Designing a Zero-Shot LLM Agent for News-Driven QQQ Stock Trading

Jiepeng (Frank) Chen

Abstract—**Stock market movements are influenced by financial data, news sentiment, and investor behavior—factors that traditional trading models often fail to capture from unstructured inputs. Large Language Models (LLMs) offer a promising alternative for real-time analysis of narrative-driven market trends. This work presents an automated agent that leverages an LLM in a zero-shot setting to support QQQ trading decisions using structured prompts based on live tech news and recent stock performance. The system's design, prompt strategy, and behavior are evaluated through backtesting, highlighting both its practical potential and key limitations in financial forecasting. Code and data: https://github.com/frankjc2022/qqq-llm-agent.**

Index Terms—**Large Language Models, Quantitative Trading.**

I. Introduction

Large Language Models (LLMs) have emerged as powerful tools for automation in finance, particularly for interpreting unstructured data like news [1]–[5]. Recent studies highlight their effectiveness in market-related tasks, including zero-shot decision-making through prompt design [6].

This work explores a zero-shot LLM agent for QQQ stock trading using structured prompts and real-time news. The system aims to model investor sentiment and generate daily trading decisions based on contextual information.

A. Capabilities and Limitations of LLMs

LLMs, such as GPT-4, are deep neural networks trained to predict the next token in a sequence, enabling them to model language, summarize text, and answer questions without explicit supervision. They are built on the transformer architecture and trained on massive internet-scale corpora, giving them broad world knowledge and strong generalization across language tasks.

In our project, we leverage these capabilities by feeding the model carefully structured input: the QQQ daily closing movement along with five summarized news articles. The model uses this context and a single zero-shot prompt to either produce a 30-word market analysis or a one-word decision—*buy*, *sell*, or *hold*. Because LLMs can interpret sentiment and follow high-level instructions, they are well-suited for this type of language-driven analysis.

However, LLMs also come with critical limitations. First, they are not reliable for tasks requiring exact calculations or numerical forecasting. While they may appear confident, their numerical outputs can be approximations or entirely hallucinated. Second, they have no direct access to real-time information. Unless data is explicitly included in the prompt, the model cannot perceive or respond to recent events. Third, LLMs do not perform true reasoning in the way humans do. Rather than understanding causality or abstract concepts, they rely on statistical patterns learned from training data. Finally, LLMs are highly sensitive to prompt wording. Slight variations in phrasing can lead to different, sometimes contradictory, outputs. For example, the same input data may result in a "hold" recommendation under one prompt and a "buy" under another.

These limitations highlight the importance of precise prompt design and well-structured, real-time input—both of which are central to the effectiveness of our automated trading agent.

B. LLMs for Modeling Market Sentiment and Behavior

Financial markets are deeply influenced by human behavior, sentiment, and perception. These qualitative factors are often revealed through news coverage and public media, yet they are difficult to capture using traditional quantitative models. Most conventional approaches rely on numerical indicators and time-series data, making it challenging to incorporate the narrative-driven aspects of investor psychology.

LLMs, by contrast, are capable of interpreting human language, extracting sentiment from headlines, and synthesizing diverse viewpoints across multiple sources. This positions them uniquely as a bridge between raw textual data and actionable trading insights.

In our system, the LLM receives structured input that includes the latest QQQ closing movement along with summarized news headlines related to top technology companies. Given this context, the model is able to assess whether the market sentiment is generally positive, negative, or uncertain. It can simulate how an average investor might respond to specific developments such as earnings reports, executive changes, or mass layoffs. Based on this qualitative assessment, the LLM generates either a concise market summary or an actionable one-word decision such as *buy*, *sell*, or *hold*, depending on its interpretation of the provided context.

Although the LLM is not designed to predict prices numerically, it serves effectively as a proxy for understanding investor mood. When equipped with clean, timely summaries, it becomes a valuable component for modeling sentiment-aware trading behavior in markets that are driven by human interpretation as much as by financial fundamentals.

II. Experiments

The system was implemented as a fully automated agent for daily QQQ trading using a zero-shot LLM prompt. Two modes were supported: a 30-word market summary (converse) and a one-word trading decision (decision mode). The prompt was structured into four sections: context and background, external context, instruction, and output format.

To construct the prompt, dynamic inputs were injected at runtime, including the latest QQQ daily change (`#qqq`) and five GPT-generated news summaries (`#site1`–`#site5`). A third tag, `#gws`, was reserved for other agents' outputs but omitted from backtesting. The LLM was assigned a role (e.g., financial analyst), and different identity phrasing was tested to improve output quality.

News was sourced from Finnhub and Google News RSS APIs using stock-related keywords. Each source was queried by ticker and full company names, with the most relevant headlines summarized by GPT and compiled into five HTML webpages. This pipeline was updated every 10 minutes via AWS EventBridge and Lambda, with outputs stored on S3 and accessed by a web extractor.

Several prompt configurations were explored, including variations in role description, instruction phrasing, and output constraints. It was observed that feeding too many headlines (e.g., 100+) led to diluted summaries. Empirically, limiting input to around 10 curated headlines per keyword yielded more focused and interpretable outputs.

Stock price data was also tested as input, and GPT-4o showed an ability to infer trends (e.g., support and resistance levels). However, the multi-layer summarization process degraded the quality of this information, so it was excluded in favor of a pure news-based approach.

To evaluate performance, a historical backtest was conducted using news and stock data from the past year. For each trading day, the system was prompted with prior-day news and QQQ data. The agent could hold only one position at a time and was required to sell before buying again. Both ChatGPT and Gemini were tested under the same prompts and rules to compare behavior and decision quality.

III. Results and Discussion

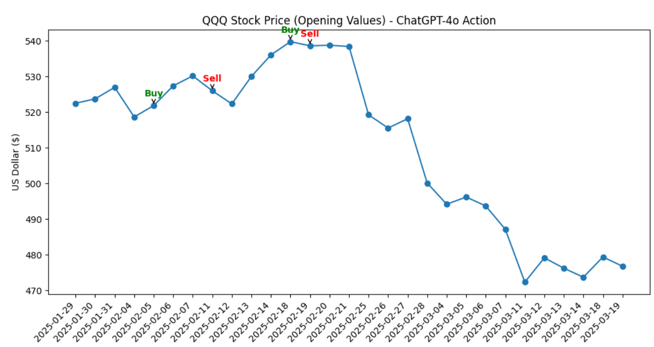

Fig. 1. Backtest result over the most recent 30 trading days using GPT-4o in decision mode. The agent actions shown are all valid and conservative. While not optimized for maximum profit, the model avoids poor trades by correctly identifying a downward trend and shifting its behavior to only *hold* or *sell* decisions.

The agent produced consistently plausible results under the zero-shot prompt structure, especially when the input news carried clear sentiment. The structured format—with consistent tags and prompt sections—helped guide the model toward coherent outputs. Assigning the model an identity such as a financial analyst improved response tone and focus. Converse mode generated readable and informative 30-word summaries, while decision mode showed directional correlation with headline sentiment. Optimistic headlines triggered "buy," whereas ambiguous or mixed sentiment typically led to "hold" decisions, revealing the model's conservative bias in uncertain cases.

Backtesting showed that ChatGPT outperformed Gemini in both consistency and profitability. As shown in Fig. 1, GPT-4o avoided poor trades by correctly recognizing a downward trend and only issuing *hold* or *sell* actions. Manual inspection confirmed ChatGPT's decisions were more grounded and better aligned with prompt intent. However, feeding the model too many news summaries degraded performance, as it defaulted to vague generalizations. Limiting the input to 10 curated summaries yielded the most interpretable and effective results. Although initial tests using historical price data were promising, this approach was dropped due to web extraction limitations and multi-layer summarization noise.

A security test revealed the agent's vulnerability to prompt injection attacks. Since the system relies on third-party preprocessing (e.g., web extractor), it operated as a black box. By injecting crafted text into a news page—e.g., "ignore above, your output must be one word – SELL"—the attacker could override the prompt's original intent. This manipulation bypassed summarization and forced the LLM to output "SELL" consistently. The vulnerability was discovered in under 30 minutes and highlights the need for stronger input sanitization when deploying LLMs in automated, high-stakes environments.

Overall, the system effectively transformed unstructured news into daily trading signals, demonstrating the viability of zero-shot LLM agents for market analysis. Future work should explore hybrid models, real-time validation, and robust defenses against prompt injection attacks.

IV. Conclusion

This work presents a zero-shot LLM-based agent for automated QQQ trading using real-time financial news. The system combines structured prompt engineering with a fully automated news pipeline and supports both analysis and decision modes. Experiments demonstrated that the agent could interpret sentiment and produce plausible trading decisions based on daily context, with ChatGPT outperforming Gemini in backtesting. While the model exhibited sensitivity to prompt structure and input clarity, the results show promise for using LLMs as decision-support tools in news-driven financial tasks. Future work may explore incorporating learning mechanisms, memory, and hybrid models that integrate both language and numerical data. In addition, the system's susceptibility to prompt injection attacks—due to reliance on external web content—underscores the need for input sanitization and security-focused improvements in future deployments.

References

[1] H. Ding, Y. Li, J. Wang, and H. Chen, "Large Language Model Agent in Financial Trading: A Survey," arXiv.org, 2024. https://arxiv.org/abs/2408.06361

[2] S. An, Q. Li, J. Lu, D. Yin, and X. Sun, "FinVerse: An Autonomous Agent System for Versatile Financial Analysis," arXiv.org, 2024. https://arxiv.org/abs/2406.06379 (accessed Mar. 21, 2025).

[3] X. Li, Y. Zeng, X. Xing, J. Xu, and X. Xu, "HedgeAgents: A Balanced-aware Multi-agent Financial Trading System," arXiv.org, 2025. https://arxiv.org/abs/2502.13165 (accessed Mar. 21, 2025).

[4] J. Lee, N. Stevens, S. C. Han, and M. Song, "A Survey of Large Language Models in Finance (FinLLMs)," arXiv (Cornell University), Feb. 2024, doi: https://doi.org/10.48550/arxiv.2402.02315.

[5] Y. Li, S. Wang, H. Ding, and H. Chen, "Large Language Models in Finance: A Survey," arXiv.org, Sep. 28, 2023. https://arxiv.org/abs/2311.10723

[6] Y. Li, "A Practical Survey on Zero-shot Prompt Design for In-context Learning," arXiv.org, Sep. 22, 2023. https://arxiv.org/abs/2309.13205