Developing an Egyptian Teacher LLM Using Retrieval-Augmented Generation

Omar Gohary[1], Omar El Abasery[2], Malak Ibrahim[3], Shahd Ahmed[4], and Dr Mohamed Taher[5]
Faculty of Computer Science, Egypt University of Informatics (EUI), Cairo, Egypt
{21-101082, 21-101173, 21-101181, 21-101034}@students.eui.edu.eg, Mohamed.taher@eui.edu.eg

***Abstract*— This paper introduces an advanced framework for training a large language model (LLM) tailored to function as an Egyptian teacher utilizing Retrieval-Augmented Generation (RAG). The implementation leverages state-of-the-art NLP techniques for text preprocessing, embedding generation, and efficient retrieval, incorporating tools such as LangChain, Chroma, and HuggingFace. By integrating multilingual support and specialized retrieval capabilities, the framework is designed to enhance educational applications, making it adept at providing accurate, contextually relevant responses to student queries. Furthermore, the system aims to assist students in their studies by providing guidance aligned with their curriculum. The data utilized in this framework are sourced from the Egyptian Knowledge Bank (EKB) and external resources relevant to the curriculum, ensuring accuracy and comprehensive coverage.**

I. INTRODUCTION

The application of artificial intelligence in education has opened new avenues for personalized and effective learning experiences. This project aims to develop a large language model (LLM) designed to serve as an Egyptian teacher, incorporating Retrieval-Augmented Generation (RAG) techniques to ensure accurate, contextual, and dynamic responses. By leveraging tools like LangChain and Chroma for embedding generation and retrieval, the framework integrates domain-specific knowledge and linguistic nuances critical for the Egyptian educational context. The system is designed not only to enhance teaching capabilities but also to assist students in their studies by offering curriculum-aligned support. Data for this framework are sourced from the Egyptian Knowledge Bank (EKB) and other external resources related to the curriculum, ensuring relevance and thorough coverage. This innovative approach combines the scalability of LLMs with the precision of RAG, addressing the unique challenges in delivering high-quality, contextually appropriate educational content.

II. METHODOLOGY

The methodology for developing the EgyptianTeacherRAG framework combines advanced retrieval techniques with generative language modeling to create a robust educational assistant tailored to the Egyptian curriculum. The steps involved in building this system are outlined below:

A. Data Collection and Preprocessing

Educational content is sourced from the Egyptian Knowledge Bank (EKB) and other resources aligned with the curriculum. The following steps are taken to prepare the data:

1) **Data Collection**: Relevant documents and text resources are gathered from trusted sources.
2) **Text Preprocessing**: The content is preprocessed using LangChain's RecursiveCharacterTextSplitter to divide it into manageable chunks suitable for embedding and retrieval.
3) **Multilingual Handling**: For multilingual support, tools like`arabic_reshaper` and `bidi.algorithm` are used to process Arabic text.

B. Embedding Generation and Storage

To enable efficient retrieval, each preprocessed text chunk is converted into vector embeddings using HuggingFace models. The embeddings are then stored in a Chroma vector database, which allows for fast similarity search during query processing.

C. System Architecture

The EgyptianTeacherRAG framework integrates the following components:

- **Retriever**: Chroma serves as the vector database, providing a high-performance mechanism to retrieve relevant content based on vector similarity.
- **Generator**: Azure OpenAI is used to generate contextually relevant and human-like responses from the retrieved content.
- **RAG Pipeline**: The retriever and generator are combined into a seamless pipeline using LangChain, allowing the system to handle user queries dynamically and accurately.

D. Dynamic Personalization

A distinguishing feature of the EgyptianTeacherRAG framework is its ability to adapt responses based on user-selected personalities. By modifying the prompt dynamically, the model tailors its tone and style to match the preferences of the user. This personalization enhances engagement and accessibility for students.

- **Personality Options**: Students can select one of three personalities:
  - **Funny**: Light-hearted and engaging.
  - **Serious**: Formal and professional.
  - **Friendly**: Approachable and conversational.

- **Prompt Engineering**: The personality setting is integrated into the system's prompt, ensuring the generated response reflects the selected style.

E. Query Processing and Response Generation

The system processes user queries through the following steps:

1) The query is embedded using the same embedding model used during preprocessing.
2) The embedding is matched against the Chroma vector database to retrieve the most relevant content.
3) Retrieved content is input into the Azure OpenAI generative model to produce a comprehensive and contextually appropriate response.

F. Implementation Details

The `EgyptianTeacherRAG` class provides the core implementation of the system, with the following key methods:

- `create_retrieval_qa_with_history`: Configures the RAG pipeline by connecting the retriever and the generative model.
- `retrieve_and_answer`: Handles user queries, performing embedding, retrieval, and response generation.
- `main`: Initializes the system, orchestrating data processing, query handling, and response generation.

G. Tools and Technologies

The development of the framework relies on:

- **LangChain**: For preprocessing and pipeline integration.
- **HuggingFace Models**: For embedding generation.
- **Chroma**: For efficient vector storage and retrieval.
- **Azure OpenAI**: For generating human-like responses.

H. Workflow Overview

- Data is preprocessed and embedded, with embeddings stored in Chroma.
- User queries are processed through embedding generation and similarity search.
- Retrieved content is passed to Azure OpenAI for response generation.
- Responses are presented to the user in a natural, contextually accurate manner.

This methodology ensures the system is scalable, accurate, and responsive, meeting the needs of students and educators in the Egyptian educational context.

III. METHODS

A. Model Architecture and Retrieval-Augmented Generation

The EgyptianTeacherRAG framework employs a Retrieval-Augmented Generation (RAG) approach, which integrates retrieval-based systems with generative language models to improve accuracy and contextuality. The architecture consists of two main components:

- **Retriever**: A Chroma vector database is used to store embeddings generated from educational content. This enables efficient and accurate retrieval of relevant information based on user queries.
- **Generator**: An Azure OpenAI-powered large language model processes retrieved information to generate coherent and contextually relevant responses tailored to the curriculum.

The system works as follows:

1) **Preprocessing and Indexing**: Educational content from the Egyptian Knowledge Bank (EKB) and other curriculum resources is preprocessed and split into smaller chunks using the LangChain RecursiveCharacterTextSplitter.
2) **Embedding Generation**: HuggingFace embeddings are generated for each text chunk and stored in the Chroma vector database.
3) **Query Processing and Retrieval**: User queries are embedded and matched against the database to retrieve the most relevant content.
4) **Response Generation**: Retrieved content is fed into the LLM, which generates contextually appropriate responses.
5) **Text-to-Speech Conversion**: Generated responses are converted into speech using the gTTS library, enhancing accessibility and engagement for students.

This architecture ensures scalability and relevance, making it ideal for educational applications.

B. Tools and Technologies

The framework utilizes the following tools and technologies:

- **Chroma**: A high-performance vector database for efficient storage and retrieval of embeddings.
- **Azure OpenAI**: Provides the generative capabilities of the LLM, ensuring high-quality text output.
- **LangChain**: Facilitates text preprocessing, embedding generation, and pipeline integration.
- **HuggingFace Models**: Used for generating embeddings of the educational content.
- **gTTS (Google Text-to-Speech)**: Enables the system to convert generated text into speech, providing an auditory response to users.

C. Key Functions in the Framework

The EgyptianTeacherRAG class includes the following core methods:

- `create_retrieval_qa_with_history`: Initializes the RAG pipeline, connecting the retriever and generator for seamless query processing and response generation.
- `retrieve_and_answer`: Processes user queries by embedding them, retrieving relevant content, and generating responses using the LLM.
- `main`: Serves as the entry point, orchestrating the system's functionality by configuring the pipeline and processing user inputs.

D. Dynamic Personality Adjustment

A key feature of the model is its ability to adapt its personality and tone based on user preferences. Students can choose from three predefined personalities:

- **Funny**: Provides responses with a humorous and engaging tone.
- **Serious**: Delivers formal and to-the-point explanations.
- **Friendly**: Mimics a conversational and approachable style, like talking to a friend.

The personality is adjusted by dynamically modifying the prompt provided to the model, ensuring that the generated responses align with the chosen personality.

E. Workflow

The workflow begins with preprocessing educational content and indexing it in the Chroma database. Once the RAG pipeline is set up, user queries are processed in real-time, leveraging the retrieval and generative capabilities of the system. Text responses are optionally converted to speech using gTTS, creating an interactive and accessible experience. The modular design allows for efficient handling of diverse queries and dynamic updates to the database.

IV. MODEL SELECTION

A. Rationale for Model Selection

The EgyptianTeacherRAG framework was designed to integrate retrieval and generative capabilities, making it essential to choose models and tools that provide both accuracy and contextual depth. Retrieval-Augmented Generation (RAG) was selected due to its ability to combine the precision of a retriever with the contextual understanding of a generative language model. This ensures the system delivers accurate, curriculum-aligned responses while maintaining scalability and robustness.

B. Tools and Libraries

- **Azure OpenAI**: Chosen for its advanced language generation capabilities, providing contextually relevant and human-like responses that align with the educational curriculum.
- **HuggingFace Models**: Used for embedding generation, ensuring high-quality vector representations of the text data.
- **Chroma**: Selected as the vector database for its efficient storage and fast retrieval of embeddings, enabling real-time query processing.
- **Streamlit**: Implemented as the user interface for its simplicity and interactivity, allowing users to engage with the system through an intuitive web-based platform.
- **gTTS (Google Text-to-Speech)**: Integrated to enable the model to generate spoken responses, enhancing accessibility for students.

C. Evaluation Criteria

The following criteria were considered during model selection:

- **Accuracy**: The ability of the system to provide correct and relevant responses to queries.
- **Efficiency**: Fast processing of user queries, including embedding generation, retrieval, and response generation.
- **Scalability**: The capacity to handle a growing volume of data and user interactions without compromising performance.
- **User Experience**: A seamless and engaging interface enabled by Streamlit and audio responses through gTTS.

D. System Integration

The final model architecture integrates these tools into a cohesive pipeline:

- Educational content is processed into embeddings using HuggingFace models and stored in Chroma.
- User queries are embedded and matched against the database to retrieve relevant information.
- The retrieved content is passed to Azure OpenAI for generating responses.
- Responses are displayed in the Streamlit interface and converted to speech using gTTS.

This combination of tools ensures the system meets educational needs while providing an engaging and accessible user experience.

V. CONCLUSION

This paper presents the development of the EgyptianTeacherRAG framework, a dynamic educational assistant tailored to the Egyptian curriculum. By integrating Retrieval-Augmented Generation (RAG) techniques with cutting-edge tools such as Azure OpenAI, Chroma, and HuggingFace, the framework achieves a robust combination of precision, scalability, and contextual understanding. The inclusion of a personalized interaction feature, allowing students to choose from funny, serious, or friendly personalities, enhances engagement and accessibility, catering to diverse learning preferences.

The framework also incorporates advanced features like text-to-speech functionality through gTTS and an intuitive user interface using Streamlit, ensuring a seamless and interactive experience. By sourcing data from the Egyptian Knowledge Bank (EKB) and other curriculum-aligned resources, the system provides accurate and relevant responses, establishing itself as a valuable educational tool.

Future work may focus on expanding the dataset, incorporating additional languages, and refining the personalization features to further enhance the user experience. This project demonstrates the potential of AI-driven systems in transforming education by bridging the gap between traditional teaching methods and modern technological advancements.

REFERENCES

[1] LangChain. [Online]. Available: https://www.langchain.com/

[2] Chroma: AI-native open-source vector database. [Online]. Available: https://www.trychroma.com/

[3] HuggingFace. [Online]. Available: https://huggingface.co/

[4] Azure OpenAI Service. Microsoft. [Online]. Available: https://azure.microsoft.com/en-us/products/cognitive-services/openai-service/

[5] Google Text-to-Speech (gTTS). [Online]. Available: https://pypi.org/project/gTTS/

[6] K. Lewis, "Retrieval-Augmented Generation: The Future of NLP," in *Proceedings of the Neural Information Processing Systems*, 2020.

[7] Recursive Character Text Splitter. LangChain Documentation. [Online]. Available: https://docs.langchain.com/docs/text-splitter/

[8] deeplearning.ai, "LangChain for LLM Application Development," Short Course. [Online]. Available: https://www.deeplearning.ai/short-courses/langchain-for-llm-application-development/

[9] Egyptian Knowledge Bank. [Online]. Available: https://www.ekb.eg/

[10] Telegram Channel: "Books and Resources." [Online]. Available: https://t.me/book_nqdir