Approach 2: Retrieval-Augmented Generation (RAG)

Rahul Manikandan

1. Approach 2: Retrieval-Augmented Generation (RAG) with DeepSeek-R1-Distill-Qwen

This approach combines semantic retrieval via FAISS with a compact, local language model for claim validation. The goal is to identify relevant clinical evidence from multiple PDF sources and confirm whether it supports the marketing claim.

1.1 Overview

- **Claim Input:** Marketing claims provided in JSON format.
- **Clinical Corpus:** Clinical PDFs are processed into paragraph, table, and OCR-based text using `preprocess.py`.
- **Embedding:** All clinical content is embedded using `sentence-transformers/paraphrase-MiniLM-`
- **Vector Store:** FAISS is used to store normalized embeddings, allowing fast top-k similarity retrieval.
- **LLM Evaluation:** Retrieved candidates are passed to a locally loaded HuggingFace pipeline using the model `deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B`.

1.2 Implementation Details

The pipeline is implemented in the `main(RAG).ipynb` notebook. We utilize the HuggingFace `pipeline` API to run the DeepSeek model locally on CPU:

```
from transformers import pipeline

model_id = "deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B"
llm_pipeline = pipeline("text-generation", model=model_id, max_length=256, devi
```

For each claim:

1. The top-k similar clinical excerpts are retrieved from the FAISS index.
2. These snippets, along with the claim, are passed to the LLM in a structured prompt.
3. The LLM infers the most relevant evidence and outputs supporting matches in a structured JSON format.

1.3 Advantages

- Fully local and open-source — no reliance on external APIs.
- Lightweight model (1.5B parameters) suitable for CPU inference.
- Output is interpretable and directly usable in downstream applications.

1.4 Limitations

- No `supports: true/false` flags or justification reasoning.
- Effectiveness highly depends on retrieval quality and prompt formatting.
- The model may miss borderline or implicit matches.

1.5 Output Format

The final output is saved in `results/rag_results.json`, containing a list of claims, each mapped to selected evidence excerpts and their source documents.

1.6 Performance Summary

This method provided improved semantic relevance compared to TF-IDF, with matches exhibiting moderate confidence. The DeepSeek-R1 model handled multi-paragraph inputs effectively and ran efficiently on CPU. While lacking explicit reasoning, the approach serves as a strong balance between simplicity, performance, and local deployment feasibility.