Unleashing AI in Education: A Pre-Trained LLMs for Accurate and Efficient Question-Answering Systems

Josie C. Calfoforo
College of Computing and Information Technologies
National University
Manila, Philippines
calfoforojc@students.national-u.edu.ph

Dr. Rodolfo C. Raga, Jr.
College of Computing and Information Technologies
National University
Manila, Philippines
rjrcraga@national-u.edu.ph

2024 21st International Conference on Information Technology Based Higher Education and Training (ITHET) | 979-8-3315-1663-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ITHET61869.2024.10837606

***Abstract*—In today's fast-paced academic environment, efficient access to professional development resources is of paramount importance. This study investigates the integration of Retrieval-Augmented Generation (RAG) and the LangChain framework to develop a question answering (QA) system using the Llama-2 large-language model. The QA system was designed to improve information retrieval accuracy and relevance for policy-related questions based on the handbook for the faculty and non-teaching staff development program at Iloilo Science and Technology University, Philippines. By leveraging advancements in artificial intelligence and natural language processing, the system processes natural language questions and retrieves relevant information from diverse data sources such as handbook and Frequently Asked Questions (FAQs) in PDF format. The QLoRA technique was employed for model fine-tuning with 4-bit precision and the optimization of VRAM usage while retaining its high performance. The fine-tuned model indicated that the training process for question and answering reached the 105th global step, along with its training loss and other training metrics, including runtime, samples processed per second, steps processed per second, total floating-point operations, and epoch information. Thus, this study demonstrates that this approach enhances information accessibility and support efficiency within academic institutions, especially during times of crisis like pandemic. Future research could further enhance QA systems to augment question-answering systems by integrating personalized features, cutting-edge natural language processing techniques, affective computing techniques, machine learning strategies, and corresponding evaluation performance metrics.**

Keywords—Question Answering, QA System, LangChain, Large Language Models, Llama, QLoRA Technique

I. INTRODUCTION

The extensive utilization of the Internet and advancements in information storage technology have empowered researchers to effortlessly access and store substantial amounts of data that can be made accessible to the general public. However, the abundance of information has rendered searching for specific data a difficult and time-consuming task. Currently, the question-answering system of the faculty and staff development (FSD) program at Iloilo Science and Technology University, Philippines, for applicants, approved scholars, and other administrative concerns is done manually through policy documents, handbooks, or FAQ sections. This methodology relies significantly on the availability and responsiveness of administrative personnel and may experience delays due to competing tasks or workloads. Researchers have created specialized tools, such as question-answering (QA) systems, to streamline the search process and provide correct answers to queries submitted by users using their natural language.

In various aspects of education, the integration of QA systems has been made possible by the advent of Artificial Intelligence technology. The application of this technology in education is growing. This technology holds the potential for QA systems to offer prompt and customized services to all stakeholders in the industry, including institutional employees and students [1]. The development of question-answering systems represents a fundamental task within the field of natural language processing (NLP) [2]. It has four (4) core components including natural language question (NLQ) processing, document processing, passage processing, and answer processing. Typically, QA systems integrate various techniques from other fields, such as information retrieval, knowledge representation, and natural language processing, to process NLQs and provide the most insightful response based on stored documents [3]. Therefore, it presents a suitable solution for querying unstructured and structured information. By prioritizing data quality and considering these aspects, developers can contribute to the creation of a comprehensive QA system through the implementation of diverse, state-of-the-art LLM-powered applications that not only comprehend and process information accurately, but also deliver a valuable and user-oriented experience. Large language models (LLMs) have emerged as powerful tools for question-answering (QA) systems, particularly in specialized domains, such as healthcare, education, and customer service. [4]. It can leverage vast amounts of data to generate comprehensive and informative responses to high-volume query scenarios that necessitate real-time processing [5] [6].

The field of natural language processing (NLP) has experienced substantial advancements with the emergence of Retrieval-Augmented Generation (RAG). This innovative technique combines the advantages of retrieval-augmented and generation-based models to produce responses that are more precise and contextually relevant. It interprets the intent of natural language questions and retrieves contextually relevant answers by comprehending the question context, thereby offering more targeted responses. Therefore, the RAG framework [4] [5] is highly effective for tasks such as question answering, summarization, and conversational artificial intelligence across various domains. In the rapidly evolving field of Natural Language Processing (NLP), Retrieval-Augmented Generation (RAG) has emerged as a promising approach that demonstrates potential to enhance the accuracy and relevance of generated responses. Similarly, LangChain appears to play a crucial role in enhancing the capabilities of a system, particularly when interacting with diverse data sources and applications [8] [5].

This study focuses on the integration of the Retrieval-Augmented Generation pipeline and LangChain framework in the development of a question answering (QA) system using a pretrained large language model (LLM) known as Llama-2. This system is designed to provide timely and accurate answers to policy-related questions and offers additional queries based on the handbook of the faculty and staff development program (FSD) of the Iloilo Science and Technology University (ISAT U) across campuses, which enables them to simplify access to information on their privileges for professional development in the context of policies and guidelines through automated question answering. Specifically, this study aimed to address the following questions:

1. How does the pre-trained large language model (LLM) perform in terms of question-answering accuracy and efficiency compared to traditional information retrieval methods in educational settings?
2. What are the improvements in user satisfaction and perceived ease of information retrieval with the new QA system of the faculty and staff development (FSD) program versus using standard policy documentation or institutional portals alone?

II. Related Work

Several researchers have used generative pre-training transformer (GPT) chatbots instead of autoregressive large language models (LLMs) for question-answering (QA) systems in the retrieval of information with Retrieval-Augmented Generation and LangChain. To address this research gap, Lu et al. [9] introduced the LLaMA-Reviewer, which utilizes LLaMA capabilities. Despite resource constraints, LLaMA-Reviewer employs parameter-efficient fine-tuning (PEFT) methods, achieving high performance with less than 1% trainable parameters. The efficacy of the LLaMA-Reviewer was assessed using two diverse publicly available datasets. Notably, even with the smallest LLaMA base model comprising 6.7B parameters and a limited number of tuning epochs, LLaMA-Reviewer demonstrates outperforms compared to existing code-review-focused models. Ablation studies were conducted to investigate the impact of various components of the fine-tuning process, including input representation, instruction tuning, and different PEFT methodologies. To facilitate ongoing advancements in this domain, the source code and all PEFT-weight plugins have been made publicly available.

A new study suggested that Llama-2-7b outperforms Mistral-7b in text generation tasks based on preliminary results using ROUGE, BLEU, and CIDEr metrics. However, Mistral-7b is known for its capabilities in reasoning and knowledge-intensive tasks. It is possible that Mistral-7b is still under development and may require larger datasets to reach its full potential for text-generation tasks [10].

Several research studies have investigated different aspects of chatbot using ChatGPT 3.5 Turbo and the LLM model with the LangChain Framework. A study by Khadija et al. [11] focused on the design of a PDF-driven chatbot that utilizes Large Language Models (LLMs) to answer faculty guideline questions. The authors used the LangChain Framework, OpenAI's Chat-GPT (GPT3.5 Turbo), and Pinecone to generate responses, and their findings demonstrating that the chatbot was capable of producing coherent responses that were closely aligned with the context of the PDF document. Another study by Ainapure et al. [12] explored the construction of a PDF chatbot using ChatGPT 3.5 Turbo and the LLM Model. This study utilized the LangChain framework and a streamlit-based homepage to enhance user interaction, and the results showed the potential of the LangChain and LLM models to create engaging chatbots that can be adapted across various domains such as customer service, education, and research. As such, Pandya and Holia (2023) also developed a new system called "Sahaay" that utilized LangChain and a custom LLM tailored for organizations to automate customer service. This study finds that Sahaay has the potential to improve customer retention, value extraction, and brand image, suggesting that LLMs have the potential to revolutionize customer service. Overall, these studies demonstrate the potential of ChatGPT 3.5 Turbo, LLM model, and LangChain framework to develop chatbots that can enhance various domains [13].

Pathak et al. (2024) explored the potential of Llama-2 through fine-tuning using the Low-Rank Adaptation (LoRA) technique, which yielded promising results [14]. These results lay the groundwork for further investigation in this area. As an open-source model, Llama-2 is poised to be a valuable tool for a variety of research applications, even when compared to advanced models, such as GPT-4. Its status as a transformer model and refined hyperparameter tuning process enhance its significance for future research and practical applications. By utilizing a text-summarization dataset, this study demonstrated the enhanced performance of fine-tuned Llama-2, suggesting its potential for broader applications. The dataset was obtained from the hugging-face space where the model was fine-tuned. Therefore, fine-tuned Llama-2 is the force to be reckoned with. However, Dettmers et al. [15] proposed an innovative fine-tuning method called QLoRA, this approach reduces memory utilization, enabling the fine-tuning of a 65 B parameter model on a single 48GB GPU while maintaining full 16-bit fine-tuning performance. This methodology employs backpropagation to transfer gradients through a frozen, 4-bit quantized, pre-trained language model into low-rank adapters (LoRA).

This study was conducted to improve the effectiveness and relevance of professional development information retrieval through the utilization of a Question Answering (QA) system by faculty and non-teaching staff at the Iloilo Science and Technology University, Philippines. To address this problem, a Retrieval-Augmented Generation (RAG) approach was employed in combination with LLM embeddings and the LangChain framework, which incorporates the QLoRA technique. This facilitates the acquisition of relevant data from PDF and FAQ documents in a more efficient and streamlined manner, thereby minimizing reliance on extensive manual searches and complex parsing techniques.

III. Methodology

This section presents the methods utilized in the development of a question answering (QA) system that facilitates convenient access to resources for the professional development of university faculty and non-teaching staff members. This study employed a Retrieval-Augmented Generation (RAG) pipeline using the LangChain framework with the Llama 2 model and the QLoRa technique. Thematic analysis was conducted through the implementation of focus groups and interviews. These methods were employed to assess client satisfaction regarding the QA system's usability

and relevance, as well as to identify areas for enhancement in subsequent development phases. As illustrated in Fig. 1, the block diagram for the proposed work starts with data collection, data preprocessing, building a QA system, fine-tuning and training a model, and analysis of the training loss and run time per second.

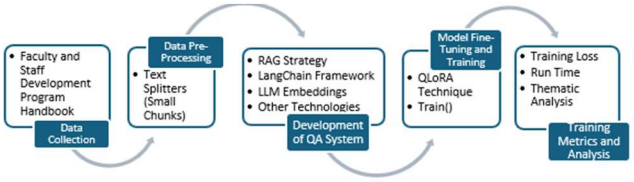

Fig. 1. Block Diagram for Proposed Work

A. Data Collection

A document in PDF format was obtained from Iloilo Science and Technology University, Philippines. The handbook and FAQs in PDF format from the faculty and staff development (FSD) program were used. This handbook contains policies and guidelines pertaining to continuing professional development opportunities for eligible faculty and staff members who demonstrate the potential to facilitate accelerated institutional advancement. It encompasses relevant local and foreign-assisted training programs, scholarship grants, seminars, conferences, workshops, immersion experiences, fellowships, and related human resource development (HRD) initiatives available within the university context. The researchers of this study adhered to ethical standards, including the informed consent, data anonymization, and confidentiality. All collected data were stored securely and accessed only by authorized researchers.

B. Data Preprocessing

This section presents a PDF format that was extracted through chunks of text using 'text splitters." In the fine-tuning process, PDF and FAQs (formatted_question) in an Excel file were utilized.

C. Development of Question Answering System

Fig. 2 shows the architecture of the question answering (QA) system by loading a variety of different types of data sources through document loaders, as shown in Fig. 3, to load the PDF format. To effectively process the text, Text Splitters by initializing the RecursiveCharacterTextSplitter in Fig. 4 to split documents into smaller chunks that need to be put into an index to retrieve them easily when answering questions on this document. Then, document data retrieval is through Retriever.

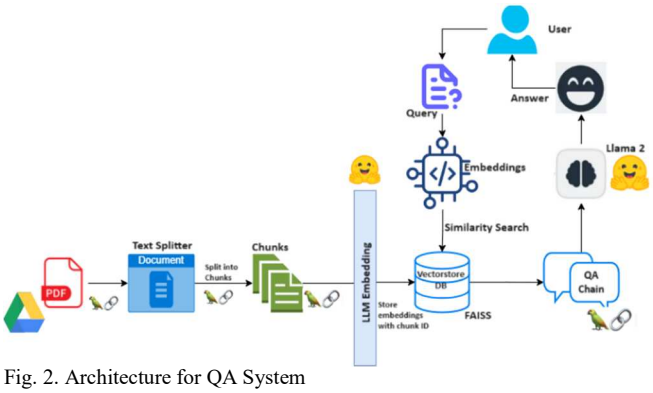

Fig. 2. Architecture for QA System

```
from langchain.document_loaders import PyPDFDirectoryLoader
loader = PyPDFDirectoryLoader(folder_path)
documents = loader.load()
```

Fig. 3. Document Loader

```
from langchain.text_splitter import RecursiveCharacterTextSplitter
# split the documents in small chunks using text splitters
text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=20)
all_splits = text_splitter.split_documents(documents)
print(len(all_splits))
```

Fig. 4. Splitting in Chunks using Text Splitter

Text splitting is followed by creating embeddings for each small chunk of text, and then storing these embeddings in a vector store, FAISS, as shown in Fig. 5. The "all-mpnet-base-v2" Sentence Transformer model was used to create embeddings from the text chunks because it converts pieces of text into vector or numerical representations, then stored in the vector stores (FAISS) for efficient retrieval based on similarity measures. The Embedding model supports HuggingFaceEmbeddings. Loading a local model via HuggingFaceEmbeddings can save on the cost of embedding calls.

```
from langchain.embeddings import HuggingFaceEmbeddings
from langchain.vectorstores import FAISS

# specify embedding model (using huggingface sentence transformer)
model_name = "sentence-transformers/all-mpnet-base-v2"
model_kwargs = {"device": "cuda"}
embeddings = HuggingFaceEmbeddings(model_name=model_name, model_kwargs=model_kwargs)

# storing embeddings in the vector store
vectorstore = FAISS.from_documents(all_splits, embeddings)
```

Fig. 5. Create Embeddings and Vector Store

Fig. 6 shows the initialization of the "ConversationalRetrievalChain" for the QA system functionality. It leverages a vector store to retrieve relevant information from PDF documents based on typically stored unstructured queries. In constructing the chain, the "return_source_documents" parameter allows the chain to return the source documents that were used to answer a question regarding the true value. Fig. 7 shows a query that answered correctly, as illustrated in Fig. 7, the follow-up questions to further enhance the conversational depth and context. By retaining this chat history, the system can provide coherent and contextually relevant responses to subsequent queries.

```
from langchain.chains import ConversationalRetrievalChain

chain = ConversationalRetrievalChain.from_llm(llm, vectorstore.as_retriever(), return_source_documents=True)
```

Fig. 6. Conversational Retrieval Chain

```
chat_history = []

query = "What are the plans of faculty and staff development?"
result = chain({"question": query, "chat_history": chat_history})

print(result['answer'])

 According to the document provided, the faculty and staff development program has plans for various types of professional development
```

Fig. 7. Question Answering with the Dataset

```
chat_history = [(query, result["answer"])]

query = "What is the policy in industry immersion?"
result = chain({"question": query, "chat_history": chat_history})
print(result['answer'])

/usr/local/lib/python3.10/dist-packages/transformers/pipelines/base.py:1157: UserWarning: You seem
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/transformers/pipelines/base.py:1157: UserWarning: You seem
  warnings.warn(
 According to the provided text, the policies governing industry immersion for faculty and staff d

* Industry immersion can only be availed by regular/permanent faculty with at least two years of t
* The experience gained from industry immersion will contribute to the faculty member's job perfor
* Recipients of the program who do not receive compensation from the company will continue to rece
* Faculty members who are paid by the company will not receive salary from the university.
* The objective of industry immersion is to provide comprehensive exposure of the faculty teaching
```

Fig. 8. Follow-up Question Answering with the Dataset

D. Model Fine-tuning and Training

The "meta-llama/Llama-2-7b-chat-hf" is being utilized in this study as a base model which is suitable for question

answering tasks. This involves adjusting model parameters to better understand and respond to faculty and staff queries. The model's tokenizer is then instantiated using the AutoTokenizer class with fast tokenization enabled. A new model, "llama-2-7b-chat-fine-tune," was defined for fine-tuning purposes. The resulting token lengths were stored in a new column called "formatted_question_tok_len" in DataFrame, and the FAQs were formatted. This process measured the length of the tokenized representation for each question in the dataset. 4-bit quantization was created via QLoRA with NF4 type configuration using BitsAndBytes. This approach facilitates efficient fine-tuning of the pre-trained LLM model and optimizes VRAM utilization while maintaining its high performance. Fine-tuning with QLoRA further enhances this relevance by enabling the model to adapt specifically to institutional language and policy details, thereby improving the accuracy of policy-related inquiries.

In this study, the transformers, accelerate, peft, trl, and bitsandbytes from the Hugging-Face ecosystem of LLM libraries were used for natural language processing tasks. A 4-bit precision was employed to load the Llama 2 model with the compute dtype "float16" from the Hugging Face for faster training. To fine-tune the model, a custom trainer called "SFTTrainer" was used to configure the training process. Supervised fine-tuning (SFT) is a critical component in the process of human feedback (RLHF). The Transformer Reinforcement Learning (TRL) library from Hugging Face offers an accessible application programming interface (API) for creating and training SFT models efficiently, utilizing minimal code on the dataset. The.train() was used to fine-tune the Llama 2 model using a formatted dataset.

E. Evaluation and Analysis

The question answering (QA) system using LangChain and a pre-trained Llama 2 model with the QLoRA technique underwent an evaluation process to determine how accurately it responds to faculty and staff queries. hyperparameters that could be used to optimize the training process. The TrainingArguments class of the Hugging Face Transformers library was used to control various aspects of the model training. Adjusting these parameters can exert a substantial influence on the training performance and final model. The training metrics were reported to the TensorBoard to be logged to monitor the training progress. The training process was observed with the number of global steps, along with the training loss and various metrics, including the runtime, samples processed per second, steps processed per second, total floating-point operations, and epoch information.

Based on the qualitative feedback obtained through focus group discussions and interviews with participants in this study, the implementation of a question-answering (QA) system powered by a pre-trained large language model (LLM), incorporating RAG and LangChain technologies with QLoRA technique, specifically for the faculty and staff development (FSD) program has demonstrated significant improvements in user satisfaction and ease of information retrieval. The system's usability, response time, and relevance of answers to questions entered based on the printed handbook were notably enhanced. In contrast to traditional methods, which involve telephone calls, chat communications, and visits to the concerned office, this approach has proven to be more efficient. The handbook Thus, users of the QA system are recommending to the researchers the development of a web-based application, as well as a user-friendly interface for the next phase of this study.

IV. Experimental Results and Analysis

The results of this study proved the effectiveness of the question-answering (QA) system using Retrieval-Augmented Generation, LangChain, and Llama 2 models with the QLoRA technique.

In Fig. 9, the question is answered before the fine-tuning of the model, while in Fig. 10, it is answered after the fine-tuning of the model, which indeed provides a comprehensive answer.

```
##Question:
What is the objective of the faculty and staff development program?

## Answer:
The main objective of the faculty and staff development program is to improve the quality of teaching
```

Fig. 9. Question Answering before the fine-tuning of model

```
prompt = "What is the objectives of the faculty and staff development program?"
pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=200)
gen_text = pipe(f"[QUESTION] {prompt} [/ANSWER]")
print(gen_text[0]['generated_text'])

[QUESTION] What is the objectives of the faculty and staff development program? [/ANSWER]  The objective
 hopefully answer the problems of the university by having a critical mass of skilled and knowledgeable
develop the skills and knowledge of the faculty and staff through training, scholarship and special acti
enhance the performance of the faculty and staff through the provision of modern equipment, tools and re
```

Fig. 10. Question Answering after the fine-tuning of model

The results indicated that the training process reached the 105th global step, the training loss was 0.231766, and various training metrics, including runtime in 88.9844, processed per second with 2.36 samples, steps processed per second of 1.18, and trained in the 5th epoch.

```
TrainOutput(global_step=105, training_loss=0.23176594348180862,
'train_steps_per_second': 1.18, 'total_flos': 762242277949440.0,
metrics={'train_runtime': 88.9844, 'train_samples_per_second': 2.36,
'train_steps_per_second': 1.18, 'total_flos': 762242277949440.0,
'train_loss': 0.23176594348180862, 'epoch': 5.0})
```

Fig. 11. Result of Training Output and Metrics

The model's performance was 4.8592 for smoothed values, in which the training process went on through five epochs, as shown in Fig. 12. a. The smoothing technique was used to reduce the variance in the loss function. Mini-batches of data were used to update the model during the training. The model was then trained on each minibatch in 105 steps. The time taken to complete the training process was 1.17 minutes. While in Fig. 12.b illustrates the maximum clipping threshold for gradients in 0.3 with the corresponding result smoothed to 1.4453 in 100 steps for 1.061 min. Gradients exceeding this value were clipped to prevent explosions during training.

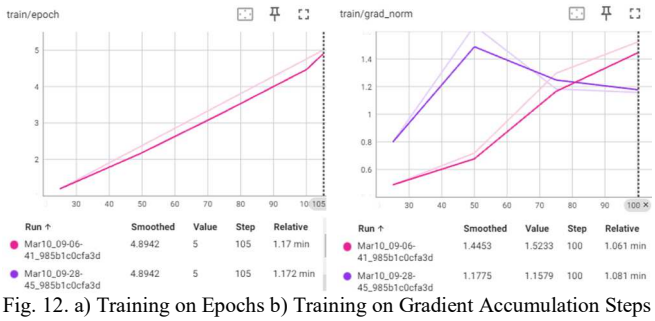

Fig. 12. a) Training on Epochs b) Training on Gradient Accumulation Steps

The learning rate was set to 0.0002, which initially controlled the magnitude of the updates to the model weights with the step size during the optimization process, as shown

in Fig. 13. a. The graph shows in Fig. 13.b the training loss function for model over 100 steps. The average loss (smoothed value) was 0.82, and the specific loss at step 100 was 0.714. It took 1.061 min to complete the 100 training steps.

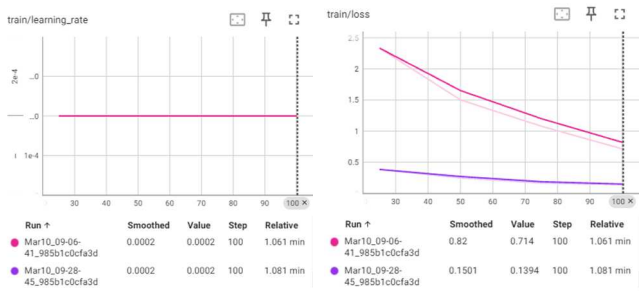

Fig. 13. a) Training on Learning Rate b) Training Loss

The system offers advanced retrieval and response capabilities and faces challenges in addressing complex or multipart questions because of its dependence on structured data, computational demands, and the intricacies of multistep parsing. Although less sophisticated, simpler query-response models often provide more predictable results for these types of queries, especially when users are willing to navigate manually through documents to piece together information. For an optimal solution, the LangChain-based system can incorporate improvements in context retention, ambiguity interpretation, and data structuring to enhance its handling of complex academic inquiries. Indeed, RAG and LangChain-based QA systems represent a substantial improvement in retrieving policy-related information by providing more accurate, consistent, and accessible responses compared to traditional manual methods. This system not only enhances the experience of faculty and staff, but also optimizes the school's administrative resources, leading to a more efficient, self-sustaining support solution.

V. Conclusion

The field of natural language processing has made remarkable progress due to the emergence of advanced technologies. This study successfully developed a Question Answering (QA) system using the RAG strategy, LangChain framework, and a large language models (LLMs) to address the information retrieval requirementsof faculty and non-teaching staff. Instead of manually searching for information, the system was designed to efficiently extract and understand relevant information from handbooks and FAQs presented in the PDF format. The QLoRA technique was employed for model fine-tuning, resulting in a high performance and optimized memory usage. The trained model demonstrated its effectiveness through various metrics, indicating a significant improvement in information accessibility and the streamlining of support processes within academic institutions. The fine-tuned model reached the 105th global training step, and various metrics, such as training loss, runtime, samples processed per second, steps processed per second, total floating-point operations, and epoch information, were analyzed to assess the performance of the system. The results indicate a notable enhancement in information accessibility and process efficiency within academic institutions. The increased adoption of digital resources and online platforms in recent years has led to seamless accessibility of relevant information and streamlining of administrative tasks. Additionally, this shift towards digital resources and online platforms has transformed the way students and educators communicate and collaborate, resulting in a more flexible and accessible learning environment.

This study contributes to advancing the capabilities of information retrieval systems, offering an intelligent solution for faculty and staff to efficiently navigate complex documentation. The outcomes underscore the positive impact of cutting-edge technologies on fostering continuous learning and accessibility in educational institutions within the dynamic landscape of academia. The integration of Retrieval-Augmented and LangChain methodologies with large language models in educational settings extends the influence of these technologies, facilitating the progression to subsequent developmental stages, as indicated by the outcomes of this research.

For future research, this study could incorporate and implement advanced natural language processing and machine learning methodologies, the QA system could be tailored to address domain-specific challenges not only in educational settings but also in different fields, such as healthcare, legal services, and corporate knowledge management. Additionally, the integration of affective computing and sentiment analysis could enable the system to better understand and respond to users' emotional states, further enhancing the overall user experience and the effectiveness of the support solution.

Acknowledgment

This research paper partially fulfills the requirements for the Conversational AI course offered at the College of Computing and Information Technologies, National University, Manila, Philippines, under the supervision of Dr. Rodolfo C. Raga, Jr. The authors acknowledge the support of the faculty and staff development (FSD) program of Iloilo Science and Technology University for their approval and provision of access to the dataset essential for this study.

References

[1] C. W. Okonkwo and A. Ade-Ibijola, "Chatbots applications in education: A systematic review," *Computers and Education: Artificial Intelligence*, vol. 2, p. 100033, Jan. 2021, doi: 10.1016/J.CAEAI.2021.100033.

[2] Y. Bian and K. Peng, "Question Answering System Analysis Based on Machine Learning," 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology, CEI 2021, pp. 279–283, Sep. 2021, doi: 10.1109/CEI52496.2021.9574542.

[3] A. Arbaaeen and A. Shah, "Natural Language Processing based Question Answering Techniques: A Survey," *7th IEEE International Conference on Engineering Technologies and Applied Sciences, ICETAS 2020*, Dec. 2020, doi: 10.1109/ICETAS51660.2020.9484290.

[4] M. A. Arefeen, B. Debnath, and S. Chakradhar, "LeanContext: Cost-Efficient Domain-Specific Question Answering Using LLMs," *Natural Language Processing Journal*, vol. 7, p. 100065, Sep. 2023, doi: 10.1016/j.nlp.2024.100065.

[5] O. Topsakal and T. C. Akinci, "Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast," *International Conference on Applied Engineering and Natural Sciences*, vol. 1, no. 1, pp. 1050–1056, Jul. 2023, doi: 10.59287/ICAENS.1127.

[6] M. L. Bernardi, M. Cimitile, and R. Pecori, "Automatic Job Safety Report Generation using RAG-based LLMs," pp. 1–8, Sep. 2024, doi: 10.1109/IJCNN60899.2024.10651320.

[7] C.-W. Sung, Y.-K. Lee, and Y.-T. Tsai, "A New Pipeline for Generating Instruction Dataset via RAG and Self Fine-Tuning," pp. 2308–2312, Aug. 2024, doi: 10.1109/COMPSAC61105.2024.00371.

[8] Z. Duan, "Application development exploration and practice based on LangChain+ChatGLM+Rasa," *2023 2nd International Conference on Cloud Computing, Big Data Application and Software Engineering*

(CBASE), pp. 282–285, Nov. 2023, doi: 10.1109/CBASE60015.2023.10439133.

[9] J. Lu, L. Yu, X. Li, L. Yang, and C. Zuo, "LLaMA-Reviewer: Advancing Code Review Automation with Large Language Models through Parameter-Efficient Fine-Tuning," *Proceedings - International Symposium on Software Reliability Engineering, ISSRE*, pp. 647–658, 2023, doi: 10.1109/ISSRE59848.2023.00026.

[10] H. Thakkar and A. Manimaran, "Comprehensive Examination of Instruction-Based Language Models: A Comparative Analysis of Mistral-7B and Llama-2-7B," *1st International Conference on Emerging Research in Computational Science, ICERCS 2023 - Proceedings*, 2023, doi: 10.1109/ICERCS57948.2023.10434081.

[11] M. A. Khadija, A. Aziz, and W. Nurharjadmo, "Automating Information Retrieval from Faculty Guidelines: Designing a PDF-Driven Chatbot powered by OpenAI ChatGPT," *Proceedings - 2023 10th International Conference on Computer, Control, Informatics and its Applications: Exploring the Power of Data: Leveraging Information to Drive Digital Innovation, IC3INA 2023*, pp. 394–399, 2023, doi: 10.1109/IC3INA60834.2023.10285808.

[12] A. Ainapure, S. Dhamane, and S. Dhage, "Embodied Epistemology: A Meta-Cognitive Exploration of Chatbot-Enabled Document Analysis," *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)*, pp. 1–6, Oct. 2023, doi: 10.1109/EASCT59475.2023.10392618.

[13] K. Pandya, B. Vishvakarma Mahavidyalaya, and I. Mehfuza Holia, "Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations," Oct. 2023, Accessed: Mar. 12, 2024. [Online]. Available: https://arxiv.org/abs/2310.05421v1

[14] A. Pathak, O. Shree, M. Agarwal, S. D. Sarkar, and A. Tiwary, "Performance Analysis of LoRA Finetuning Llama-2," pp. 1–4, Feb. 2024, doi: 10.1109/IEMENTECH60402.2023.10423400.

[15] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient Finetuning of Quantized LLMs," May 2023, Accessed: Mar. 11, 2024. [Online]. Available: https://arxiv.org/abs/2305.14314v1