Fine-Tuned RAG Chatbot with Streaming Responses – Amlgo Labs

Submitted by: Md Asher
Role Applied: Junior AI Engineer – Amlgo Labs

---

1. Document Preprocessing & Chunking Strategy

Initial Input: A legal document (~10,500+ words) in PDF format.

Cleaning Steps:

- Removed headers, footers, special characters, and HTML tags.
- Normalized line breaks for sentence continuity.

Chunking Approach:

- Used LangChain's RecursiveCharacterTextSplitter.
- Chunk size: ~1000 characters, with 200-character overlap.
- Sentence-aware splitting to maintain coherence.

Output:

- ~100 semantic chunks saved in /chunks/
- Used as basis for vector retrieval.

2. Embedding Model & Vector Database

Embedding Model:

- **Model:** all-MiniLM-L6-v2 (384D embeddings)
- **Reason:** Lightweight, accurate for semantic search.

Vector DB:

- **Tool:** FAISS
- **Storage:** /vectordb/
- **Retrieval Method:** Cosine similarity
- **Top K:** 100 chunks retrieved per query

3. Language Model & Prompt Design

LLM Used: TinyLlama (1.1B parameters), run locally via **Ollama**

Why TinyLlama?

- Fast responses (~10s)
- CPU-friendly
- Small memory footprint

Prompt Template:

You are a helpful AI assistant. Answer the user query strictly based on the context below:

<context>

[retrieved chunks]

</context>

Query: [user_query]

Answer:

Purpose:

- Prevent hallucinations
- Keep answers grounded in document

4. Streamlit UI Deployment (Chatbot)

Tech Stack: Python + Streamlit + Ollama

Features Implemented:

- Chat layout using st.chat_input and st.chat_message
- Real-time token streaming
- "Thinking..." spinner shown during response generation
- Source chunks displayed in an expandable panel
- **Mixed formatting:** Responses use both paragraph and bullet points
- **Sidebar added:**
    - Current model in use (TinyLlama via Ollama)
    - Number of indexed chunks (~100)
    - Clear chat button with st.rerun() support

User Experience:

- UI loads in <3 seconds after first run
- Response time: ~10–12 seconds on CPU
- Smooth streaming token-by-token

5. Sample Questions & Responses

Q1: ***What happens if the user violates the terms?***

Response (Formatted):

- If a user violates the terms in the User Agreement, actions may include:
    - Account suspension
    - Listing removal
    - Legal proceedings

Paragraph Context:
Users are also prohibited from scraping, using bots, or bypassing security measures.

Q2: ***Can the company change the terms?***

- Yes. The platform may unilaterally amend the terms with notice.

Q3: ***Are bots allowed to access the services?***

- No. The use of automated tools is strictly forbidden in the agreement.

Limitations

Failure Cases:

- "What is the refund policy for electronics?" → Not present in document.
- "How many users opt for arbitration?" → Stats not mentioned.

6. Challenges & Limitations

Key Challenges:

- Handling long context with small LLMs
- Preventing hallucinations
- Keeping UI responsive despite streaming

Mitigation:

- Used strict prompt templates
- Controlled chunk overlap
- Optimized retrieval with MiniLM embeddings

Final Thoughts & Future Scope

Successes:

- Full RAG pipeline with:
  - Chunking
  - Embedding
  - Retrieval
  - LLM-based generation
- Local deployment with fast inference
- Clean UI, real-time streaming

Future Additions:

- Model dropdown (Mistral, Zephyr, etc.)
- Upload PDF & live process
- Active learning from user feedback
- Filter chunks based on metadata or section headings

Developed by: *Md Asher*
Email: iamasher786@gmail.com
GitHub: github.com/iamasher
Contact: +91 9334358522