First Progress Report - *TCM Sage*

1. Report Objective

This report aims to demonstrate the progress of the project *TCM-Sage*, to showcase the basic research that has been done, the optimized system architecture compared to the initial one mentioned in the project statement, the achievements so far, and the plan for the upcoming stage.

2. Proposed system solution

This project aims to develop a proof finding and summarizing tool for TCM practitioners, to offer them suggestions based on classical texts, and offer source-able citation, to enrich their confidence in giving suggestions to the patients.

- **Core Architecture**: System will be based on **Modular RAG** Structure. The reason for choosing it is because of its flexibility and scalability, able to handle the complexity in the domain of Traditional Chinese Medicine(TCM).
- Key Components
  - Hybrid Retriever: This component aims to solve the unclear term definition issue in TCM together with Vector Search, as one character may contain multiple meanings in ancient Chinese.
  - Knowledge Base: The classical TCM text Huang Di Nei Jing（黃帝內經）is selected for the MVP as it is comprehensive yet not overly complex, which makes it good to be a starting point. The text has been processed and stored in a ChromaDB vector storage.
  - Reflective Generator: This design is inspired by the Self-RAG framework, aiming to build trust via a double "glass box" mechanism instead of the usual "black box" to ensure the answer is factually correct and source-traceable, hence gaining trust from the TCM practitioner.
    1. Controllable Inference (Query Routing): Before the prompt is being sent to the actual processing model, a smaller model will first be used to determine the clinical severity of the user's query. For query with lower severity, the model will then pass it to the actual processing model with a higher temperature setting to generate a more flexible and detailed answer; for query with high severity, the query will be passed to the bigger model with absolute 0 temperature to ensure the answer is trustworthy.
    2. Self-Critique (Answer Validation): After processing user's query and generated answer, the main model will then self critic its answer, check its own output for factual support([ISSUP]) and provide a citation that links to the original text for user's manual verification.
  - Changes comparing to the original statement: More advanced and state-of-art technique are used(i.e. Self-RAG, the Knowledge Graph(KG) used in MedRAG) to optimize the initial idea, and gained extra inspiration from the bi-weekly meeting(make the citation traceable, allowing users to click on the citation and being brought to the

original text being cited).

3. Proposed schedule & Achieved schedule

In short, the progress are going as expected, with the first stage(research, scoping, and design) completed, moving on to the next stage.

- Literature Review: Reviewed 4 literature related to RAG, RAG's usage in TCM, and some RAG framework.
  - Literature reviewed related to RAG:
    - Retrieval-Augmented Generation for Large Language Models: A Survey
    - A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions.
  - Literature reviewed related to Self-RAG:
    - SELF-RAG: LEARNING TO RETRIEVE, GENERATE, AND CRITIQUE THROUGH SELF-REFLECTION.
  - Literature reviewed related to KG-RAG Implication in TCM:
    - MedRAG: Enhancing Retrieval-augmented Generation with Knowledge Graph-Elicited Reasoning for Healthcare Copilot
- **MVP and Architecture Design**: Detailed documentation created and stored in GitHub repo for reference.
- **Data processing**: Full text of *Huang Di Nei Jing(黃帝內經)* downloaded, cleaned, chunked, and stored into ChromaDB.
- **Environment Setting**: All necessary and related Python environment(i.e. LangChain, ChromaDB, SentenceTransformers) are configured with requirements updated in GitHub repo.

4. Problems with the project

- **Data source**: Many data found online are either incomplete, inconsistent, or combined with modern translation which may lead to misunderstanding. To ensure the consistency, a script was built to do the data cleansing, removing unnecessary formatting, modern translation, and any other useless information to ensure the data integrity.
- Technical Challenge:

  Response Latency: During the meeting with supervisor, latency issue is brought, which vast amount of text like Huang Di Nei Jing(黃帝內經) may lead to significant delay in retrieving source and generating reply, which may cause a unhappy user experience(UX). Attempts will be made to solve this issue using ChromaDB Vector Storage; its fast similarity search performance will not be impacted by the original file size.

  - Citation Accuracy: The meeting also brought up the importance of accurate citation, which allow user to go to the original cited text with one click, can significantly improve user's confidence on the answer, and allow human-in-the-loop to reduce the risk of hallucination in Artificial Intelligence(AI). This will be attempted to solve by adding source metadata(e.g. chapter name) in chunks, and prompt the AI to include the metadata when

answering the query.

5. Engaging activities

The project is transiting to stage 2(Core App Development) from stage 1(research, scoping, design, and data preparation). The upcoming task will be building the core logic, which is a step towards a functional MVP.

Core Logic for the First Implementation Trail:

1. Load the created vector database.
2. Accept a user's query.
3. Perform a retrieval.
4. Generate a response backed database.