Large Language Model Agent in Financial Trading: A Survey

Han Ding*
hd2412@columbia.edu
Columbia University
New York, NY, USA

Yinheng Li*
yl4039@columbia.edu
Columbia University
New York, NY, USA

Junhao Wang*
jw3668@columbia.edu
Columbia University
New York, NY, USA

Hang Chen*
hc2798@nyu.edu
New York University
New York, NY, USA

ABSTRACT

Trading is a highly competitive task that requires a combination of strategy, knowledge, and psychological fortitude. With the recent success of large language models(LLMs), it is appealing to apply the emerging intelligence of LLM agents in this competitive arena and understanding if they can outperform professional traders. In this survey, we provide a comprehensive review of the current research on using LLMs as agents in financial trading. We summarize the common architecture used in the agent, the data inputs, and the performance of LLM trading agents in backtesting as well as the challenges presented in these research. This survey aims to provide insights into the current state of LLM-based financial trading agents and outline future research directions in this field.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**; **Information extraction**; **Intelligent agents**.

KEYWORDS

Large Language Models, Agent, Asset Management, Quantitative Trading

1 INTRODUCTION

Recent advances in large language models (LLMs) have revolutionized research in natural language processing and demonstrated significant potential in powering autonomous agents [46]. LLM agents have been applied across various domains, such as healthcare [32] and education [59]. In addition, the finance sector has seen lots of exploration of LLM applications [23, 26]. There has been a emerging trend of developing LLM powered agents for trading in financial markets. Professional traders are required to process amount of information from various sources and quickly make decisions. Therefore, LLMs are well-suited for this role due to their ability to process large amounts of information quickly and produce insightful summaries.

In this survey, we conduct a systematic analysis of the research into using LLMs as agents for financial trading. Our goal is to identify common areas of research and offer insights into future directions. Specifically, we aim to address the following questions:

- What are the common architectures in LLM powered trading agents?
- What types of data are used for LLMs to make informed trading decisions?
- What is the current performance of LLMs in financial trading, along with their potential and limitations?

As LLM powered agent is an emerging research topic, relatively few studies have explored applying this technique in financial trading. In this survey, we reviewed 27 papers that study using LLMs for financial trading with seven of which explicitly include the term "agent" in their titles. We identified these papers through multiple Google Scholar search using keywords such as "LLM for trading" and "GPT stock agent." Each paper was manually assessed to confirm its relevance to financial trading with LLM agents. To the best of our knowledge, this is the first paper to review the contributions in the domain of LLM agents for financial trading.

2 ARCHITECTURE

The architecture is a crucial aspect when designing an LLM-based agent, and it's often determined by the agent's objective. Generally, the primary objective of a trading agent is to optimize returns through its trading decisions over a specific period. Besides, other risk-related metrics are also crucial in evaluating agent performance, which we will explore further in Section 4. While there are LLM-powered agents designed for various financial tasks such as summarizing financial news [1] or acting as financial advisors [21], our focus will be on trading agents aiming to achieve investment returns, as this constitutes the majority of research in this area.

The architectures can be broadly categorized into two types: LLM as a Trader and LLM as an Alpha Miner. LLM trader agents leverage LLMs to directly generate trading decision(i.e. BUY, HOLD, SELL).On the other hand, Alpha Miner agents utilize LLM as efficient tools to produce high quality alpha factor, which are subsequently integrated into downstream trading systems. A tree diagram demonstrating the hierarchical structure and development of all these architectures is presented in Figure 1.

2.1 LLM as a Trader

The architecture of LLM trader agents focuses on utilizing LLM to directly make trading decisions. These systems are designed to analyze vast amounts of external data, such as news, financial reports, stock price and refine information from the these data to generate buy or sell signals. This section discusses different sub-types of LLM as a trader agents, including news-driven, reflection-driven, debate-driven and reinforcement learning(RL)-driven agents.

*All authors contributed equally to this research. Order is random.

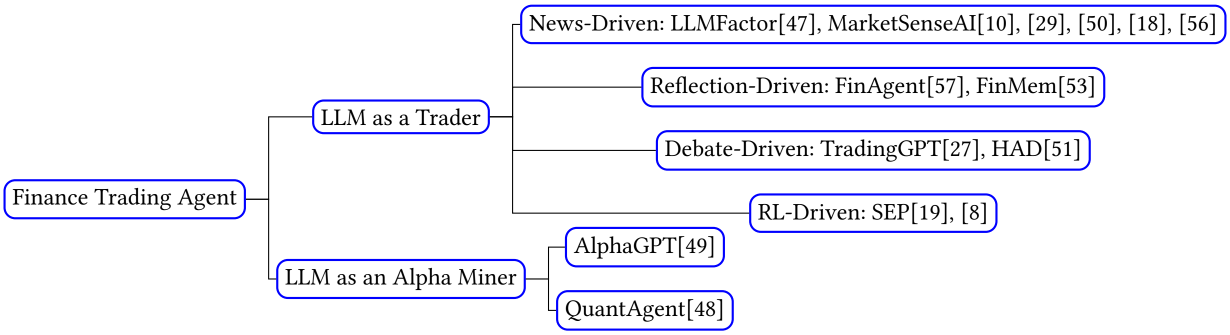

Figure 1: Overview of architectures of finance LLM agents.

2.1.1 *News-Driven.*

News-driven architecture is the most fundamental type, where individual stock news and macroeconomic updates are integrated into the prompt context. The LLMs are then instructed to predict stock price movements for the next trading period. Existing works [29, 50] evaluate both close-source LLMs(e.g GPT3.5/4) and open-source LLMs(e.g Qwen[3], Baichuan[52], etc.) in financial sentiment analysis. They also backtest simple long-short strategy based on these sentiment scores, demonstrating the effectiveness of trading using such strategy. Additionally, [18, 56] researched the performance of LLMs(FinGPT, OPT, etc.) that specifically fine-tuned with financial related dataset and demonstrated further improvement by aligning LLMs with domain-specific knowledge.

More advanced architectures involve summary, refinement of news data and reasoning of the relationship between news data and stock price movement. [10] developed several summary modules, including progressive daily news summary, fundamental and macroeconomic summary, and stock price momentum summary. These summaries are managed by a memorization module and referred as "memory". During the trading stage, relevant "memory" is retrieved as "recommendation" context to generate final trading decision. The authors also found that the general-purpose LLMs such as GPT4[36] has great in-context learning capability in financial oriented tasks. LLMFactor[47] first utilizes LLM's reasoning capability to identify important factors by asking the LLM to analysis relationship between historical news and corresponding stock price movements. Then, the agent extract these factors from daily news and make predictions of stock price during trading.

Note that some approaches are more akin to advanced LLM-based methods, with zero-shot prompting or in-context learning[24, 42], rather than agent-based systems, but they are included here for completeness.

2.1.2 *Reflection-Driven.*

Reflection[38] is built on extracted memory using LLMs summarization. It is high-level knowledge and insights progressively aggregated from raw memories and observations. Such reflection is used to make trading decisions. In this section, we survey finance LLM agents that incorporate reflection into their architectures.

FinMem [53] introduces a trading agent with layered memorization and characteristics. The raw inputs, such as daily news and financial reports, are summarized into memories. Upon the arrival of new observations, the relevant memories are retrieved and integrated with these observations to produce reflections. Both memories and reflections are stored in a layered memory bucket. During the trading phase, these memories and reflections are retrieved and utilized by the decision-making module to generate the final trading decisions. The retrieval method considers the recency, relevancy, and importance of the information.

FinAgent[57] proposed the first multimodal agent with similar layered memory and layered reflection design, with an additional multimodal module that takes in numeric, text and image data. Furthermore, the decision making module incorporates technical indicators such as Moving Average Convergence/Divergence(MACD)[9] and Relative Strength Index(RSI)[11] as well as analyst guidance to effectively capture market dynamics. This framework has demonstrated superior performance in backtesting compared to other agents including FinMem.

The design of memory and reflection can also find its root in cognitive science[6]. Analogous to human learning, where human beings interact with the environment, absorb feedback, generate memories and apply learned lessons to solve tasks, memory and reflection in LLMs based trading agents share similar mechanisms. The inclusion of memory and reflection in LLM-based algorithms offers significant benefits such as mitigating the risk of hallucinations[15] and obtaining high-level understanding of the environment[39].

2.1.3 *Debate-Driven.*

Debating among LLMs is proven to be an effective method to enhance the reasoning and factual validity. This approach is also widely adopted in LLM financial agent. [51] proposed a heterogeneous debating framework where LLM agents with different roles (i.e. mood agent, rhetoric agent, dependency agent, etc.) debate with each other, which improves the news sentiment classification performance. TradingGPT[27] proposed a similar architecture as FinMem[53] with one extra step that agents debate on each other's actions and reflections, thereby improving the robustness of reflections.

2.1.4 *Reinforcement Learning Driven.*

Reinforcement learning methods, such as RLHF [37] and RLAIF [22], have proven effective in aligning LLM outputs with expected behaviors. One challenge, however, is obtaining high-quality feedback efficiently and systematically. In financial trading, backtesting provides a cost-effective method for generating high-quality feedback on trading decisions and, intuitively, can serve as a source of rewards in reinforcement learning. SEP [19] has proposed leveraging reinforcement learning with memorization and reflection modules in trading agents. This approach utilizes a series of correct and incorrect predictions derived from financial market history to refine the LLM's predictions in real-world markets.

Furthermore, Reinforcement Learning is well known as a classical method for decision making in games and trading, due to its nature[16]. [8] developed a RL-based framework consists of Local-Global (LG) model and Self-Correlated Reinforcement Learning (SCRL), which are made of multi-layer perceptrons. An LLM is used to generate embeddings from news headlines, which are then projected into the stock feature space. These embeddings are integrated with existing stock features to serve as inputs for the LG model. The LG model, functioning as the policy network, is trained via Proximal Policy Optimization (PPO) [43] with trajectories sampled from the training trading period.

2.2 LLM as an Alpha Miner

Another important category involves agents using LLMs as Alpha Miners, where the LLM generates alpha factors instead of directly making trading decisions. QuantAgent [48] demonstrated this method that leverages the LLM's capability to produce alpha factors through an innerloop-outerloop architecture. In the inner loop, the writer agent takes in a general idea from human trader and generates a script as its implementation. The judge agent provides feedback to refine the script. In the outer loop, the committed code are tested in real world market and the trading results are used to enhance the judge agent. It has been proved that that this approach enables the agent to progressively approximate optimal behavior with reasonable efficiency. In a subsequent research, AlphaGPT[49] propose a human in the loop framework for alpha mining. This approach instantiated an alpha mining agent on a similar architecture and an experimental environment. Both studies demonstrates the effectiveness and efficiency of the LLM powered alpha mining agent system, which is especially valuable as alpha mining is an resource intensive job.

2.3 LLM Selection in Agent

Lastly, to investigate the use of different LLM models, we have included a histogram (see Figure 2) of the LLM models to provide an overview of the spectrum. It is worth noting that OpenAI's models, particularly GPT-3.5 and GPT-4, dominate research usage due to their outstanding general performance. Also there is a long tail distribution of open source model selection, catering to needs for more flexible and specialized development. Notably, GPT3.5 is even used more frequently than GPT4, indicating a preference on its cost-effectiveness and lower latency.

3 DATASET

LLM-powered trading agents rely heavily on diverse data sources to generate trading signals. In our survey, we identified a wide range of data types utilized by various agents, which we have categorized into four major groups:

- Numerical Data: Includes numbers or statistics, such as stock prices and trading volumes.

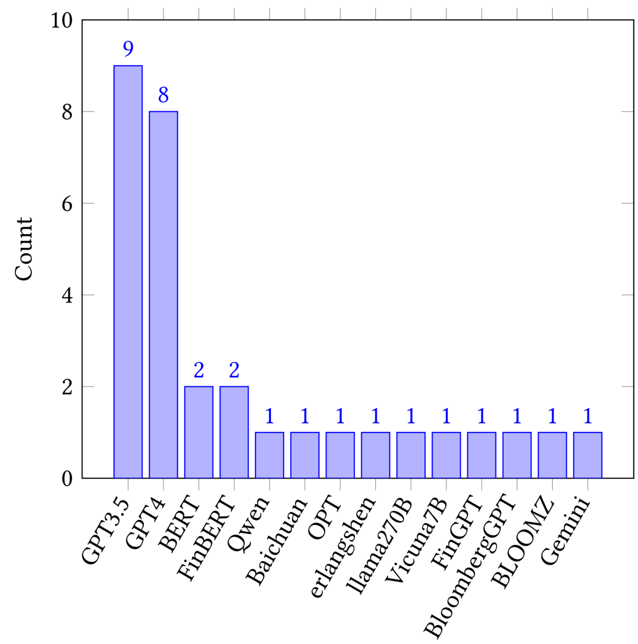

Figure 2: Histogram of base LLM used by Finance Agent (one paper may contain multiple agent)

- Textual Data: text-based information, such as stock news, financial reports.
- Visual Data: Consists of charts and images related to the financial markets.
- Simulated Data: Consists of data from simulated stock markets and news events.

3.1 Numerical Data

In conventional quantitative trading models, numeric data has played a crucial role[2, 31]. However, LLMs are inherently designed to process textual data. To accommodate numerical data, it must be converted into text strings to ensure compatibility with LLM architectures. Despite LLMs' known weaknesses in arithmetic problem and reasoning[12], numerous studies have successfully incorporated numerical data into into LLM-based trading agents. [57] calculates common stock price features, such as three-day price changes, from raw stock price data. These features are then described and summarized by the LLM to form short-term, mid-term, and long-term signals. These signals contribute to the low-level reflection processes of the LLM agent.

In [48], the authors further used additional numerical market data such as open and close price, high and low price to create trading ideas that guide the outer feedback loop. This data also serves as a means of evaluating the generated alpha strategy. Our findings suggest that incorporating numerical data is essential, as it inherently reflects the characteristics of the financial market. For example, high trading volumes and rising prices typically signify positive market expectations, often correlating with company performance.