Untitled 6

94-815 Spring 2025, CO3, Ivan Wiryadi (iwiryadi)

1. Introduction

This report provides analysis of the AI-Powered Financial Portfolio Rebalancer, a system designed to assist investors in maintaining balanced portfolios. The implementation leverages LangChain's agent framework to integrate multiple Large Language Models (LLMs) and compare their effectiveness for financial analysis tasks.

The system was built to perform three primary functions:

1. Retrieve real-time stock prices
2. Analyze portfolio balance based on an equal-weight strategy
3. Provide market trend insights

2. Implementation Details

2.1 Setup

1. Clone / copy the repository
2. Create a python virtual enviroment and install required packages (`requirements.txt`)
3. Put in OPENAI and GROQ API keys into `.env copy` and rename it to `.env`
4. `python financial_agent.py`

2.2 Core Components

The implementation consists of three main tools that is combined/used together to provide financial analysis:

Stock Price Lookup Tool

This tool fetches the latest stock price data using the Yahoo Finance API. It returns information including:

- Current price
- Previous close price
- Daily change (absolute and percentage)
- Timestamp of the data

The implementation includes simple data cleaning to handle potential formatting issues with ticker symbols (removing '$' symbols or quotes). The tool returns structured data in a

dictionary format to be further used by the agent.

Portfolio Rebalancing Tool

This tool analyzes a given portfolio against an equal-weight strategy and suggests necessary adjustments. It:

- Parses the portfolio input
- Calculates target weights based on the number of assets
- Determines necessary buy/sell actions for each asset
- Returns detailed recommendations in a structured format

The tool implements error handling and formatting to handle format variations.

Market Trend Analysis Tool

This tool analyzes the S&P 500 (using SPY as a proxy) to provide insights on market trends over the past week. It calculates:

- 5-day returns
- Volatility (standard deviation of daily returns)
- Market trend characterization (strongly bullish, mildly bullish, mildly bearish, strongly bearish)
- Start and current prices

A note in this implementation is the use of a dummy input parameter to mitigate runtime errors due to the Agent passing a parameter even though it is unnecessary.

2.3 LLM Integration

Three different LLM configurations were implemented:

1. **OpenAI GPT-4o-mini**

- Used as the default option
- Implemented via `langchain_openai.ChatOpenAI`

2. **Groq LLaMA3-8B**

- Implemented via `langchain_groq.ChatGroq`
- Using model "llama3-8b-8192"

3. **Groq LLaMA3-70B**

- Implemented via `langchain_groq.ChatGroq`
- Using model "llama3-70b-8192"

All models were configured with a temperature of 0 to maximize deterministic responses.

3. Challenges Encountered and Solutions

Several significant challenges were encountered during development:

LLM Output Parsing and Tool Selection

One of the most persistent challenges was ensuring that LLMs correctly called tools with valid parameters. This manifested in several ways:

1. **Inconsistent Input Formatting**: The LLMs, particularly LLaMA3-8B and GPT-4o-mini, sometimes included extra characters in stock symbols (e.g., '$AAPL' instead of 'AAPL').

- **Solution**: Implemented input cleaning in the `get_stock_price()` function to remove the extra characters.

2. **Invalid Tool Calls**: The LLMs occasionally attempted invalid actions or provided incorrectly formatted parameters, especially for `market_trend_analysis()`.

- **Solution**: Added a dummy parameter to the Market Trend Analysis tool to ensure compatibility with the agent framework's requirements.

3. **Output Parsing Errors**: As seen in the error logs, LLaMA models sometimes generated output formats that LangChain's parser couldn't interpret correctly:

```
langchain_core.exceptions.OutputParserException: Could not parse LLM
output: `Thought: I have all the current prices of the stocks in the
portfolio. Now, I need to recalculate the weights based on the current
prices.

Action: (No action needed, just calculation)
```

- **Solution**: A more robust approach would be to use `handle_parsing_errors=True` in the AgentExecutor configuration, though this wasn't implemented in the current version.

4. Analysis of LLM Performance

4.1 Comparative Metrics

To evaluate the agents, the code was run 4 times (results are stored in `runs/`). The first run however ended in an error, and so only the remaining three is considered for this analysis.

Metric	OpenAI GPT-4o-mini	Groq LLaMA3-8B	Groq LLaMA3-70B
Response Time (avg)	16.64s	4.01s	71.49s
Response Accuracy	6/6	6/6	6/6
Tool Selection Efficiency	2/6	3/6	3/6
Quality of financial advice	Average	Average	Fine

Additional explanation:

- Response time is simply elapsed time by the agent from receiving input to generating the final answer
- Response accuracy is number of correct final answer out of the two tests across 3 runs (totaling to 6). Here I do not consider the model's intermediary outputs / reasoning and only evaluate the final answers.
- To determine the results (in this case balance equally all portfolio), the model only need to use `rebalance_portfolio`. If the model do not use it, it scores 0 and if it uses others it scores 0.5 for that single use test in a run.
- Quality of financial advice is determined by whether the model sensibly provide additional information and consideration (rather arbitrarily and subjectively decided).

4.2 Performance Analysis

Overall observation

Each model tends to overuse tools, even though to rebalance the portfolio they only need `rebalance_portfolio`. This indicates that even with LLMs becoming more capable, there's still room for improvement in tool selection efficiency.

OpenAI GPT-4o-mini

Strengths:

- Correctly concluding that all portfolios are already balanced.
- Added market context for recommendations by checking market trends. For example it noted in run2.txt, it noted: "However, due to the current strongly bearish market trend, it is advisable to monitor the portfolio closely."
- Produced concise, actionable recommendations, consistently formatting suggestions in clear bullet points (e.g., in run2.txt: "To rebalance the portfolio, sell 17% of AAPL, buy 3% of TSLA, and buy 13% of GOOGL").

Weaknesses:

- Occasionally failed to retrieve stock data. In run1.txt, it attempted to retrieve stock prices but received "No data found" for all stocks.
- Occasionally performed redundant analysis. For example, in run2.txt, it performed calculations: "MSFT: 0.655 shares * $381.26 = $249.99" despite having assumed earlier that it allocates $250 to each stocks.
- Medium response times averaging 16.64 seconds, making it slower than LLaMA3-8B.

Notable Behavior:

In run1.txt, despite failing to retrieve stock data, GPT-4o-mini still provided a correct recommendation:

```
It seems that I am unable to retrieve stock prices for any of the stocks
in the portfolio. However, I can still analyze the portfolio based on the
weights provided. The current weights are AAPL (50%), TSLA (30%), and
GOOGL (20%), which are clearly imbalanced compared to an equal-weight
strategy of approximately 33.33% for each stock.
```

This demonstrates its own reasoning capabilities is able to 'take over' for simple tasks when the tools are not available.

Groq LLaMA3-8B

Strengths:

- Significantly faster response time averaging just 4.01 seconds across all runs, making it more than 4 times faster than GPT-4o-mini and 17 times faster than LLaMA3-70B.
- Correctly concluding that all portfolios are already balanced.

Weaknesses:

- Less thorough analysis compared to other models, often skipping market trend analysis altogether. Unlike the other models, it never referenced the MarketTrendAnalyzer tool's outputs in its recommendations.
- Occasionally struggled with tool selection, as seen in all runs where it attempted invalid actions. In run2.txt, run3.txt, and other runs, it tried "Action: None" which resulted in errors: "None is not a valid tool, try one of [StockPriceLookup, PortfolioRebalancer, MarketTrendAnalyzer]."
- Limited market context in recommendations, focusing only on the immediate rebalancing task without providing broader financial context or considerations.

Notable Behavior:

LLaMA3-8B consistently provided the most concise responses, maintaining the exact same format and wording across all runs for both balanced and imbalanced portfolios. This suggests it may be relying more on pattern matching than deep reasoning.

Groq LLaMA3-70B

Strengths:

- Correctly concluding that all portfolios are already balanced.
- Most comprehensive analysis of all models, consistently getting all steps of the financial analysis process. In all runs, it checked portfolio balance, retrieved current prices for all stocks, and analyzed market trends before providing recommendations.
- Consistently included specific price data in recommendations. For example, in run3.txt: "Sell 0.17 of AAPL (currently priced at 190.13) to reach the target weight of 0.33."
- Provided detailed market context in recommendations. In run2.txt, it advised: "considering the current market trends, which are strongly bearish, it may be a good idea for the user to consider diversifying their portfolio or adjusting their investment strategy to mitigate potential losses."

Weaknesses:

- Extremely slow response times averaging 71.49 seconds, almost 18 times slower than LLaMA3-8B and over 4 times slower than GPT-4o-mini. In run2.txt, it took over 103 seconds to complete analysis.
- Occasional parsing errors due to non-standard output formatting, as seen in the failed run: "Could not parse LLM output: `Thought: I have all the current prices of the stocks in the portfolio. Now, I need to recalculate the weights based on the current prices.`"

Notable Behavior:

LLaMA3-70B performed the most thorough and consistent analysis across all runs. Even for the already balanced portfolio in run3.txt, it still checked all stock prices and market trends before concluding:

```
Based on the analysis, the portfolio is already balanced according to the
equal-weight strategy, and no rebalancing actions are necessary. However,
considering the current market trends, which are strongly bearish, it may
be a good idea for the user to consider diversifying their portfolio or
adjusting their investment strategy to mitigate potential losses.
```

This demonstrates superior analytical thoroughness, albeit at significant time cost.

5. Recommendations for LLM Selection

Based on the comparative analysis, the following recommendations can be made for different financial analysis scenarios:

5.1 For Quick Portfolio Checks

Recommended LLM: Groq LLaMA3-8B

The 8B model's extremely fast response time (averaging just 4.01 seconds) makes it ideal for quick portfolio balance checks where detailed analysis isn't required. While it lacks the market context provided by larger models, it consistently delivers accurate rebalancing recommendations. This makes it particularly well-suited for:

- High-volume, low-complexity rebalancing operations
- Time-sensitive trading scenarios
- Mobile applications where response time is critical
- Budget-conscious implementations where API costs need to be minimized

5.2 For Comprehensive Financial Analysis

Recommended LLM: Groq LLaMA3-70B

The 70B model provides the most thorough responses, as evidenced by its systematic approach to checking all stocks and market trends in every run. Its responses include specific price data and actionable market insights that go beyond simple rebalancing recommendations. This depth makes it suitable for:

- Complex financial decision-making requiring nuanced understanding
- Financial advisory services needing comprehensive analysis
- Investment platforms offering premium analysis features
- Educational applications demonstrating proper financial analysis methodology

The significantly longer processing time (71.49 seconds on average) and higher API costs are justified in scenarios where analysis quality outweighs speed considerations.

5.3 For Balanced Performance

Recommended LLM: OpenAI GPT-4o-mini

For users needing a balance between speed and quality, the GPT-4o-mini model offers the best compromise. With an average response time of 16.64 seconds, it's significantly faster than LLaMA3-70B while still providing market context and clearly formatted recommendations. Its robust error handling (as demonstrated in run1.txt) makes it particularly suitable for:

- Everyday portfolio management tasks for individual investors
- Financial applications targeting general users
- Scenarios requiring reliable performance under varying API conditions

6. Discussion and Future Improvements

- **More Targeted Tool Selection**: Because of the inefficient use of tools, further improvements can be made so the agents specifically use tool that is required to perform the task.
- **Improved Error Handling**: Add backoff/retry logic for API failure, e.g. API rate limiting, false input by agents.
- **Multi-Strategy Support**: Extend rebalancing beyond equal-weight to include other strategies (market-cap, risk-parity).
- **Enhanced Output Parsing**: Implement better LLM output parsing to handle non-standard formats.

In context of creating more advanced or deploy-able LLMs:

- **Ethical and responsible disclosures and deployment**: Need to consider how to provide accountability to such agents that provide recommendations, and to measure or evaluate their performance over time. Financial advisors usually need certification or license. Other factors include mitigating attacks or abuse, third party data transfers over API calls, etc.

Appendix / Generative AI Use Disclosure

Please see `appendix.pdf` or `appendix.md`. After the initial prompt, I reviewed and updated the contents of the report.

References

Anthropic. (2025). Claude 3.7 Sonnet (20250219 version) [Large language model]. https://www.anthropic.com/