AI-Powered Document-Grounded QA Chatbot: Technical Report

1. Introduction

This report details the technical implementation and performance of a Document-Grounded Question Answering (QA) Chatbot. It leverages Retrieval-Augmented Generation (RAG) principles to provide accurate, context-aware responses from a specific document. A key focus was ensuring high factual accuracy and mitigating hallucination, especially under CPU-only hardware constraints. This report outlines the architecture, implementation choices, and observed performance, including solutions to significant challenges.

2. Document Structure and Chunking Logic

The chatbot's knowledge base is the **"AI Training Document.pdf" (eBay User Agreement)**, a 19-page legal document.

- **Objective**: To break the document into small, semantically coherent chunks for effective retrieval and to fit within the LLM's context window.
- **Methodology**: A **sentence-aware splitter** was used.
    - **Max Token Length**: Chunks were limited to **120 tokens** for concise context.
    - **Overlap**: A **20% overlap** (approx. 24 tokens) maintained continuity across chunk boundaries.
- **Output**: This yielded **98 distinct chunks**, stored in chunks/AI Training Document.jsonl.

3. Embedding Model and Vector Database

For semantic search, an embedding model and vector database were utilized.

- **Embedding Model**: **BAAI/bge-small-en-v1.5** (384-dimensional) was chosen for its performance and CPU compatibility. Embeddings are **cosine-normalized**.
- **Vector Database (Index)**: A **FAISS IndexFlatIP** was used for efficient similarity search of the 98 embeddings.
- **Storage**: The index is stored as vectordb/index.faiss.

4. Prompt Format and Generation Logic

The generation component uses a locally hosted LLM with a meticulously crafted prompt strategy.

- **LLM Used**: **Phi-3-mini-4k-instruct-q4.gguf** (3.82 billion parameters, 4-bit Q4_K_M GGUF) run via llama-cpp-python. This was critical for memory optimization on constrained hardware.
- **Prompt Format (Phi-3 Instruction Template)**: The prompt adheres to Phi-3's specific chat template <s><|user|>\n{message}<|end|>\n<|assistant|>\n. System instructions and context are embedded within the user message to maximize instruction adherence.
- **System Prompt for Grounding and Conciseness**:
- You are a highly accurate, concise, and helpful assistant.
- Your responses MUST be based ONLY on the provided Context.
- Do NOT use any external knowledge or make up information.
- If the answer is not explicitly found in the provided Context, you MUST respond with: "I cannot find the answer in the provided context."
- Provide answers directly and concisely. For questions asking for lists (e.g., "What does X cover?"), present the answer as a clear, numbered list.
- Avoid any introductory phrases, conversational filler, or apologies.
- Be direct and to the point.

This explicitly enforces context grounding, controls verbosity, and dictates output formatting.

- **Generation Parameters**:
    - max_tokens=512: Limits response length.
    - stream=True: Enables token-by-token streaming.
    - stop=["<|end|>", "</s>"]: Ensures generation ceases at end-of-sequence tokens.
    - temperature=0.0, top_p=1, top_k=0: Ensure deterministic and consistent output.

5. Example Queries and Responses

These examples demonstrate the chatbot's performance, highlighting success and mitigation of previous failure cases:

Success Case 1: Grounded Information Extraction & Formatting

- **Query**: "What does the Introduction cover?"

- **Expected from Document**: Specifics about the eBay User Agreement's scope, contracting entities, dispute resolution provisions, and the Agreement to Arbitrate.
- **Observed Response**:
    1. The scope of the User Agreement, including Mobile Application Terms of Use and other policies.
    2. Agreement to comply with terms when using eBay Services.
    3. The specific eBay entity you are contracting with, based on your residence.
    4. Advice to read the agreement carefully due to provisions on claim resolution, disclaimer of warranties, limitation of liability, and legal disputes.
    5. Highlights an Agreement to Arbitrate, requiring claims to be submitted to binding arbitration unless opted out, and waiving rights to class action or jury trials.
- **Evaluation**: **Success**. Accurate, document-direct, and formatted as a numbered list.

Success Case 2: Specific Detail Retrieval

- **Query**: "What does eBay do if a seller fails to provide a shipping label?"
- **Expected from Document**: eBay may charge the seller for an eBay-generated label if they fail to provide one.
- **Observed Response**: "If a seller fails to send the buyer a return shipping label, eBay may generate and provide a label instead, and then charge the seller for the cost of that eBay-generated label."
- **Evaluation**: **Success**. Precise, accurate, and directly extracted.

Mitigated Failure Case: Overcoming Generic Responses / Verbosity

- **Previous Problem**: Model hallucinated generic academic introduction and was verbose/repetitive.
- **Resolution**: Iterative refinement of the system prompt to enforce context grounding, concise output, and specific formatting successfully guided the model to accurate, document-specific responses.

Failure Case Example (Expected Behavior for Out-of-Context Questions)

- **Query**: "What is the capital of France?"

- **Expected Response (based on grounding prompt)**: "I cannot find the answer in the provided context."
- **Evaluation**: **Success (as a grounding mechanism)**. Demonstrates correct identification of out-of-context queries.

6. Notes on Hallucinations, Model Limitations, and Slow Responses

Developing this RAG system on constrained local hardware presented significant challenges.

- **Hallucinations Mitigation**: Achieved through **aggressive prompt engineering**, strictly instructing the model to *only* use provided context and explicitly fall back to "I cannot find the answer in the provided context."
- **Model Limitations on Hardware**: The **system's memory (8GB RAM, 2GB VRAM on NVIDIA MX330 GPU)** was the primary bottleneck.
    - **Segmentation fault Issues**: Initial attempts to load open-source LLMs (even 4-bit quantized via transformers) consistently resulted in Segmentation fault errors due to insufficient VRAM/RAM.
    - **Solution: GGUF and llama-cpp-python**: This combination provided superior memory efficiency, allowing the 3.82B Phi-3-mini model (~2.23 GB disk size) to load into system RAM and run entirely on the CPU (n_gpu_layers=0).
- **Slow Responses**: Running on CPU-only results in slower inference.
    - **Observed Latency**: First responses take **6-8 seconds**; subsequent responses are **2-3 seconds**. This latency is a bottleneck for real-time interaction.

7. Conclusion

The InfoStream-AI Chatbot successfully demonstrates accurate, document-grounded RAG, mitigating hallucinations despite significant hardware limitations. The strategic choice of the Phi-3-mini GGUF model with llama-cpp-python and meticulous prompt engineering proved to be a functional and robust solution. Future improvements should prioritize hardware upgrades (8GB+ VRAM GPU) or migration to cloud-based LLM APIs for faster inference and larger model support.