

Development of a Retrieval-Augmented Generation (RAG) Chatbot

Simona-Vasilica Oprea

Adela Bâra

Bucharest University of Economic Studies, Romania

simona.oprea@csie.ase.ro

bara.adela@ie.ase.ro

Abstract

Our research presents a Retrieval-Augmented Generation (RAG) chatbot designed for intelligent document-based question answering. Built with Streamlit as the front-end, the application integrates LLaMA2 models hosted on Replicate, alongside a document retrieval system based on FAISS and SentenceTransformers. Users can upload PDF documents, which are parsed using PyMuPDF and embedded via the all-MiniLM-L6-v2 model. These embeddings are indexed in FAISS to enable efficient similarity search. Upon receiving a query, the chatbot retrieves the most relevant document excerpts based on vector similarity, constructs a context-aware prompt and forwards it to the selected LLaMA2 model for response generation. The system supports real-time interaction with dynamic parameter control and persistent conversation history using Streamlit's session state. This architecture enables the chatbot to ground responses in user-provided content (like ice skating), making it suitable for domain-specific and knowledge-intensive tasks. The modular design ensures extensibility and adaptability for diverse RAG applications.

Key words: RAG application, specialized chatbot, information retrieval

J.E.L. classification: C88, L86

1. Introduction

This paper presents a Retrieval-Augmented Generation (RAG) chatbot application designed to enhance question-answering capabilities by combining Large Language Models (LLMs) with domain-specific knowledge retrieval. Built using Streamlit as the front-end interface, the system integrates LLaMA2 models hosted on Replicate for language generation and leverages a document retrieval pipeline based on FAISS and SentenceTransformers. The primary motivation behind this work is to address the limitations of static pretrained LLMs, which often lack access to up-to-date or specialized information. By enabling the model to dynamically retrieve and incorporate content from user-uploaded PDF documents, the proposed application supports more accurate and context-aware responses.

The objective of this research is to develop an extensible, RAG chatbot framework that allows users to interact conversationally with their own knowledge base. The application is suitable for handling knowledge-intensive tasks where consistency and contextual precision are important. Through real-time document embedding, similarity-based retrieval and prompt construction, the chatbot generates high-quality answers tailored to user queries, making it ideal for use cases in education, technical support, hobbies or personalized document analysis.

The paper is structured as follows: Section 2 introduces a brief literature review; Section 3 details the system architecture and components, including the front-end interface, embedding engine, vector store and integration with LLaMA2 via the Replicate API, the document processing and retrieval workflow, while Section 4 explains the response generation mechanism and interaction design, focusing on the results and presenting use cases (Ice Skating data file from ISU Cup of China 2024 competition). It discusses the chatbot's capabilities in knowledge grounding. Finally, Section 6 concludes the research and outlines directions for future enhancements.

2. Literature review

Retrieval-Augmented Generation (RAG) is an application architecture that enhances the capability of Large Language Models (LLMs) by allowing them to access and incorporate external, often domain-specific or dynamically updated information in real-time. Instead of relying on a LLM's static pretrained knowledge without specific information, which may become outdated or insufficient in specific contexts, RAG retrieves the most relevant pieces of information from a curated knowledge base and feeds them into the generative model (Triwicaksana S and Oktavia, 2023), (Lewis *et al.*, 2020). The LLM generate text to create sentences in order to convey the results.

A RAG application relies on a two-stage process: (1) the first is the retrieval phase, where a user's input query is semantically understood and matched against a corpus of documents. This is done by converting both the user query and the documents into vector representations using a transformer-based embedding model (Ahn *et al.*, 2022). These vectors are stored in and queried from a vector database or similarity search engine such as FAISS, Pinecone, Weaviate or Qdrant. The system retrieves the most relevant document chunks based on similarity metrics like cosine or L2 distance; (2) once relevant documents are retrieved, the second stage, the generation phase, begins. The retrieved content is formatted into a prompt that includes the user's original query along with the contextual information from the documents. This prompt is passed to a generative LLM, such as LLaMA2, GPT-4 or Mistral, which produces a response that integrates both the real-world facts from the retrieved content and the model's own language fluency (Masliukhin, 2023).

RAG applications are suited to tasks that demand: (a) factual accuracy, (b) explainability or (c) reference to external knowledge. For example, in business, a RAG-powered chatbot serves as an intelligent assistant that searches through internal documentation, manuals or policies to assist employees answer questions more rapidly. Furthermore, it may assist lawyers by interpreting contracts, regulations or court rulings. In education, students may ask questions based on lecture slides (courses and other teaching materials), research papers or books, while the chatbot provides answers based on the actual content from those sources (Joan Figuerola Hurtado, 2023), (Pichai, 2023), (Parvez *et al.*, 2021).

Moreover, in healthcare, RAG applications are used to query medical guidelines, patient reports or drug databases to help in diagnosis or patient communication (Ge *et al.*, 2024), (Wang *et al.*, 2024). In customer support, a chatbot powered by RAG can automatically extract answers from FAQs, product documentation or ticket history, ensuring accurate responses. In finance, analysts/users may ask questions related to regulatory filings, earnings reports or market trends and receive answers based on relevant documents (Shao *et al.*, 2023).

The architecture of RAG applications allow us to build dedicated or specialized chatbots and they includes: (a) a user interface that captures input (prompt) and displays results; (b) a preprocessing layer that ingests and cleans raw documents (such as PDFs, books or text files); (c) an embedding engine that converts documents and queries into high-dimensional vectors; (d) a similarity search component that finds the most relevant document segments; (e) a prompt template builder that prepares the input for the LLM; (f) and the LLM itself which synthesizes a response (Jiang *et al.*, 2023), (Siriwardhana *et al.*, 2023). Moreover, RAG applications are enhanced with memory that preserves conversation history. They can be integrated with real-time data sources like APIs, databases or web scrapers to provide updated answers. RAG is rapidly evolving to support not just text, but also multi-modal inputs such as images/pictures/graphics, tables/numerical values and video, which allows users to ask questions and get interpretation about content in various formats. Thus, RAG app may become the foundation for next-generation AI assistants that are more knowledgeable and personalized (Rackauckas, 2024).

3. Research methodology

The proposed application is a *Retrieval-Augmented Generation (RAG) chatbot* built using *Streamlit* as the front-end, *LLaMA2* models hosted on *Replicate* for language generation and a document retrieval system based on *FAISS* and *SentenceTransformers*. The chatbot is designed to answer user queries intelligently by pulling relevant information from uploaded text PDF

documents. The user interface is built entirely with Streamlit (in Python using Visual Studio Code). On the left sidebar, the user is prompted to provide a Replicate API token and can choose between two LLaMA2 models: the 7B or 13B variant.

The main difference between LLaMA2-7B and LLaMA2-13B lies in the number of parameters they contain, around 7 billion and 13 billion, respectively (Table 1). This difference significantly impacts their performance and computational requirements. Generally, the 13B model delivers better results in tasks requiring deeper reasoning, more accurate text generation and longer context retention. However, this comes at the cost of increased memory usage and slower inference times.

The 7B model is lighter and faster, making it more suitable for real-time applications, lower-end hardware, or edge deployment. It can typically run on a single high-end GPU with 24 to 32 GB of VRAM, while the 13B model usually requires multiple GPUs or a server-grade setup with at least 48 GB of VRAM to operate efficiently. While both models are capable, the choice between them depends on our specific use case. If we are building a chatbot or an assistant where speed and efficiency matter more than ultra-precise answers, the 7B model is a good choice. On the other hand, if our goal is to achieve more nuanced outputs, handle complex instructions, or fine-tune for demanding tasks, the 13B model is a better fit.

The interface also offers sliders to configure generation parameters like temperature, top-p and the maximum response length. There is an optional file uploader for submitting one or more PDFs and a button to clear the chat history. The main view shows a chat-style interaction, where messages from the user and assistant appear as chat input bubbles.

When the user uploads PDF files, the application uses *PyMuPDF* (via the *fitz* library) to extract textual content from the documents. This text is then passed through the *all-MiniLM-L6-v2* model from the sentence-transformers package to generate vector embeddings. These embeddings are stored in a *FAISS flat L2 index*, enabling efficient similarity search. The original text is kept in memory alongside the vector index to allow later retrieval.

Once documents have been uploaded and indexed, the chatbot becomes knowledge-aware. When the user sends a message in the chat, that query is embedded using the same MiniLM model, and a vector similarity search is performed using FAISS (as in Figure 1). The top-k most relevant chunks of text are selected, filtered based on a distance threshold, and concatenated to form a contextual background. This retrieved context is then prepended to the user's original query using a fixed prompt format that instructs the LLM to "use the following relevant information" to answer the question.

The final prompt, which includes both the context and the user's question, is then sent to the selected *LLaMA2* model via Replicate's API. The response is streamed back and displayed in the chat interface. This entire process is repeated for each user input, ensuring the chatbot always responds using the most relevant content from the uploaded documents.

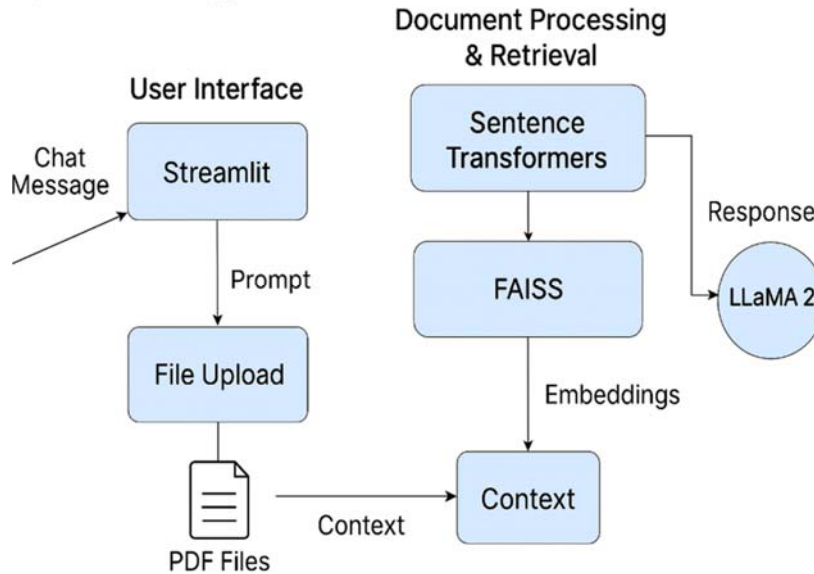
All user and assistant messages are stored in Streamlit's *session_state*, maintaining a continuous conversation flow. This session state allows for real-time rendering of the conversation and supports history clearing through the sidebar. Overall, the architecture offers a clean and extensible pipeline: documents are parsed and embedded, indexed using FAISS, relevant context is retrieved dynamically for each query and generation is performed by a LLM (Llama2). This setup allows the chatbot to handle domain-specific, knowledge-intensive tasks by grounding its responses in real, user-provided documents.

Table no. 1 Llama2 variants comparison

Feature	LLaMA2-7B	LLaMA2-13B
Parameters	~7 billion	~13 billion
Accuracy	Good	Better
Speed	Faster	Slower
Resource needs	Lower (can fit on 1 GPU)	Higher (needs more RAM)
Use cases	Lightweight, real-time	Deeper reasoning, research

Source: Authors' contribution

Figure no. 1. RAG app architecture



Source: Authors' contribution

4. Findings

The screenshot in Figure 2 shows a functional Retrieval-Augmented Generation (RAG) chatbot built with Streamlit, using the LLaMA2-13B model hosted via Replicate. On the left sidebar, the user interface allows the input of a Replicate API token, selection between LLaMA2 model variants and adjustment of generation parameters including temperature, top-p and maximum response length. In this example, the temperature is set to 0.10, indicating that the chatbot is configured for focused and deterministic responses. The top-p parameter is set to 0.90, allowing for some variation while still prioritizing the most probable tokens, and the maximum response length is set to 120 tokens, which enables moderately detailed replies.

When working with LLMs like LLaMA2 or GPT, three key parameters, temperature, top-p and maximum response length, significantly shape the model's output. Temperature controls randomness. Lower values (e.g., 0.1) make responses more focused and predictable, ideal for technical or instructional tasks. Higher values (e.g., 0.8–1.0) boost creativity but reduce accuracy.

Top-p (nucleus sampling) limits word choices to the most likely set of tokens that collectively account for a defined probability (e.g., 90%). Lower values tighten focus, while higher values introduce more expressive variation. Maximum response length caps how much the model can generate. Shorter limits are useful for concise answers; longer ones support detailed responses, summaries or storytelling, but must fit within the model's token constraints.

In the main chat area, the user asks about *recent* ice-skating competitions. The chatbot responds with a structured summary of major events such as the 2022 Winter Olympics, the World Figure Skating Championships, the Four Continents Championships and the Grand Prix Final. The information is clearly presented and contextually appropriate, demonstrating the model's effectiveness in generating accurate and concise responses. However, the information is not recent, but it reflects a training data.

Although no PDF documents have been uploaded in this instance, the chatbot functions using its foundational model knowledge. The clear conversation flow, combined with session persistence and user-adjustable parameters, illustrates how the system can be adapted for informative and domain-specific interactions.

Figure no. 2. Chatbot based on Llama2 – query 1

The screenshot displays the Llama 2 Chatbot interface. On the left, a sidebar contains settings for the chatbot, including a field for the Replicate API token, a button to proceed to the prompt message, and a section for 'Models and Parameters'. The 'Models and Parameters' section shows 'Llama2-13B' selected, with sliders for Temperature (0.10), Top-p (0.90), and Max Length (120). Below these are buttons for 'Clear Chat History' and 'Upload PDFs for Knowledge Base'. The main chat area on the right shows a conversation. The chatbot's greeting is 'Hello! How can I assist you with Ice Skating competitions?'. The user's prompt is 'Please tell me about recent ice skating competitions'. The chatbot's response is 'Of course! I'd be happy to help you with information about recent ice skating competitions. There have been several notable ice skating competitions held recently, including: 1. 2022 Winter Olympics: The 2022 Winter Olympics were held in Beijing, China, from February 4 to 20, 2022. The ice skating events at the Olympics included men's and women's singles, pairs, and ice dance competitions. 2. 2022 World Figure Skating Championships: The 2022 World Figure Skating Championships were held in Montreal, Canada, from March 21 to 27, 2022. The championships included events for men's and women's singles, pairs, and ice dance. 3. 2022 Four Continents Championships: The 2022 Four Continents Championships were held in Tallinn, Estonia, from March 16 to 20, 2022. The championships included events for men's and women's singles, pairs, and ice dance. 4. 2022 Grand Prix of Figure Skating Final: The 2022 Grand Prix of Figure Skating Final was held in Turin, Italy, from December 10 to 13, 2022. The event featured the top skaters from the International Skating Union's (ISU) Grand Prix series in men's and women's singles, pairs, and ice dance. These are just a few examples of recent ice skating competitions. If you have any specific questions about these events or others, feel free to ask!'

Source: Authors' contribution

Figure 3 captures another interaction with the Retrieval-Augmented Generation (RAG) chatbot interface built using Streamlit and powered by the LLaMA2-13B model via the Replicate API. The user has submitted a question asking for more information about famous male skaters. The chatbot responds with a well-structured list of notable figures in figure skating and extreme sports, including Yuzuru Hanyu, Nathan Chen, Yuma Kagitama, Shoma Uno, Patrick Chan, and Tony Hawk. Each entry includes a brief summary of the individual's achievements, emphasizing technical skill, artistry, and accolades. The left sidebar remains consistent with previous configurations. The model in use is LLaMA2-13B, with a low temperature value of 0.10 and a top-p value of 0.90, prioritizing accurate, deterministic output with a balance of variation. The maximum response length is set to 120 tokens, which allows the model to provide fairly detailed responses without overwhelming verbosity. The chatbot effectively delivers factual and coherent responses, reflecting the influence of the chosen parameters. Despite no external PDFs being uploaded, the model draws from its pretrained knowledge to generate informative content. This further demonstrates how the system supports accurate and structured conversations in knowledge-focused domains, making it useful for educational or informative use cases.

Figure 4 shows the PDF file with music selections and coaching teams for ice dance competitors in the ISU Grand Prix Cup of China 2024. Each listed team consists of two skaters, their coaches, and the music they've chosen for the Short Program/Rhythm Dance (SP/RD) and Free Dance (FS/FD).

Figure no. 3. Chatbot based on Llama2 – query 2

The screenshot shows the Llama 2 Chatbot interface. On the left, there's a sidebar with 'Enter Replicate API token:' and a 'Proceed to entering your prompt message!' button. Below this, 'Models and Parameters' are listed: 'Choose a Llama2 model' (Llama2-13B), 'Temperature' (0.10), 'Top-p' (0.90), and 'Max Length' (128). A 'Clear Chat History' button is at the bottom. The main chat area on the right shows a user query: 'please tell me more about famous male skaters'. The chatbot response lists five famous male skaters: Yuzuru Hanyu (Japan), Nathan Chen (USA), Yuma Kagiyama (Japan), Shoma Uno (Japan), and Patrick Chan (Canada). It also mentions 'In extreme sports, some famous male skaters include:' and lists Tony Hawk (USA).

Source: Authors' contribution

Figure no. 4. PDF file - ice dance competitors in the ISU Grand Prix Cup of China 2024

ISU GP Cup of China 2024		
Ice Dance		
Music Selection		
Marjorie LAJOIE / Zachary LAGHA		CAN
Coach	Romain Haguenauer, Marie-France Dubreuil, Pascal Denis, Patrice Lauzon, Josee Piche	
Music SP/RD	Soul Bossa Nova by Quincy Jones and His Orchestra Shining Star by Earth, Wind & Fire Boogie Wonderland by Earth, Wind & Fire & The Emotions	
Music FS/FD	The Sound of Silence by Disturbed, Paul Simon Murky Solitude by Karl Hugo - The Sound of Silence by Disturbed, Paul Simon	
Junfei REN / Jianing XING		CHN
Coach	Xun Zheng	
Music SP/RD	September by Earth, Wind & Fire Sing a Song by Earth, Wind & Fire Boogie Wonderland by Earth, Wind & Fire	
Music FS/FD	Silver Linings by Christopher Lawrence Henderson, Landon David Hook An Isolated Moment by Keith Merrill The Power of Mind by Gabriel Saban, Philippe Briand, Anne Sophie Versnaeyen Catastrophic by Adrián Berenguer Seasons by Adrián Berenguer	
Zixi XIAO / Linghao HE		CHN
Coach	Patrice Lauzon, Pascal Denis, Zhuoming Sun	
Music SP/RD	Fever by Eddie Cooley, John Davenport Hit the Road Jack by Mayfield Hit the Road Jack (remix) by Mayfield	
Music FS/FD	Overture (from "Romeo and Juliet" soundtrack) by Gerard Presgurvic Verone (from "Romeo and Juliet" soundtrack) by Gerard Presgurvic Aimer by Gerard Presgurvic O Verona (from "Romeo and Juliet" soundtrack)	
Olivia SMART / Tim DIECK		ESP
Coach	Marie-France Dubreuil, Roman Hagenauer, Patrice Lauzon	
Music SP/RD	Move Over by Janis Joplin Piece of my Heart by Big Brother, The Holding Company & Janis Joplin Black Betty by Ram Jam	
Music FS/FD	Resurrection (Dune soundtrack) by Hans Zimmer Seduction (Dune soundtrack) by Hans Zimmer House Atreides (Dune soundtrack) by Hans Zimmer A Time of Quiet Between the Storms (Dune soundtrack) by Hans Zimmer Trailer 3 Music Dune Part Two (Dune soundtrack) by Hans Zimmer The Feeling Begins by Peter Gabriel	
Julia TURKKILA / Matthias VERSLUIS		FIN
Coach	Maurizio Margaglio, Neil Brown	
Music SP/RD	I Got You (I Feel You) by James Brown Land Of 1000 Dances by Wilson Pickett. Christopher Kenner	
Music FS/FD	Bewitched by Per Störby Jutbring Obcy Astronom (Remastered 2021) by Grzegorz Ciechowski. Zbigniew Krzyskowski	

Source: <https://www.isu.org/events/isu-gp-cup-of-china-2024/>

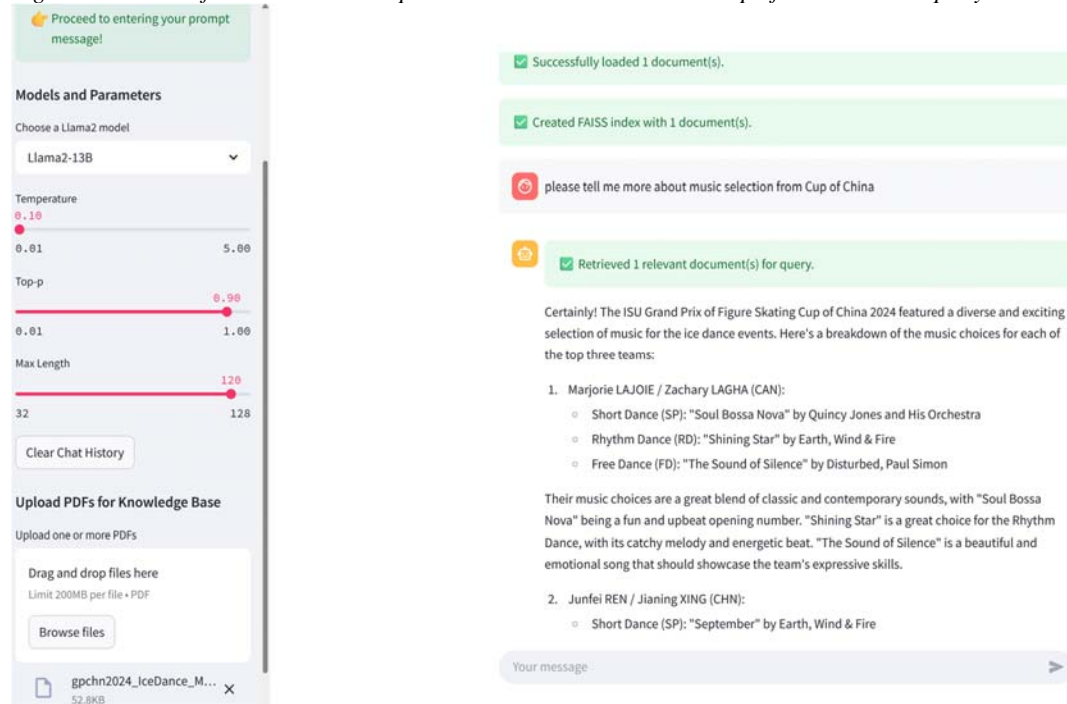
Figure 5 demonstrates a complete and successful use of the Retrieval-Augmented Generation (RAG) pipeline in the Streamlit-based chatbot application. In this case, the user uploaded a PDF document titled "gpchn2024_IceDance_Music.pdf", which was processed, embedded, and indexed using FAISS. Confirmation messages at the top of the interface indicate that the document was loaded correctly and a FAISS index was created. The user then queried the chatbot with a question about music selections from the Cup of China figure skating competition. The chatbot retrieved the relevant document content, as confirmed by the message "Retrieved 1 relevant document(s) for query" and responded with an accurate and structured summary.

The model provides detailed information about the music selections for the top ice dance teams, including:

- Marjorie Lajoie / Zachary Lagha (CAN): Music includes "Soul Bossa Nova", "Shining Star", and "The Sound of Silence".
- Junfei Ren / Jianing Xing (CHN): Music includes "September".

The chatbot not only lists the music tracks but also adds commentary on how these choices reflect each team's style and emotional expression. This context-rich response by LLM capabilities showcases the system's ability to ground its answers in uploaded document content using sentence embeddings and similarity search via FAISS. Model parameters remain consistent with earlier examples: temperature is low (0.10) to ensure factual accuracy, top-p is set to 0.90 for a slight variation in word choice, and maximum length is set to 120 tokens, enabling informative but concise outputs. Overall, this example highlights the chatbot's core functionality: combining local document knowledge (PDF file) with generative capabilities (Llama2) to provide accurate, context-aware answers to user queries.

Figure no. 5. PDF file - ice dance competitors in the ISU Grand Prix Cup of China 2024 – query 1



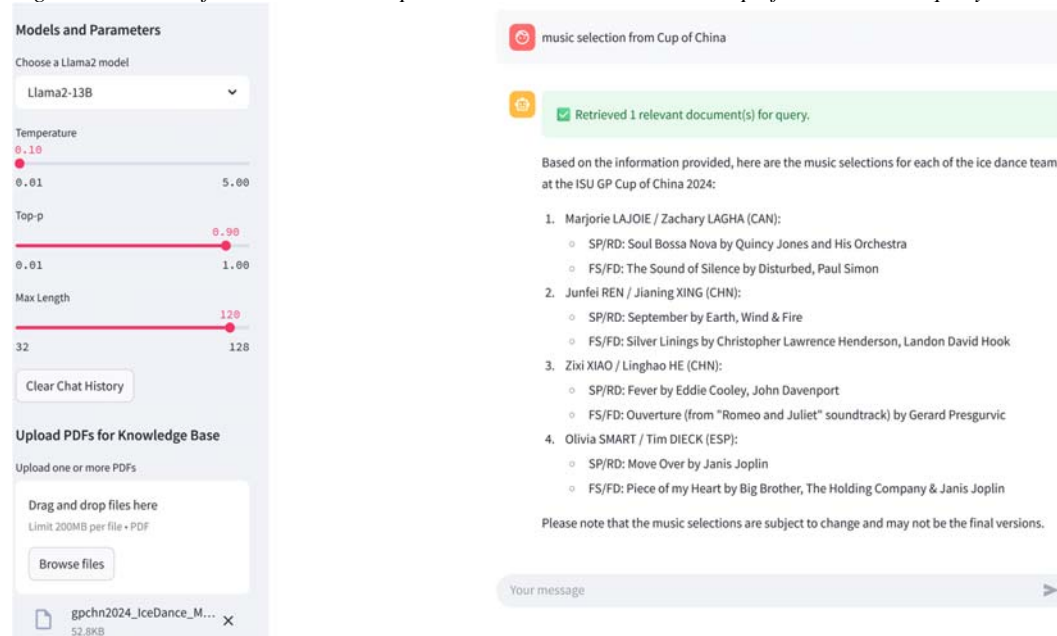
Source: Authors' contribution

Figure 6 further demonstrates the chatbot's ability to retrieve and generate responses grounded in uploaded PDF documents using a Retrieval-Augmented Generation (RAG) architecture. The user has asked for the music selection from the Cup of China, and the chatbot has correctly pulled relevant data from a previously uploaded document titled *gpchn2024_IceDance_Music.pdf*. The system confirms that it retrieved one relevant document for the query, and the chatbot then outputs a well-organized response listing the ice dance music selections for four teams participating in the ISU Grand Prix Cup of China 2024:

1. Marjorie Lajoie / Zachary Lagha (CAN)
 - SP/RD: *Soul Bossa Nova* by Quincy Jones and His Orchestra
 - FS/FD: *The Sound of Silence* by Disturbed, Paul Simon
2. Junfei Ren / Jianing Xing (CHN)
 - SP/RD: *September* by Earth, Wind & Fire
 - FS/FD: *Silver Linings* by Christopher Lawrence Henderson, Landon David Hook
3. Zixi Xiao / Linghao He (CHN)
 - SP/RD: *Fever* by Eddie Cooley, John Davenport
 - FS/FD: *Overture* (from *Romeo and Juliet* soundtrack) by Gerard Presgurvic
4. Olivia Smart / Tim Dieck (ESP)
 - SP/RD: *Move Over* by Janis Joplin
 - FS/FD: *Piece of My Heart* by Big Brother, The Holding Company & Janis Joplin

The response also includes a disclaimer noting that the music selections may be subject to change, highlighting the chatbot's ability to deliver not only facts but also contextual qualifiers from the source material. However, in this example, the LLM did not include its comments on each team music selection. As with previous examples, the chatbot is running with a LLaMA2-13B model, a temperature of 0.10, top-p of 0.90 and maximum response length of 120 tokens. These parameters continue to ensure focused, accurate, and coherent outputs suitable for knowledge-intensive tasks. Overall, this screenshot effectively demonstrates the chatbot's end-to-end RAG workflow, from document ingestion and indexing to grounded, query-specific response generation, making it a valuable tool for domain-specific question answering.

Figure no. 6. PDF file - ice dance competitors in the ISU Grand Prix Cup of China 2024 – query 2



Source: Authors' contribution

5. Conclusions

The development and deployment of the Retrieval-Augmented Generation (RAG) chatbot using Streamlit and LLaMA2 models demonstrates the effectiveness of integrating document retrieval with generative LLMs for domain-specific question answering. The chatbot is capable of intelligently responding to user queries by grounding its answers in uploaded PDF documents, which are processed using PyMuPDF for text extraction, embedded with the all-MiniLM-L6-v2 model from SentenceTransformers and indexed via FAISS for efficient similarity search. This architecture enables accurate and contextually relevant responses, making the system highly applicable for knowledge-intensive tasks.

One of the core strengths of the application lies in its flexible model configuration. Users can select between the LLaMA2-7B and LLaMA2-13B variants depending on their specific requirements. The 7B model offers faster inference and is suitable for lightweight, real-time applications, while the 13B model provides higher accuracy and deeper reasoning capabilities, albeit with increased computational demands. This flexibility supports deployment across a wide range of environments, from personal desktops to research-grade servers. The chatbot's performance is further enhanced by its configurable generation parameters, including temperature, top-p, and maximum response length. These parameters give users control over the output style, balancing between creativity and determinism. In the evaluated use cases, a low temperature (0.10), a top-p of 0.90, and a response length cap of 120 tokens were shown to deliver accurate, focused and concise outputs.

The chatbot successfully demonstrated its capabilities through multiple examples, including answering domain-specific questions related to figure skating competitions. It accurately retrieved and summarized music selections and coaching information from a PDF document about the ISU Grand Prix Cup of China 2024. In doing so, it showed the capacity to extract relevant information from structured documents and respond in a format that is both informative and contextually appropriate. Even in the absence of uploaded documents, the chatbot leverages the foundational knowledge of the underlying LLaMA2 model to deliver relevant answers, albeit potentially outdated. The proposed application validates the viability of an end-to-end RAG pipeline that combines document ingestion, embedding, vector indexing and LLM inference within a single, cohesive system.

6. Acknowledgements

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS/CCCDI - UEFISCDI, project number COFUND-CETP-SMART-LEM-1, within PNCDI IV.

7. References

- Ahn, Y. *et al.*, 2022. Retrieval-Augmented Response Generation for Knowledge-Grounded Conversation in the Wild, *IEEE Access* [Preprint]. Available at: <https://doi.org/10.1109/ACCESS.2022.3228964>.
- Ge, J. *et al.*, 2024. Development of a liver disease-specific large language model chat interface using retrieval-augmented generation, *Hepatology* [Preprint]. Available at: <https://doi.org/10.1097/hep.0000000000000834>.
- Jiang, Z. *et al.*, 2023. Active Retrieval Augmented Generation, in *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings*. Available at: <https://doi.org/10.18653/v1/2023.emnlp-main.495>.
- Joan Figuerola Hurtado, 2023. Harnessing Retrieval-Augmented Generation (RAG) for Uncovering Knowledge Gaps, *arXiv* [Preprint].
- Lewis, P. *et al.*, 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks', in *Advances in Neural Information Processing Systems*.
- Masliukhin, S.M., 2023. Dialogue system based on spoken conversations with access to an unstructured knowledge base, *Scientific and Technical Journal of Information Technologies, Mechanics and Optics* [Preprint]. Available at: <https://doi.org/10.17586/2226-1494-2023-23-1-88-95>.
- Parvez, M.R. *et al.*, 2021. Retrieval Augmented Code Generation and Summarization, in *Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*. Available at: <https://doi.org/10.18653/v1/2021.findings-emnlp.232>.
- Pichai, K., 2023. A Retrieval-Augmented Generation Based Large Language Model Benchmarked On a Novel Dataset, *Journal of Student Research* [Preprint]. Available at: <https://doi.org/10.47611/jsrhs.v12i4.6213>.
- Rackauckas, Z., 2024. Rag-Fusion: A New Take on Retrieval Augmented Generation, *International Journal on Natural Language Computing* [Preprint]. Available at: <https://doi.org/10.5121/ijnlc.2024.13103>.
- Shao, Z. *et al.*, 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy, in *Findings of the Association for Computational Linguistics: EMNLP 2023*. Available at: <https://doi.org/10.18653/v1/2023.findings-emnlp.620>.

- Siriwardhana, S. *et al.*, 2023. Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering, *Transactions of the Association for Computational Linguistics* [Preprint]. Available at: https://doi.org/10.1162/tacl_a_00530.
- Triwicaksana S, M.B. and Oktavia, T., 2023. BUILDING A RETRIEVAL-AUGMENTED GENERATION SYSTEM FOR ENHANCED STUDENT LEARNING: CASE STUDY AT PRIVATE UNIVERSITY, *Journal of Theoretical and Applied Information Technology* [Preprint].
- Wang, Calvin *et al.*, 2024. Potential for GPT Technology to Optimize Future Clinical Decision-Making Using Retrieval-Augmented Generation, *Annals of Biomedical Engineering* [Preprint]. Available at: <https://doi.org/10.1007/s10439-023-03327-6>.