Proof of Concept: Arabic-Speaking LLM System with Document Parsing and Question-Answering

By: Husnain & Abdur Rehman

1. Introduction

The purpose of this proof of concept (PoC) is to demonstrate the feasibility of developing an Arabic-speaking language model system with document upload, parsing, and question-answering capabilities. The PoC aims to showcase the core functionalities and user experience of the proposed system.

2. Implementation Details

2.1 User Interface

- The PoC utilizes Streamlit, a framework for building interactive AI applications, to create a modern and sleek user interface.
- The UI consists of components such as file upload, chat interface, and document statistics area (sidebar) providing an intuitive and visually appealing user experience.

2.2 File Upload and Document Parsing

- The system supports file upload for both Arabic and English documents in formats such as PDF.
- Uploaded files undergo text extraction and document parsing.
- Recursive chunking is applied to break down the extracted text into segments.

2.3 Vector Database and Retrieval-Augmented Generation (RAG)

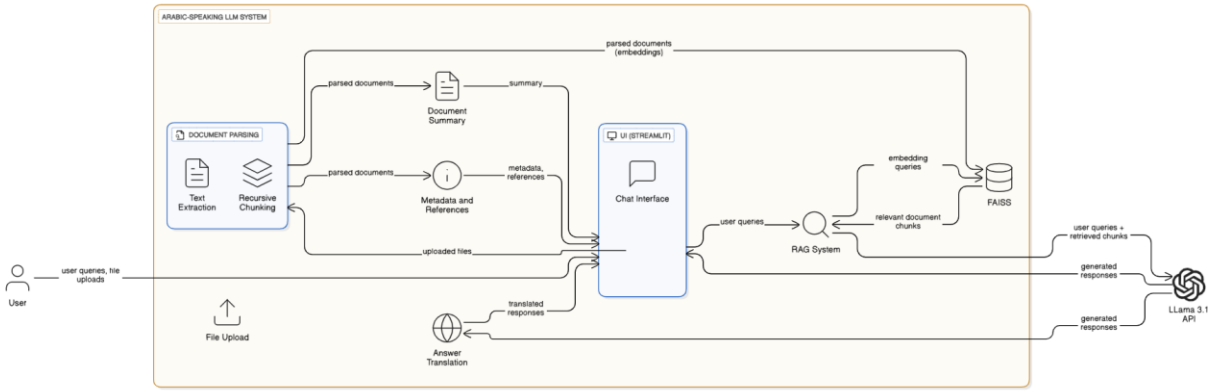
- The PoC utilizes FAISS, a library for efficient similarity search and clustering of dense vectors, to store the semantic representations of document chunks.

- A Retrieval-Augmented Generation (RAG) system is implemented to retrieve relevant document chunks based on user queries and generate appropriate responses.

2.4 Language Model Integration

- For the PoC, the LLama 3.1 API is used as the underlying language model.
- The system integrates with the LLama 3.1 API to process user queries, translate them if necessary (for English queries), and generate responses.

2.5 Architecture Diagram



2.6 Features

The system features:

1. Q/A chat interface.
2. Summary of uploaded pdf.
3. Multiple file uploads (up to 3) 5MB each.
4. File reference in the response.
5. English translation of answer.
6. File statistics such as:
   a. No. of pages.
   b. Words/characters in the uploaded pdf.

3. Suggestions for Final System

3.1 Language Model Selection

- For the final system, we recommend using either the JAIS family of models or the Qwen2.5 model, which have shown excellent performance on the Arabic LLM leaderboard.
- These models offer improved language understanding, generation capabilities, and support for Arabic language processing.

3.2 System Requirements

- The final system should be deployed on a server with sufficient computational resources to handle the large tensor sizes of the recommended language models and process large documents efficiently.
- Minimum recommended system specifications:
    - CPU: Intel Xeon or AMD Ryzen processor with at least 16 cores
    - RAM: 256 GB or higher to accommodate the tensor sizes of the language models
    - Storage: SSD with at least 1 TB capacity
    - GPU: NVIDIA GPU with at least 24 GB VRAM for accelerated inference
- The system should be able to scale vertically to handle the memory-intensive requirements of the language models.

4. Conclusion

The PoC demonstrates the feasibility of developing an Arabic-speaking LLM system with document parsing and question-answering capabilities. By leveraging modern UI frameworks, advanced parsing, and state-of-the-art language models, the system provides an intuitive and efficient way for users to interact with Arabic documents and obtain relevant answers to their queries.

For the final system, we recommend using the JAIS family of models or the Qwen2.5 model to achieve optimal performance and language understanding. The system should be deployed on a server with high-performance computational resources to handle the large tensor sizes of the language models and process large documents efficiently.

Further development and refinement of the system can include additional features such as document summarization, sentiment analysis, and multi-lingual support to enhance its functionality and user experience.