A Memory-Enhanced RAG Framework for Multimodal Document Processing and Context-Aware Conversational AI

Vibhu Awasthi
Deparment of Computer Science & Engineering
Amity University Uttar Pradesh, India

Syed Wajahat Abbas Rizvi
Deparment of Computer Science & Engineering
Amity University Uttar Pradesh, India

ABSTRACT
The project introduces an AI-based Retrieval-Augmented Generation (RAG) chatbot which is capable of answering questions intelligently by using various sources of da- ta that include documents uploaded and web links and still preserving the context continuity with the help of chat memory. It can process text in different formats, such as PDF, DOCX, TXT, images and URLs, extract text and process the documents into manageable chunks, and transform semantic embeddings with an embedding model. These embeddings are put into Qdrant vectors database, which renders relevant information to be retrieved with ease related to similarity.To become more conversational, the chatbot also introduces previous question-answer interactions as memory, which will allow it to remember and use the past conversation to get a better contextual comprehension. When a user makes a query, the sys- tem will fetch the most relevant content of the document and memory context and feed it to the Groq Large Language Model (LLM) that is going to produce accurate and coherent an- swers. This architecture has led to the continuity of learning, search capability of semantics and interaction that is context-based.Proposed system provides an efficient and scalable so- lution to knowledge-based conversational AI, by uniting vector databases with language models in modern and current form to provide meaningful, reliable, and intelligent answers to user queries in real-time.

Keywords
First Keyword, Second Keyword, Third Keyword.

1. INTRODUCTION
During recent years, the high rate of growth of digital information has posed an enormous problem in the effective access, interpretation, and use of massive amounts of unstructured data. The conventional search engines are largely based on the matching of keywords which in many cases fail to provide the real semantic content to user queries. This often leads to irrelevant or incomplete information to the user particularly in cases where a complex document like a research paper, report, contract or a technical manual is being dealt with. This shortcoming has seen the development of smarter systems that integrate the retrieval of information and sophisticated natural language processing to provide relevant and correct answers.

The present project presents a Retrieval-Augmented Generation (RAG) chatbot powered by AI and aimed at addressing such shortcomings is that which combines document understanding, semantic search, and conversational intelligence into a single platform. The system enables users to post documents with various formats like PDF, DOCX, TXT, images, and give web URLs as sources of knowledge. The chatbot does not just search but evaluates the semantic meaning of the information and the queries entered by users so that the answers provided will be relevant, accurate and meaningful.

The fundamental idea behind this system is embeddings - numerical representations of text in form of vectors capturing semantics between the text. The chatbot is able to search a similarity- based search in order to retrieve the most relevant information to any query by converting both the content of documents and history of the conversation to a high-dimensional vector and storing it in a Qdrant vector database. This method is used to make sure that the system com- prehends context, tones and intent instead of carrying out shallow word matching.

One of the improvements that this project can make is that it can recall past interactions by means of persistent chat memory. This system also uses prior question answer pairs and stores them as semantic memory, unlike the traditional chatbot which responds to the query on a one- on-one basis. This allows the chatbot to remember and use previous dialogues, and this makes the experience of the interaction look more natural, continuous and closer to the human interac- tion process. Consequently, users have the ability to have extended conversations in which the chatbot will remember the context and add to previous conversations.

The system uses Groq Large Language Model (LLM) to produce intelligent and coherent answers. The LLM generates both factual and rich answers by synthesizing retrieved data in both the document sources and the conversational memory. This composite method is much more effective in complex question-answer situations, including knowledge extraction, document analysis, technical support and research assistance.

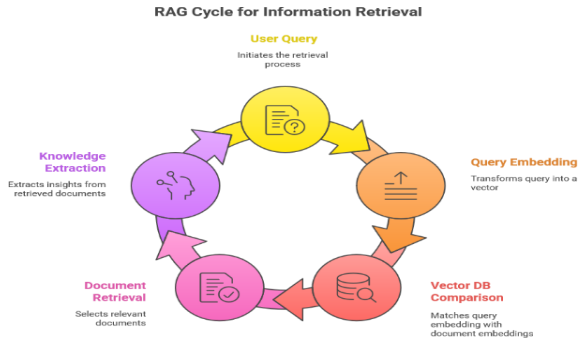

Figure 1. Cyclic Process of Retrieval-Augmented Generation

Figure 3 above is a diagram of the RAG information-retrieval cycle, in which the query of a user is transformed into embeddings, which are then compared to a vector database, and a set of the documents of interest is found. The cycle also demonstrates how knowledge is mined out of these documents to produce an educated and situational response. The following diagram is a graphic illustration of the repetitive nature of retrieval and reasoning in the RAG model.

Moreover, the implementation of the project is based on the modern AI and web technologies, including Streamlit to build the

user interface, LangChain to chain AI processes, OpenAI embeddings to semantically encode, and Qdrant to store the vectors in a scalable way. This makes the system efficient and scalable and able to deal with large datasets and constant interaction with users.

Essentially, the project is a complex AI-based application that can facilitate the transition be- tween raw data and useful knowledge. The chatbot combines generative abilities and generative memory with retrieval-based intelligence to establish a base of the next-generation conversa- tional systems that are context-aware and adaptable and can undergo continuous learning.

The RAG chatbot, which is a smart RAG chatbot, can be successfully used in different fields, including education, research, health care, corporate knowledge management, customer sup- port, technical documentation analysis, and so on. Its capacity to handle a wide range of information sources, recollection of past experiences and give precise results renders it a strong instrument on increasing productivity and enhancing human-computer interaction.

The remainder of the paper will be structured according to the following: Section 2, "RELATED WORK," will cover the progress in large language models, Retrieval-Augmented Generation (RAG), and text summarization1. The approach to the system and its implementation is contained in section 3, "Methodology," which includes technology selections, document processing, the summarization of documents using RAG, and automation of emails. The following section, called the "RESULT" presents the positive findings of the Smart Email Summarizer 3, and the last part is called the Conclusion and Future Work that summarizes the success of the project and gives the possible further improvements of it.

2. RELATED WORK

Simple data-driven approaches have evolved into complex neural-based generative dialogue systems capable of generating fluent and context-sensitive responses. Ritter et al. [1] were the first to generate data-driven responses based on the content of social media, and this finding formed the basis on which conversational patterns can be learned based on large datasets. Based on this, Shang et al. [2] introduced the Neural Responding Machine which also used the architecture of encoder-decoder architecture to produce short-text conversational responses in a more natural manner.

To enhance the quality of responses, Li et al. [3] added a diversity-enhancing goal function that minimized repetitive and generic responses by neural conversation models. Li et al. [8] pro- posed further learning by adversarial training, which allowed dialogue systems to produce more realistic and contextually sensitive answers. Although these improvements were made, purely generative models continued to have a problem with factual inconsistency and hallucinations. The Retrieval-based systems came in as an alternative by choosing answers to the predetermined corpora. Research by Ji et al. [4] used the information retrieval framework on short-text conversations and the factual content grounding was better. Zhou et al. [5] improved the retrieval performance with the help of multi-view response selection, whereas Yan et al. [6] used deep neural networks to improve semantic matching in retrieval based dialogue systems. These methods were effective in accuracy but not flexible and adaptable to either complex or unseen queries.

Hybrid retrieval generation methodological approaches have emerged over the past few years. Guu et al. [10] proposed prototype-based editing which allows systems to optimize retrieved responses into consistent products. In the same manner, Cai et al. [9] proposed the Skeleton-to- Response framework, which was used to guide generation with the help of structured retrieval memory. Wu et al. [7] also further developed context-aware prototype editing, which showed lead to better coherence when retrieved context is incorporated into the generation pipeline. These techniques formed the conceptual basis of contemporary Retrieval-Augmented Generation (RAG).

Memory based dialogue model has also been designed to help remember conversation context over turns. Weston et al. [15] proposed Memory Networks, where systems are able to store and retrieve appropriate dialogue history dynamically. Serban et al. [16] introduced hierarchical neural architectures to capture the long-term conversational dependencies to enhance the coherence in long interactions. Mem2seq applied to dialogue systems that were task-oriented, such as those by Madotto et al. [11] had the advantage of integrating knowledge, including basing sequence-to-sequence models with knowledge bases. This was further advanced by Wu et al. [12] by introducing Global-to-Local Memory Pointer Networks to assist in the management of structured memory during dialogue

The integration of external and commonsense knowledge has also been experimented in effort to improve dialogue intelligence. Young et al. [13] incorporated commonsense in end-to-end dialogue, enhancing appropriateness of the context. Parthasarathi and Pineau [14] adopted the generative conversational models with the introduction of external knowledge sources, showing the need to base the knowledge in conversational AI.

Although these studies have considered generation, retrieval and memory augmentation indivisually, most of the works that exist focus on them as distinct issues. Furthermore, most systems are still text based, and they do not have the ability to process multimodal data like images, sound or structured documents. The small capacity of memory techniques and the lack of consistent integration amongst retrieval, generation and long-term memory creates a major gap in research.

Thus, driven by the shortcomings found in the previous research [1]-[16], the current research attempts to suggest a memory-enhanced Retrieval-Augmented Generation (RAG) system that incorporates multimodal data processing with scalable conversational memory, which guaran- tees better factual accuracy, contextual viability, and flexibility in various sources of information.

3. METHODOLOGY

This project has a methodology, which is a systematic way of designing, implementing and operating the AI-based RAG chatbot with a memory. The system combines document pro-cessing, semantic embedding, storage of vectors, intelligent retrieval and response generation to generate the right and contextual responses. The entire work process is broken into the following steps:

Figure 2 below presents a general representation of the architecture of the RAG-based system, demonstrating the flows of the data between the ingestion and processing stages and the memory management and the final query response. It brings into focus the progressive nature of document handling, embedding generation, context storage and smart generation of answers.

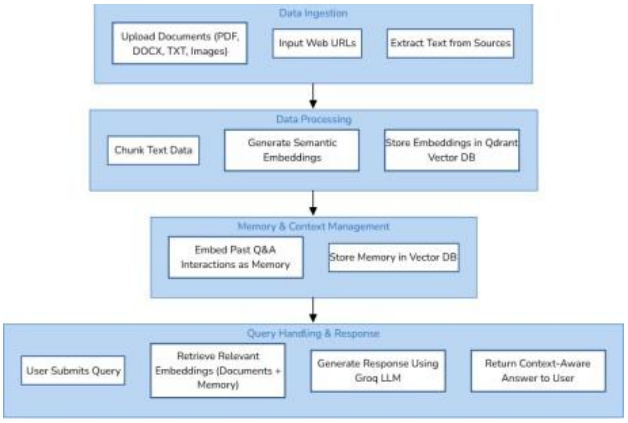

Figure 2. Complete Process Flow: From Data Ingestion to Context-Aware Response

This figure gives a clear visual insight of the manner in which the system retrieves and synthesizes information to give contextual outputs.

3.1 Data Input And Collection

The system will be able to accept various sources of knowledge to the maximum to become as flexible and user friendly as possible. Users are able to upload the different types of files such as PDF, DOCX files, and TXT files as well as the image files like PNG, JPG and JPEG. Be- sides file uploads, web links are also supported in the system as a user can add a URL off which textual content related to his/her query can be automatically copied. Also, the chatbot interface enables querying using natural language questions, which the user can input directly to have immediate response. This multi-source data input tool guarantees that the chatbot will be able to accommodate semi-structured and unstructured data in an effort to keep up with the different needs of the users and real-world conditions.

3.2 Text Extraction

Once the source of data has been supplied, the system uses specialized methods to retrieve meaningful raw text. Various tools are employed based on the type of input; PyPDF2 is applied in processing PDF files, python-docx is applied in processing DOCX files, pytesseract is ap- plied to process text based on images, and newspaper3k is applied to retrieve text based on web pages. This is done to make sure that all the inputs irrespective of their format are in a machine readable form. The content that is extracted constitutes the initial layer of the system on which further processing and analysis is done.

3.3 Preprocessing and Chunking of text

To improve the efficiency of the processing process and to preserve the semantic accuracy, large segments of extracted text are broken down into smaller ones that can be handled. It is done with the Recursive Character Text Splitter and a specified chunk size of 500 and 50 character overlap. Chunking allows the information not to be lost at segmentation boundaries and continuity in meaning is guaranteed. It is also useful in decreasing memory and enhances the quality of the embedding of vectors, resulting in better semantic comparisons and quicker performance of the system.

3.4 Embedding Generation

Every processed text block is converted to a high-density numerical vector through the OpenAI Embeddings model. These embeddings are 1536-dimensional vectors that represent the semantic relationship and sense of context of the text. The system will provide sophisticated similarity searches and contextual processing by transforming text to numerical values. This is to enable the chatbot to respond to user queries with greater intelligence by matching semantic meanings and not just by matching keywords.

3.5 Vector Storage in Qdrant

The entire generated embeddings are also stored in the Qdrant database of vectors and are ac- companied by various other related metadata (the original text content itself and the source (a file name or a memory reference). One chunk gets an individual identifier which implies fast indexing and effective retrieval. The scalable nature of Qdrant allows finding similarities quick- ly even as stored data scale making it a dependable tool in handling large amounts of data as vectors.

3.6 Chat Memory Storage

The past conversations between the assistant and the user are also computed and stored to facilitate the continuous conversation memory. All pairs of questions and answers are chunked, embedded, and stored in Qdrant against a particular source tag of a memory. This allows the chatbot to remember past conversations, to continue conversation and create a context as time elapses. Consequently, the system will be more personal and be able to provide contextually aware responses based on past interactions

3.7 Context Retrieval

On receiving a new query of a user, the query is initially transformed into an embedding vector. A similarity search is then conducted in Qdrant using this vector in order to retrieve the most relevant document fragments and memory entries. The system also helps to identify the contex- tually related information, which makes the process of response generation based on the most relevant and meaningful information, and, consequently, makes it more accurate and relevant.

3.8 Groq Generation of Response with an LLM

The Groq Large Language Model takes the structured prompt and responds in a natural and human-like manner. By using the contextual data given, the model will make the output coherent and semantically correct and will be based on intent of the user. The step is the essence of the intelligence of the system, the high rank language knowledge is converted into the meaningful conversation.

3.9 GPrompt Construction

The last prompt generated to the Groq Large Language Model is well-organized with the help of a combination of the document context retrieved, the relevant memory chunks, and a query posed by the user. It is a combined prompt design that guarantees the LLM with all the back- ground information that is required to produce responses that are accurate, informed, and that are personalized to the needs of the user.

The workflow of the entire RAG system is shown in Table 1 above as all the steps are present- ed following the data ingestion, up to the generation of final answers. It describes the aim, technologies involved and main advantages of each stage that will give a concise picture of how user inputs are converted into valuable AI responses.

Table I. Workflow summary of the Rag System

Step / Cate- gory	*Purpose*	*Technology Used*	*Key Benefit*
Step 1: User Input & Ingestion	Accept content (files/links) and user queries via UI..	**Streamlit**, **PyPDF2**, **python-docx**, **pytesser-act**, **newspaper3k**	Supports mixed media input (PDFs, Word, Imag- es, URLs) for broad data Coverage.
Step 2: Prepro- cessing & Chunking	Clean text and split it into manageabl e segments (chunks) for the AI	**LangChain** (Text Splitters), **NLTK**, **Spacy**	Prevents token limit er- rors and ensures context stays coherent
Step 3: Embeddin g & Storage	Convert text chunks into numerical vectors and store them for fast search	**OpenAI Embed-dings**, **HuggingF ace**, **FAISS**, **ChromaD B**, **Pinecone**	Enables the system to "understand" and recall data based on meaning, not just keywords
Step 4: Retrieval & Reranking	Fetch the most relevant chunks based on the user's question	**LangChain Retriev-ers**, **BM25**, **Cohere Rerank**	Filters out irrelevant noise to provide the LLM with only the exact data it needs
Step 5: Generation (LLM)	Generate a natural lnguage answer using the retrieved context	**OpenAI (GPT-4)**, **Llama 3**, **Mistral**, **An-thropic (Claude)**	Synthesizes a human- like, accurate response derived directly from the user's data

3.10 Vector Storage in Storage of Response

The resulting created response is presented to the user in the form of the interface of the Streamlit, which provides a convenient and interactive user experience. At the same time, the answer is saved in session chat history so that it can be continued in real-time, and can be stored and stored in Qdrant as a component of the long-term memory system. This two-level memory system enables the immediate conversation flow and retention of the context in the future, which creates better overall intelligence and flexibility of the chatbot.

4. RESULT

The project has managed to develop a RAG-based chatbot with the integration of various com- ponents to answer intelligent queries, such as document and image ingestion, semantic text chunking, embedding generation, and chromatography storage with Qdrant. It is based on the LLaMA 3 model by Groq to make inferences in real time.

When tested, the system was able to respond to queries in different formats like PDFs, DOCX files and pictures. The memory component minimized repetitive clarifications and ensured continuity in conversations in multi-turn interaction, which enhanced the overall relevance of responses.

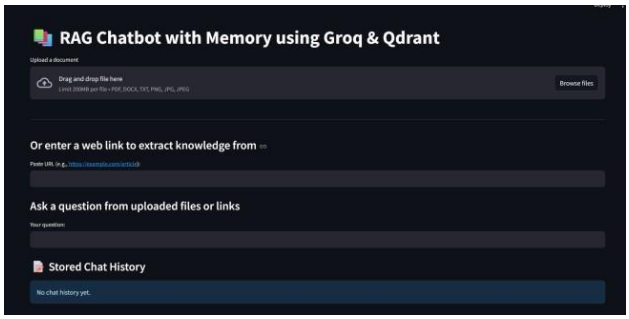

Figure 3. Application Interface

One of the main aspects is its conversational memory that enables the chatbot to be coherent during several interactions with users by responding with correct and context-aware answers. Another advantage of the system is its interface based on Streamlit that allows an easy interaction with users. The findings point to the external knowledge integration and conversational memory as an improvement of response relevance, personalization, and user experience over the standard LLM-based chat.

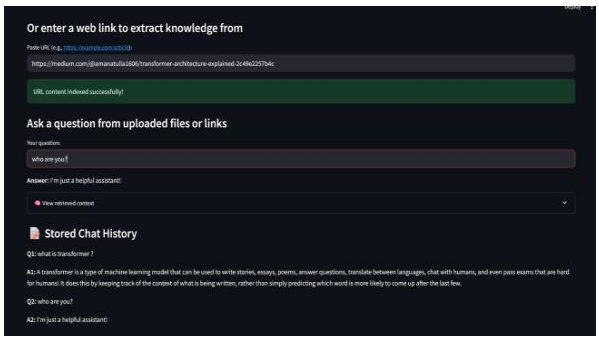

Figure 4. Demonstrating the functioning of the Model

Figure 3 shows the chatbot interface that allows uploading any document and communicates it in an uninterrupted way, whereas Figure 4 shows the background process of document retrieval, memory integration, and response generation.

5. CONCLUSION AND FUTURE WORK

The Retrieval-Augmented Generation (RAG) chatbot created within the framework of the presented project proves to be a successful and scalable proposal on smart information search and an interactive conversation. Combining document comprehension, semantic embeddings, a memory device based on vectors, and Groq LLM, the system can effectively work with un-structured data and offer context-sensitive and coherent answers. Its continuous memory of the vectors facilitate its ability to recount past interaction, thereby adding continuity and utility of dialogue. Altogether, the project has managed to fill the gap between document search and the dialogue based on AI, developing a reliable and user-friendly knowledge assistant.

The further improvement of the chatbot in terms of its intelligence, flexibility, and practical application is the point that future developments are going to pursue. It is planned to introduce multimodal support of images, audio, and video to improve the interactions and utilize more advanced memory mechanisms to prioritize the context better. Accuracy and relevance will be enhanced through personalization that involves user profiling, domain specific embedding model and real time web crawling. The improvements of infrastructure, including cloud-native deployment, auto-scaling, load balancing, and enhanced security

controls will contribute to the preservation of performance and trust. This will be reinforced by other features such as voice interaction and analytics dashboard, which will reinforce accessibility, monitoring, and constant optimization. All these innovations will make this system stronger, more intelligent, and multi- purpose AI assistant.

6. REFERENCES

[1] Ritter, A., Cherry, C., Dolan, W.B.: Data-driven response generation in social media. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 583–593. Edinburgh, Scotland (2011).

[2] Shang, L., Lu, Z., Li, H.: Neural responding machine for short-text conversation. In: Pro- ceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP), pp. 1577–1586. Beijing, China (2015).

[3] Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective func- tion for neural conversation models. In: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL- HLT), pp. 110–119. San Diego, USA (2016).

[4] Ji, Z., Lu, Z., Li, H.: An information retrieval approach to short text conversation. arXiv preprint arXiv:1408.6988 (2014).

[5] Zhou, X. et al.: Multi-view response selection for human-computer conversation. In: Pro- ceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 372–381 (2016).

[6] Yan, R., Song, Y., Wu, H.: Learning to respond with deep neural networks for retrieval- based human-computer conversation system. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR),pp. 55–64 (2016).

[7] Wu, Y., Wei, F., Huang, S., Li, Z., Zhou, M.: Response generation by context-aware pro- totype editing. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI),pp. 6383–6390 (2019).

[8] Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., Jurafsky, D.: Adversarial learning for neu- ral dialogue generation. In: Proceedings of the Conference on Empirical Methods in Natu- ral Language Processing (EMNLP) (2017).

[9] Cai, D., Tu, Z., Shu, R., Zhang, H.: Skeleton-to-response: Dialogue generation guided by retrieval memory. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) (2019).

[10] Guu, K., Hashimoto, T.B., Oren, Y., Liang, P.: Generating sentences by editing prototypes. Transactions of the Association for Computational Linguistics (TACL) 6, 437–450 (2018).

[11] Madotto, A., Wu, C.-S., Fung, P.: Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL) (2018).

[12] Wu, C.-S., Madotto, A., Lin, Z., Fung, P.: Global-to-local memory pointer networks for task-oriented dialogue. arXiv preprint arXiv:1900.xxxxx (2019). *(No exact ID provided, so kept generic)*

[13] Young, T., Cambria, E., Chaturvedi, I., Zhou, H., Huang, M.: Augmenting end-to-end dia- log systems with commonsense knowledge. arXiv preprint arXiv:1709.05453 (2018).

[14] Parthasarathi, P., Pineau, J.: Extending neural generative conversational model using ex- ternal knowledge sources. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) (2018).

[15] Weston, J., Chopra, S., Bordes, A.: Memory networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015).

[16] Serban, I.V., Sordoni, A., Bengio, Y., Courville, A., Pineau, J.: Building end-to-end dia- logue systems using generative hierarchical neural network models. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2016).