Reducing Hallucination in Financial AI Through Retrieval-Augmented Agentic Design

Abstract

This project presents **Project B.A.N.A.N.A.** (**B**rokerage **A**nalytic **N**etwork & **A**utonomous **N**ode **A**ssistant), a production-grade multi-agent platform designed for autonomous financial research and personalized recommendation workflows. The system integrates **Retrieval-Augmented Generation (RAG)** to ingest unstructured financial data (SEC filings, transcripts) and **Fine-tuned Transformer models** to deliver high-precision sentiment analysis and domain-specific content generation. Orchestrated via **LangGraph**, specialized agents leverage the **Model Context Protocol (MCP)** to interact with real-time market tools and data servers. The architecture achieves sub-second latency for complex reasoning tasks and ensures explainability through a state-aware agentic workflow.

Modern financial ecosystems generate large volumes of structured and unstructured data, including corporate filings, technical indicators, and social sentiment signals. Human analysts struggle to synthesize these signals in real time. Meanwhile, generic large language models (LLMs) frequently produce hallucinated or ungrounded outputs when operating without domain constraints.

BANANA addresses these limitations through the integration of Retrieval-Augmented Generation (RAG), a fine-tuned financial transformer (FinBERT), and an autonomous multi-agent workflow with confidence gating.

Layer 1: The Outer Peel (Data Ingestion & MCP Gateway)

This layer handles the connectivity to the outside world through standardized **MCP Servers**.

- **SEC-Edgar MCP:** Standardized retrieval of 10-K and 10-Q filings.
- **Market-Data MCP:** Live price feeds and historical OHLC data.
- **Social-Sentiment MCP:** Real-time streams from financial news and social media.
- **Functionality:** Normalizes disparate data formats into a unified JSON-RPC structure for agent consumption.

Layer 2: The Pulp (Agentic Orchestration & RAG Pipeline)

The "reasoning" layer is orchestrated by **LangGraph**, where specialized agents operate within a shared state machine.

- **The Researcher Agent (RAG Specialist):** When a query is received, this agent performs a semantic search against the Vector Core. It retrieves specific "chunks" of text (e.g., "Company Risks" from a 10-K) to augment the generation process.
- **The Analyst Agent (Fine-tuned Specialist):** Uses a fine-tuned RoBERTa model to classify sentiment and detect financial entities with high precision.
- **The Scribe Agent:** Merges the RAG context with sentiment scores to generate a structured financial blog post ("The Daily Slip").

Layer 3: The Core (Vector Store & Fine-tuned Backbone)

The intelligence backbone of the system.

- **Vector Database (Pinecone/Milvus):** Stores document embeddings generated by a pre-trained transformer. This is the source for all RAG operations.
- **Fine-tuning Strategy:** * **Domain Adaptation:** The base model is fine-tuned on the FiQA (Financial Question Answering) and SST-2 datasets to improve accuracy in financial contexts.
    - **RLHF (Reinforcement Learning from Human Feedback):** The Scribe agent's output is refined through RLHF to ensure the generated blogs match the tone and reliability of professional analysts.

Operational Workflow

1. **Query Entrance:** The user asks: *"Is the latest Tesla 10-K suggesting growth or risk?"*
2. **Task Decomposition:** The LangGraph **Orchestrator** splits the task:
    - *Task A:* Fetch the 10-K via **SEC MCP**.
    - *Task B:* Perform **RAG** semantic search on growth/risk sections.
3. **Context Synthesis:** The **Researcher Agent** retrieves the relevant chunks from the **Vector Core**.
4. **Inference:** The **Analyst Agent** processes the retrieved text using the **Fine-tuned model** to calculate a "Risk vs. Growth" score.
5. **Generation:** The **Scribe Agent** synthesizes all findings into a personalized response.