Team 86

Pathway

End Term Appendix

Table of Contents

Introduction

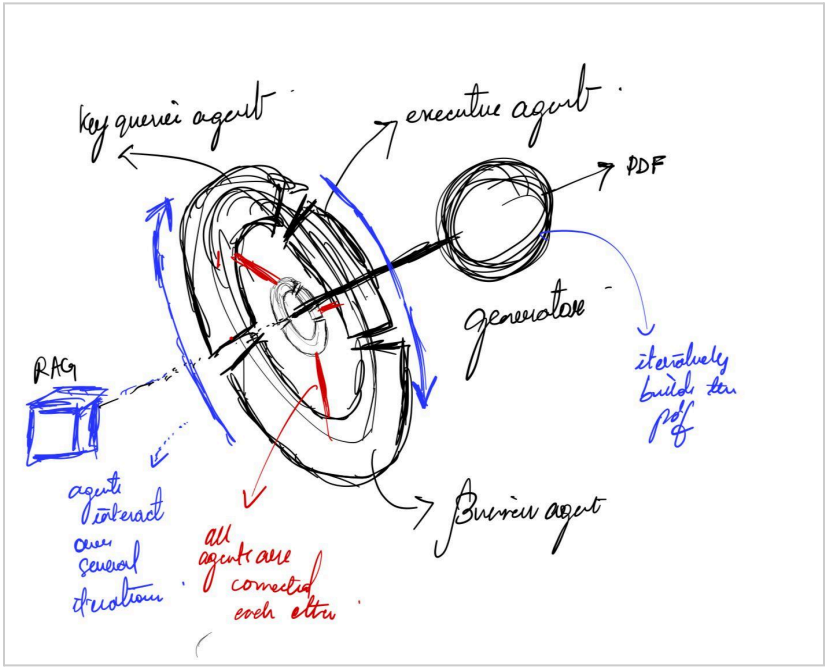
System Architecture - Iterations and Reasoning

Autonomous Layer:

1. Operational Mechanism:

The system functions in two main modes: **QnA Mode** and **Discussion Mode**.

QnA Mode: Agents generate domain-specific, in-depth questions targeting critical aspects of the company's financial and operational state. These questions are processed by the RAG system, which retrieves answers from integrated data sources to provide meaningful insights for due diligence.
Discussion Mode: Agents review the answers from the RAG system, assessing their quality and alignment with the financial report's standards.



- If answers meet expectations, a **Generator Agent** compiles them into a comprehensive Financial Due Diligence report.
- If answers fall short, agents collaboratively identify gaps, suggest improvements, or reframe questions. These suggestions are passed to a **Reframing Agent**, which refines the original queries and re-injects them into the RAG system for better results.

Iterative Process:

- **Switching Modes**:
  The system alternates between QnA Mode and Discussion Mode. After each round of

discussion, the suggestions from agents are compiled into a new set of queries for the next Q&A phase.

- **Feedback and Consensus**:
  At the end of each discussion round, all agents evaluate the responses and suggestions from others. A majority vote is taken to assess the satisfiability of the agents—determining whether the agents are aligned on the insights and decisions derived so far.
- **Stopping Criteria**:
  The iteration continues until the majority of the agents are satisfied with the answer provided by the RAG.

This iterative process ensures high-quality, precise, and actionable financial analysis.

2. Autonomy and Workflow:

The autonomous layer operates independently, executing all tasks without human intervention. It is designed to manage every step, from generating targeted questions to producing a finalized Financial Due Diligence (FDD) report. The process ensures efficiency, accuracy, and iterative improvement.

- **Question Generation**:
  Each specialized agent (Key Metrics, Business, Executive) generates domain-specific questions tailored to its focus area, such as financial health, market conditions, and governance. These questions are submitted to the RAG system, which retrieves answers from integrated data sources.
- **Answer Evaluation**:
  Responses are rigorously assessed by the agents for quality, relevance, and alignment with industry standards. The agents verify the completeness of information, cross-checking against their internal knowledge bases and predefined benchmarks.
- **Report Generation or Refinement**:
  If the responses meet quality standards, the system compiles a comprehensive FDD report, summarizing key findings and insights. If the answers are incomplete, ambiguous, or below standard, a Reframing Agent revises the questions to address specific gaps or discrepancies. These refined questions are then resubmitted to the RAG system, ensuring iterative improvement in data accuracy and relevance.
- **Final Report**:
  Once the agents are satisfied with the responses, a final FDD report is produced. This report encapsulates all critical financial details, including metrics, risk assessments, and strategic evaluations, providing actionable insights for stakeholders.

- **Quick Overview panel:**
  The information extracted from the final report is also converted into structured JSON files, enabling seamless integration with an Overview panel. This dashboard offers a quick and engaging way to visualize key financial data, metrics, and insights. It provides stakeholders with an intuitive interface to interact with the report's findings, facilitating informed and easy decision-making at a glance.

Dynamic Document Retrieval as a Cost-Effective Alternative to Knowledge Graphs:

A Knowledge Graph (KG) agent was initially considered, but its high setup cost led us to opt for Dynamic Relevant Document Retrieval instead. This optional workflow enhances retrieval performance without the complexity and overhead of a KG. It remains optional, as **query transformation often** suffices, simplifying the system and reducing costs, though our system is flexible for switching into D-RAG too.

Financial Tools description:

The financial agent in our system is equipped with 18 robust financial tools, each of which is designed to perform distinct financial tasks efficiently. These tools integrate with various APIs and libraries to deliver varied functionalities, supporting our objective of retrieval and analysis. The tools used have been defined below:

1. **Find Similar Companies:** Identifies companies similar to a given stock symbol, optionally filtered by country.
2. **Get Earnings History:** Retrieves the quarterly earnings history of a stock, including reported and estimated EPS, surprises, and percentages.
3. **Get Upcoming Earnings:** Lists stocks with upcoming earnings announcements within a specified date range and country.
4. **Get Current Gainer Stocks:** Provides a list of the top gainers of the day across all market caps.
5. **Get Current Loser Stocks**: Retrieves data on the top losers of the day across all market caps.
6. **Get Current Undervalued Growth Stocks:** Finds stocks with low price-to-earnings ratios and growth rates exceeding 25%.
7. **Get Current Technology Growth Stocks**: Lists technology stocks with strong growth metrics.
8. **Get Current Most Traded Stocks:** Compiles data on stocks with the highest intraday trading volumes.
9. **Get Current Undervalued Large-Cap Stocks**: Identifies potentially undervalued large-cap stocks.
10. **Get Current Aggressive Large-Cap stocks:** Retrieve a list of high-performing large-cap stocks with strong growth potential.
11. **Get Current Hot Penny Stocks:** Fetch trending penny stocks with significant trading volume or price action.
12. **Get Top-k Trending News:** Extract the top-k most popular and trending news articles across various financial topics.
13. **Get Google Trending Searches:** Retrieve the latest trending search queries from Google.
14. **Get Google Trends For Query:** Analyze and display interest trends for a specific query over time using Google Trends.
15. **Get Latest NEWS For Stocks:** Fetch the most recent news articles and updates related to specific stocks.
16. **Get Current Stock Price Info:** Retrieve real-time stock price information and related financial data.
17. **Web Search Tool:** Executes searches using the Tavily API & DUCKDUCKGO to retrieve relevant financial content with a limit of three results.
18. **Calculator:** Evaluates mathematical expressions using the Wolfram Alpha API for precise calculations.

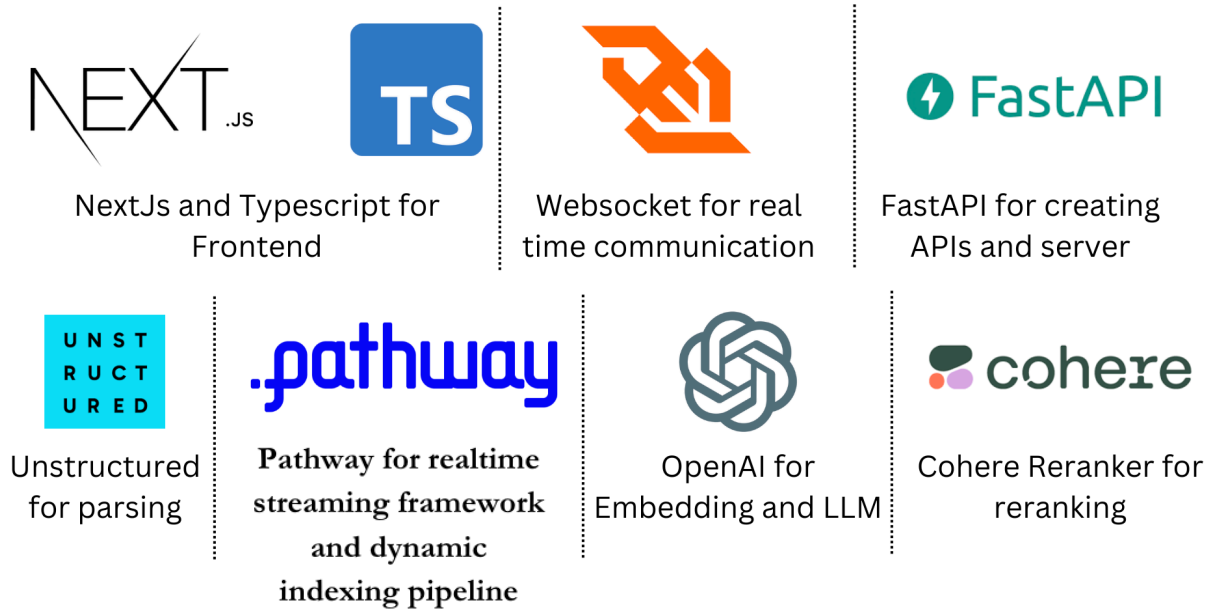
Chart Agent: Although not exactly a tool, Charting Agent acts like a tool to the Reasoning agent for extracting the information related to the creation of the chart and uses its own chart-creating tools to generate contextually relevant charts.

Each tool seamlessly integrates into the system, enabling the financial agent to deliver real-time insights and perform advanced analytics.

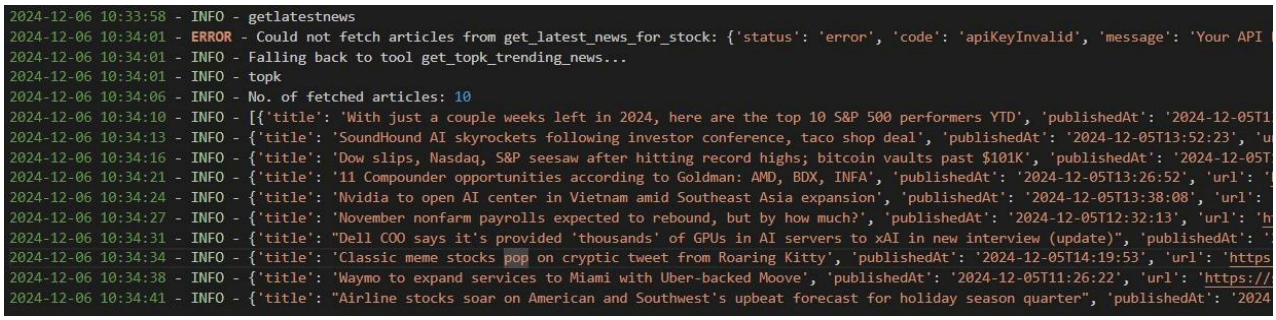
Each of the above tools is backed by a set of certain APIs that are mentioned below:

1. **Alpha Vantage API**: Provides real-time and historical financial data, including stock prices, earnings, and trends.
2. **Finnhub API**: Offers financial data such as peer analysis, company profiles, and market news.
3. **Financial Modeling Prep API**: Supplies data on gainers, losers, and the most traded stocks.
4. **News API**: Enables retrieval of relevant financial news articles for sentiment and trend analysis.
5. **WolframAlpha API**: Performs advanced mathematical and computational queries.
6. **Polygon API:** Delivers in-depth market data and insights about stock symbols.
7. **Tavily API:** Summarizes web content and financial search results efficiently.
8. **DuckDuckGo API:** Facilitates secure and private web searches for relevant financial information.

Tech Stack:



Resilience in Error Handling and Fallbacks:



To ensure uninterrupted functionality, the system incorporates fallback mechanisms for critical components. If Tavily search fails, the system defaults to DuckDuckGo for search queries. Similarly, if Google News retrieval fails, it falls back on top-k news articles from SeekingAlpha.com. In case the Polygon API fails to find similar companies for a given query, the system seamlessly switches to the Finhub API as a backup. These fallback strategies ensure robustness and reliability in handling failures.

In addition to tool failures, the system accounts for scenarios where a query is raised before the Pathway VectorStoreServer is fully initialized. To prevent crashes in such cases, the initial retrieval is skipped and instead, Tavily search is used. This ensures seamless functionality even during startup delays, maintaining system reliability and user experience.

Conclusion

In conclusion, the appendix highlights the systematic exploration and integration of various RAG methodologies that motivated the development of FA^{3}STER, our agentic RAG-based solution for FDD. Over the course of development we experimented with diverse approaches, including Self-RAG for enhanced query introspection, contextual RAG for maintaining semantic relevance, adaptive RAG for real-time response optimization, modular RAG for architectural flexibility, and light RAG for efficiency in resource-constrained scenarios. Furthermore, we also experimented with advanced prompting techniques such as plan-and-solve strategies to enhance the system's decision-making and response-generation capabilities. The insights gained through this process contributed significantly, helping us to come up with the Financial Agentic Autonomous and Accurate System Through Evolving(Dynamic) Retrieval(FA^{3}STER) augmented generation system for Financial Due Diligence.

References

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2023). Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.[1]

Jeong, S., Baek, J., Cho, S., Hwang, S. J., & Park, J. C. (2024). Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*.[2]

Yan, S. Q., Gu, J. C., Zhu, Y., & Ling, Z. H. (2024). Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.[3]

Gao, Y., Xiong, Y., Wang, M., & Wang, H. (2024). Modular rag: Transforming rag systems into lego-like reconfigurable frameworks. *arXiv preprint arXiv:2407.21059*.[4]

Guo, Z., Xia, L., Yu, Y., Ao, T., & Huang, C. (2024). LightRAG: Simple and Fast Retrieval-Augmented Generation. *arXiv preprint arXiv:2410.05779*.[5]

Yepes, A. J., You, Y., Milczek, J., Laverde, S., & Li, R. (2024). Financial report chunking for effective retrieval augmented generation. *arXiv preprint arXiv:2402.05131*.[6]

Anthropic, T. (2024, September 19). *Introducing Contextual Retrieval*. Anthropic. Retrieved December 6, 2024[7]

Xu, B., Peng, Z., Lei, B., Mukherjee, S., Liu, Y., & Xu, D. (2023). Rewoo: Decoupling reasoning from observations for efficient augmented language models. *arXiv preprint arXiv:2305.18323*.[8]

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K. W., & Lim, E. P. (2023). Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.[9]

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2024). Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, *36*.[10]