PATHWAY MIDTERM REPORT

Team 84

ABSTRACT

This report presents our progress in developing a multi-agent Retrieval-Augmented Generation (RAG) system for efficient financial data processing. Designed to manage complex reasoning tasks, our architecture balances retrieval and contextual understanding with modules optimized for dynamic data changes. Our approach includes a hybrid RAG model incorporating LightRAG and GraphRAG for improved data access, and integrates FinGPT—a language model fine-tuned for financial terminology. Additionally, our system includes schedulers, guardrails, and fallback mechanisms to ensure robustness. Initial tests using financial datasets indicate enhanced data retrieval and processing accuracy, showcasing the system's potential for reliable financial data analysis.

Keywords: *financial data; RAG; FinGPT; LightRAG; multi-agent system.*

1. APPROACH OUTLINE

1.1. Problem Understanding

We are working to develop a multi-agent RAG system capable of handling complex financial data processing tasks. The focus is on designing an efficient, scalable, and intelligent system to process financial data, manage reasoning tasks, and handle numerical analysis, using an architecture that balances retrieval, reasoning, and adaptability to real-time data changes.

1.2. Use-Case

Our chosen use case centers around the financial sector, where we are implementing a system for retrieval, complex reasoning and mathematical computation, tailored for handling dynamic financial data.

1.3. Novelty

In terms of innovation, the solution involves fine-tuning a specialized language model, FinGPT, designed for financial data, enhancing it to operate more accurately with the financial terminology and data structures. We propose a multi RAG hybrid architecture composing of LightRAG, GraphRAG, and Pathway as core of our knowledge base, that enable efficient information retrieval as well as better contextual understanding from structured data. Furthermore, our solution involves implementing a dynamic multi-agent system with additional layers like concurrent scheduler, input-output guardrails and fallback mechanisms to ensure robust performance in various operational contexts.

1.4. Solution Overview

The process flow in this financial multi-agent system begins with a Query input, which is processed by the Query Agent. This agent, equipped with guardrails and a ReAct Decomposer, ensures the input is safe and effectively broken down for analysis. The MultiAgent Core then handles the query through specialized agents managed by the Agent Manager: the Math Analyst, Financial Analyst, News Analyst, and Reasoning Agent, each contributing their expertise. All agents can access data through the retriever agent that works in conjunction with the Light RAG/Pathway hybrid pipeline. The LLM Manager gives access to the agents various large language models (LLMs) to enhance processing. All agents can access a set of various global tools that help the model run codes, access the web, get up-to-date data. If any issues arise, the Fallback Manager offers additional safeguards, using mechanisms like human oversight, a reviver tool to resume the process at the last safe stage, and guardrails to ensure accurate outputs. Finally, the processed information is directed to the Output stage.

1.5. Background Study

The design of financial multi-agent systems (TALEBIRAD; NADIRI, 2023), inspired by FinRobot (YANG et al., 2024), FinVerse (AN et al., 2024), and customized FinGPT (YANG; LIU; WANG, 2023), aims to address complex financial decision-making by leveraging specialized, autonomous agents. The current implementation of the financial multi-agent system utilizes LightRAG (GUO et al., 2024) for efficient graph-based retrieval. The idea of fallback mechanisms is inspired by C-RAG (YAN et al., 2024), ensuring robust handling of failed queries. Additionally, the system incorporates query complexity assessment, drawing from the principles of Adaptive RAG, to dynamically evaluate complexity of queries. Hybrid RAG (SARMAH et al., 2024) integrates pathway vector stores and graph RAG for flexible data retrieval, enhancing agents' ability to analyze, predict, and respond to diverse financial scenarios. Inspiration from the LLM Compiler (KIM et al., 2023) helped create the Agent Manager that oversees all operations.

2. ANALYSIS AND INITIAL EXPERIMENTS

2.1. System Architecture

- **Guardrails** are implemented using GuardrailsAI library to ensure that queries are executed safely and efficiently
- **ReAct Decomposer** is used to categorize queries as simple or complex. Complex (multi-step) queries are further decomposed for effective processing.
- **Agent Manager** oversees all the agents, identifies the agents, decides when to activate them and manage their workflow, executing tasks parallelly allowing them to run on multiple threads. Also comprises of a state management system that stores current execution state that communicates with the reviver if anything goes left.
- **Hybrid RAG Mechanism** is a mechanism to select the optimal retrieval approach out of Graph RAG, suitable for global context or Pathway VectorStore, suitable for

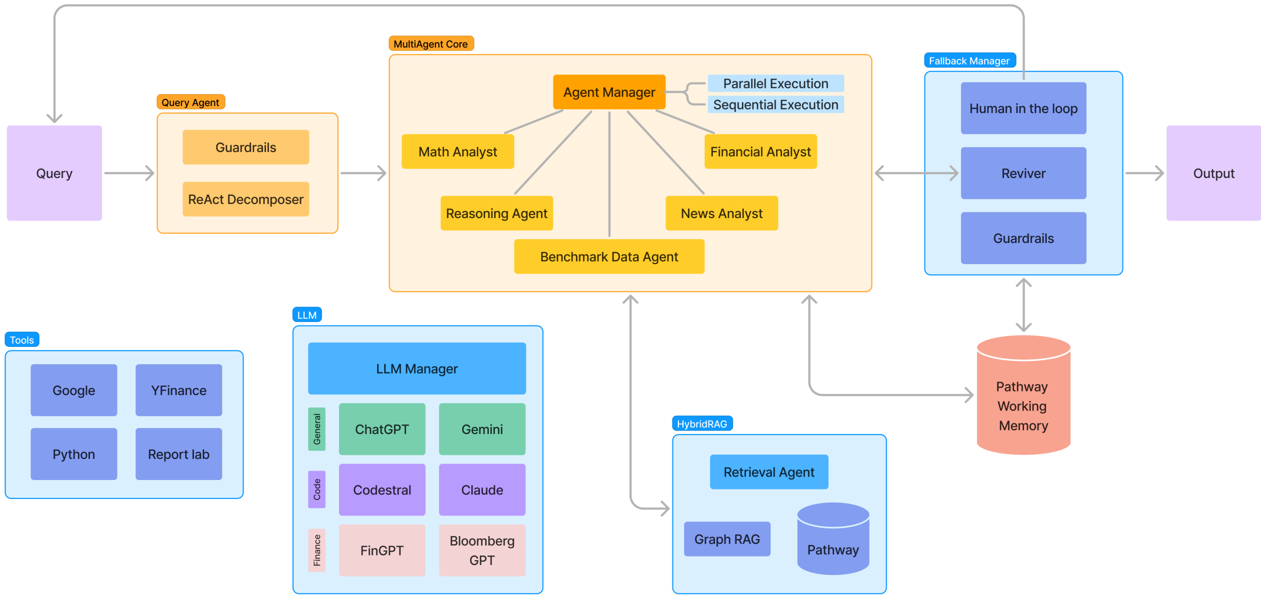

Figure 1. System Architecture

mathematical data.

- **Fallback Mechanism** allows a process to restart or recover if an error or failure occurs. It is enhanced by the Reviver Tool, which optimizes the recovery by resuming the output generation from the last successful state. If multiple failures happen in the pipeline, the system can prompt the user for additional confirmations or input using a Human-in-the-Loop (HITL) tool. This ensures that the process can either continue seamlessly or adjust based on user preferences.

The system architecture is outlined through a flowchart detailing the interactions between core components: data retrieval, reasoning, and response generation.

2.2. Implementation Progress

We have set up the Pathway infrastructure and successfully tested it for real-time pipeline processing. Basic agents have been integrated using the LangChain framework, allowing for initial functionality testing. Both the Pathway vector store and LightRAG have been tested for retrieval, providing insights into optimal data access and retrieval strategies. Additionally, a basic agentic workflow has been implemented, establishing a foundation for further development. This foundational setup will be expanded in future phases to incorporate advanced agent capabilities and enhance system performance.

2.3. Issues Addressed

- **Task Decomposition for Parallel Execution:** By breaking down complex prompts into simpler, more manageable tasks, our approach creates streamlined instruction flows that can be executed in parallel. This decomposition preserves time efficiency while ensuring that each step is completed effectively.
- **Enhanced Efficiency with LightRAG:** LightRAG optimizes the performance of traditional Graph RAG by significantly improving efficiency and processing speed. This advancement overcomes the inherent slowness of standard Graph RAG.
- **Global Context Integration:** Current approaches often suffer from a lack of a unified global context. Our method addresses this gap by improving state management and enabling robust fallback mechanisms, ensuring a more cohesive and resilient system.

2.4. Areas of Novelty

The novel contributions of this project are primarily in the use of specialized techniques for handling financial data: (1) using fine-tuned FinGPT for enhanced financial reasoning, (2) implementing Light RAG to balance fast data retrieval and (3) integrating schedulers, parallel processing, guardrails, and fallback mechanisms. Together, these innovations enable a system that not only retrieves data efficiently but also understands and processes it with an intelligence designed specifically for financial applications.

2.5. Preliminary Results

Our initial experiments show promising results in retrieving and aggregating data using the Retrieval-Augmented Generation Architecture System (RAGAS). We conducted thorough experiments on a specialized finance domain QA-type Ultra Domain finance dataset.

We tested the core parts of our RAG pipeline, i.e., naive vector RAG and graph-based LightRAG, on various metrics presented below. In addition, we also experimented with Pathway's dynamic RAG capabilities on various data sources within our pipeline. LightRAG outperformed Pathway on global textual datasets, leading us to implement a hybrid RAG approach

Metric	Light RAG	Pathway
Faithfulness	0.779187	0.180853
Answer Relevancy	0.962549	0.781300
Context Precision	0.863636	0.318182
Context Recall	0.772727	0.102273
Context Entity Recall	0.198394	0.043290
Semantic Similarity	0.926183	0.743458
Answer Correctness	0.576453	0.218168

Table 1. Comparison of Light RAG and Pathway Results

for optimal results. This architecture has proven effective in sourcing relevant financial information and contextualizing it in a meaningful manner, with notable improvement in the system's response accuracy.

3. CHALLENGES AND NEXT STEPS

3.1. Technical Challenges

A key technical challenge we have encountered is integrating Graph RAG with Pathway's dynamic processing nature. Graph RAG provides excellent data retrieval and context-building capabilities, but Pathway's abstractions create difficulties in fully leveraging these strengths. The primary challenge lies in synchronizing Pathway's real-time adaptability with the structured, contextual approach of Graph RAG, requiring us to find a solution that blends both systems' advantages without compromising functionality.

3.2. Next Steps

Moving forward, we aim to:

- Develop More Agents: Integrate additional agents to improve task management and data handling within the system.
- Robust Testing: Conduct more comprehensive testing to evaluate the system's performance, especially in dynamic financial data environments.
- Expand Data Testing: Broaden the range and type of financial data inputs for a more comprehensive evaluation of the system's capabilities.
- Using FinGPT: Implementing use of FinGPT for improved financial data reasoning and mathematical computation.
- Explore General Financial Multi-Agent RAG System: We also aim to explore the legal and compliance aspects of a general-purpose financial multi-agent system, ensuring it can operate within financial data governance regulations and cater to a wider range of financial applications.

References

AN, S. et al. Finverse: An autonomous agent system for versatile financial analysis. *arXiv preprint arXiv:2406.06379*, 2024.

GUO, Z. et al. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024.

KIM, S. et al. An llm compiler for parallel function calling. *arXiv preprint arXiv:2312.04511*, 2023.

SARMAH, B. et al. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. *arXiv preprint arXiv:2408.04948*, 2024.

TALEBIRAD, Y.; NADIRI, A. Multi-agent collaboration: Harnessing the power of intelligent llm agents. *arXiv preprint arXiv:2306.03314*, 2023.

YAN, S.-Q. et al. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

YANG, H.; LIU, X.-Y.; WANG, C. D. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023.

YANG, H. et al. Finrobot: An open-source ai agent platform for financial applications using large language models. *arXiv preprint arXiv:2405.14767*, 2024.