Labeling Financial Time Series with Large Language Models

Maximino DiGiacomo-Castillo
Department of Computer Science
Stanford University
Stanford, USA
maxdigi@stanford.edu

***Abstract*—Data labeling is a well-studied issue within artificial intelligence. One overlooked area in this space is labeling time series data. Labeling time series data can help quickly explain trends that identify anomalies or can highlight significant events. In the research presented here, we used large language models to label historical stock performance, leveraging knowledge graphs and context to improve baseline performance. Specifically, we measured how effectively large language models generate labels that explain a stock's performance in each year between 2018 and 2022. We generated labels for a random sample of companies in the Standard and Poor's 500 index. We found that newer large language models outperformed older models, but that supplementing language models with our knowledge graph and company relevant context did not meaningfully improve model performance. We did not rule out the possibility that more finely tuned knowledge graphs or data context could improve model performance.**

I. Introduction

Without relevant context, explaining trends in time series data can be difficult. For example, in February 2020 global equity prices fell more than 30% in a month due to uncertainty in the early days of the COVID-19 pandemic. Without prior knowledge of the pandemic or past events such as the Dot-Com Bubble, 9/11, or the Global Financial Crisis, equity price data would appear to have random and sharp fluctuations.

Leveraging large language models (LLMs) to integrate context from diverse sources can enhance our understanding of the causal factors that drive different time series [1]. To further improve the causal reasoning of popular LLM models, we can introduce a knowledge graph [2] [3] [4].

A knowledge graph uses a graph structure to capture relationships between different entities. For example, we can make nodes for different entities like "Federal Reserve" and "Interest Rates" and then create a directed edge between the nodes that captures the relationship. In this case, one possible edge might be "sets." With two nodes and an edge, we can form a *triplet* that maps to a subject-predicate-object statement. In this case, we would read the triplet as (Federal Reserve) - (sets) - (Interest Rates).

Given a task, like labeling a time series, we can query our knowledge graph for relevant supplemental information. Specifically, we could use a keyword, semantic, or hybrid search to retrieve triplets from the knowledge graph that relate to the stock labeling task.

By creating a knowledge graph with widely accepted causal relationships, we aim to develop methods capable of explaining and predicting behavior in different fields.

Finance is an ideal test case for this project due to the availability of extensive time series data, and related contextual information such as news articles and economic reports. However, the methodology developed can be applied to other fields. For example, we could ask a language model to generate labels for a time series that tracks GMAT scores over time.

II. Methods

A. Models

For our experiment we used two large language models, GPT-4o and GPT-4o mini. Older models like GPT-3.5 Turbo did poorly in our experiments. For example, when asked to generate a yearly "catalyst" for a stock's performance between 2018 and 2022, GPT-3.5 Turbo outputs a single catalyst for the year 2023. On the other hand, the newer GPT models correctly and consistently handle this task.

B. Data Pre-Processing

Before our data labeling step, we did a series of pre-processing steps. First, we began by fetching all relevant price data. Then, we computed the performance of each stock relative to our benchmark index, the S&P 500. Computing relative performance allowed us to isolate a given stock's performance, which will inform our model's reasoning. For example, knowing that Delta Air Lines' stock massively underperformed the S&P 500 index in 2020, during the COVID-19 Pandemic, could inform how a model performs.

These steps are summarized below:

1) Fetch historical price data for SP500 index and all constituent stocks from January 1, 2018 to December 31, 2022.
2) Compute daily return of benchmark (percentage points) using raw price data.
3) Compute daily return of stock (percentage points) using opening and closing price data.
4) Compute yearly returns for benchmark and individual stocks.
5) Compute individual overperformance or underperformance for each stock against the benchmark.

C. Knowledge Graph Construction and Querying

A knowledge graph was constructed by hand with the help of subject matter experts with financial services industry experience. The knowledge graph aims to capture key relationships in the movement of stock prices. There are many reasons for a stock's price to change, so we constructed a knowledge graph that captured the highest leverage causes that are commonly agreed upon. We began by collecting many triplets, and then applied entity resolution to identify and merge duplicate entities, ensuring that our dataset contained unique and consistent entries. Our final graph has 42 unique entities, and 68 edges. The nodes with the highest degree in our graph are

1) stock prices
2) corporate investment
3) consumer spending
4) interest rates
5) earnings per share

Some example triplets are

1) (analyst) - (issues) - (upgrade)
2) (earnings per share) - (drive) - (stock prices)
3) (share buybacks) - (increase) - (earnings per share)

We used Neo4j and LlamaIndex, to store and query our knowledge graph. When querying our knowledge graph, we used a hybrid retrieval mode that combined both semantic and keyword search. For a given query, we fetched the top 5 most relevant triplets.

D. Context Construction and Querying

In addition to our knowledge graph, we included company-specific information for additional context. We used the following data for context construction:

1) Company summary: Ticker, exchange (NYSE, Nasdaq, BATS), short description, country, currency, sector, and industry.
2) Earnings data: Historical annual and quarterly earnings. Quarterly earnings include report date, reported earnings per share, and expected earnings per share.
3) News articles: Title, url, publication date, and summary. Articles are filtered by relevance (Only for 2022).

All of the above data was obtained using the Alpha Vantage API. When analyzing model performance using supplementary context, we used a ticker lookup to fetch the above data for a given company. Specifically, when processing data for Apple stock, we searched our database to get Apple's company summary, earnings data, and news articles. No semantic search was used at this stage.

We observed minor data quality issues during the retrieval of news articles. For the following tickers the Alpha Vantage API returned an "invalid input" error:

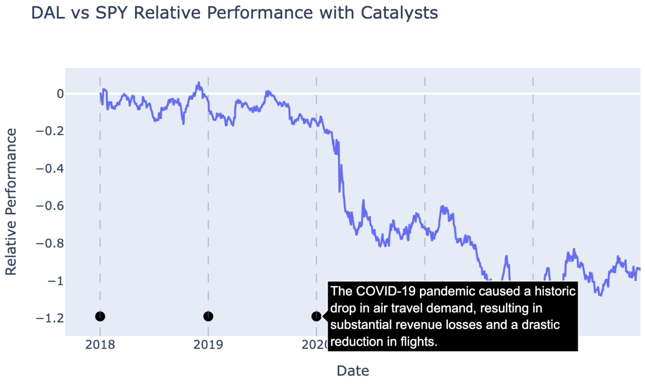
1) BRK.B (Berkshire Hathaway)
2) JPM (JPMorgan Chase & Co)
3) MTB (M&T Bank)
4) HBAN (Huntington Bancshares)
5) BF.B (Brown-Forman)
6) CE (Celanese)

For Berkshire Hathaway and Brown-Forman, we used their alternative tickers, BRK-A and BF-A respectively, to retrieve the relevant news articles. For the other companies listed above, there are no alternative tickers or listings, so it is unclear why the API fails to return data. As of the time of writing we are investigating this issue. Still, given that news data is missing for only 4 companies out of 500, we have good coverage here.

E. Prompt

In this section we describe the prompt used for the task. At a high level, our prompt asked the model to generate a "catalyst" for each stock's performance in every year from 2018 to 2022. We asked the model to produce a sentence-long catalyst for each year.

This format makes it easy to generate plots where the model's output labels are displayed over price data. For human evaluation, this is a quicker and more intuitive way to measure performance (Figure 1).



The COVID-19 pandemic caused a historic drop in air travel demand, resulting in substantial revenue losses and a drastic reduction in flights.

Fig. 1. Example Labels

For consistency in output, we enforced a JSON schema. Our JSON schema for model output is

```
{
    "catalysts": [
        {
            "date": "string",
            "catalyst": "string"
        }
    ]
}
```

Initially, we tried different methods for label placement, like placing labels on dates where a stock's price deviated from a moving average or when the the daily price movement was much larger than usual. When trying to place labels based on such metrics, we found that it was difficult to isolate idiosyncratic changes in a company's performance from broader market trends or sector-wide movements. When asked to label the type of events described above, the model tends to produce general labels like "growth stocks underperform."

Our complete prompt, as well as all of our code, is available online at our GitHub repository .

F. Evaluation

Finally, we analyzed our results using human annotation. Our human annotators, subject matter experts with at least 5 years of experience in the asset management industry, analyzed model performance over the given metrics:

1) Accuracy: Rate the truthfulness of the model's performance. For example, determine if a given label is a hallucination.
2) Reasoning: Evaluate whether the label reasonably explains relative performance during a given period. For example, if a stock massively outperforms its benchmark over a given period, then we would expect the catalyst over that period to be *positive*.

For each year in the period, the evaluator was asked to generate a score from 0-1 (inclusive) for both accuracy and reasoning. For example, they might say that a label for Apple's 2018 performance had 0.95 accuracy and 0.90 reasoning. When aggregating results, we sum over the yearly labels. So for a given stock, the model's total accuracy score is the sum of its accuracy in each year from 2018 to 2022. Therefore, the total possible accuracy score is 0-5. We measure reasoning scores using the same method.

We utilized two different language models, and two different context setups. Our context setups were baseline, and knowledge graph plus company-specific context. Our two human annotators were given all outputs from each model and parameter combination. The model and parameters for a given output were hidden from the annotators.

We tested two LLM models under two different data augmentation settings. Therefore, we had 4 different scenarios, and 500 total companies in each scenario. In order to maintain human level model evaluation, while keeping the evaluation stage tractable, we sampled a random subset of our data for each scenario.

We desired a 95% confidence interval and assume a 5% margin of error. For our population size of 500, we had to sample $n = 218$ different data points. Our random sample included at least 30 companies from each sector of the S&P 500 (Table 1). This gave us a reasonably large sample, allowing us to isolate and compare model performance by sector.

Sector	**Count**
Manufacturing	82
Technology	72
Trade & Services	72
Finance	62
Energy & Transportation	62
Life Sciences	54
Real Estate & Construction	32

TABLE I
SAMPLE SECTOR COUNTS

III. RESULTS AND ANALYSIS

A. Results

Overall, our results show that GPT-4o outperforms GPT-4o-mini in both accuracy and reasoning, in both the baseline and data augmented scenario (Table II and Table III). The magnitude of the improvement is small in every case. In the most extreme scenario, of baseline data and reasoning assessment, GPT-4o outperforms GPT-4o-mini by just 0.12 points.

TABLE II
ACCURACY

	Baseline	**Knowledge Graph**
GPT-4o-mini	3.84	3.85
GPT-4o	3.94	3.92

TABLE III
REASONING

	Baseline	**Knowledge Graph**
GPT-4o-mini	3.48	3.48
GPT-4o	3.60	3.54

B. Aggregation

In this section we show our analysis of how the models performed on subsets of our data. Specifically, we aggregate performance by features like year, market capitalization, or market beta.

When aggregating by year, we found that both models, and both contexts, perform better on accuracy and reasoning when asked to identify catalysts for 2020 (Table IV and Table V).

Year	Accuracy	Reasoning
2018	0.73	0.63
2019	0.79	0.72
2020	0.85	0.78
2021	0.76	0.69
2022	0.72	0.65

TABLE IV
GPT-4O MINI: ACCURACY AND REASONING BY YEAR

Year	Accuracy	Reasoning
2018	0.76	0.66
2019	0.79	0.72
2020	0.85	0.79
2021	0.78	0.70
2022	0.76	0.69

TABLE V
GPT-4O: ACCURACY AND REASONING BY YEAR

Aggregating by market capitalization, market beta, and price-earnings ratio, did not produce statistically meaningful differences in accuracy or reasoning performance. Accuracy and reasoning are highly correlated across all parameter settings. The correlations are summarized below (Table VI, VII, VIII, IX).

mini (context)	accuracy	reasoning	beta	pe_ratio	market_cap
accuracy	1.00	0.89	0.02	0.04	0.21
reasoning	0.89	1.00	0.09	0.05	0.24
beta	0.02	0.09	1.00	-0.15	-0.01
pe_ratio	0.04	0.05	-0.15	1.00	0.02
market_cap	0.21	0.24	-0.01	0.02	1.00

TABLE VI
Correlation Matrix for Mini (Context)

mini	accuracy	reasoning	beta	pe_ratio	market_cap
accuracy	1.00	0.88	-0.04	0.04	0.14
reasoning	0.88	1.00	0.00	0.02	0.18
beta	-0.04	0.00	1.00	-0.15	-0.01
pe_ratio	0.04	0.02	-0.15	1.00	0.02
market_cap	0.14	0.18	-0.01	0.02	1.00

TABLE VII
Correlation Matrix for Mini

gpt-4o (context)	accuracy	reasoning	beta	pe_ratio	market_cap
accuracy	1.00	0.87	-0.02	-0.01	0.19
reasoning	0.87	1.00	0.02	-0.03	0.24
beta	-0.02	0.02	1.00	-0.15	-0.01
pe_ratio	-0.01	-0.03	-0.15	1.00	0.02
market_cap	0.19	0.24	-0.01	0.02	1.00

TABLE VIII
Correlation Matrix for GPT-4o (Context)

gpt-4o	accuracy	reasoning	beta	pe_ratio	market_cap
accuracy	1.00	0.88	0.06	0.02	0.10
reasoning	0.88	1.00	0.08	0.01	0.14
beta	0.06	0.08	1.00	-0.15	-0.01
pe_ratio	0.02	0.01	-0.15	1.00	0.02
market_cap	0.10	0.14	-0.01	0.02	1.00

TABLE IX
Correlation Matrix for GPT-4o

IV. Future Work

In order to better capture the performance of a given stock, we could measure performance against a factor model instead of the S&P 500 index. For example, in 2024 Intel has underperformed the S&P 500 index, but in the context of recent semiconductor and growth performance, Intel's underperformance is magnified.

It is likely that additional contextual data, like historical commodity prices or CPI announcements, could enhance model performance. For this reason, future work could add additional datasets and implement semantic or sector-specific RAG to retrieve relevant context given a stock.

Given more time and resources to recruit a wider range of subject matter experts, we could assign annotators a subset of the data that aligns with their expertise within finance. For example, a hedge fund analyst who covers the healthcare space could label stocks specifically in the healthcare industry.

Finally, we could experiment with the latest generation of large language models, like OpenAI's o1 model. Similarly, we could fine-tune a language model to perform our labeling task. Given our initial results presented in this paper, we have a small but high quality dataset for fine-tuning.

V. Limitations

One major limitation with our study is that it relies on human evaluation. Prior work has shown that human evaluation is prone to bias based on the assertiveness of model output [5]. In this paper, we did not attempt to control for human bias in evaluating model performance.

In terms of data quality, we were limited to easily accessible and inexpensive datasets. This limited our ability to programmatically access highly curated and trusted data sources for financial news, such as Bloomberg, The Wall Street Journal, or the Financial Times.

Similarly, we were not able to leverage internal datasets that financial industry practitioners might use. Examples of interesting proprietary datasets that might improve model performance are expert-network call transcripts, internal macroeconomic forecasts, and social media sentiment analysis.

VI. Conclusion

In this paper, we present our use of LLMs to label stock price data. We found that the newer generation of language models performs this task reasonably without any fine-tuning or data augmentation. We also find that our specific attempt to improve model performance, using a knowledge graph and relevant stock context, was unsuccessful. Still, our results show that the models performed better at their task during the calendar year 2020, a time when the COVID-19 pandemic was an obvious catalyst for stock prices. Finally, we show that model performance was consistent across sector, market capitalization, market beta, and price-earnings ratios, suggesting that the models do not perform any better on subsets of assets. This paper lays the groundwork for future research in labeling time series data, specifically in the financial sector.

References

[1] X. Zhang, R. R. Chowdhury, R. K. Gupta, and J. Shang, "Large language models for time series: A survey," 2024.

[2] E. Blomqvist, M. Alirezaie, and M. Santini, "Towards causal knowledge graphs - position paper," in *KDH@ECAI*, 2020.

[3] H. Abu-Rasheed, C. Weber, and M. Fathi, "Knowledge graphs as context sources for llm-based explanations of learning recommendations," *2024 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1–5, 2024.

[4] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Trans. on Knowl. and Data Eng.*, vol. 36, p. 3580–3599, Jan. 2024.

[5] T. Hosking, P. Blunsom, and M. Bartolo, "Human feedback is not gold standard," 2024.