Product Overview and Target User

Our RAG system streamlines competitive research for insurance industry financial analysts by enabling efficient analysis of 10K reports and strategic initiatives across multiple carriers. The system specifically targets experienced financial analysts specializing in competitive intelligence and strategic planning who require rapid insights from complex financial documents.

Target User Profile

Financial analysts in the insurance industry are highly specialized professionals who navigate complex regulatory filings with precision. They rely on established competitive research frameworks to assess market trends, benchmark competitors, and extract insights from vast amounts of financial data including publicly available financial fillings such as 10K reports. Given the iterative nature of their work, these analysts frequently refine their analyses, requiring tools that preserve context across multiple research sessions. Operating in time-sensitive environments, they must ensure their findings are both reliable and actionable.

Target User Workflow

Step 1: Document Collection - Analysts gather 10-K reports from multiple sources, a straightforward process not requiring optimization.
Step 2: Information Extraction - currently, analysts manually review documents, requiring intense focus with risk of missing key information.
Step 3: Analysis - processing data to identify patterns demands deep expertise.
Step 4: Insight Synthesis - converting insights into structured findings often leads to "blank page syndrome." AI-assisted generation will streamline this step. We expect that step 2 to step 4 will be iteratively repeated until the user is satisfied with the results.
Step 5: Report Formatting - standardizing deliverables is a lower-priority concern not addressed in initial implementation.

The RAG system will focus on Stages 2-4, addressing key challenges in data extraction, analysis, and insight synthesis, ensuring analysts can derive competitive insights accurately and swiftly. The iteration of the target steps is captured in the form of the user iteratively prompt engineering.

Technical Decisions and Tradeoffs

Model Architecture

Considering the Jupyter Notebook interface, a unified GPT-4 API approach is implemented across the entire pipeline, rather than specialized models for different stages. While specialized models could optimize individual components and reduce costs, the unified approach simplifies development and maintenance. The consistency and reliability of GPT-4 outweigh the additional API costs for this financial analysis PoC. Additionally GPT-4 was chosen for its reliability and unified workflow, which can result in higher token usage costs. As usage scales, we plan to explore model caching or hybrid approaches with smaller specialized models (e.g., Llama 2 or GPT-3.5) for certain tasks.

Document Processing

Among PyPDF2, PDFPlumber, and Adobe PDF Services API options, PDFPlumber was selected for its superior handling of financial tables, crucial for MD&A and Financial Statement extraction. While Adobe's API offers higher accuracy and PyPDF2 provides faster processing, PDFPlumber provides the optimal balance of precision and implementation simplicity.

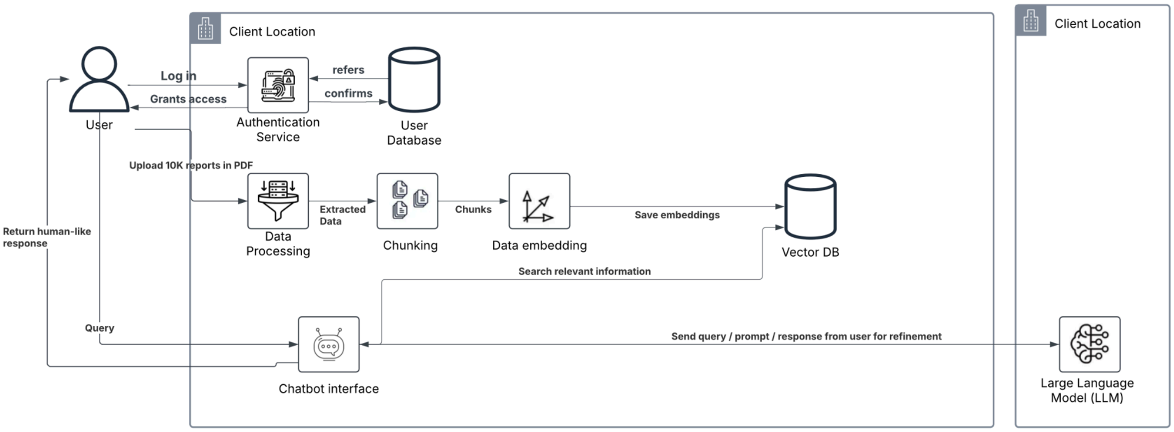
Chunking Strategy
A hybrid approach is implemented instead of pure fixed-size or semantic chunking. This method preserves financial table integrity while dividing narrative sections into 500-1000 token chunks. While pure semantic chunking would offer better context preservation, the hybrid approach reduces complexity while maintaining accuracy for financial analysis.

Vector Storage
Local FAISS implementation with L2 distance metrics was chosen over cloud solutions like Pinecone or Weaviate. Though cloud solutions offer better scalability, FAISS provides adequate functionality without external dependencies. L2 distance metrics were selected over cosine similarity for better financial content performance.

Context Management
An in-memory Python dictionary tracks recent exchanges instead of a persistent database solution. While this sacrifices long-term persistence, it provides sufficient functionality while avoiding database complexity. The system maintains 3-4 previous interactions, balancing context preservation with simplicity.

Success Metrics - To be addressed in a separate deck.

System Architecture Diagram



- Authentication Layer secures access by validating user credentials against a database through a client-side interface, ensuring only authorized users can interact with the system.
- Data Processing Pipeline manages document ingestion, transformation, and storage. PDFs are uploaded, text is extracted and cleaned, then segmented into structured chunks. These chunks are embedded as vector representations and stored in a searchable knowledge base for efficient retrieval.
- Query and Response System powers user interaction through a chatbot. It retrieves relevant data from the vector database, assembles context, and leverages an LLM to generate accurate and coherent responses.

Additional Considerations

Context Window Size: GPT-4 may truncate responses if too many large sections are appended.

Hallucination Risk: LLMs can fabricate data. A future plan is to implement numeric cross-checking with the original PDF tables.

Multi-Carrier Depth: Currently, the retrieval logic occasionally focuses on only one carrier if chunk overlaps are ambiguous. We will refine FAISS labeling to ensure more balanced retrieval across multiple carriers.