Extended Abstract: Enhancing Accessibility to MODTRAN Documentation: A Chatbot Framework Using Retrieval-Augmented Generation (RAG)

Zarin T. Shejuti
Department of Computer Science
Winston-Salem State University
Winston-Salem, USA
zshejuti123@rams.wssu.edu

Debzani Deb
Department of Computer Science
Winston-Salem State University
Winston-Salem, USA
debd@wssu.edu

Emily R. Dunkel
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, USA
emily.dunkel@jpl.nasa.gov

***Abstract*—This work presents the development of a chatbot using Retrieval-Augmented Generation (RAG) to streamline access to MODerate resolution atmospheric TRANsmission (MODTRAN)'s extensive documentation. By combining document retrieval with a Large Language Model (LLM), the chatbot delivers accurate, context-aware answers to user queries. Key resources, including the MODTRAN6 User Manual, Algorithm Theoretic Basis Document (ATBD), and the frequently asked questions section from the MODTRAN website, were processed into a searchable vector database. The evaluation showed that the chatbot effectively provides satisfactory and accurate responses, though occasional extraneous information highlights the need for refinement.**

***Keywords*—MODTRAN, OpenAI, Langchain, Chatbot**

I. Introduction

MODTRAN (MODerate resolution atmospheric TRANsmission) [1] is a renowned atmospheric radiative transfer model used extensively for simulating the propagation of electromagnetic radiation through the Earth's atmosphere. Our research focuses on developing a chatbot specifically designed for MODTRAN users. The objective is to leverage a Large Language Model (LLM) to enable efficient and intuitive access to the MODTRAN documentation, including its user manual. Manually searching through extensive documentation can be time-consuming and tedious, which our chatbot aims to address by providing accurate, context-aware responses to user queries.

To achieve this, we utilized Retrieval-Augmented Generation (RAG) [2], a cutting-edge approach that combines document retrieval with generative capabilities. The MODTRAN documents used in the RAG pipeline include the MODTRAN6 User Manual, the MODTRAN6 Algorithm Theoretic Basis Document (ATBD), and an HTML page containing the MODTRAN FAQ, processed into a vector database by extracting and splitting the text into manageable chunks, generating embeddings using OpenAI models [3], and indexing them with LangChain [4], a powerful framework for building LLM-powered applications. Users can query this database for instant and precise information retrieval. This combination ensures the chatbot delivers reliable and dynamic interactions, streamlining the process of obtaining information from MODTRAN's comprehensive resources.

II. Methodology

The methodology for developing the MODTRAN chatbot is structured into several key steps: i) Document selection, ii) Data preparation, iii) Embedding generation, iv) Vector database creation, and v) Retrieval-Augmented Generation (RAG) Pipeline.

A. Document Selection

The first step in developing the MODTRAN chatbot was to identify the key resources that would serve as the knowledge base for the system. The documents selected include the MODTRAN6 User Manual, the Algorithm Theoretic Basis Document (ATBD), and the MODTRAN FAQ page. These resources were chosen because they encompass comprehensive information about MODTRAN, ranging from technical details and theoretical foundations to practical usage and common questions.

B. Data Preparation

The selected documents were then processed to extract and clean their text content. For the MODTRAN6 User Manual and the ATBD, which are provided in PDF format, *PyPDF2* [5] was used to extract text while handling formatting inconsistencies. The MODTRAN FAQ, available as an HTML page, was parsed using *BeautifulSoup*, allowing the text to be extracted while preserving its semantic structure. Once the text was extracted, it was divided into manageable chunks using LangChain's *CharacterTextSplitter* [6]. Each chunk was limited to 1000 characters, with an overlap of 200 characters.

C. Embedding Generation

After preparing the data, the text chunks were transformed into vector representations using HuggingFace's *SciBERT-NLI* model. This model was chosen for its efficiency and effectiveness in handling scientific documents. By encoding the text into embeddings, the system ensures optimized retrieval during the query process.[7]

D. Vector Database Creation

The generated embeddings were indexed in a *FAISS* (Facebook AI Similarity Search) vector database. Metadata such as the source document and the chunk index were also stored alongside the embeddings to facilitate traceability.

SoutheastCon 2025 | 979-8-3315-0484-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/SOUTHEASTCON56624.2025.10971575

	1. How do you call the MODTRAN GUI from the command line?	**2. If you want to run MODTRAN from the command line using an input json file, what is the command?**	**3. What file includes the full radiance data?**	**4. What file includes the radiance data convolved with a user-supplied SRF?**	**5. What range of solar zenith angle are allowed?**
Answers provided	mod6gui	mod6c_cons	tp7	chn	0 to 180
Answers generated by LLM	To call the MODTRAN GUI from the command line, you can use the command: <modtran_directory>/bin/linux/**mod6gui**	To run MODTRAN from the command line using an input json file, the command is: <modtran6_directory>/bin/linux/**mod6c_cons** <ROOTNAME>.json.	The file that includes the full radiance data is the **.tp7** file.	The file that includes the radiance data convolved with a user-supplied SRF is the **.tp6** file.	The range of solar zenith angle allowed in MODTRAN is from **0 to 180** degrees.

Fig. 1. Queries with Corresponding Expert and LLM Answers

E. *Retrieval-Augmented Generation (RAG) Pipeline*

The RAG pipeline integrates retrieval and generative capabilities into a seamless workflow. When a user submits a query, the vector database retrieves the most relevant text chunks using similarity scoring. The top *k* chunks (*k=13*) are then passed to an LLM through LangChain's *load_qa_chain*. The LLM synthesizes the retrieved information to generate a coherent and contextually accurate response.

III. Results and Discussion

A set of query-and-answer pairs was generated by a MODTRAN expert. These queries were then run through the LLM to evaluate the type and quality of results it generated. The answers produced by the LLM were compared against the expert-provided answers, and the generated responses were sent back to the expert for evaluation. A summary of the analysis is presented in Figure 1, which lists the queries, the corresponding answers provided by the expert, and the answers generated by the LLM.

The answers generated by the LLM are mostly accurate and relevant compared to the expert-provided answers. For the first question, the LLM provides a more detailed response by including the full path to the command, though the core answer, *mod6gui*, matches the expert's answer. Similarly, for the second question, the LLM correctly identifies *mod6c_cons* as the command to run MODTRAN from the command line using a JSON file but provides additional clarification on the expected file structure, which enhances the response. The third answer is completely accurate as *.tp7* is indeed the file containing the full radiance data. However, for the fourth question, the expert indicates *chn* as the correct file, while the LLM states *.tp6*, which introduces a discrepancy and suggests a possible inaccuracy in the LLM's response. Finally, the LLM correctly states that the range of solar zenith angles allowed in MODTRAN is *0 to 180* degrees, consistent with the expert's response. Overall, the LLM provides detailed and mostly accurate responses, though the inconsistency in the fourth question requires verification.

We also ran the queries through ChatGPT for comparison, noting that its responses were verbose but lacked specificity, whereas our model provided concise answers, better identifying specific keywords and aligning with expert-provided answers.

IV. Conclusion

In this work, we developed a chatbot utilizing Retrieval-Augmented Generation (RAG) to make accessing the extensive and complex MODTRAN documentation more efficient. By combining document retrieval techniques with a Large Language Model (LLM), the chatbot provides accurate and context-aware answers to user queries. Our evaluation demonstrated that the chatbot produces satisfactory and accurate responses for most queries, effectively meeting user needs. Future improvements include refining retrieval methods, optimizing embeddings, and enabling the LLM to evaluate its own responses for better accuracy.

Acknowledgment

This research was supported by the NASA MUREP/DEAP IMPACT (Institute for Multi-agent Perception through Advanced Cyber-Physical Technologies) Award, No. 80NSSC23M0054. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

[1] "Spectral Sciences Inc.," 2016. [Online]. Available: http://modtran.spectral.com/modtran_about. [Accessed Jan'25].

[2] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." Available: https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[3] "OpenAI," [Online]. Available: https://platform.openai.com/docs/guides/embeddings. [Accessed Dec24].

[4] "LangChain," [Online]. Available: https://python.langchain.com/v0.1/docs/get_started/introduction. [Accessed Dec'24].

[5] "The Python Package Index," 2022 31 December. [Online]. Available: https://pypi.org/project/PyPDF2/. [Accessed Jan'25].

[6] "LangChain," 09 December 2024. [Online]. Available: https://api.python.langchain.com/en/latest/character/langchain_text_splitters.character.CharacterTextSplitter.html. [Accessed Dec'24].

[7] B. Nilesh et al., "Strengthening Sentence Similarity Identification Through OpenAI Embeddings and Deep Learning," IJACSA) International Journal of Advanced Computer Science and Applications, vol. 15, no. 4, 2024, Accessed: Jan. 31, 2025. [Online]. Available: https://thesai.org/Downloads/Volume15No4/Paper_85-Strengthening_Sentence_Similarity_Identification.pdf