Preprint of full paper accepted for CONVERSATIONS 2024 – the 8th International Workshop on Chatbots and Human-Centred AI, hosted by CERTH, Thessaloniki, Greece, December 4-5, 2024. The final version of the paper will be published in the forthcoming Springer LNCS post-workshop proceedings.

Development and Evaluation of a University Chatbot Using Deep Learning: A RAG-Based Approach

Kabir Olawore[1], Michael McTear[2], and Yaxin Bi[3]

Faculty of Computing, Engineering and the Built Environment, Ulster University,
Belfast Campus, Northern Ireland, United Kingdom
{Olawore-B,mf.mctear,y.bi}@ulster.ac.uk

Abstract. In university systems, traditional methods of information retrieval often prove inefficient, leading to frustration among students and staff. This paper presents the development and evaluation of a university-specific chatbot that employs the Retrieval-Augmented Generation (RAG) approach to improve the accuracy and relevance of its responses. Unlike conventional chatbots that depend on intent classification and pre-designed system responses and conversation flows, the proposed chatbot integrates Large Language Models (LLMs) with local university data, enhancing its ability to handle complex queries with context-aware responses and dynamically generated conversation flows. The system architecture includes components such as LangChain for orchestration, a vector store for embedding external knowledge, and a user interface developed using Streamlit. Evaluation results demonstrate that the RAG-based chatbot substantially outperforms traditional LLMs, including GPT-3.5, GPT-4 mini, and GPT-4, in terms of answer accuracy and reliability. In this paper we also reflect on the lessons learned during the chatbot's development and deployment in a real-world university setting.

Keywords: Chatbot, retrieval-augmented generation, large language models, information retrieval, university information systems, LangChain, GPT-3.5, GPT-4, vector store, streamlit

1 Introduction

In university systems, conventional methods for retrieving information, such as emails, phone calls, and website browsing, can be time-consuming and frustrating due to delayed responses and the need for repeated inquiries. Students and staff often seek information about fees, faculties, departments, campus facilities, or administrative processes. This influx of routine inquiries puts a strain on administrative staff, reducing their overall productivity.

To address this challenge, institutions have deployed various solutions, including chatbots. However, traditional chatbots have significant limitations in contextual natural language understanding and managing complex queries [1, 2]. In these systems, developers design intents to interpret user inputs and train classifiers using examples of potential user inputs. However, this process is resource-intensive, making it difficult for developers to create and maintain a comprehensive set of intents. Moreover, inputs that were not anticipated during the design phase cannot be effectively interpreted. Another problem is that the system's responses and the conversation flow are pre-designed, which limits their flexibility and adaptability. Finally, due to their inability to process and integrate structured as well as unstructured information in real-time, traditional chatbots have often struggled to keep up with dynamically changing data, and this has led them to sometimes deliver inaccurate and inconsistent responses to users. These limitations have led to user dissatisfaction and hinder effective information retrieval.

Following the launch of OpenAI's ChatGPT in November 2022, Conversational AI has seen significant advancements [3]. These models, trained on extensive datasets, can provide dynamic, accurate responses across diverse topics, making them valuable in chatbot development [4]. Despite their strengths, LLMs have limitations; while they respond fluently, their knowledge is fixed to their training data and can lack real-time accuracy.

In this paper we present an adaptive LLM-based chatbot that can mitigate the challenges posed by traditional chatbots as well as addressing the limitations of LLMs, thereby improving the efficiency and accuracy of information retrieval in university settings and helping to reduce the burden on administrative staff. Our approach leverages a modern generative AI framework known as Retrieval-Augmented Generation (RAG) which integrates the ability of LLMs to respond to complex queries with the ability to fetch and incorporate specific, contextually-relevant information in their responses [5].

As one of the first applications of RAG in university systems, this approach addresses both traditional chatbot limitations and the static knowledge constraints of LLMs. We also reflect on the challenges in integrating dynamic knowledge sources, scaling the system for complex queries, and improving user experience in a real-world setting.

To realize this adaptive chatbot, we have developed a technology architecture comprising five key components:

1. Large Language Models (LLMs) for natural language understanding and generation.
2. LangChain for orchestrating the flow of information.
3. A Vector store for storing embeddings and serving as a retriever.
4. A University-specific dataset.
5. Streamlit for the front-end interface.

LangChain [6] is an open-source framework that we use to manage the RAG process through specialized chains that connect the LLM with external data sources. The front-end interface was developed using Streamlit (a Python framework for developing interactive data applications) [7], which provides users with a user-friendly, conversational interface to interact with the chatbot.

In the following sections, we will explore the implementation details of this architecture and discuss how it addresses the unique challenges of university information systems.

2 Related Work

Early chatbots relied primarily on pattern matching techniques to facilitate conversation, with notable examples including ELIZA, PARRY, and ALICE [8]. ELIZA used simple pattern matching to simulate a psychotherapist, responding to user inputs with scripted replies [9], while PARRY, designed to mimic the behavior of a paranoid schizophrenic, showed how specific psychological profiles could be modeled through pattern recognition [10].

ALICE advanced this approach by employing AIML (Artificial Intelligence Markup Language), which allowed developers to define patterns and responses using XML tags, enabling more sophisticated interactions [11], while ChatScript introduced a more flexible scripting language that allowed developers to define complex conversation flows and manage dialogue context effectively [12]. Cleverbot introduced statistical methods by analyzing vast datasets of previous conversations, matching user inputs with historical interactions to generate more contextually relevant responses [13].

The field of Conversational AI has advanced significantly with the introduction of deep learning techniques [14]. This evolution in deep learning has paved the way for the development of LLMs like Google’s BERT [15], OpenAI’s GPT models [16], Google’s Gemini models [17], and Meta’s Llama models [18]. These and similar models have transformed natural language processing in conversational agents. See [19] for a comprehensive review of the evolution of conversational AI prior to the launch of OpenAI’s ChatGPT in November 2022.

2.1 Applications of LLMs in Education

LLMs are being used in a wide range of domains, including healthcare, education, finance, customer support, legal systems, entertainment and media, marketing and sales, and scientific research. In the following, we outline some current work in which LLMs are being applied in education.

A chatbot designed to answer questions about faculty guidelines is presented in [20]. This system utilized OpenAI's GPT-3.5 Turbo model within the LangChain framework and employed Pinecone to generate responses. Evaluation results indicated that the chatbot produced coherent answers that were closely aligned with the context of the PDF documents it referenced.

In [21], a chatbot system specifically for Vietnamese universities is described, employing deep learning techniques for consultancy purposes. User intent recognition combines a Pattern Matching method with a Text Classification model utilizing Bidirectional LSTM and an Attention mechanism. Additionally, Deep Reinforcement Learning was implemented to train an agent for Dialogue Management. In a Proof of Concept study, the system achieved an impressive average F1-score of 89% across three classes in the Text Classification task, while the Dialogue Management task scored 86%. A chatbot developed for a college website is detailed in [22], aimed at addressing user queries related to course content, extracurricular activities, and directions to various campus locations. A basic performance test revealed that the chatbot achieved an efficiency rate of 89%.

In another work, [23] developed a chatbot for simplifying academic papers by embedding arXiv papers in a vector store and using RAG to support semantic searching of the documents. The use of RAG to support LLMs in information retrieval is discussed in more detail in the following section.

2.2 Using LLMs and RAG to Retrieve Information

As mentioned earlier, LLMs have revolutionized the field of Artificial Intelligence (AI), especially in Conversational AI, but differ from search engines in delivering factual answers. While search engines provide explicit, current information, LLMs rely on implicit knowledge limited by their training data and the date of the training. To address this, Retrieval Augmented Generation (RAG) enables LLMs to produce more accurate, up-to-date responses by retrieving relevant information from a vector database [5]. In RAG systems, user queries are embedded and matched with stored vectors, retrieving pertinent information that, when combined with the query, enhances the accuracy and timeliness of LLM-generated answers.

RAG has been used in a number of LLM-based chatbots. Aeyeconsult, a chatbot developed by Singer et al. [24], used GPT-4, LangChain, and a vector store to extend the model's knowledge in ophthalmology. Other work has explored RAG concepts in customer service automation [25] and energy system decision support [26].

Sun Kai et al. at Meta [27] examined the factual knowledge abilities of Large Language Models (LLMs) against Knowledge Graphs, revealing LLMs' limitations in factual accuracy, with even advanced models like GPT-4 achieving around 60% accuracy on popular facts. This highlights the need for new strategies beyond model size increases to improve LLMs' factual knowledge.

In a related work, Lehmann et al. at Amazon Science [28] proposed HumanIQ, a method that integrates structured and unstructured data in LLM prompts to mimic human reasoning, achieving over a 50% error reduction for GPT-4 in tasks requiring external information, thus demonstrating significant improvement over previous approaches.

Agarwal et al. [29] presented a novel toolkit, LitLLM, designed to assist researchers in conducting literature reviews. LitLLM automates scientific literature reviews using RAG and LLMs. The system takes a user's abstract, retrieves and re-ranks relevant papers, and generates a literature review.

2.3 Evaluating LLM-Based Chatbots

Evaluating information retrieval in LLM-based chatbots is crucial, especially for applications demanding high accuracy and adherence to ethical standards. Traditional metrics like precision, recall, and F1 score often fall short for LLM systems, missing contextual accuracy when responses don't exactly match ground-truth answers [30]. Alternative metrics such as Faithfulness, Correctness, Relevance, Fluency, and Consistency have been introduced [31], along with ethical considerations like Fairness, Bias, and Privacy [32].

There is a lack of consensus on effective evaluation methods for LLM-based chatbots. Various approaches have been employed, including human evaluation, automated evaluation, and LLM-based evaluation methods [33]. Human evaluation, while ideal, can be costly and inconsistent due to LLM variability [34]. Automated tools like BLEURT assess response similarity to reference texts [35]. Recently, LLMs themselves have been used as evaluators, with frameworks like G-Eval offering improved accuracy by capturing response semantics [36]. Galileo's LUNA excels in hallucination detection and efficiency, while ARES evaluates RAG systems by generating synthetic data to reduce human annotation needs [37, 38].

Ragas [39] is an evaluation library for LLM-based applications, using a secondary LLM to assess responses on metrics like context precision, faithfulness, context recall, relevance, correctness, and answer similarity. This evaluation method, applied to the current system, is further detailed in later sections. Evaluations of LLM-based chatbots reveal issues with reliability and accuracy in information retrieval, suggesting a need for systems like RAG.

In contrast, Banerjee et al. [40] proposed an "End-to-End" benchmark that focused specifically on accuracy and usefulness. The E2E benchmark employs semantic similarity metrics to compare chatbot responses against expert-generated "golden answers". The authors evaluated the benchmark using cosine similarity of embeddings from the Universal Sentence Encoder (USE) and Sentence Transformers (ST), comparing these results with traditional ROUGE scores. Their experiments on a product support chatbot demonstrated that the E2E benchmark, particularly when using ST embeddings, is more sensitive to improvements from prompt engineering compared to ROUGE scores.

3 Methodology

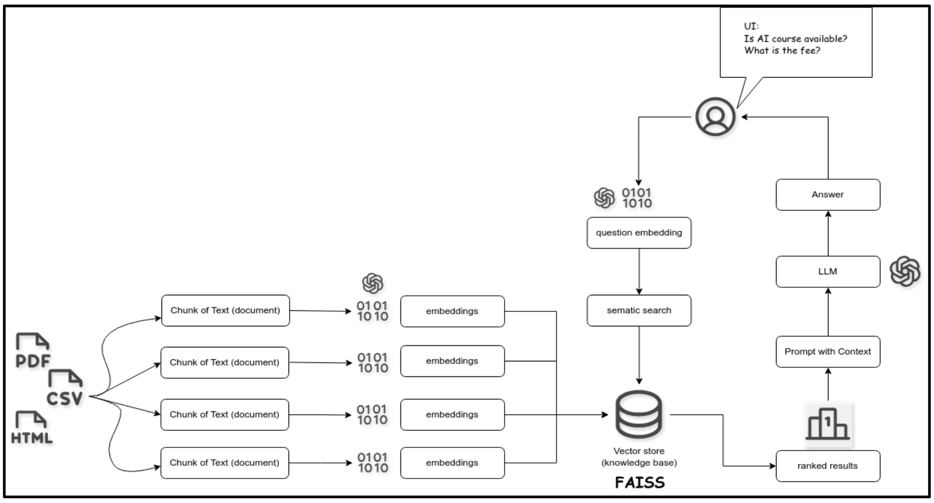


Fig. 1. High Level RAG architecture

The methodology section of this paper outlines the software architecture and tools used in the development of the RAG-based chatbot, as well as the research methods employed to evaluate its performance. This section begins by detailing the proposed architecture in Fig. 1. The specific tools and technologies used in the implementation are described, followed by an explanation of the research methodology used in assessing the chatbot capability and effectiveness. This includes details on the data sources, evaluation metrics, and experimental setup employed to test the chatbot's responses and accuracy.

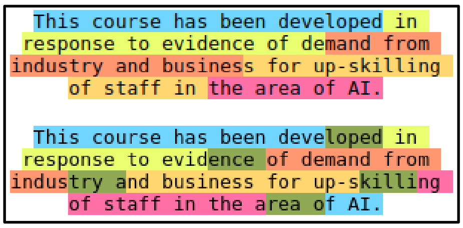
The chatbot's software architecture adopts a client-server model. The user-facing component is a Streamlit-based web application, providing a chat-style interface where users can initiate conversations, clear the chat history, and engage with the system. On the server side, the LangChain framework is leveraged to power the chain that processes and manages the conversational interactions.

3.1 Dataset

The data for this study was sourced from the Ulster University website and converted to a PDF format. The dataset contains information about the Faculty of Computing, Engineering, and the Built Environment (CEBE), including details on fees, courses, modules, and assessment. Prior to use, the data was pre-processed to remove noise, and ensure quality and consistency.

3.2 Chunking

The text content was extracted from the PDF file using the LangChain text splitter, a method to create 1000 chunks of text, and with a 200-character overlap between each chunk. The chunking process involves dividing the textual data into more compact, digestible segments to facilitate efficient processing by language models [41]. These individual chunks are then transformed into vectorized units of information and stored in a vector store, in this case, Facebook AI Similarity Search (FAISS) [42].



This course has been developed in
response to evidence of demand from
industry and business for up-skilling
of staff in the area of AI.

This course has been developed in
response to evidence of demand from
industry and business for up-skilling
of staff in the area of AI.

Fig. 2. The first text shows an example text of 30 chunks apart, with each colors representing a chunk. The second text shows 30 chunks with green overlaps of 5.

Chunking is used to give language models relevant context for answering questions, while overlap preserves semantic connections across boundaries. A 1,000-size chunk optimizes context and computational efficiency, as shorter inputs suit OpenAI embeddings best (see Fig. 2).

3.3 Embeddings

In this phase, each text chunk is converted into a numerical vector representation using OpenAI's "text-embedding-ada-oo2" model. This model was chosen for its high accuracy compared to other alternatives. The resulting embeddings capture the semantic meaning and context of the text, and are processed as vectors.

The vectors are stored in FAISS, which provides an efficient framework for indexing and managing vector-based data. To assess the relationship between entities, the system employs cosine similarity, measuring the angular difference between pairs of vectors. This method helps the algorithm recognize common attributes between text segments that share similar semantic meaning. The formula for cosine similarity is defined as in Fig. 3.

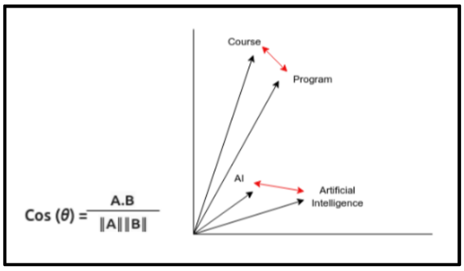


Fig. 3. Cosine similarity formula with an example of word embeddings in a 2D space. It is defined as the dot product of vector A and B divided by the product of their magnitudes.

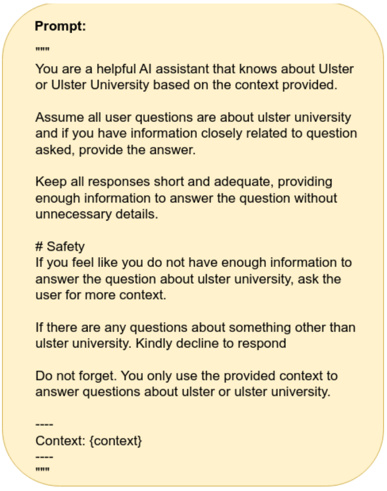
3.4 Retriever

The retriever component in the RAG system extracts relevant documents in response to user queries, whether conversational or keyword-based, by fetching pertinent data from the vector store to supply context for answers [41]. Queries are embedded as vectors and matched to stored document vectors using cosine similarity to locate the most semantically similar text chunks. The FAISS vector store functions as both the database and retriever, allowing for efficient retrieval of relevant information for context. This context is then provided to the language model, enabling it to generate informed, accurate responses.

3.5 Prompt Engineering

This involves crafting system prompts that determine how a language model behaves in response to a user query [41]. In this project, we employed the technique of prompt engineering with the goal of getting responses that are tailored to the shared context of Ulster University from the RAG system.

It was critical to carefully choose the right words, phrases, symbols, and formats to provoke the desired responses from the LLMs. For example, if a user asks a question on topics outside the context of Ulster University, the prompt design enables the model to politely decline to answer. It was a crucial component in optimizing the chatbot's performance and ensuring it stayed within the bounds of its intended conversational domain. Fig. 4 shows the system prompt used in the Ulster University chatbot.



```
Prompt:
"""
You are a helpful AI assistant that knows about Ulster
or Ulster University based on the context provided.

Assume all user questions are about ulster university
and if you have information closely related to question
asked, provide the answer.

Keep all responses short and adequate, providing
enough information to answer the question without
unnecessary details.

# Safety
If you feel like you do not have enough information to
answer the question about ulster university, ask the
user for more context.

If there are any questions about something other than
ulster university. Kindly decline to respond

Do not forget. You only use the provided context to
answer questions about ulster or ulster university.

----
Context: {context}
----
"""
```

Fig. 4. System Prompt used to instruct the model.

3.6 Memory and Generation

In the final stage of the RAG system, the language model generates responses by combining the user query with structured and unstructured information retrieved from the FAISS vector store, enhancing the response accuracy beyond the LLM's parametric knowledge. This process involves constructing prompts that blend the user's question, retrieved context, and any necessary guidance to reduce hallucinations and ensure factual responses. Additionally, the memory module archives conversation history and relevant context from previous interactions, enabling a cohesive flow across multiple exchanges within a session [6].

3.7 Chaining

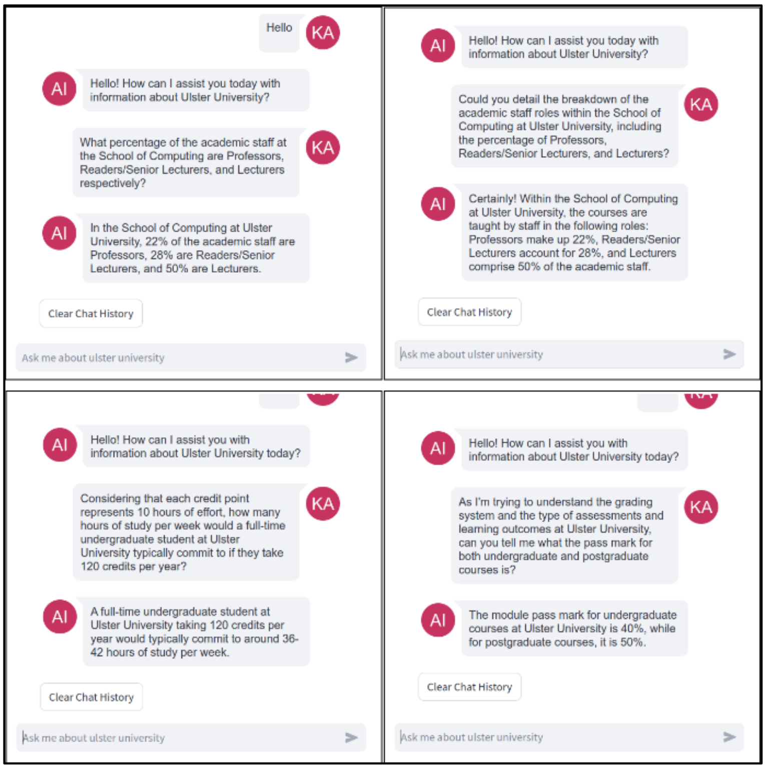
In the LangChain framework, a chain is a sequence of actions designed to build an efficient RAG workflow by linking processes like handling user queries, embedding text, searching the vector store, retrieving documents, and generating responses with an LLM [6]. This study utilized three main chains:

- create_retrieval_chain: Processes the user's query, retrieves relevant documents, and uses the language model to generate a response, essential for the core retrieval stage.
- create_history_aware_retriever: Incorporates chat history to form additional queries, allowing the system to follow up on previous conversations.
- create_stuff_document_chain: Compiles retrieved contexts into a single prompt and feeds it to the language model.

By utilizing these predefined chains together, we create a workflow that can process the user's query, consider the conversation history, retrieve relevant information, compile this information into a coherent prompt, and generate an in-formed response.

3.8 Chatbot UI

The user interface for interacting with the chatbot is provided by Streamlit, a framework for developing conversational-based applications [7]. The Streamlit-based interface, shown in Fig. 5, offers a chat-style platform where users can enter questions, clear previous conversations, and receive responses in natural language.



Hello

Hello! How can I assist you today with information about Ulster University?

What percentage of the academic staff at the School of Computing are Professors, Readers/Senior Lecturers, and Lecturers respectively?

In the School of Computing at Ulster University, 22% of the academic staff are Professors, 28% are Readers/Senior Lecturers, and 50% are Lecturers.

Hello! How can I assist you today with information about Ulster University?

Could you detail the breakdown of the academic staff roles within the School of Computing at Ulster University, including the percentage of Professors, Readers/Senior Lecturers, and Lecturers?

Certainly! Within the School of Computing at Ulster University, the courses are taught by staff in the following roles: Professors make up 22%, Readers/Senior Lecturers account for 28%, and Lecturers comprise 50% of the academic staff.

Hello! How can I assist you with information about Ulster University today?

Considering that each credit point represents 10 hours of effort, how many hours of study per week would a full-time undergraduate student at Ulster University typically commit to if they take 120 credits per year?

A full-time undergraduate student at Ulster University taking 120 credits per year would typically commit to around 36-42 hours of study per week.

Hello! How can I assist you with information about Ulster University today?

As I'm trying to understand the grading system and the type of assessments and learning outcomes at Ulster University, can you tell me what the pass mark for both undergraduate and postgraduate courses is?

The module pass mark for undergraduate courses at Ulster University is 40%, while for postgraduate courses, it is 50%.

Fig. 5. Streamlit Chatbot Interface, with the images depicting simple, complex, distracting, and situational question to test how the chatbot responds in different approaches.

3.9 Evaluation

The ground truth dataset for evaluating the chatbot was generated using Giskard [43], a platform for testing AI model performance. This dataset includes the 60 questions (example in Fig. 9), the reference answers, the reference contexts from which the answers should be derived, the chatbot's conversation histories, and other relevant metadata.

To assess the chatbot’s performance, we employed human evaluation, cosine similarity metric and the RAGAS evaluation framework [39]. RAGAS is designed specifically for evaluating retrieval-augmented generation applications like our chatbot. It measures the system’s performance across various metrics, including context precision, context recall, faithfulness, answer relevance, answer similarity, answer correctness, and potential harmfulness. Fig. 6 shows the RAGAS architecture.

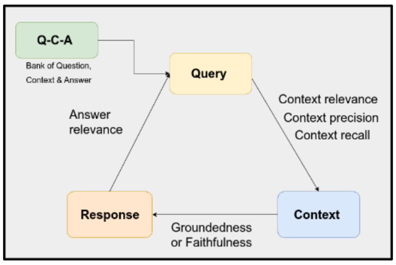


Fig. 6. The RAGAS architecture

The evaluation process involves analyzing the query, expected response, and context from the Giskard-generated dataset, comparing it with the chatbot's output to calculate metrics. RAGAS employs language models and cosine similarity to assess the alignment of the chatbot’s response with the expected answer. Each metric yields a score from 0 to 1, offering a detailed view of the chatbot's performance and highlighting areas for improvement. This thorough evaluation aids in identifying opportunities for further enhancement.

4 Results

The evaluation results demonstrate the effectiveness of the RAG-based chatbot system. Fig. 7 shows the plot of the chatbot’s responses against the ground truth answers, as measured by cosine similarity. The results show that deploying the RAG-based chatbot with different LLMs (GPT 4, GPT 4 mini and GPT 3.5) produced satisfactory results.

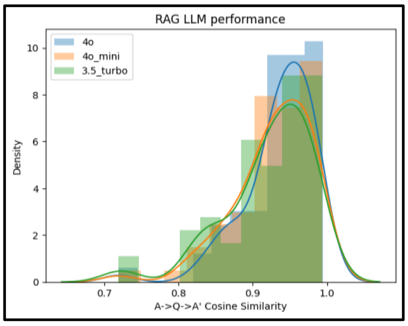


Fig. 7. Cosine similarity plot showing the performance of RAG on GPT-4o, GPT-4o mini and GPT-3.5

The density is the number of answers that share similar cosine similarity. This suggests that the RAG-based chatbot using GPT 4 consistently produces responses that closely match the reference answers.

The RAG-based model with GPT 4 mini also performs well, but exhibits more variability in its results. GPT-3.5, while still effective, demonstrates a wider range of cosine similarity scores, implying that it is somewhat less reliable in generating precise answers compared to the RAG-based models.

The RAGAS evaluation further reinforces the chatbot's strong performance. The radar chart in Fig. 8 showcases the system's scores across six key metrics:

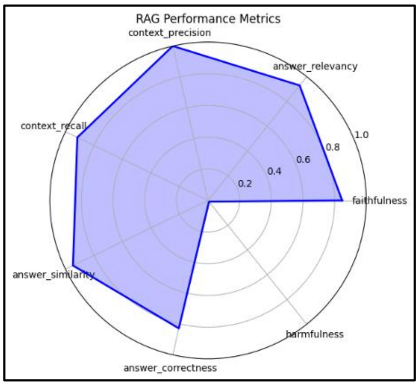
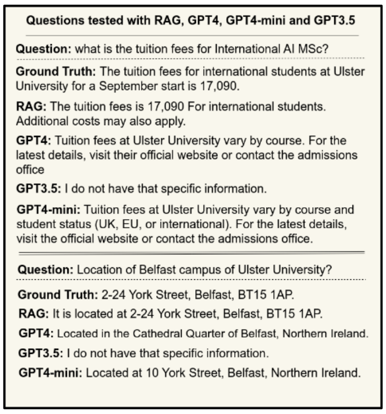


Fig. 8. Radar plot of RAGAS evaluation

- Context precision, faithfulness, and context recall nearly reach perfect scores (0.8-0.9), highlighting excellent information retrieval and contextual understanding.
- Answer relevancy and correctness score highly (0.8-0.9), indicating pertinent and accurate responses. Answer similarity is also good (0.8-0.9), suggesting consistency in the generated answers.
- The plot also suggests that the model has a reasonable performance in minimizing harmful content.



Questions tested with RAG, GPT4, GPT4-mini and GPT3.5

Question: what is the tuition fees for International AI MSc?

Ground Truth: The tuition fees for international students at Ulster University for a September start is 17,090.

RAG: The tuition fees is 17,090 For international students. Additional costs may also apply.

GPT4: Tuition fees at Ulster University vary by course. For the latest details, visit their official website or contact the admissions office

GPT3.5: I do not have that specific information.

GPT4-mini: Tuition fees at Ulster University vary by course and student status (UK, EU, or international). For the latest details, visit the official website or contact the admissions office.

Question: Location of Belfast campus of Ulster University?

Ground Truth: 2-24 York Street, Belfast, BT15 1AP.

RAG: It is located at 2-24 York Street, Belfast, BT15 1AP.

GPT4: Located in the Cathedral Quarter of Belfast, Northern Ireland.

GPT3.5: I do not have that specific information.

GPT4-mini: Located at 10 York Street, Belfast, Northern Ireland.

Fig. 9. Test questions and answers

Overall, the results confirm the RAG-based chatbot's strong ability to retrieve relevant information and produce accurate responses. Unlike standalone LLMs like GPT-3.5 or GPT-4, which are limited to their training data and may hallucinate information, the RAG system offers up-to-date and grounded responses (see Fig. 9).

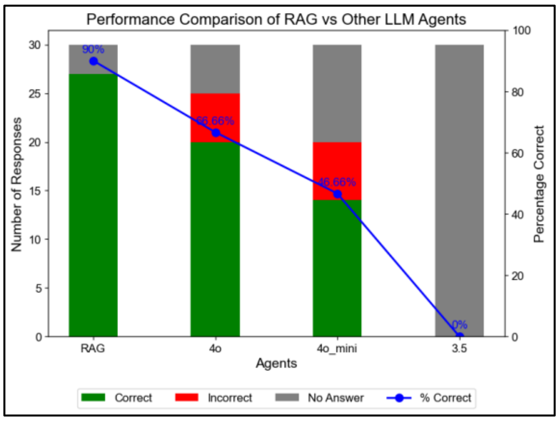


Fig. 10. Stacked bar plot of RAG performance against GPT-4, GPT-4 mini and GPT-3.5

The stacked bar plot in Fig. 10 compares the performance of RAG against GPT-4, GPT-4 mini and GPT-3.5. In a human evaluation of 30 questions, RAG achieved 90% accuracy with 27 correct answers, outperforming GPT-4 (66.66%) with 20 questions, GPT-4 mini (46.66%) with 14 questions, and GPT-3.5, which failed to answer correctly. These findings highlight RAG's superior capacity for handling grounded questions, with some minor areas for refinement in answer consistency and accuracy.

5 Discussion

The study has examined the development of a chatbot solution designed to enhance information retrieval and accessibility within a university system. While large language models (LLMs) like ChatGPT-3.5 and 4 have made question-answering tasks more accessible, we have demonstrated that they come with limitations and risks. A major concern is the quality, reliability, and freshness of the data used to train these models, which can lead to the dissemination of incorrect, seemingly correct but incorrect, or incomplete information, especially on domain-specific topics like information about a university.

Our proposed solution addresses these limitations by integrating LLMs with an external information source, in this case, the university data. The results demonstrate that this approach is more effective, with the RAG-based chatbot outperforming standalone powerful LLMs like chatGPT-4 in terms of accuracy. Additionally, our system allows for the tracing of each response back to its original source within the university dataset, providing transparency and accountability.

5.1 Ethical Considerations

Data privacy and security were key priorities in developing the chatbot, particularly given its focus on university-related information from the Faculty of Computing, Engineering, and the Built Environment (CEBE). Only non-sensitive data was processed, strictly adhering to the faculty's terms of use to protect intellectual property, including licensed datasets and software. For transparency and reproducibility, we documented our methods thoroughly and clearly disclosed the chatbot's capabilities and limitations to build trust. The RAGAS assessment further confirms minimal risk of harmfulness, demonstrating our commitment to accuracy and reliability.

5.2 Limitations and Future Directions

This study encountered several limitations. As RAG-based solutions are still emerging, academic resources and advanced methods are limited. Cost was another constraint, as the language models used, especially GPT-4, incur high expenses.

The dataset was a small, manually curated subset from the Ulster University website, focusing on Faculty of Computing, Engineering, and the Built Environment (CEBE), which restricted testing to a narrow context. Efforts to scrape more data encountered excessive noise, risking unwanted results.

Additionally, with evaluation methods still debated in the research community, selecting an optimal approach was challenging. Although the RAGAS framework was used, its reliability is not fully established, so we supplemented it with human evaluations, which, while effective for a small sample, would be challenging to scale.

Moving forward, future research directions include:

- Developing a reliable solution for evaluating a chatbot's performance in terms of memory and chat history.
- Testing the proposed solution with open-source language models to compare its performance against a broader range of alternatives.
- Exploring more advanced and modular approaches to the RAG-based architecture.
- Evaluating the system's performance with structured datasets hosted in databases, such as Graph or SQL, to assess its capabilities in a wider range of data scenarios.
- Exploring the use of LangChain agents to make the chatbot smarter in its decision making.

6 Conclusions

In this research, we developed a Retrieval-Augmented Generation (RAG)-based chatbot using the LangChain framework, tailored for a university information system. Key lessons include the value of integrating external knowledge sources to improve response accuracy and relevance.

Our evaluation showed that the RAG-based chatbot outperformed traditional LLMs like GPT-3.5 and GPT-4 in accuracy and reliability, thanks to the inclusion of university-specific data. The RAGAS metrics indicated strong performance in context precision and relevance, suggesting our approach effectively eases the administrative load and boosts user satisfaction. While built for a university, this adaptable framework holds potential for broader commercial applications needing reliable information retrieval.

Throughout the development process, we learned the critical importance of:

- Data Quality: The relevance and accuracy of the underlying dataset directly impact the chatbot's performance. Ensuring high-quality, up-to-date information is essential for maintaining trust and reliability.
- User-Centric Design: Engaging with end-users during development helped us tailor the chatbot to meet their specific needs and expectations, enhancing the overall user experience.
- Evaluation Methodology: The challenges associated with evaluating LLM performance prompted us to adopt a multifaceted evaluation framework that combines automated metrics with human assessments, providing a more comprehensive view of the chatbot's capabilities.

Looking ahead, expanding the dataset to cover more university departments could increase the chatbot's versatility across various contexts. Advanced, automated evaluation tools will also be essential for consistently measuring performance in dynamic environments. Integrating structured data sources and exploring methods for managing conversation history could further improve the chatbot’s handling of complex queries and context maintenance.

In conclusion, our research demonstrates the potential of RAG-based chatbots for enhancing information retrieval in university systems, while emphasizing the need for continued refinement. As AI progresses, solutions like ours pave the way for more accurate, context-aware, and reliable information retrieval systems.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Caldarini, G., Jaf, S., McGarry, K.: A Literature Survey of Recent Advances in Chatbots. Information 13(1), 41 (2022)
2. Patel, N. P., Parikh, Patel, D. A., Patel, R.R.: AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT). In: 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 148–150, Coimbatore, India (2019)
3. McTear, M., Ashurkina, M.: Transforming Conversational AI: Exploring the Power of Large Language Models in Interactive Conversational Agents. Apress, Berkeley, CA (2024)
4. Alammar, J., Grootendorst, M.: Hands-On Large Language Models: Language Understanding and Generation. O’Reilly Media, Sebastopol, CA (2024)
5. Lewis, P., Perez, E., Piktus, A., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 34th International Conference on Neural Information

Processing Systems (NIPS '20), pp. 9459–9474. Curran Associates Inc., Red Hook, NY, USA (2020)

6. LangChain Homepage, https://www.langchain.com/, last accessed 2024/10/30
7. Streamlit Homepage, https://streamlit.io/, last accessed 2024/10/30
8. Adamopoulou, E., Moussiades, L.: An Overview of Chatbot Technology. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) Artificial Intelligence Applications and Innovations (AIAI 2020). IFIP Advances in Information and Communication Technology, LNCS, vol 584, pp. 373–383. Springer, Cham. (2020)
9. Weizenbaum, J.: ELIZA–A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), 36–45 (1966)
10. Colby, K.M., Hilf, F.D., Weber, S., Kraemer, H.: Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes. Artificial Intelligence 3, 199–221 (1972)
11. Wallace, R. S.: The Anatomy of A.L.I.C.E. In: Epstein, R., Roberts, G., Beber, G. (eds.). Parsing the Turing test. London: Springer Science+Business Media, pp. 181–210 (2009)
12. Arsovski, S., Cheok, A., Muniru, I., Raffur, M.R.: Analysis Of The Chatbot Open Source Languages AIML and ChatScript: A Review. In: 9th DQM International Conference on Life Cycle Engineering and Management (2017)
13. Cleverbot Homepage, http://www.cleverbot.com, last accessed 2024/10/30
14. Tur, G., Celikyilmaz, A., He, X., Hakkani-Tür, D., Deng, L. Deep Learning in Conversational Language Understanding. In: Deng, L., Liu, Y. (eds.) Deep Learning in Natural Language Processing, pp. 23–48. Springer, Singapore (2018)
15. Ravichandiran, S.: Getting Started with Google BERT: Build and train state-of-the-art natural language processing models using BERT, Packt Publishing, Birmingham, UK (2021)
16. GPT-4 Homepage, https://openai.com/index/gpt-4/, last accessed 2024/10/30
17. Gemini Homepage, https://gemini.google.com/app, last accessed 2024/10/30
18. Llama 3.2 Homepage, https://www.llama.com/, last accessed 2024/10/30
19. McTear, M.: Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots. Synthesis Lectures on Human Language Technologies. Springer, Cham. (2021)
20. Khadija, M.A., Aziz, A., Nurharjadmo, W.: Automating Information Retrieval from Faculty Guidelines: Designing a PDF-Driven Chatbot powered by OpenAI ChatGPT. In: 2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA), pp. 394–399. Bandung, Indonesia (2023)
21. Le-Tien, T., Nguyen-D-P, T., Huynh-Y.V.: Developing a Chatbot system using Deep Learning based for Universities consultancy. In: 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), pp. 1–7. Seoul, Republic of Korea (2022)
22. Shivashankar, B., Sundari, A. M. A., Surendra, H., Atul Sai, S.S., Moharir, M.: Deep Learning based Campus Assistive Chatbot. In: 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), pp. 1–4. Bangalore, India (2021)
23. Dean, M., Bond, R. R., McTear, M.., Mulvenna, M. D.: ChatPapers: An AI Chatbot for Interacting with Academic Research. In: 31st Irish Conference on Artificial Intelligence and Cognitive Science (AICS2023), pp. 1-7. Letterkenny, Ireland (2023)
24. Singer, M. B., Fu, J. J., Chow, J., Teng, C. C.: Development and Evaluation of Aeyeconsult: A Novel Ophthalmology Chatbot Leveraging Verified Textbook Knowledge and GPT-4. Journal of Surgical Education 81(3), 438–443 (2024)

25. Pandya, K., Holia, M.: Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations, https://arxiv.org/abs/2310.05421, last accessed 2024/10/30
26. Gamage, G., Mills, N., Silva, D., et al.: Multi-Agent RAG Chatbot Architecture for Decision Support in Net-Zero Emission Energy Systems. In: International Conference on Industrial Technology (ICIT), pp. 1–6. Bristol, United Kingdom (2024)
27. Sun, K., Xu, Yifan Y.E., Zha, H., Liu, Y., Dong, X.L.: Head-to-Tail: How Knowledgeable are Large Language Models (LLM)? A.K.A. Will LLMs Replace Knowledge Graphs? https://arxiv.org/abs/2308.10168, last accessed 2024/10/30
28. Lehmann, J., Bhandiwad, D., Gattogi, P.;,Vahdati, S.: Beyond Boundaries: A Human-like Approach for Question Answering over Structured and Unstructured Information Sources. Transactions of the Association for Computational Linguistics 12, 786–802 (2024)
29. Agarwal, S., Laradji, I.H., Charlin, L., Pal, C.: LitLLM: A Toolkit for Scientific Literature Review, https://arxiv.org/abs/2402.01788, last accessed 2024/10/30
30. Alinejad, A., Kumar, K., Vahdat, A.: Evaluating the Retrieval Component in LLM-Based Question Answering Systems, https://arxiv.org/abs/2406.06458, last accessed 2024/10/31
31. Yu H, Gan A, Zhang K, Tong S, Liu Q, Liu Z. Evaluation of Retrieval-Augmented Generation: A Survey. arXiv preprint arXiv:2405.07437. 2024 May 13.
32. Gallegos IO, Rossi RA, Barrow J, Tanjim MM, Kim S, Dernoncourt F, Yu T, Zhang R, Ahmed NK. Bias and fairness in large language models: A survey. Computational Linguistics. 2024 Jun 11:1-79.
33. Abeysinghe, B., Circi, R.: The Challenges of Evaluating LLM Applications: An Analysis of Automated, Human, and LLM-Based Approaches, https://arxiv.org/abs/2406.03339, last accessed 2024/10/31
34. Song, Y., Wang, G., Li, S., Lin, B.Y.: The Good, The Bad, and The Greedy: Evaluation of LLMs Should Not Ignore Non-Determinism, https://arxiv.org/html/2407.10457v1, last accessed 2024/10/31
35. Sellam, T., Das, D., Parikh, A.P.: BLEURT: Learning Robust Metrics for Text Generation, http://arxiv.org/abs/2004.04696, last accessed 2024/10/31
36. Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., Zhu, C.: G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment, https://arxiv.org/abs/2303.16634, last accessed 2024/10/31
37. Galileo LUNA, https://galileo.ai/blog/introducing-galileo-luna-a-family-of-evaluation-foundation-models, last accessed 2024/10/31
38. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. https://github.com/stanford-futuredata/ARES, last accessed 2024/10/31
39. Ragas Homepage, https://docs.ragas.io/en/stable/, last accessed 2024/10/31
40. Banerjee, D., Singh, P., Avadhanam, A., Srivastava, S.: Benchmarking LLM powered chatbots: methods and metrics. https://arxiv.org/abs/2308.04624, last accessed 2024/10/30
41. Alto, Valentinam, Building LLM Powered Applications: Create intelligent apps and agents with large language models , Packt Publishing, Birmingham, UK (2024)
42. Faiss: A library for efficient similarity search, https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search, last accessed 2024/10/30
43. Giskard AI Homepage, https://docs.giskard.ai/en/stable/index.html, last accessed 2024/10/30