Paper IC08

AI-Enhanced Chatbot for Streamlined Clinical Trials Analysis and Document Management

Jaime Yan, Merck & Co., Inc., Rahway, NJ, USA
Chao Su, Merck & Co., Inc., Rahway, NJ, USA
Changhong Shi, Merck & Co., Inc., Rahway, NJ, USA

ABSTRACT

The rapid escalation in data volume demands effective utilization and safeguarding of confidential documents, such as FDA guidelines, SOPs, and protocols. We introduce a chatbot framework employing the open-source Large Language Model (LLM) from Hugging Face and LangChain/Vector database technique to streamline summary extraction and rapid searches in private documents through natural language inquiries. Additionally, we fine tune the LLM with domain-specific knowledge, empowering the framework to adapt to various tasks, leverage existing company macro libraries, and generate SAS® code; this elevates data usage efficiency and facilitates SAS® code tasks while guaranteeing data security and privacy. The framework provides additional privacy protection by operating locally. This groundbreaking approach addresses the industry's requirement for proficient data management and privacy conservation, presenting a novel, valuable, and flexible solution. In conclusion, implementing a chatbot framework using LLM and LangChain/Vector database optimizes data utilization, supports SAS® code tasks, and fortifies data privacy.

INTRODUCTION

In the realm of clinical trial research, the role of statistical programming cannot be overstated. It is crucial for achieving precise and consistent analysis and interpretation of data. This field primarily encompasses two significant activities: the creation of standardized datasets that comply with CDISC standards, specifically SDTM and ADaM, and the development of Tables, Listings, and Figures (TLFs) tailored to the specific requirements of study protocols or mock-ups.

One of the main challenges in dataset creation lies in enabling programmers to quickly understand and apply these standards, ensuring accurate data extraction from documentation. Traditionally, this has involved discussions and sharing of knowledge through academic papers. In terms of TLF generation, there is a push towards automation, such as creating user interfaces for selecting conditions or using macros to interpret conditions from Excel files to produce TLFs. Yet, the automation of extracting information directly from mock-ups is still a work in progress.

To address these pivotal challenges, this paper introduces innovative solutions aimed at:

- Employing Large Language Models (LLMs) in a secure manner to offer domain-specific insights or to extract information from sensitive documents.
- Facilitating the exploration and analysis of CDISC standard datasets through simple natural language queries.
- Developing a comprehensive system for auto-generating TLFs from mock templates, significantly improving efficiency and precision.

Through these initiatives, the paper seeks to foster further development of these solutions in an open, collaborative environment across the industry, ultimately enhancing the efficacy and reliability of statistical programming in clinical research.

Gaps to directly using pre-trained LLMs

The deployment of pre-trained Large Language Models (LLMs) within pharmaceutical statistical programming presents three notable challenges:

Domain Knowledge: Out-of-the-box LLMs often lack the nuanced understanding necessary for pharmaceutical statistical programming tasks. This includes a grasp of complex regulatory standards and the ability to interpret clinical data accurately — critical for developing compliant and impactful research materials.

Security Concerns: The use of LLMs for processing and generating answers from data files can pose security risks, particularly with confidential clinical trial data. Upholding data protection and adhering to privacy laws such as HIPAA are of utmost importance within the healthcare sector.

Industry Optimization: Unmodified LLMs may not fully exploit their capabilities nor cater to the unique requirements of the pharmaceutical industry, such as integrating with a company's macro library for seamless end-to-end TLF reporting. This could lead to less-than-ideal performance and an inability to meet the specific needs of pharmaceutical statistical programming.

Proposed Solutions:

We envision the amalgamation of Large Language Models (LLMs) with techniques such as Retrieval-Augmented Generation (RAG), Parameter-Efficient Fine-Tuning, and SAS® kernel into a unified chatbot interface. This integration is poised to revolutionize the daily tasks of statistical programmers by offering solutions to the following key challenges:

Review the Document:

Method: Utilizing advanced RAG, optimized for PDF and other formats, we enhance the LLMs' contextual comprehension, facilitating the precise retrieval of relevant information. This proves especially beneficial when working with intricate ADaM dataset structures like ADSL, BDS, and OCCDS.

Benefit: The process significantly streamlines key point identification and content summarization, enabling direct interaction with the document, and aids in navigating and interpreting a wide array of pharmaceutical datasets and research papers.

Explore the Data (ADaM Data Analysis):

Method: Extract schema of each ADaM dataset from spec, use LLM to generate the PROC SQL query as source to RAG process. Then by augmenting RAG with a SAS® kernel, we establish seamless integration between LLMs and SAS®'s robust statistical analysis capabilities, creating a powerful tool for data exploration.

Benefit: This empowers users to efficiently tackle most tasks through natural language dialogue, substantially decreasing the time required to comprehend and analyze ADaM data structures.

Generate the Report:

Method: A meticulously crafted SAS® reporting macro system, in sync with a finely tuned LLM, taps into the pattern recognition abilities of LLMs to align mock templates with SAS® macros, thereby streamlining TLF generation.

Benefit: The system enables an end-to-end automated report generation workflow that rapidly transforms mock-ups into detailed reports, significantly boosting the automation, efficiency, and regulatory compliance of the reporting process.

By implementing these methods, we aim to enhance the LLMs' functionality for pharmaceutical statistical programming, overcoming prevalent hurdles and customizing the technology for the specific needs of the industry. This paper will focus on shedding light on the RAG and fine-tuning processes, with a lesser emphasis on deployment and security aspects. Our objective is to demonstrate how advanced retrieval techniques and model fine-tuning can forge more sophisticated LLM applications in pharmaceutical research, thus raising the bar for statistical programming and reporting standards within the industry.

Chatbot Framework

Our chatbot framework, powered by Retrieval-Augmented Generation (RAG) and fine-tuning techniques, is designed to deliver precise and contextually relevant responses within the highly specialized domain of pharmaceutical statistical programming. Here's how the framework is structured:

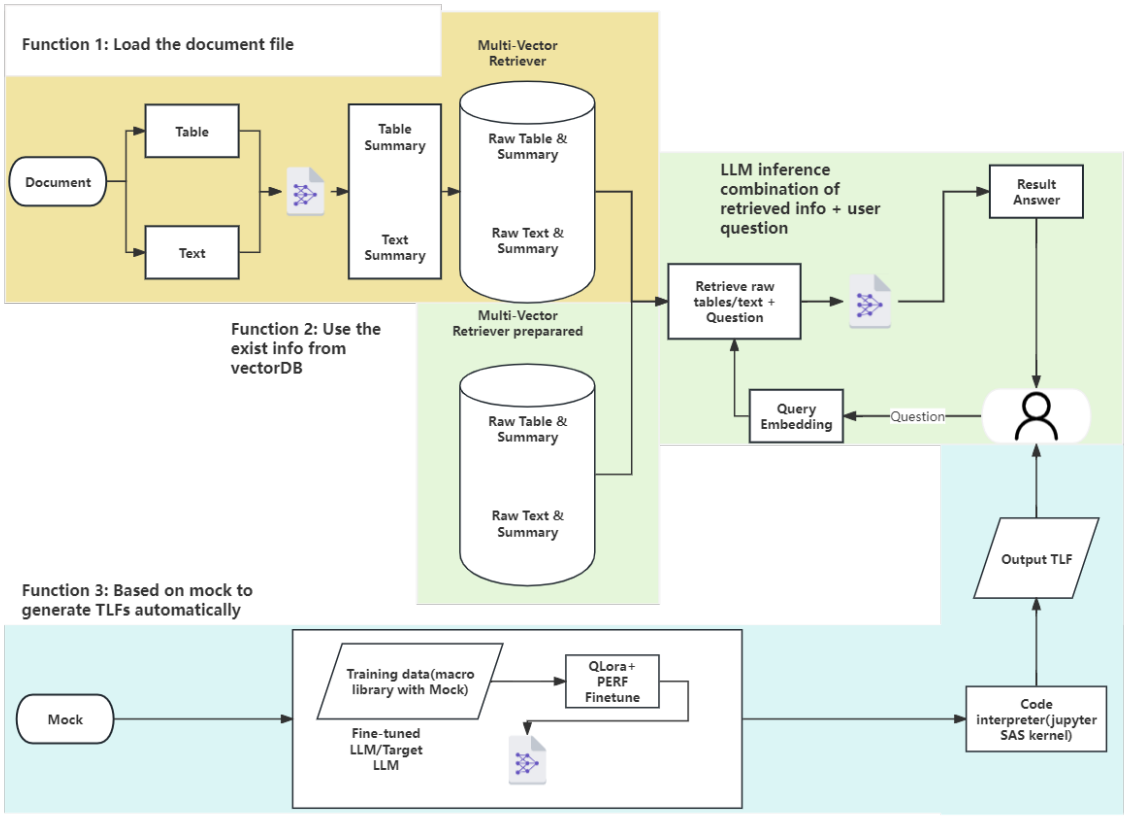

Figure 1. Chatbot workflow and design

Front-End:
User Interaction Module:
The chatbot interface offers two modes of interaction:

- *Document Upload*: Users can upload documents which are then processed as described above. Subsequent questions from the user prompt the retrieval of related information, with the LLM providing the answers.
- *Direct Query*: Users may also directly query the vector database. The system retrieves the relevant information based on the question and the LLM offers answers infused with domain expertise.

User Interface Options:
The front-end UI may be developed with tools like Gradio UI or Streamlit, designed to ensure a straightforward and engaging user experience.

Back-End:
Document Processing Module:
This module encompasses PROC SQL query generation using JSON-formatted ADaM dataset schema inputs for the LLM to generate query pairs. It also involves document embedding, where documents are segmented into tabular and textual parts. A pre-trained LLM summarizes these sections, aiding a multi-vector retriever like Langchain in indexing both original and summarized content.

SAS® Code Interpreter Module:
Operates as the back-end, employing a JUPYTER SAS® kernel to handle the generated or retrieved code from the vector database, returning the output or result.

End-to-End TLF Generation Module:
This component inputs a mock template and uses a fine-tuned LLM to interpret it and generate SAS® macro calls. These calls are processed in a SAS® kernel, outputting TLFs to a specified location. The system iteratively adjusts the code based on any errors or warnings encountered.

Model Selection and Deployment:

For the RAG Pipeline and TLF Generation, a selection of models from platforms like Hugging Face is made, depending on the deployment environment. A fine-tuned 7b LLM model, supported by high-quality training data and a robust SAS® macro reporting system, is used for TLF generation.
Deployment may involve hosting on cloud services with API access or local operation alongside a SAS® kernel, depending on computational resources and security considerations.

This chatbot framework is a convergence of advanced AI techniques with domain-specific knowledge, aiming to revolutionize the way pharmaceutical statistical programming tasks are performed. By integrating RAG for content retrieval and employing fine-tuning for task-specific LLMs, the framework promises to enhance the accuracy and efficiency of TLF generation, while ensuring a seamless user experience through intuitive interfaces.

Enhancing Domain Knowledge in LLM with Retrieval-Augmented Generation

Integrating Retrieval-Augmented Generation (RAG) into Large Language Models (LLMs) marks a significant stride towards specialized proficiency in statistical programming. By utilizing RAG, we enhance LLMs with the ability to access a wealth of domain knowledge, drawing from established frameworks like Llamindex or Langchain. This integration is crucial in two key areas:

Curated Content for Statistical Programming:
The RAG pipeline incorporates a handpicked selection of resources pivotal to statistical programming, ensuring that the content is not only authoritative but also highly relevant. This collection includes pharmaceutical research papers, CDISC standards, and custom PROC SQL queries that are aligned with the ADaM/SDTM data structures. By interfacing with an integrated SAS® kernel, this system facilitates immediate and accurate analysis, delivering essential support for industry professionals.

- Processing Semi-structured Data Within Document: We adopt a sophisticated approach to managing semi-structured data, treating tabular and textual elements distinctly. Specialized packages for handling unstructured data are employed to efficiently process and integrate varied data types within the RAG framework.

- Data Generation for Exploration and Analysis: For data generation, a high-performance LLM is utilized to create PROC SQL queries. The process begins with Python code extracting schema details from standard ADaM specifications, formatted in JSON. These schemas are then paired with custom prompts that instruct the LLM to generate corresponding questions and PROC SQL query pairs. This methodology uses the combined data schema and prompts as inputs for the LLM to infer and produce datasets tailored for user queries.


```
{
    "ADAE": {
        "Structure": "one record per subject per each AE recorded in SDTM AE domain",
        "Variables": [
            {
                "VARIABLE": "STUDYID",
                "LABEL": "Study Identifier",
                "DATA TYPE": "text",
                "Codelist": "ADSL.STUDYID",
                "Code": "Study Identifier",
                "Terms": [
                    {
                        "Order": 1.0,
                        "Term": "MKxxxx",
                        "Decoded Value": null
                    }
                ]
            },
            {
                "VARIABLE": "USUBJID",
                "LABEL": "Unique Subject Identifier",
                "DATA TYPE": "text",
                "Codelist": null,
                "Code": null,
                "Terms": [
                    {
                        "Order": null,
                        "Term": null,
                        "Decoded Value": null
                    }
                ]
            },
```

```
{
  "question": "How many subjects in the study experienced any adverse event?",
  "code": "proc sql; select count(distinct usubjid) as AE_Count from ADAE where AEFL = 'Y'; quit;"
},
```


Figure 2. Sample ADAE schema and question pair snip with JSON format

Crafting Effective Prompts for Retrieval:
Crafting well-structured prompts is vital for successful information retrieval. These prompts are formulated to match the retrieved content and are subsequently paired with user queries. The aim is to direct the LLM to provide responses that are both contextually relevant and technically sound, utilizing the information extracted from the RAG system. These prompts act as conduits, merging user questions with the retrieved data to enable the LLM to generate outputs that are accurate and informatively rich.

Through these focused efforts, RAG significantly enhances LLMs, equipping them to become advanced instruments for the intricate requirements of pharmaceutical statistical programming. This strategy ensures LLMs not only gain a deeper grasp of specialized content but also enhance their interaction with diverse data structures essential to the field.

Fine-Tuning LLM for End-to-End TLF Reporting Generation

The refinement of Large Language Models (LLMs) to generate end-to-end Tables, Listings, and Figures (TLFs) reporting requires a strategic redesign of the macro library to be LLM-compatible. This section outlines the approach for fine-tuning LLMs to interface effectively with macro libraries and produce executable code:

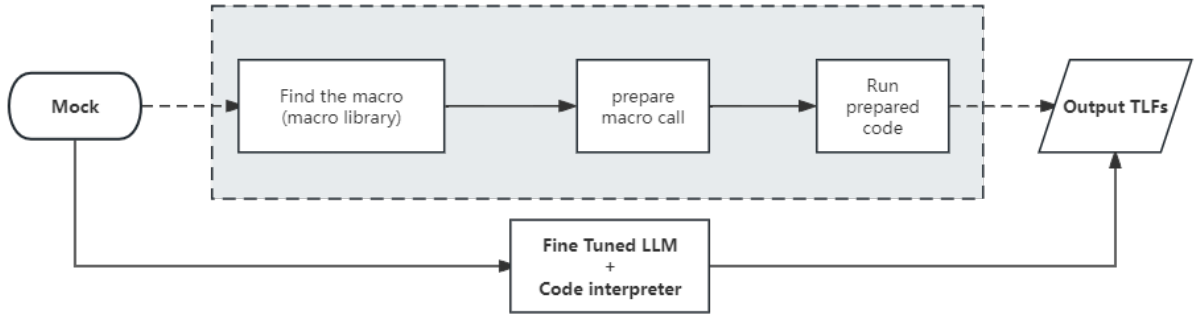

Figure 3. LLM empowered end to end TLFs generator workflow

Macro Library Compatibility:
The macro library must be restructured to facilitate interaction with LLMs. Macros that are inherently more compatible with LLM architectures, such as those that function modularly and allow for dynamic report system design, will be prioritized. These include macros that map directly to the mock structure, allowing for a seamless mock-to-macro relationship.

Data Importance in Fine-Tuning:
The fine-tuning process, possibly utilizing frameworks like Lora, emphasizes the critical role of data. Not just any data, but data that effectively reflects the reporting system's requirements. Here, the focus is not merely on the volume of data but on its quality and structure. For LLMs to generate accurate reporting systems, the design and encapsulation of the macro library are paramount.

Pattern Learning and Inference:
A core aspect of fine-tuning involves teaching the LLM to discern and replicate the patterns between mock templates and SAS® macro calls. If a rough mapping relationship can be established, the LLM can be fine-tuned to recognize these patterns and generate corresponding runable code efficiently.

In essence, the performance of a fine-tuned LLM in generating TLFs will depend on the intricacies of the macro library design, the dataset used for fine-tuning, and the specificity of hyperparameter settings, which are often determined on a case-by-case basis. The library design should facilitate a clear understanding of the mock-to-macro translation process for the LLM. Moreover, the quantity and representativeness of the records used for fine-tuning can significantly impact the model's ability to generalize and perform accurately when generating TLFs.

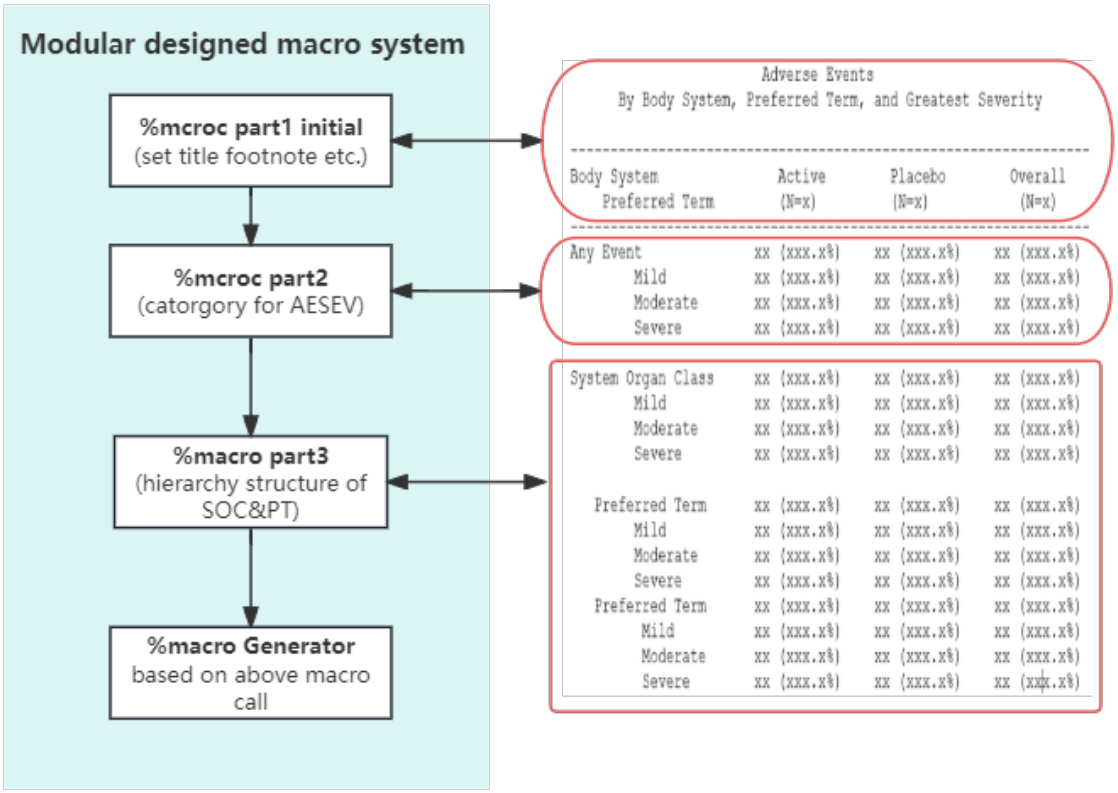

Figure 4. Design SAS® macro report macro call with mapping relationship with mock module

Ultimately, the synergy between a well-designed macro library and a finely tuned LLM holds the potential to revolutionize TLF reporting systems, making them more efficient and responsive to the dynamic needs of pharmaceutical statistical programming.

CONCLUSION

In conclusion, integrating Retrieval-Augmented Generation (RAG) and fine-tuning into a chatbot framework marks a significant advancement for pharmaceutical statistical programming. This innovative approach not only delivers precise and context-aware responses but also simplifies the extraction and utilization of complex data through user-friendly interfaces and automated TLF generation. Tailoring Large Language Models to specific tasks and deploying them flexibly across cloud or local servers showcases the framework's adaptability. This strategic application of AI is poised to enhance the accuracy, efficiency, and accessibility of pharmaceutical research, heralding a new era of industry standards.

Moving forward, our focus is on enhancing our AI solution through industry-wide collaboration, aiming to broaden its capabilities with insights from diverse professionals in pharmaceutical and statistical programming. Our goal is to advance AI's role in pharmaceutical research, setting new standards for efficiency and effectiveness. This collaborative approach encourages idea sharing, ensuring that AI and statistical programming advancements benefit the entire industry.

REFERENCES

1. Jeff Cheng, Shunbing Zhao, Guowei Wu, and Suhas R. Sanjee(2023) Automated Mockup Table and Metadata Generator https://www.lexjansen.com/pharmasug/2023/SD/PharmaSUG-2023-SD-230.pdf
2. Chase, H. (2022). LangChain. Retrieved from https://github.com/langchain-ai/langchain (Accessed on [01/10/2024]).
3. Busa, B., Murugesan P., Malcolm S. (2020) “Automation of TFL Generation using CDISC 360 Enriched Metadata” October 7th CDISC US Interchange. Retrieved from https://www.cdisc.org/sites/default/files/2021-08/Automation_of_TFL_Generation_using_CDISC_360_Enriched_Metadata.pdf

4. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., ... Rush, A. M. (2020). HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771. Available at: https://arxiv.org/abs/1910.03771
5. Clinical Data Interchange Standards Consortium, Inc. CDISC Analysis Data Model Team. Analysis data model (ADaM) version 2.1. 2009. Accessed April 19, 2022. https://www.cdisc.org/standards/foundational/adam
6. Clinical Data Interchange Standards Consortium, Inc. CDISC Analysis Data Model Team. Analysis data model implementation guide version 1.2. 2019. Accessed April 19, 2022. https://www.cdisc.org/standards/foundational/adam

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jaime, Yan	Chao Su,	Changhong Shi
	+1 (732) 5946459	+1 (732) 5941383
mingyu.yan1@merck.com	chao.su@merck.com	changhong_shi@merck.com