Report Skeleton - Building a Clinical Depression AI Assistant Based on RAG

Ai Ka* **Jiao*** **Renan*** **Yiyang***

1 Summary

Encyclopedia is of great use, but imagine carrying an encyclopedia around with you every day. Heavy? Not portable? People might say, "Google it!". True! That could be a solution. But tons of information online, how can we be sure that the information we get after searching is the one with authority? For example, someone is suffering from some symptoms and needs the correct medication. The consequences are unbearable if the medication is not delicately accurate. People might also say "just go to the doctors, then". Of course! But doctors are humans too! We can't expect the doctors to remember everything from medical school textbooks. Of course, they can google too, but wouldn't a more specialized and well-confirmed question and answer information bank be more useful in this case? Just like a recipe for the doctors.
With this thought, we set out to build our Retrieval Augmented Generation (RAG) system called "TherapyRecipe" to come to the rescue of this issue. It has a highly specialized information bank for precise searches in depression symptoms, medications, and medication procedures. There are mainly information retrieval and generator sections in this system, with other supporting sections, for example, data processing and evaluation. The generator is a large language model that generates answers with proper prompts.
To dive into the details, keep reading with interest.

2 Data

The data that serves as the information database is the paper *"Canadian Network for Mood and Anxiety Treatments (CANMAT) 2023 Update on Clinical Guidelines for Management of Major Depressive Disorder in Adults: Réseau canadien pour les traitements de l'humeur et de l'anxiété (CANMAT) 2023: Mise à jour des lignes directrices cliniques pour la prise en charge du trouble dépressif majeur chez les adultes"* by Lam *et al.* (2024). The data mainly contains clinical guidelines for managing depressive disorders.
Since the data is in the format of a research paper, we performed data wrangling to build an organized database. Our chosen format is a JSON file. We extracted the texts directly into structured chunks with the following categories:

- `text`
- `metadata`
- `section`
- `type`
- `headings`
- `referenced_tables`
- `referee_id`
- `chunk_id`

The paper also includes images and tables, for which we applied both vision-language model parsing and manual human interpretation. The finalized file can be found in our GitHub repository under the name `guideline_db.json`.
Besides the research paper as the database, 33 question-and-answer pairs were provided as the gold standards for evaluation purposes. Following the sample question-and-answer pairs, we self-generated 99 similar pairs as our silver data for more evaluation practice. Below are the detailed steps of processing the data:

1. **Save HTML page**
    - Download and save the HTML page of the article.
2. **Text Parsing**

*Equally contributed authors.

- Delete the head and tail of the file:
  - Start from the title.
  - Delete the sections after `## Conclusion` (from supplementary materials).
  - Delete the French abstract.
- Observe and identify the link of each section.
- Extract <h1>, <h2>, <h3> tags to identify the hierarchy of the article structure.
- Get headings, concatenate all the headings together, and prepend them to the actual paragraph texts.
- Abbreviation substitution: Detect all abbreviations in the text and replace them with their full definitions.
- Detect what tables/figures are mentioned in the texts.

3. **Analyze Special Elements in the Text**

- **Table image**
  - Tables embedded in the page are accessed via links. Get the links from the HTML source page.
  - Grab the link and send it to the Vision model (LLaMA-11B). Convert the table into a natural language description.
  - There are many duplications added to improve model understanding.
  - Green circle images represent Level 1, 2, 3, 4. These pictures are saved as links in the HTML. Search for these links and replace them with plain text.
  - Extract the table name and store it in the `referee_id` field.
- **HTML table**
  - Detect the HTML structure of the table and parse it manually using custom Python code.
  - Extract the table name and store it in the `referee_id` field.
- Manually merge figure captions that appear below images.
- Some paragraphs inside a <box> are split across multiple chunks. Merge them into one chunk.
  - Longest of these chunks is around 250 words (approximately 350 tokens, which is acceptable).

Sample schema of the JSON structure database:

```
{
 "text": "The results are summarized in Table 2...",
  "embedding": [...],
  "metadata": {
    "section": "Results",
    "type": "paragraph",
    "chunk_id": 42,
    "headings": "Abstract/Background",
    "referenced_tables": ["table_2"],
    "referee_id": "table_2"
  }
}
```

3 Methods

Figure 1: Pipeline of the RAG system: Document Preprocessing, Retriever, Generator, Output

As shown above, the system follows this workflow pattern. In the previous section, we mentioned data wrangling, which is one part of the document preprocessing step. Another important part of preprocessing is converting text into vector embeddings. A suitable embedder is key in this step.
We experimented with a variety of embedding models. The ones we tested include:

- `sentence-transformers/all-MiniLM-L6-v2`
- `BAAI/bge-large-en-v1.5`
- `jinaai/jina-embeddings-v3`
- `abhinand/MedEmbed-large-v0.1`
- `NeuML/pubmedbert-base-embeddings`
- `emilyalsentzer/Bio_ClinicalBERT`
- `pritamdeka/S-PubMedBert-MS-MARCO`
- `microsoft/BiomedNLP-PubMedBERT-base-uncased-abstr`

Among them, `abhinand/MedEmbed-large-v0.1` and `emilyalsentzer/Bio_ClinicalBERT` performed the best, while `MiniLM-L6-v2` served as the baseline and fallback option. The web interface allows users to select from different embedder options.

The last major component in document processing is image parsing. Arrows and icons in a picture or table are the challenging part. We have tried Llama 4, Llama 3.2, and Donut to parse, and they show edges in different areas. Parsing by vision-language models solely cannot solve the complicated components in an image or table; therefore, human parsing was in place eventually.

The retriever performs the search function once a query is typed. The length of the retrieved results can be customized. Considering the answers in the golden sets are only one or two words, which are not ideal for evaluation metrics, we customized the length of the retrieved results so that they look more like the real answers. These answers are also what we used as the golden answers when evaluating.

Similar to embedders, we also tried different large language models as the generator. The models we have tried are Llama-3.3-70B-Instruct, meta-llama/Meta-Llama-3-8B, GPT-4o, and mistralai/Mistral-7B-Instruct-v0. Answers from Llama models are a bit lengthy, not as concise as Mistral. GPT-4o generally has good answers, but with more responses like" I don't know". Prompt engineering was conducted for better results. Few-shot learning was found to be useful for a higher accuracy score.

4 Evaluation

Large language model evaluation and human evaluation are both in place for evaluation, based on the same evaluation metrics as shown below:

The large language model evaluation is based on the AlignScore in the SummaC metric, and the human evaluation is conducted by a specialist from the partner's team. With the combimaniton of jinaai/jina-embeddings-v3 as the embedder and meta-llama/Llama-3.3-70B-Instruct as the generator, we have the following evaluation results. The large language model evaluation with GPT-4o shows an an average of 3.18 on faculty/accuracy, 3.15 on completeness, 4.36 on safe and ethical, and 4.15 on clinical applicability/ generalization/practicality. The overall score in percentage is 74.24%. The human evaluation shows an average of 3.94 on faculty/accuracy, 3.76 on completeness, 4.79 on safe and ethical, and 4.42 on clinical applicability/ generalization/practicality. The overall score in percentage is 84.55%.

5 Analysis

Summarize system strengths and weaknesses based on evaluation results, identify what kinds of problems the model handles well and under which conditions it still makes errors (e.g., lack of context, difficulty understanding terminology). Analyze error examples and outline areas for potential improvement.

6 Future Work

Content Overview: List directions we haven't completed but are worth exploring, such as fine-tuning the embedding model and building a user feedback loop. Also include suggestions for future improvements by the partner organization.

7 Conclusion

Concisely revisit the project goals, system implementation, and key outcomes, highlighting our contribution to building a clinical AI assistant and its practical value.

A Appendix A: Project Timeline

List the 6-week project progress and division of labor among team members.

Week	Person	Tasks
Week 1	Ai Ka Agatha Lee	Submitted to Canvas; Schedule
Week 1	Jiao Zeng	Submitted to Canvas; Introduction; Data
Week 1	Yiyang Du	Submitted to Canvas; Deliverables; Methods
Week 2	Ai Ka Agatha Lee	Document table parsing; HTML & image tables; Make slides (Thu); Schedule
Week 2	Jiao Zeng	Build JSON structure; Link text and tables; Script (Thu); Introduction & Data
Week 2	Rennan Wang	Join AWS; Propose embedding model; Image table; Research embedder; Deliver presentation
Week 2	Yiyang Du	Join AWS; Zoom setup; Implement & test search; Presentation outline; Methods
Week 3	Everyone	Team check-in; Meeting with Takeoff; Embedding test; Read code; Table result $\rightarrow$ DB; Abbreviations & references; Script & slides
Week 4	Ai Ka Agatha Lee	Continue table parsing
Week 4	Jiao Zeng	Finish search function; Query rewrite
Week 4	Rennan Wang	Prompt engineering
Week 4	Yiyang Du	Develop UI
Week 4	Everyone	Contact Yash; Template to JJ; Question pair evaluation; UI development; Model selection; Table image work; History function; Abbreviation replacement
Week 5	Everyone	Friday slides; Update Streamlit frontend; Design evaluation metrics; Finish table parsing; Develop UI; RAG backend integration; Graph.json merge; Query results $\rightarrow$ Markdown; Answer streaming; Evaluation tool; Understand frontend code; BLEU/ROUGE research; Evaluate GPT answers; Kayoung evaluates GPT questions
Week 6	Everyone	Review blog post
Week 6	Jiao Zeng	Sort QA pairs; Kayoung evaluates GPT QA; Update evaluation site; Extract scores; Query rewrite (optional); Try Ollma (optional)
Week 6	Rennan Wang	Write report & skeleton; Make initial presentation slides
Week 6	Yiyang Du	Write report & skeleton; Write blog; LLM memory feature

References

Lam, R. W., Kennedy, S. H., Adams, C., Bahji, A., Beaulieu, S., Bhat, V., ... & Milev, R. V. (2024). Canadian Network for Mood and Anxiety Treatments (CANMAT) 2023 Update on Clinical Guidelines for Management of Major Depressive Disorder in Adults: Réseau canadien pour les traitements de l'humeur et de l'anxiété (CANMAT) 2023: Mise à jour des lignes directrices cliniques pour la prise en charge du trouble dépressif majeur chez les adultes. The Canadian Journal of Psychiatry, 69(9), 641-687.