Medical RAG Chatbot for Respiratory Infection Insights

Problem Statement

Our project aims to develop a command-line chatbot leveraging Retrieval-Augmented Generation (RAG). It will assist doctors by providing evidence-based insights for diagnosing respiratory infections, such as pneumonia, influenza, and bronchitis, based on patient symptoms. The chatbot will retrieve relevant information from a curated medical knowledge base and generate concise suggestions such as: *"Symptoms may indicate pneumonia; consider chest X-ray"*. With this chatbot, we aim to enhance medical diagnostic accuracy, speed, and confidence. The system will not provide diagnoses but will serve as a decision-support tool, delivering contextual information to support medical professionals in their practice.

Ethical and Social Impact

This project engages with significant ethical and social considerations inherent in medical AI applications. Ethically, the chatbot is designed to avoid direct diagnoses, incorporating a mandatory disclaimer in every generated response to clearly distinguish AI-provided support from actual medical decision-making. This disclaimer will be included in all outputs to prevent misuse and ensure it remains a supportive tool rather than a replacement for professional judgment. By focusing on insights, it respects the expertise of doctors and mitigates the risk of over-reliance on AI. Data privacy is prioritized through the exclusive use of public, de-identified datasets, safeguarding patient confidentiality.

Socially, the chatbot has the potential to improve access to medical knowledge in under-resourced healthcare settings, where doctors may lack immediate access to updated guidelines. Summarizing complex medical texts into actionable insights promotes equitable healthcare, particularly for primary care providers serving diverse populations. We will carefully curate a diverse and authoritative knowledge base to ensure fair and reliable outputs.

Expected Learning Outcome

Through this project, we will gain hands-on experience with RAG, practice the integration of vector-based retrieval and large language model (LLM) generation tailored for medical natural language processing (NLP). The project will deepen our understanding of preprocessing and normalization of medical texts, particularly in handling domain-specific jargon. We expect to develop proficiency with pre-trained models, such as BioBERT (for embedding) and LLaMA 3.2:1B-Instruct (for generation).

Additionally, we will learn to design ethical AI systems, especially for medical applications, by implementing safety disclaimers and rigorously evaluating output relevance. Collaboration will be enhanced through task division, splitting responsibilities between data curation and model integration, and using GitHub for version control. Finally, we will employ evaluation techniques for RAG systems, applying both manual scoring and retrieval metrics to assess performance.

Data Set

The chatbot will rely on a filtered subset of the PubMed QA dataset, accessible via HuggingFace (1), which contains approximately 211,000 Q&A pairs derived from PubMed abstracts. We will filter this dataset for respiratory infections, including pneumonia, influenza, and bronchitis, using keywords such as “pneumonia,” “influenza,” “cough,” and “fever,” to yield 1,000-2,000 relevant pairs. The QA contexts will serve as the knowledge base for retrieval.

Technical Approach

Our RAG-based chatbot will be implemented in Python. We will begin by preprocessing the PubMed QA dataset to ensure consistency and relevance. This involves normalizing texts by converting them to lowercase, removing punctuation, while preserving medical terms with BioBERT’s tokenizer. Texts will be chunked into ~500-token segments using langchain.text_splitter to facilitate efficient retrieval.

For the retrieval component, we will embed ~2,000-4,000 chunks using BioBERT (2) via the HuggingFace transformers library. BioBERT’s pre-training on biomedical texts, such as PubMed abstracts, ensures high relevancy by capturing nuanced medical semantics, making it ideal for retrieving precise contexts for respiratory infection queries. This domain specificity enhances the chatbot’s ability to provide accurate insights, aligning with our goal of supporting doctors effectively. The resulting embeddings will be indexed using faiss-gpu cosine similarity search, retrieving the top-3 most relevant texts per query. Subword tokenization, inherent to BioBERT, will preserve subword nuances in the medical jargon, further boosting retrieval quality.

For generation, we will employ LLaMA 3.2:1B-Instruct (3), a 1-billion-parameter instruction-tuned model accessed via Ollama. LLaMA 3.2:1B-Instruct is best suited for our project due to its alignment for dialogue and task-specific responses, ensuring coherent, medical-toned insights that directly address queries. Its instruction-tuning enhances safety by guiding the model to follow strict response formats and avoid speculative outputs. Our prompt will include a mandatory disclaimer: *"Based on {context}, provide insights for a doctor about symptoms: {query}. Include: 'This is not a diagnosis; consult a physician.'"* This safety feature is critical for ethical medical AI, preventing misinterpretation as a diagnostic tool. The model's efficiency fits comfortably within our compute power, enabling local execution. Prompt engineering techniques will be applied to guide the model's responses, promoting consistency, contextual relevance, and adherence to medical tone.

The RAG pipeline will be integrated using langchain, combining retrieval and generation into a cohesive system. A simple chat interface built with Python's Streamlit library will allow users to enter symptom queries (e.g., "cough, fever") and receive generated insights. This approach ensures a streamlined, local solution optimized for our hardware and timeline.

Evaluation Metrics

Retrieval Evaluation

To evaluate the retrieval performance of our RAG system, we will use multiple established metrics implemented via the HuggingFace evaluate library. Specifically:

- **Recall@K**: Measures whether at least one relevant document appears in the top K retrieved contexts. We will compute Recall@3 with a target score > 0.8.
- **Precision@K**: Evaluates the proportion of retrieved documents that are relevant in the top K documents. We will use Precision@3 to quantify contextual relevance.
- **Mean Reciprocal Rank (MRR)**: Captures how highly ranked the first retrieved document is, averaged across queries. Higher MRR indicates better prioritization of useful context.

These metrics will be computed by comparing the predicted retrieved document IDs (from FAISS) with reference document IDs known to contain the correct answers, derived from the filtered PubMedQA dataset.

We will use ~30 manually verified queries to ensure accurate evaluation. Retrieval outputs will be validated using HuggingFace's evaluate.load("retrieval_recall"), retrieval_precision, and mean_reciprocal_rank tools.

Generation Evaluation

To quantify the quality of the chatbot's responses, we will use automated NLP metrics including:

- **BLEU**: Measures n-gram overlap with a reference answer. This ensures that the chatbot's responses include key medical terms and phrases drawn directly from authoritative sources.
- **ROUGE-L**: Captures longest common subsequence between generated and reference answers. It is particularly useful for verifying that the chatbot preserves the structure and sequence of clinically relevant information when summarizing evidence.
- **BERTScore**: Computes semantic similarity using contextual embeddings from a pre-trained language model. This is ideal for evaluating whether the chatbot's responses capture the correct meaning and nuance of medical insights, even if phrased differently.

These scores will be calculated using the HuggingFace evaluate library. Reference outputs will be constructed from PubMedQA answers or rewritten to reflect expert-style medical insight.

References

1. PubMed QA Dataset: http://huggingface.co/datasets/qiaojin/PubMedQA
2. BioBERT Embedding Model: https://huggingface.co/dmis-lab/biobert-v1.1
3. Llama 3.2-1B-Instruct Generative Model: https://ollama.com/library/llama3.2