Hybrid Documents RAG based Chatbot

Maharun Afroz, *2211023642,* Sanjida Amin Nadia, *2122319642,* Tasnia Hossain, *2121480642,* and Tahmina Mozumdar, *2111482642*

***Abstract*—State-of-the-art commercial conversational models are typically limited to public data, lacking direct access to personal or specialized documents due to security concerns about exposing sensitive information. To bridge this gap, the Hybrid Documents RAG based Chatbot empowers users to securely tap into their own document collections for personalized information retrieval. By seamlessly combining rapid local indexing, context-sensitive ranking, and reranking, the system enables fast, precise searches without relying on cloud-based services. This privacy-centric design ensures that sensitive data remains safely on-device, while also offering online mode for accessing more computational power. Ultimately, the chatbot offers a groundbreaking paradigm in personalized information discovery, striking an ideal balance between performance, flexibility, and security.**

***Index Terms*—Retrieval Augmented Generation (RAG), Chunking, Local Indexing, Document Retrieval, Vector Database, Context-Sensitive Ranking, Data Privacy, NLP Testing and evaluation**

I. Introduction

In the contemporary digital landscape, the challenge of accessing pertinent information from extensive document collections persists. Traditional artificial intelligence models, which depend on pre-existing knowledge, often encounter difficulties in processing newly introduced content in real time. This limitation poses a significant obstacle for users seeking precise, document-specific insights rather than generic responses.A chatbot based on Retrieval-Augmented Generation (RAG) technology effectively addresses this issue by facilitating direct interaction with documents that extend beyond the capabilities of standard AI models. By dynamically retrieving and integrating information from user-provided documents, it ensures responses that are more relevant and contextually aware. Moreover, the emphasis on data privacy through an offline mode enhances security, rendering it a dependable solution for handling sensitive information. The capability to manage multiple documents concurrently further increases efficiency, allowing users to extract knowledge seamlessly without the need to switch between various sources.This initiative aspires to develop an intelligent chatbot that revolutionizes user engagement with documents, providing a more interactive, secure, and accessible experience.

II. Problem Statement

A RAG-based chatbot allows users to interact with documents that state-of-the-art models do not have prior knowledge of. It enhances security by offering an offline mode, ensuring data privacy. Additionally, it enables seamless conversations across multiple documents simultaneously, improving efficiency and accessibility.

III. Literature Review

Chatbots have evolved from simple rule-based systems to intelligent AI-driven assistants capable of retrieving and generating human-like responses. Recent advancements in **Natural Language Processing and Machine Learning have enabled chatbots to process large amounts of text from different sources, including PDFs and online databases. Offline chatbots, which extract information from PDF documents, rely on text extraction techniques, Term Frequency-Inverse Document Frequency, and embeddings-based search (e.g., FAISS, ChromaDB) to retrieve relevant information efficiently. Studies show that using LLM fine-tuning on domain-specific documents significantly improves the accuracy of chatbot responses. On the other hand, online chatbots leverage web search engines and APIs (such as OpenAI, Google Search, or Wikipedia) to fetch real-time information. Hybrid models that combine offline document retrieval with online search capabilities are gaining attention as they ensure both data privacy (offline mode) and updated information (online mode). However, challenges remain in seamlessly integrating both approaches. Issues such as **response selection optimization, reducing AI hallucinations, and improving retrieval accuracy need further research. Developing an efficient hybrid chatbot that can extract precise answers from PDFs and online sources while ensuring fast and accurate responses is a promising area of development.

IV. Methodology

This section outlines the methodological approach used in the development of the Hybrid Documents RAG-based Chatbot. The key components include document preprocessing, chunking, vector database integration, reranking, and the use of small/large language models (SLM/LLM). Additionally, testing and evaluation methods are discussed to ensure the system's effectiveness.

A. Data Preprocessing

Before embedding documents into the retrieval system, preprocessing is essential to enhance information extraction efficiency. The preprocessing pipeline includes:

- **Text Extraction**: Extracting text from PDFs, DOCX, and other document formats.
- **Cleaning**: Removing unwanted characters, stopwords, and special symbols.
- **Normalization**: Converting text to lowercase, stemming, and lemmatization to reduce dimensionality.
- **Tokenization**: Splitting text into meaningful units for efficient chunking.

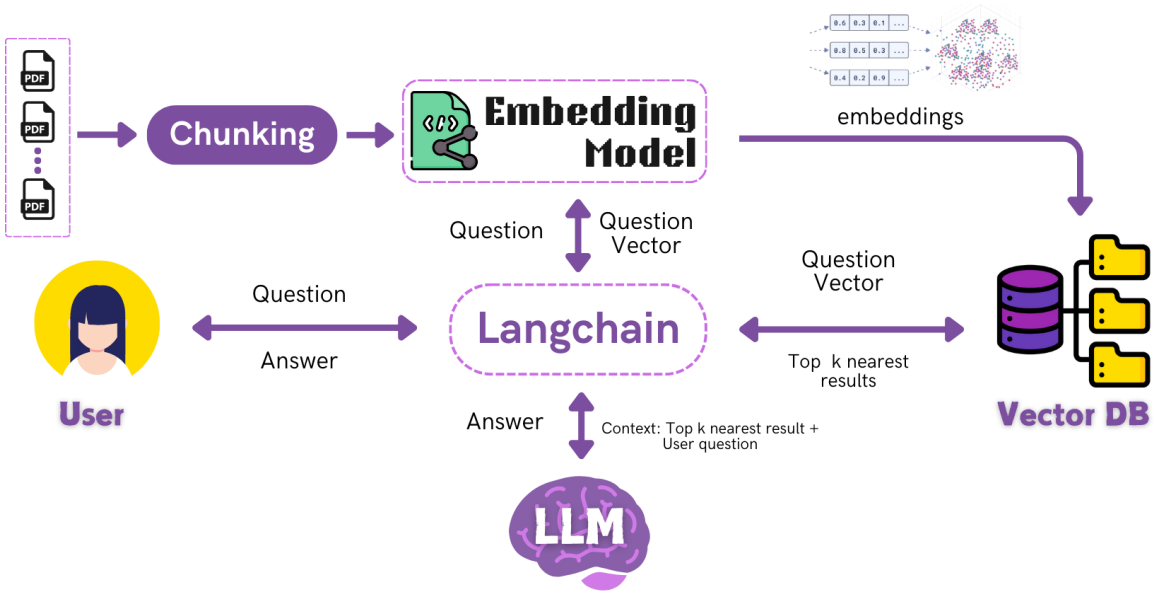

Fig. 1: RAG Pipeline

B. Chunking

Chunking plays a crucial role in breaking down large documents into manageable and semantically meaningful pieces. The method used includes:

- **Fixed-size Chunking**: Splitting text into equal-sized segments (e.g., 512 tokens per chunk).
- **Semantic Chunking**: Using NLP models to segment text based on meaning and context rather than fixed lengths.
- **Sliding Window Technique**: Overlapping chunks to retain context across different segments.

The choice of chunking technique impacts retrieval accuracy and ensures that relevant information is retained during processing.

C. Vector Database

To facilitate efficient document retrieval, embeddings of text chunks are stored in a vector database. The following components are considered:

- **Embedding Model**: A pre-trained model (e.g., Hugging Face Transformers, OpenAI) generates dense vector representations of text.
- **Vector Store**: FAISS is used for fast similarity search.
- **Indexing Strategy**: Hierarchical clustering and Approximate Nearest Neighbors (ANN) improve retrieval speed and accuracy.
- **Metadata Storage**: Additional metadata such as document source, timestamps, and keywords are stored for contextual ranking.

D. Reranking

To improve retrieval precision, a reranking mechanism is implemented:

- **BM25 Ranking**: A traditional term-frequency-based ranking method to filter initial results.
- **Cross-Encoders**: Transformer-based models are applied to re-score search results based on relevance.
- **Hybrid Retrieval**: Combining sparse (BM25) and dense (vector-based) retrieval to enhance performance.

E. Small and Large Language Models (SLM/LLM)

A hybrid approach is used for query answering:

- **SLM for Efficiency**: Lightweight models (e.g., MiniLM, DistilBERT) provide quick responses for straightforward queries.
- **LLM for Complex Queries**: Large models (e.g., GPT, Mistral) handle more complex reasoning tasks and provide contextualized responses.
- **Adaptive Model Selection**: The system dynamically selects between SLM and LLM based on query complexity and computational cost.

F. Query Processing Flow

The chatbot processes queries through the following stages:

1) **User Query Input**: The user submits a question.
2) **Embedding & Retrieval**: The query is embedded and compared against stored vectors to retrieve the most relevant chunks.
3) **Reranking**: Retrieved results are reordered for improved accuracy.
4) **SLM/LLM Response Generation**: A model generates the final answer.
5) **User Feedback Loop**: Users can rate responses, improving future retrieval quality.

V. Aim and Object

To develop a Hybrid Document Rag-based Chatbot that provides users with fast and reliable access to information from multiple PDFs and documents. The chatbot will offer both online and offline functionality through a conversational search interface.

VI. Objectives

1) **Document Retrieval System Design:** Design a practical document retrieval system capable of gathering data from various document formats (e.g., PDF, DOCX, TXT).
2) **Ranking System Development:** Create a ranking system to highlight the most relevant search results based on user queries, considering factors like keyword frequency, semantic similarity, and document structure.
3) **Offline Functionality Implementation:** Ensure offline functionality, enabling document searching without internet connectivity. This may involve local indexing and storage of document data.
4) **Interactive System Development:** Develop a user-friendly interactive system with a conversational interface for dynamic query refinement. This includes features like query suggestions, clarification prompts, and contextual understanding.
5) **Hybrid Database Integration:** Integrate a structured database (e.g., SQL) and a vector database (e.g., Faiss, Chroma) for efficient document storage and retrieval. The structured database can store metadata and the vector database can handle semantic search.
6) **Data Privacy and Security:** Prioritize data privacy and security by minimizing reliance on cloud-based services and implementing appropriate data protection measures. This could involve local processing and storage of sensitive information.

VII. Technical Approach (Optional - Add more details as needed)

This section would detail the specific technologies and methodologies used to achieve the objectives. For example:

- **Document Parsing:** Libraries like PyPDF2, Tika, or similar tools for extracting text from various document formats.
- **Indexing:** Techniques like TF-IDF, BM25, or word embeddings for creating an index of the documents.
- **Vector Embeddings:** Models like Sentence-BERT, or other embedding models for generating vector representations of text for semantic search.
- **Chatbot Framework:** Frameworks like Rasa, Dialogflow, or similar for building the conversational interface.
- **Database Technologies:** Specific choices for SQL and vector databases (e.g., PostgreSQL, SQLite, Faiss, Chroma).

VIII. Evaluation Metrics (Optional)

This section would outline how the chatbot's performance will be evaluated. For example:

- **Precision and Recall:** Measuring the accuracy of search results.
- **Mean Average Precision (MAP):** Evaluating the ranking of search results.
- **User Satisfaction:** Gathering feedback from users on the chatbot's usability and effectiveness.
- **Response Time:** Measuring the speed of the chatbot's responses.

IX. Testing and Evaluation

A. Unit Testing

Each module is tested in isolation to ensure correctness. This includes:

- **FAISS Vector Store**: Ensures similarity search and retrieval efficiency.
- **Hugging Face Embeddings**: Verifies the accuracy and consistency of embeddings.
- **Ollama LLM**: Tests response generation and contextual understanding.

B. Integration Testing

Testing the interaction and data flow between components to ensure smooth operation:

- **Sequential and Parallel Integration**: Test sequences of components that work together, or test multiple components in parallel.
- **Data Flow Testing**: Ensures data is correctly passed and transformed between components.
- **Interface Testing**: Verifies that the interfaces between components handle data correctly and manage errors appropriately.

C. End-to-End (E2E) Testing

E2E testing simulates real-world conditions to verify that the entire system operates as intended. This includes a full chat session with multiple steps.

D. Functional Testing

1) NLU Testing:

- **Named Entity Recognition (NER)**: Identifies specific data points.
- **Intent Recognition**: Identifies the user's intent behind a query.

2) NLP Testing:

- **Response Generation**: Evaluates accuracy, grammatical correctness, and coherence of chatbot responses.
- **Paraphrasing Quality**: Ensures long responses are summarized accurately without losing key information.

E. Load Testing

Simulates multiple sessions with the chatbot to test its performance under high demand.

F. A/B Testing

Tests multiple variations of the chatbot to determine which version performs better.

G. User Acceptance Testing (UAT)

Final approval phase before deployment to validate real-world usability:

- **Select Representative Users**: Gather insights from diverse testers.
- **Collect Qualitative Feedback**: Evaluate user experience and satisfaction.
- **Iterative Testing**: Implement improvements based on feedback.

H. Accuracy Metrics

- **BLEU Score**: Measures n-gram overlap between generated and reference responses.
- **ROUGE Score**: Evaluates precision of overlapping n-grams.
- **Exact Match**: Measures the percentage of responses that exactly match predefined answers.
- **Hit Rate**: Determines whether at least one relevant document is retrieved.

I. Latency and Efficiency Metrics

- **Response Time (Latency)**: Measures the time taken by the chatbot to process and generate a response.
- **Throughput**: Evaluates the number of requests the chatbot can handle without performance degradation.
- **Query Retrieval Time**: Assesses how long FAISS takes to retrieve relevant documents.

J. Dataset

Data is essential for **evaluating** and **tracking improvements** in our project.

It helps us assess performance, identify weaknesses, and refine our approach. We will create a dataset of ***1,000 samples*** to ensure thorough evaluation and optimization, leading to a more accurate and reliable system.

X. Conclusion

The Hybrid Documents RAG-based Chatbot is designed for efficient and accurate interaction with documents. By incorporating preprocessing, advanced chunking, vector-based retrieval, and reranking, it ensures precise information extraction. The hybrid approach of using both Small and Large Language Models strikes a balance between speed and accuracy.Thorough testing, including unit, integration, and NLP-based assessments, guarantees reliability. Metrics such as accuracy, latency, and user feedback facilitate continuous improvements. This methodology results in a secure, efficient, and user-friendly chatbot that enables seamless document retrieval and interaction.

References

[1] "Evaluation of Retrieval-Augmented Generation: A Survey," arXiv, 3 Jul 2024. Available: https://arxiv.org/abs/2405.07437.

[2] "Evaluating Retrieval Quality in Retrieval-Augmented Generation," ACM, 11 Jul 2024. Available: https://dl.acm.org/doi/abs/10.1145/3626772.3657957.

[3] Y. Xi, W. Liu, X. Dai, R. Tang, W. Zhang, Q. Liu, X. He, and Y. Yu, "Context-aware reranking with utility maximization for recommendation," *arXiv preprint arXiv:2110.09059*, Feb. 2022.