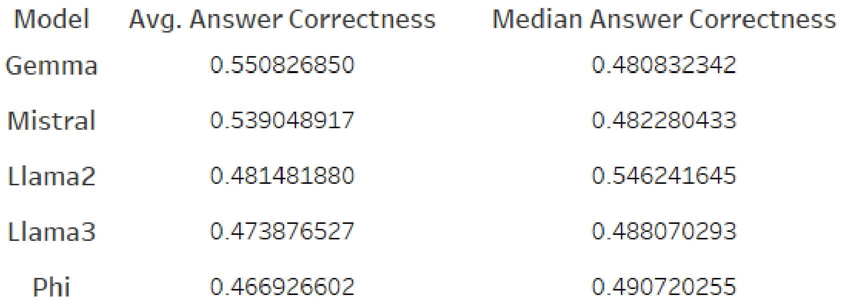
Answer Correctness for each Question broken down by Model. Colour shows details about Model. The marks are labelled by Answer Correctness.

Figure 6: Answer Correctness for each question

In contrast, Llama2, Llama3, and Mistral exhibited similar performance levels, though individually, Llama3 maintained the shortest threshold in terms of response correctness. Phi, on the other hand, performed less effectively compared to the other models, though it still impressed given its relatively small size of just 2.7 billion parameters. However, Phi did produce a highly inaccurate, hallucinated answer for one of the questions, completely out of context.

To identify the most optimal model among these, it would be beneficial to calculate both the average and median scores across all the questions and models. This approach would provide a more comprehensive understanding of each model's overall performance.

Here in Figure 7, the observation is that Gemma has an average answer correctness score of 0.55. However, the model tends to produce extreme results—either providing highly accurate answers or failing to generate a relevant response. This inconsistency suggests that Gemma's performance may not be ideal for all scenarios. By analyzing other metrics, such as context Precision, it can be determined whether Gemma is effectively using the



Model	Avg. Answer Correctness	Median Answer Correctness
Gemma	0.550826850	0.480832342
Mistral	0.539048917	0.482280433
Llama2	0.481481880	0.546241645
Llama3	0.473876527	0.488070293
Phi	0.466926602	0.490720255

Figure 7: Average and Median of Answer correctness for each model

context compared to other models.

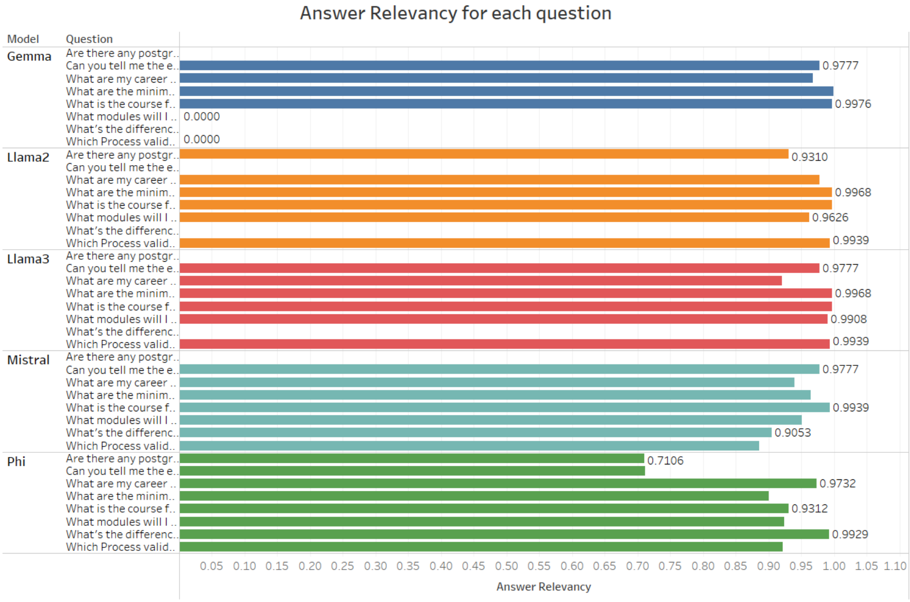
On the other hand, Mistral and Llama2 have maintained very strong median scores, indicating a more consistent performance. This consistency suggests that these models have the potential to deliver reliable results across a broader range of queries, making them promising candidates for further consideration.

4.2.2 Answer Relevancy

This section explores the Answer Relevancy metric for each model in relation to the corresponding questions. This metric evaluates not only how closely the generated answer aligns with the question but also examines the presence of any incomplete or irrelevant information in the response. The assessment is based on the generated answers, the questions posed, and the context provided. The chart below displays the scores for each model, highlighting their performance in terms of answer relevancy.

From Figure 8, it is evident that Gemma scored 0 on four out of the eight questions, indicating that the answers generated by Gemma were irrelevant to the questions more often than not. This occurs because Gemma tends to respond only when a direct answer is found in the retrieved context, rather than attempting to correlate information or generate a response using the context as effectively as other LLM models.

The question *”Are there any postgraduate courses related to Marketing at ATU St.*

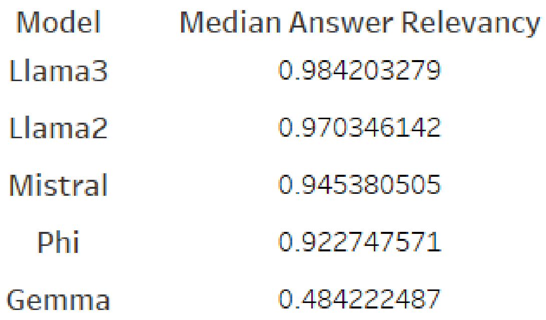


Answer Relevancy for each Question broken down by Model. Colour shows details about Model. The marks are labelled by Answer Relevancy.

Figure 8: Answer Relevancy for each question

Angela's?" should ideally not retrieve any relevant context since the ground truth indicates that no such course exists. In this case, Mistral provided a relevant answer for all questions except the one mentioned, demonstrating its ability to handle such nuances. Llama3 followed closely behind in terms of relevancy. On the other hand, Phi generated responses that included information that should not have been there, which can be interpreted as hallucination or a failure to adhere to the model instructions provided. This further highlights the inconsistencies in Phi's performance.

In this context, the average of the answer relevancy scores might not provide a meaningful assessment, particularly since both Gemma and Phi have demonstrated issues with generating relevant answers. Given their tendency to produce irrelevant responses, relying on the average could skew the results. Instead, calculating the median will likely offer a more accurate representation of each model's performance, as it provides the best mid-range value, filtering out the extremes and giving a clearer picture of consistent relevancy across the models.



Model	Median Answer Relevancy
Llama3	0.984203279
Llama2	0.970346142
Mistral	0.945380505
Phi	0.922747571
Gemma	0.484222487

Figure 9: Median of Answer Relevancy for each model

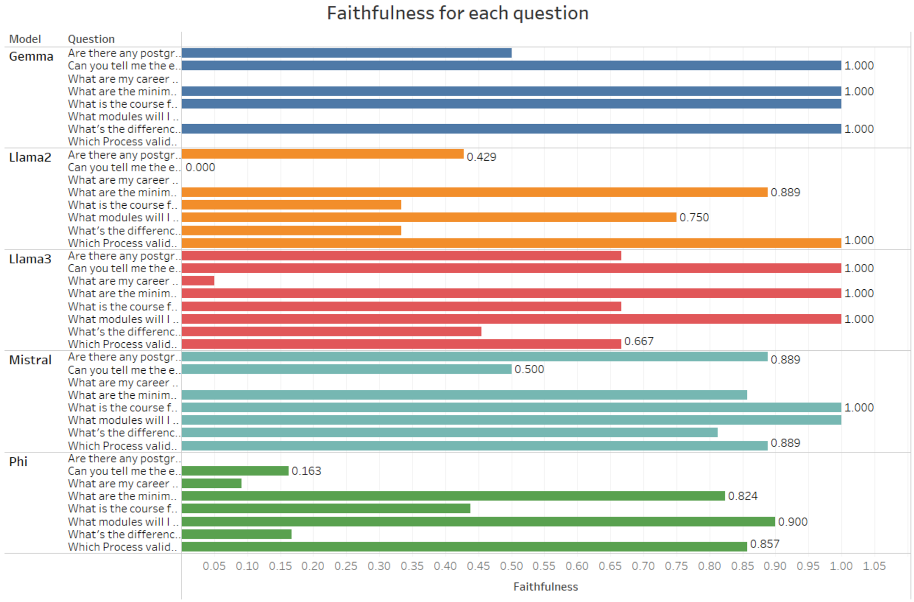
From Figure 9, it can be seen both the Llama models and Mistral performed very well in generating relevant answers. Among them, Llama3, the latest model, provided highly relevant responses, outperforming both Llama2 and Mistral in terms of answer relevancy.

4.2.3 Faithfulness

Faithfulness scores based on the factual consistency between the generated answer and the context retrieved. This shows how the answer is dependent on the context.

Figure 10 reinforces the observation that Gemma tends to be either highly faithful to the context or not at all. Llama3 stands out by consistently demonstrating a high level of faithfulness to the context in its responses. Mistral also performed well, though not quite as faithfully as Llama3, it still proved to be strong compared to the other models.

Up to this point, the metrics have focused on evaluating the models' text generation abilities. However, the next two metrics, Context Precision, and Context Recall shift the focus to assessing the retrieval capabilities of the ChromaDB and embeddings. These metrics evaluate how well the retrieved context aligns with the question and the ground truth, highlighting the effectiveness of the retrieval process in providing relevant information for the models to generate accurate responses.



Faithfulness for each Question broken down by Model. Colour shows details about Model. The marks are labelled by Faithfulness.

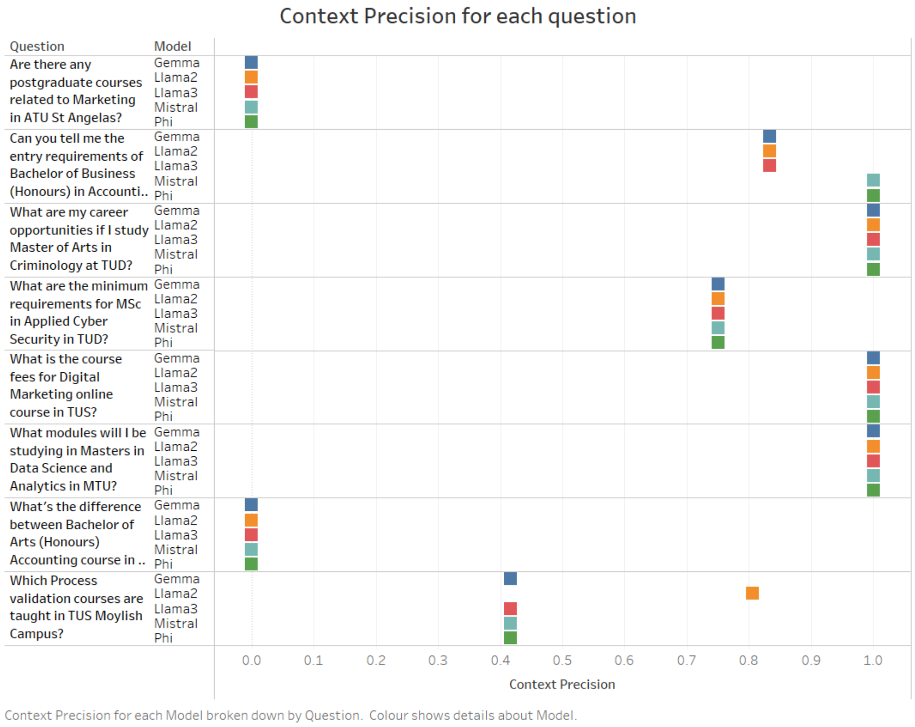
Figure 10: Faithfulness for each question

4.2.4 Context Precision

Context Precision assesses whether the relevant items from the ground truth are ranked highly in the retrieved contexts. This metric is calculated using the question, ground truth, and contexts. Ideally, the Context Precision score should remain consistent for a given combination of question and ground truth, as the same context will be fetched regardless of the model used.

From Figure 11, three exceptions are noticeable: Mistral and Phi scored slightly higher than other models on Question 2, and Llama2 outperformed others on Question 8. The reason for these discrepancies is unclear. According to the RAGAS framework, Context Precision should rely solely on the question, ground truth, and context, so these variations could be due to a bug in the framework or other hidden factors considered during calculation.

Additionally, two of the questions resulted in contexts with very little relevant data. One of these cases was intentional, while the other suggests a potential issue with the process


Context Precision for each question

Context Precision for each Model broken down by Question. Colour shows details about Model.

Figure 11: Context Precision for each question

used, possibly pointing to the need for a more refined chunking method to improve context retrieval.

4.2.5 Context Recall

Context Recall measures the relevancy between the ground truth and the retrieved context. Like Context Precision, this metric is calculated using the question, ground truth, and contexts.

In Figure 12, the only notable exception occurs with the question, *What modules will I be studying in Masters in Data Science and Analytics in MTU?* The ground truth for this question is significantly longer compared to others, as it includes a comprehensive list of all the modules over three years. However, it remains uncertain whether this is the sole reason for the observed discrepancy in the Context Recall score.Apart from that exception,

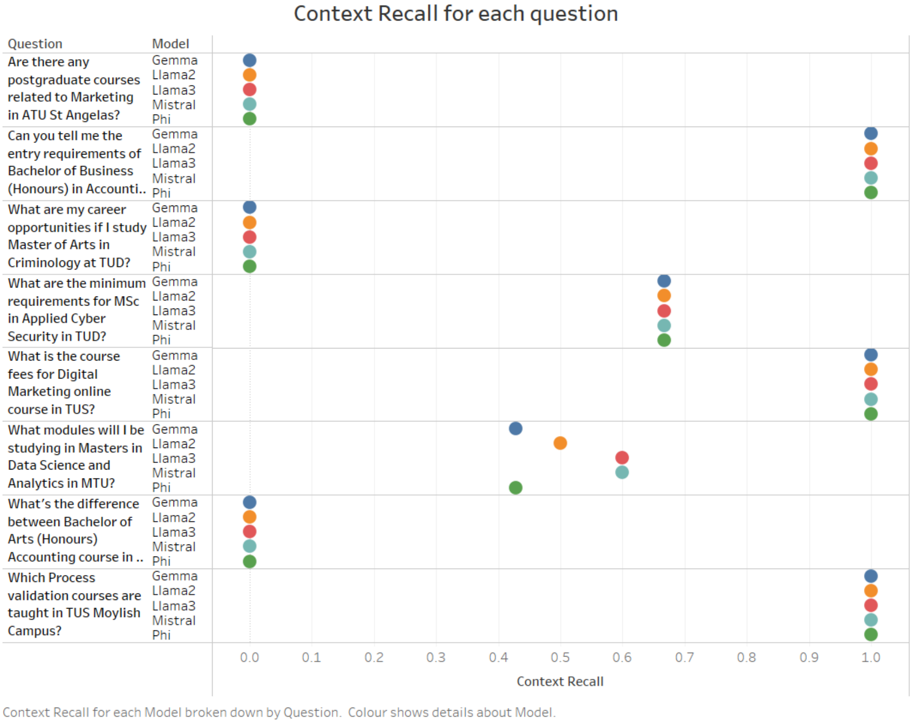


Figure 12: Context Recall for each question

the other questions show consistent results. Three of the ground truths did not align well with the retrieved context, while the other four demonstrated a considerably good match between the ground truth and the retrieved context. This consistency across most questions suggests that the retrieval process generally performs well, with a few areas needing further refinement.

4.3 Summary

This section provides a comprehensive analysis of five open-source LLM models Llama3, Gemma, Mistral, Phi, and Llama2 using the RAGAS framework to determine the best model for a Retrieval Augmented Generation (RAG) system designed to help students access course information from Technological Universities in Ireland. The evaluation focuses on Answer Relevancy, Answer Correctness, Faithfulness, Context Precision, and Context Recall.

A series of structured questions, each with corresponding ground truths, were used to assess the models. The generated answers and retrieved contexts were recorded in a CSV file for analysis. The findings showed that Gemma often produced highly accurate or irrelevant answers, scoring 0 on several questions due to its reliance on direct context matches. Mistral and Llama3 consistently performed well, with Llama3 standing out for its high relevancy and faithfulness.

Answer Correctness revealed that while Gemma had high accuracy, its inconsistency made it less reliable. Mistral and Llama2 demonstrated strong, consistent performance, making them promising candidates. Answer Relevancy indicated that Gemma and Phi struggled with irrelevant responses, while Mistral and Llama3 excelled.

Faithfulness scores highlighted Llama3's consistent alignment with context, while Gemma showed extreme variability. The evaluation then shifted to retrieval capabilities, where Context Precision and Context Recall were assessed. Some anomalies were noted, particularly with Mistral, Phi, and Llama2 in specific questions, potentially due to framework issues. Overall, the retrieval process showed consistency, though improvements in chunking methods could enhance performance.

In summary, Llama3 emerged as a strong performer across metrics, while Mistral and Llama2 also showed potential, particularly in generating consistent and relevant answers.

5 CONCLUSION

The primary objective of this research was to develop an AI chatbot capable of assisting students in accessing course information from various Technological Universities across Ireland. The chatbot aimed to streamline the process of finding relevant course details, eliminating the need for students to manually navigate through multiple university websites. This involved a comparative analysis of different open-source Large Language Models (LLMs) to identify the most suitable one for the Retrieval Augmented Generation (RAG) system. The evaluation criteria included Answer Relevancy, Answer Correctness, Faithfulness, Context Precision, and Context Recall.

5.1 Key Findings and Insights

The analysis of the findings revealed several key insights. Gemma, despite its straightforwardness and accuracy in some cases, exhibited inconsistencies in generating relevant responses. It often fails to correlate information or utilize the context effectively, leading to irrelevant answers. Llama2, Llama3, and Mistral demonstrated similar performance levels, with Llama3 maintaining the shortest threshold in terms of response correctness. Phi, although impressive for its small size, produced less accurate results and even generated a hallucinated answer in one instance.

In terms of answer relevancy, Llama3 outperformed both Llama2 and Mistral, providing highly relevant responses. Mistral also performed well, handling nuances effectively and generating relevant answers for most questions. The context precision and context recall metrics highlighted the effectiveness of the retrieval process, with consistent results across most questions, suggesting that the retrieval process generally performs well, with a few areas like splitting data, where adopting different chunk sizes depending on the different university websites the data being fetched from could be ideal to get more relevancy in one context document.

The research objective was to develop an AI chatbot for accessing course information and to evaluate different open-source LLMs for this RAG system development. The chatbot successfully streamlined the process of finding relevant course details, eliminating the need for manual navigation through multiple university websites. The comparative analysis of LLMs identified Llama3 as the most suitable model for the RAG system due to its high performance in terms of answer relevancy, answer correctness and faithfulness to the context.

5.2 Discussion of Findings

The findings of this research align closely with existing literature on the use of chatbots in educational institutions and the comparative analysis of open-source LLMs. For instance, the study by Galstyan et al. (2024) utilized a RAG framework for their SmartAdvisor chatbot, which successfully demonstrated the effectiveness of this approach in delivering accurate and informative academic counseling. Similarly, Wei et al. (2022) conducted a significant project aimed at understanding students' needs and recommending courses for them. The integration of AI models in such projects has significantly simplified these processes, enhancing their efficiency and accessibility.

One of the most relevant works to this study is by Neupane et al. (2024), who developed the BARKPLUG v.2 system. This system incorporates a RAG framework that uses OpenAI API calls to generate course-related information. In contrast, the current research aims to achieve similar outcomes using open-source alternatives, offering a more affordable solution. While the accuracy levels of the models used in this research may not fully match those of BARKPLUG v.2, they are commendably close, and importantly, they incur no cost.

Similarly, Odede and Frommholz (2024) developed Jaybot, a university chatbot that leverages GPT-3.5 Turbo, highlighting the trend of integrating advanced AI models into educational tools. In another significant contribution, Chang et al. (2024) emphasized LLaMA as one of the leading open-source models for text generation, a finding corroborated by the evaluations in this research. Zhao et al. (2023) further supports this view in his survey, which tracks the evolution of LLaMA within the open-source AI community. While ChatGPT continues to lead the AI industry, LLaMA has emerged as a front-runner in the realm of open-source AI models, underscoring the model's growing importance and reliability in diverse applications.

The findings of this research contribute to the growing body of knowledge on AI chatbots in education and the evaluation of open-source LLMs. The development of the chatbot and the comparative analysis of LLMs provide valuable insights for future research and development in this field. The study highlights the importance of carefully selecting the appropriate LLM model for specific applications, considering factors such as accuracy, relevancy, faithfulness, and context handling capabilities. The inconsistencies observed in some models emphasize the need for further refinement and improvement in LLM technology.

5.3 Limitations

This research encountered several limitations that present valuable opportunities for future exploration and improvement. One significant challenge was the data scraping process, which was hampered by outdated links and inaccessible sitemaps on some university websites. Overcoming these obstacles and expanding data collection to encompass all campuses of ATU would greatly enhance the chatbot's comprehensiveness and utility. Additionally, future research could explore the use of alternative evaluation frameworks and metrics to gain a more nuanced and detailed understanding of the models' performance.

A particularly intriguing limitation observed in this study is related to the context retrieval process. Currently, when a context is fetched, the system searches the database for relevant keywords and returns the top-ranking contexts. However, there is potential to improve this process by introducing a secondary context retrieval mechanism that captures related data from subsequent chunks. For instance, on a course page where the first 500 words contain the course title and related information, while the next chunk includes details about the modules, the RAG model currently struggles to maintain this continuity. Increasing the context window or incorporating a semantic chunker might address this issue, but whether these solutions can be implemented effectively at the context retrieval level remains a question for further investigation.

Another area for improvement is the response speed of the system. This project was conducted on a personal system, which may have hindered the speed of the LLM model. Moving to a cloud-based server could significantly enhance response times, making the system more efficient. Additionally, using a native LLM instead of a framework like Ollama would be more ideal for deployment in a real-world environment.

5.4 Future Research

Future research could explore the integration of personality into the chatbot, which has been shown to boost student trust and engagement, as suggested by Kuhail et al. (2022). Neupane et al. (2024) stated that multi-lingual support for their chatbot would be beneficial for international students, this is a feature to explore as the universities in Ireland are attracting students from all over the world (The Irish Times 2023). Additionally, examining the impact of model size on fine-tuned LLMs in data-to-text generation tasks, as investigated by Mahapatra and Garain (2024), could further enhance the chatbot's capabilities, enabling it to deliver more personalized and accurate responses.

Expanding the chatbot's knowledge base to assist students in accessing a broader range of relevant information about universities, beyond just course data, represents an excellent avenue for future research. Although this idea was initially considered, the primary focus of the project was to first achieve the milestone of providing comprehensive course-related information. Future enhancements could involve integrating additional data, such as student-related resources and information about the city where the university is located. This would further enrich the chatbot's functionality and enhance the overall student experience by helping them better understand and navigate the local environment.

It is also important to recognize the rapid advancements in LLM models. At the start of this thesis, Llama3 and Gemma were the most current models available. However, during the research, Llama 3.5 and Gemma 2 were released, highlighting the fast-evolving nature of the LLM landscape. This dynamic field of LLM research, as discussed by Chen et al. (2024) and Hemberg, Moskal, and O'Reilly (2024), presents opportunities to continuously incorporate these advancements into the chatbot. Future research should actively integrate the latest models to ensure that the system remains current, competitive, and capable of leveraging the most recent developments in LLM technology.

References

[1] Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. *BLEU: a Method for Automatic Evaluation of Machine Translation*. Oct. 2002. DOI: 10.3115/1073083.1073135.

[2] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[3] Claire Brown, Peter Varley, and John Pal. “University course selection and services marketing”. In: *Marketing Intelligence Planning* 27 (May 2009). DOI: 10.1108/02634500910955227.

[4] Zehra Yerlikaya and Pınar Onay Durdu. “Usability of University Websites: A Systematic Review”. In: (2017), pp. 277–287. DOI: 10.1007/978-3-319-58706-6_22.

[5] Nurul Liyana Hussin, Nurulhayah Muhamad, and Muhammad Khalil Tarmizi Abdul Sukor. “DETERMINANTS OF STUDENTS’CHOICE OF COURSES AND UNIVERSITY SELECTION”. In: *Journal of Business Innovation* 4.2 (2019), p. 71.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. “Language Models are Few-Shot

Learners". In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[7] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. "BERTScore: Evaluating Text Generation with BERT". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.

[8] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. URL: https://doi.org/10.1145/3442188.3445922.

[9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. arXiv: 2005.11401 [cs.CL]. URL: https://arxiv.org/abs/2005.11401.

[10] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*. 2021. arXiv: 2107.13586 [cs.CL]. URL: https://arxiv.org/abs/2107.13586.

[11] Mohammad Amin Kuhail, Justin Thomas, Salwa Alramlawi, Syed Jawad Hussain Shah, and Erik Thornquist. "Interacting with a Chatbot-Based Advising System: Understanding the Effect of Chatbot Personality and User Gender on Behavior". In: *Informatics* 9.4 (2022). ISSN: 2227-9709. DOI:

10.3390/informatics9040081. URL: https://www.mdpi.com/2227-9709/9/4/81.

[12] Soner Polat and Çağlar Çelik. “University Websites: Attractive or Casual?” In: *Change The Magazine of Higher Learning* 3 (June 2022), pp. 16–28. DOI: 10.55993/hegp.1079380.

[13] Tan Wei, Mohd Hijazi, Suraya Alias, Ag Ibrahim, and Mohd Fairuz Iskandar Othman. “Intelligent Course Recommender Chatbot Using Natural Language Processing”. In: *International Journal on Advanced Science, Engineering and Information Technology* 12 (Sept. 2022), p. 1915. DOI: 10.18517/ijaseit.12.5.14798.

[14] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. *Automatic Chain of Thought Prompting in Large Language Models*. 2022. arXiv: 2210.03493 [cs.CL]. URL: https://arxiv.org/abs/2210.03493.

[15] Tira Nur Fitria, Nurmala Elmin Simbolon, and Afdaleni Afdaleni. “Chatbots as online chat conversation in the education sector”. In: *International Journal of Computer and Information System (IJCIS)* 4.3 (2023), pp. 93–104.

[16] Sa Gao and Andrew Gao. *On the Origin of LLMs: An Evolutionary Tree and Graph for 15,821 Large Language Models*. 2023.

[17] ES Shahul, Jithin James, Luis Espinosa Anke, and Steven Schockaert. “RAGAS: Automated Evaluation of Retrieval Augmented Generation”. In: *arXiv.org* abs/2309.15217 (2023). DOI: 10.48550/arxiv.2309.15217.

[18] Timm Teubner, Christoph Flath, Christof Weinhardt, Wil Aalst, and Oliver Hinz. “Welcome to the Era of ChatGPT et al.: The Prospects of Large Language Models”. In: *Business Information Systems Engineering* 65 (Mar. 2023). DOI: 10.1007/s12599-023-00795-x.

[19] Krishna Thakkar. "Exploring the capabilities and limitations of GPT and Chat GPT in natural language processing". In: *Journal of management research and analysis* 10.1 (2023), pp. 18–20. DOI: `10.18231/j.jmra.2023.004`.

[20] The Irish Times. *International student numbers in higher education climb to new high*. Oct. 2023. URL: `https://www.irishtimes.com/ireland/education/2023/10/04/international-student-numbers-in-higher-education-climb-to-new-high/`.

[21] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. *A Survey of Large Language Models*. 2023. arXiv: `2303.18223 [cs.CL]`. URL: `https://arxiv.org/abs/2303.18223`.

[22] Zarif Bin Akhtar. "Unveiling the evolution of generative AI (GAI): a comprehensive and investigative analysis toward LLM models (2021–2024) and beyond". In: *Journal of Electrical Systems and Information Technology* 11.1 (2024). DOI: `10.1186/s43067-024-00145-1`.

[23] Hanieh Alipour, Nick Pendar, and Kohinoor Roy. "ChatGPT Alternative Solutions: Large Language Models Survey". In: *arXiv preprint arXiv:2403.14469* (2024).

[24] Md. Al Amin, Mohammad Shazed Ali, Abdus Salam, Arif Khan, Ashraf Ali, Ahsan Ullah, Md Nur Alam, and Shamsul Kabir Chowdhury. "History of generative Artificial Intelligence (AI) chatbots: past, present, and future development". In: *arXiv.org* abs/2402.05122 (2024).

[25] Jinjie Bai. “Exploring techniques and overcoming hurdles in generative AI”. In: *Applied and Computational Engineering* (2024). DOI: 10.54254/2755-2721/36/20230455.

[26] Shruti Bansal, Punjika Rathi, Rachit Agarwal, and Ramneek Ahluwalia. “Leveraging AI to Enhance the Global Mobility Experience”. In: *Advances in finance, accounting, and economics book series* (2024), pp. 108–133. DOI: 10.4018/979-8-3693-1503-3.ch006.

[27] Cara Burgan, Josiah Kowalski, and Weidong Liao. “Developing a Retrieval Augmented Generation (RAG) Chatbot App Using Adaptive Large Language Models (LLM) and LangChain Framework”. In: *Proceedings of the West Virginia Academy of Science* (2024). DOI: 10.55632/pwvas.v96i1.1068.

[28] Bowen Cao, Chengbin Deng, Zhisong Zhang, Yuexian Zou, and Wai Lam. *On the Worst Prompt Performance of Large Language Models*. 2024. DOI: 10.48550/arxiv.2406.10248.

[29] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. “A Survey on Evaluation of Large Language Models”. In: *ACM Trans. Intell. Syst. Technol.* 15.3 (Mar. 2024). ISSN: 2157-6904. DOI: 10.1145/3641289. URL: https://doi.org/10.1145/3641289.

[30] Bo Chen, Xinyi Dai, Huifeng Guo, Leyi Wei, Weiwen Liu, Yong Liu, Jiarui Qin, Ruiming Tang, Yichao Wang, Chuhan Wu, Yaxiong Wu, and Hao Zhang. *All Roads Lead to Rome: Unveiling the Trajectory of Recommender Systems Across the LLM Era*. 2024. DOI: 10.48550/arxiv.2407.10081.

[31] Philip Feldman, James R. Foulds, and Shimei Pan. “RAGged Edges: The Double-Edged Sword of Retrieval-Augmented Chatbots”. In: *arXiv.org* abs/2403.01193 (2024). DOI: 10.48550/arxiv.2403.01193.

[32] Lilit Galstyan, Hovhannes Martirosyan, Elen Vardanyan, and Khajak Vahanyan. “SmartAdvisor University Chatbot Spring 2024”. In: (2024).

[33] Erik Hemberg, Stephen Moskal, and Una-May O’Reilly. “Evolving Code with A Large Language Model”. In: *arXiv.org* abs/2401.07102 (2024). DOI: 10.48550/arxiv.2401.07102.

[34] Pingli Jiang, Rong Fan, and Yong Yu. *Retrieval Augmented Generation via Context Compression Techniques for Large Language Models*. 2024. DOI: 10.31219/osf.io/ua6j5.

[35] Apoorva Joshi. *How to Choose the Right Chunking Strategy for Your LLM Application*. 2024. URL: https://www.mongodb.com/developer/products/atlas/choosing-chunking-strategy-rag/.

[36] Sheetesh Kumar. “Chatbots: A Comprehensive Review of Functionality and Development”. In: *Indian Scientific Journal Of Research In Engineering And Management* (2024). DOI: 10.55041/ijsrem30316.

[37] Joy Mahapatra and Utpal Garain. *Impact of Model Size on Fine-tuned LLM Performance in Data-to-Text Generation: A State-of-the-Art Investigation*. 2024. arXiv: 2407.14088 [cs.CL]. URL: https://arxiv.org/abs/2407.14088.

[38] Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba□Nabende. “Prompt Engineering in Large Language Models”. In: *Algorithms for intelligent systems* (2024), pp. 387–402. DOI: 10.1007/978-981-99-7962-2_30.

[39] Maryamah Maryamah, Muhammad Maula Irfani, Edric Boby Tri Raharjo, Netri Alia Rahmi, Mohammad Ghani, and Indra Kharisma Raharjana. *Chatbots in Academia: A Retrieval-Augmented Generation Approach for Improved Efficient Information Access*. 2024. DOI: 10.1109/kst61284.2024.10499652.

[40] Subash Neupane, Elias Hossain, Jason Keith, Himanshu Tripathi, Farbod Ghiasi, Noorbakhsh Amiri Golilarz, Amin Amirlatifi, Sudip Mittal, and Shahram Rahimi. *From Questions to Insightful Answers: Building an Informed Chatbot for University Resources*. 2024. arXiv: 2405.08120 [cs.ET]. URL: https://arxiv.org/abs/2405.08120.

[41] Julius Odede and Ingo Frommholz. *JayBot – Aiding University Students and Admission with an LLM-based Chatbot*. 2024. DOI: 10.1145/3627508.3638293.

[42] Qiyao Peng, Hongtao Liu, Hongyan Xu, Qing Yang, Minglai Shao, and Wenjun Wang. *Review-LLM: Harnessing Large Language Models for Personalized Review Generation*. 2024. arXiv: 2407 . 07487 [cs.CL]. URL: https://arxiv.org/abs/2407.07487.

[43] Phillip Schneider, Manuel Klettner, Elena Simperl, and Florian Matthes. *A Comparative Analysis of Conversational Large Language Models in Knowledge-Based Text Generation*. 2024. arXiv: 2402 . 01495 [cs.CL]. URL: https://arxiv.org/abs/2402.01495.

[44] Aamir Sohail and Lei Zhang. *Using large language models to facilitate academic work in psychological sciences*. 2024. DOI: 10.31234/osf.io/a4thd.

[45] Shuting Wang, J. J. Song, Jianhua Cheng, Yuqi Fu, Peidong Guo, Kun Fang, Yutao Zhu, and Zhicheng Dou. *DomainRAG: A Chinese Benchmark for Evaluating Domain-specific Retrieval-Augmented Generation*. 2024. DOI: 10.48550/arxiv.2406.05654.

[46] Zheng Wang, Shu Xian Teo, Jian Ouyang, Yan Xu, and Wei Shi. *M-RAG: Reinforcing Large Language Model Performance through Retrieval-Augmented Generation with Multiple Partitions*. 2024. DOI: `10.48550/arxiv.2405.16420`.

[47] Song Yang, Ying Dong, and Zhonggen Yu. "ChatGPT in Education". In: *International Journal of Information and Communication Technology Education* (2024). DOI: `10.4018/ijicte.346826`.

[48] Xiao Yang, Sun Kim, Xin Hao, Yaru Sun, Nikita Bhalla, Xiangsen Chen, Saroj Choudhary, Rongze Daniel Gui, Z. Jiang, Ziyu Jiang, Long Kong, Brian Moran, Jiaqi Wang, Yifan Xu, An Yan, Chuang Yang, Eting Yuan, Hanwen Zha, Nan Tang, L. Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen-tau Yih, and Xin Luna Dong. *CRAG – Comprehensive RAG Benchmark*. 2024. DOI: `10.48550/arxiv.2406.04744`.

[49] Gokul Yenduri, M Ramalingam, Govardanan Chemmalar Selvi, Y. Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H. Jhaveri, B. Prabadevi, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. "GPT (Generative Pre-trained Transformer) – A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions". In: *IEEE Access* 12 (2024), pp. 54608–54649. DOI: `10.1109/access.2024.3389497`.

[50] Jeff Huber and Anton Troynikov. *Chroma Docs*. URL: `https://docs.trychroma.com/`.