Southern Adventist University

Senior Project

Algorithmically Trading like a Human with GPT-J

Joel PECKHAM
EMAIL: joelskyler@gmail.com

supervised by
Willard MUNGER, PhD.
Robert ORDÓÑEZ, M.S.

Abstract

We explore the viability of using GPT-J, a 6 billion parameter natural language processing model, as a stock market trading indicator. This novel approach leverages both GPT-J's ability to process large, data-rich inputs, and GPT-J's deep understanding of the world. We study the viability of using GPT-J as a natural language based stock predection algorithm by creating and evaulating a small scale test dataset. Our test dataset produced no significant results.

December 6, 2021

CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. Introduction

A. Background

Algorithmic trading has become the dominant way of buying and selling securities. In the U.S. stock market, algorithms account for 70-80% of trading volume [1]. The current generation of trading algorithms now use neural networks to improve prediction accuracy. However, some believe neural networks are incapable of inventing novel trading ideas and therefore can only increase trading efficiency by approximately 10% [2]. On the other hand, we propose that recent increases in the scale of neural networks can unlock the ability to algorithmically invent winning trades.

Specifically, we propose that large-scale natural language processing models such as OpenAi's GPT-3 [3] have captured a large enough understanding of the world, that given the proper input, they can generate novel, winning trades. Transformer networks of this type are designed to both ingest and output text. Taking into account the impressive knowledge displayed by GPT-3 and others in text-completion tasks, we propose that with adequate fine-tuning and well engineered prompts, such networks can learn to trade like a human by reading the news, social media, and other sources such as SEC filings. In theory, by properly engineering our input text prompts, we can leverage the massive amount of real-world knowledge encoded in the network's weights.

Because of the complications inherit in working with OpenAi and licensing GPT-3, we have instead chosen to use the open-source project GPT-J [4]. GPT-J is a transformer network [5] that has been trained on an 825GiB language modeling data set called The Pile [6]. Compared to GPT-3-Davinci's 175 billion network parameters, GPT-J's 6 billion parameters seems small, however it has proved itself to be capable enough to outperform GPT-3 in some tasks such as code generation [7]. Therefore, we believe that GPT-J will be sufficient to explore the viability of our novel approach to trading.

B. Problem Statement

As algorithmic traders seek to employ increasingly competitive and complex trading algorithms, we propose to study the viability of using GPT-J as a natural language based stock predection algorithm. This work will provide a foundation for developing and testing such an algorithm by building and evauluating a small-scale test dataset.

II. Review of Literature

A. Previous Methods

Many neural network structures and methods have been used to create trading algorithms. Some types include:

- Recurrent neural networks (RNNs) & Long short-term memory (LSTMs) [8][9].
  These are the most widely used network architectures for trading [10]. An LSTM trained on 900,000 sequences of length 30 days of Chinese stock market data yielded an improvement of 12.9% in prediction accuracy over a random guess [11].
- Convolutional Neural Networks (CNNs) [10]
  CNNs can be used by converting time-series data into images [12]. Or they can be used to extract sentiment features from text [13].
- Deep reinforcement learning (DRN).
  - Deep Q-learning [14] [15].
  - Deep robust reinforcement learning [16].
- Conventional deep learning [17].
- Transformer networks [18].

Most relevant to our work are methods that incorporate sentiment analysis of news sources. [9] evaluated sentiment analysis methods and found that LSTMs could correctly classify news tweets as indicative of positive or negative price movement 92% of the time. [15] found that adding sentiment analysis to a Deep-Q learning algorithm could improve the Sharpe Ratio (a measure of profit as compared to risk) of the agent by a factor greater than 2 in their test cases.

B. Relevance & Novelty

By our estimation, the vast majority of previous works involving sentiment analysis used a pre-processing step to extract sentiment from the news, and then embedded those features into a time-series dataset. News sources were often limited to headlines, tweets and small snippets because of the memory limitations of RNN sentiment classifiers. With the introduction of large transformer networks [5], capable of processing large amounts of text like OpenAi's GPT-3 [3] or Wang & Komatsuzaki's GPT-J [19], we believe a new class of trading network can be created. Readily available implementations of GPT-J allow for inputs with up to 2048 input tokens or words. Our method will encode the current world state in a large text-input which combines sentiment, real-world facts, and stock price data into a single input. We believe that combining rich input information with the network's pre-existing understanding of the world, the network will be able to produce significantly better trading signals than previous methods.

III. Long-term Plan

To guide our execution, we have outlined the goals, requirements, steps, and processes we intend to implement in the long-term. This road-map will guide the creation and evaluation of our small-scale test dataset.

A. Requirements & Goals

1) Functional requirements (user stories):

- As finance researchers, we want to quantify the ability of news releases, SEC filings, and social media posts to move stock prices.
- As AI researchers, we want to evaluate the viability of using GPT-J as a stock movement indicator so that we can understand the power of GPT-J to understand complex real-world interactions.

- As algorithmic traders, we want to evaluate the viability of using GPT-J as a stock movement indicator so that we can make more-informed trading decisions.

2) Non-functional requirements: Our overall flow of execution as outlined in figure 1 is as follows:

- We will evaluate the correlations between the following items:
  a) The release of SEC filings for company X and large movements in the stock price of company X.
  b) The release of news stories mentioning company X and large movements in the stock price of company X.
  c) The posting of tweets or other social media posts mentioning company X and large movements in the stock price of company X.
- We will evaluate the best ways of formatting prompts for GPT-J to increase output accuracy and consistency across varying inputs. Some options might include:
  a) Providing a form for GPT-J to fill out appended to the end of the input data.
  b) Asking GPT-J a direct question appended to the end of the input data.
  c) Appending a universe current stock prices to the beginning of the input.
  d) Appending multiple news stories from the past days and weeks at the beginning of the input.
- We will deploy the best model from my previous evaluations and test it on live stock market data. We will compare the model's performance against market indices like the S&P 500.

B. Dataset Options

This section outlines the available options for dataset creation. Although, we do not currently have the resources to create a dataset for each of these options, we have left them here as an aid for future work.

1) Stock Price Data: Stock price data is publicly available Yahoo Finance API [20]. This source can be used to gather general technical information about a given ticker symbol. For historical price data, the Alpaca Data API v2 [21] can be used. This API gives access to 5 years of historical data for training and live price data for live model inference testing.

2) SEC Filings: The SEC provides public access to their EDGAR database of public company filings [22]. The EDGAR API can be used to download the most recent SEC filings for a given company for live trading. Or for training, bulk datasets are available for download. The release dates of each filing can then be labeled with the stock prices of the company before and after the filing.

3) News: Theoretically, a huge amout of financial news can be scraped from the following sources, although developing web-scrapers for each source is a massive time commitment. However, a complete dataset would include a backlog of articles from the sources listed in Table I.

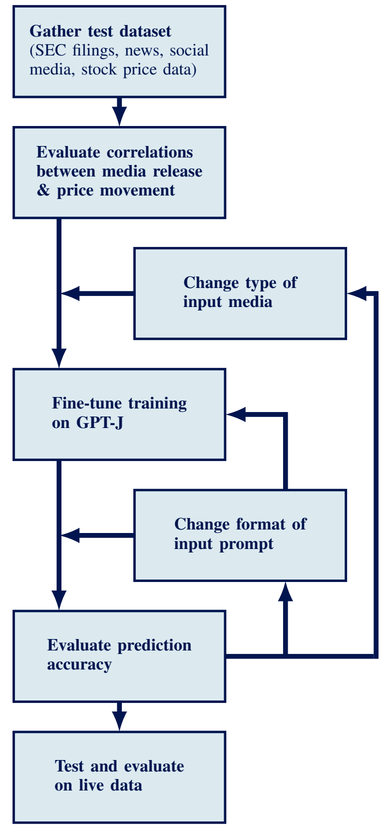

Fig. 1. Execution Flow

TABLE I
NEWS SOURCES

Source	URL
Bloomberg	www.bloomberg.com
CNBC Finance	www.cnbc.com/finance
Reuters	www.reuters.com
Forbes	www.forbes.com
CNN Business	www.cnn.com/business
Yahoo Finance	www.finance.yahoo.com/news
Wall Street Journal	www.wsj.com/

4) Social Media: Twitter is the best option for social media data because the Twitter API [23] will allow us to gather timely opinions on a given ticker symbol. After filtering for a specific time period, tweets can be added to our training dataset.

C. Testing & Evaluation Plan

1) Media Release Correlations: To measure the correlation between the release of a media item such as an SEC filing or news story, we will use the standard event study method as detailed in [24]. This method uses abnormal returns in a given period to calculate the effect of a certain event on a stock's price. Abnormal return for a given day is defined as

$$AR_{i,t} = R_{i,t} - (\alpha_i + \beta_i R_{m,t}) \tag{1}$$

for firm i at time t where α_i and β_i represent the relationship between a given stock and it's reference market. And where $R_{m,t}$ is the return of the actual reference market. This method uses the historical relationship of a stock to it's reference market to estimate normal returns. Abnormal returns are therefore the difference between the actual return of the stock and the normal return of the stock.

We can then take large sample of media release events of the same type and calculate the average abnormal return as follows:

$$AAR = \frac{1}{N}\sum_{i=1}^{N} AR_{i,t} \tag{2}$$

Finally, we can measure the total impact of the event over a given period of time by using cumulative abnormal return:

$$CAR(t_1, t_2) = \sum_{t=t_1}^{t_2} AR_{i,t} \tag{3}$$

where t_1 and t_2 are the start and end dates of the event window.

Because we are limited in the granularity of data we have access to and can easily process, our research will focus on day-by-day events, returns, and correlations.

2) GPT-J Model prediction accuracy: To evaluate the accuracy of our model given different input data, we will use two metrics. First is a simple ratio of well-formed, parseable outputs to malformed outputs. This first metric we refer to as format-correctness.

$$F_c = \frac{N_{parseable}}{N_{malformed}} \tag{4}$$

The second metric is a simple ratio of the actual stock price to the predicted stock price. This second metric we refer to as price-accuracy.

$$P_a = \left|\frac{P_{actual}}{P_{predicted}}\right| \tag{5}$$

We will use the price accuracy metric as our loss-function in fine-tune training of GPT-J. The format-correctness metric will be used to evaluate the reliability and robustness of our prompt. We may experiment with using the format-correctness metric as a weighted portion of our loss function.

IV. Small-Scale Implementation

A. Data Gathering

1) News Data: Using the freely available New York Times Archive API Citation Needed. We downloaded the associated metadata for each New York Times article from January 2006 through October 2022. This subset of the New York Times archive consists of 1,446,289 articles. We then filtered the articles based on the keywords associated with the article, to get articles relevant to a company in our list of top 50 U.S. companies. Our resulting dataset contains 23428 unique articles from the Times.

B. Stock Data Gathering

The next step was to collect a dataset of historical stock prices that can be correleated with the news articles at our disposal. Using the Alpaca Data API v2 [21], we downloaded the historical stock prices for our top 50 U.S. companies.

TABLE II
Stock Sources

Company	Ticker
Apple	AAPL
Microsoft	MSFT
Alphabet	GOOG
Amazon	AMZN
Facebook	FB
Tesla	TSLA
Berkshire Hathaway	BRK.A
Nvidia	NVDA
JP Morgan	JPM
Visa	V
JOHNSON & JOHNSON	JNJ
United Health	UNH
Walmart	WMT
Bank of America	BAC
Home Depot	HD
Mastercard	MA
PROCTER & GAMBLE	PG
Disney	DIS
Adobe	ADBE
Salesforce	CRM
Netflix	NFLX
Exxon	XOM
Oracle	ORCL
Nike	NKE
Comcast	CMCSA
Coca-Cola	KO
Cisco	CSCO
Pfizer	PFE
Fisher Scientific	TMO
Accenture	ACN
Eli Lilly	LLY
Intel	INTC
Pepsi	PEP
Verizon	VZ
Danaher	DHR
Chevron	CVX
Abbott Laboratories	ABT
Broadcom	AVGO
Costco	COST
Merck	MRK
Wells Fargo	WFC
Abbvie	ABBV
Morgan Stanley	MS
AT&T	T
McDonald's	MCD
Texas Instruments	TXN
Medtronic	MDT
UPS	UPS
Nextera	NEE

C. Data Preprocessing

Using the Alpaca API, we were limited to 5 years of past data. This means we must filter our news dataset to start in November of 2016. We were then able to attach stock data with 8,053 news articles over 1,222 weeks 2. The dataset was naturally skewed to a certain few companies like Facebook, Netflix, and Apple which account for over 45% of the news articles in our event dataset dispite them only being 3/50 companies or 6% of the companies inlcuded in our news search 3.

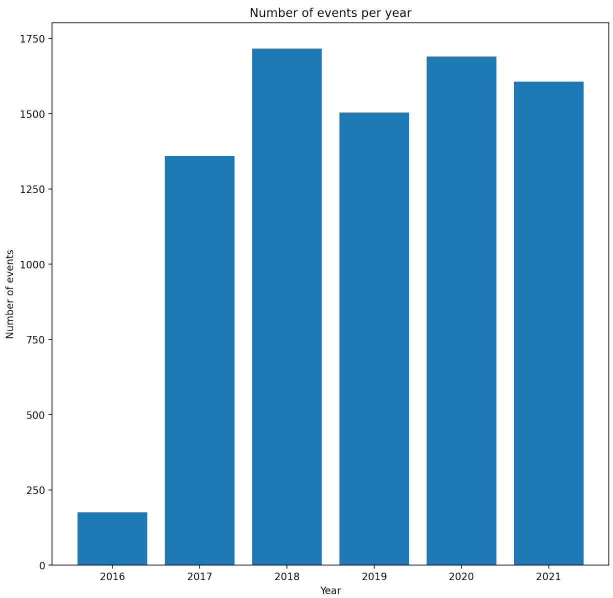

Fig. 2. Number of Events/Year

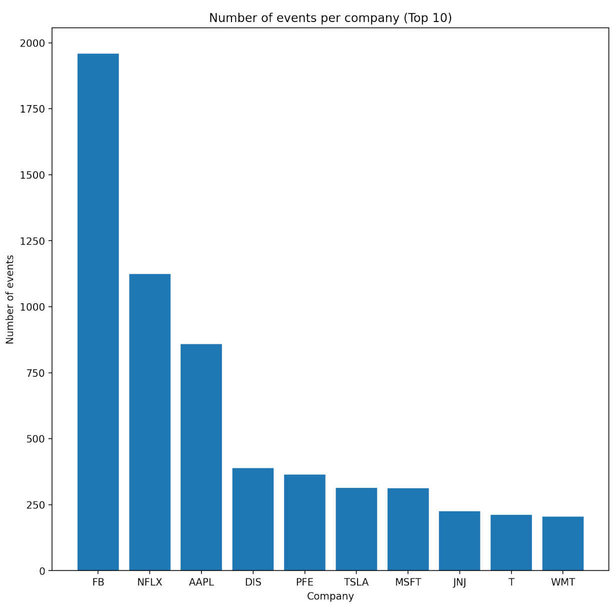

Fig. 3. Number of Events/Company (Top 10)

V. Results

A. Event Study

1) Standard Methodology: Our event study had the goal of invetsigating if the news releases in our datset were correlated with abnormal returns in stock price. We use the event study methodology as described above in section III-C1 for each stock in our dataset. Figure 4 shows the expected return of Facebook compared to the its actual retrun and reference index. Averaging over all the stocks in our dataset, the event study methodology showed that the events in our dataset were not correlated with abnormal returns in stock price.

Fig. 4. Facebook: Actual vs. Expected Return

2) Simple Correlation Test: To confirm our suspicions that our event data in fact had no correlation with price movements, we ran a simple correlation test on a frequecy of news releases for a given company and the absolute acceleration of stock price. Figure 5 gives an example of this data on Microsoft. The average correlation coefficient over all stocks is 0.035, which means that there is no significant correlation between the news releases and the stock price in our dataset.

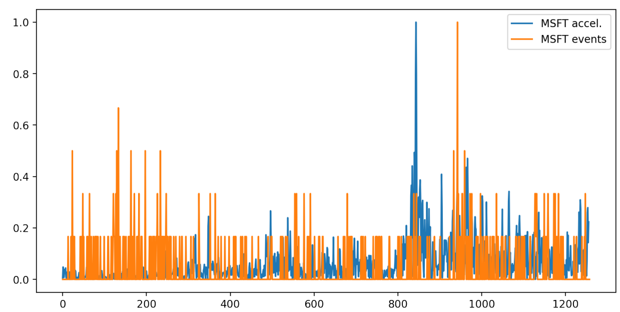

Fig. 5. Abs. Acceleration of MSFT vs. News Releases

B. Inference

Running inference on models of this scale requires large amounts of parallel compute. With the limited resources of this study, renting Google TPU time to set up our own training and inference environment was intractable. However, online services such as HuggingFace, provide APIs to interface with pre-trained models. However, even this service is expensive, so we were unable to perform fine-tune testing. We were also limited to only performing inference on a small subset of our dataset: about 4,000 events.

1) Format-Correctness Metric: Suprisingly, GPT-J was always able to retrun a parseable output given our test prompt. **Give example prompt.** We believe that adding a "$" to the end of the prompt made GPT-J more likely to return a parseable output.

2) Price-Accuracy Metric: Our limited inference method was unable to predict stock prices with any accuracy. In our dataset, a simple linear regression over the past week of stock prices was able to generate a prediction of the next day's price 52% of the time. By comparision, the GPT-J model was correct

47.5% of the time. Both of these numbers are not significantly better from a random guess.

VI. Conclusion

In conclusion, our study was unable to determine if language models like GPT-J contain any predective power in the stock market. Further study is needed to determine if natural-language-processing based models can be used to predict stock prices.

A. Future Work

Future work to improve the significane of our results will include:

- Including more data sources (news, and other) to increase the number of events we can analyze.
- Including more stocks in our analysis.
- Varying prompt format.
- Increasing the size of our inferernce dataset.

References

[1] H. Samuelsson, "What percentage of trading is algorithmic?" 2021. [Online]. Available: https://therobusttrader.com/what-percentage-of-trading-is-algorithmic/

[2] D. Vonko, "Neural networks: Forecasting profits," 2021. [Online]. Available: https://www.investopedia.com/articles/trading/06/neuralnetworks.asp

[3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[4] B. Wang, "Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX," https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[6] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The pile: An 800gb dataset of diverse text for language modeling," *CoRR*, vol. abs/2101.00027, 2021. [Online]. Available: https://arxiv.org/abs/2101.00027

[7] "Gpt-j use cases." [Online]. Available: https://www.helloforefront.com/use-cases

[8] G. Chen, Y. Chen, and T. Fushimi, "Application of deep learning to algorithmic trading," 2017. [Online]. Available: http://cs229.stanford.edu/proj2017/final-reports/5241098.pdf

[9] P. Mehta, S. Pandya, and K. Kotecha, "Harvesting social media sentiment analysis to enhance stock market prediction using deep learning," *PeerJ Computer Science*, vol. 7, p. e476, Apr. 2021. [Online]. Available: https://doi.org/10.7717/peerj-cs.476

[10] Y. Gu, T. Shibukawa, Y. Kondo, S. Nagao, and S. Kamijo, "Prediction of stock performance using deep neural networks," *Applied Sciences*, vol. 10, no. 22, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/22/8142

[11] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 2823–2824.

[12] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach," *Applied Soft Computing*, vol. 70, pp. 525–538, Sep. 2018. [Online]. Available: https://doi.org/10.1016/j.asoc.2018.04.024

[13] Y. Shi, Y. Zheng, K. Guo, and X. Ren, "Stock movement prediction with sentiment analysis based on deep learning networks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 6, Nov. 2020. [Online]. Available: https://doi.org/10.1002/cpe.6076

[14] Y. Wang, D. Wang, S. Zhang, Y. Feng, S. Li, and Q. Zhou, "Deep q-trading," 2017. [Online]. Available: http://cslt.riit.tsinghua.edu.cn/mediawiki/images/5/5f/Dtq.pdf

[15] A. Nan, A. Perumal, and O. R. Zaiane, "Sentiment and knowledge based algorithmic trading with deep reinforcement learning," 2020.

[16] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, vol. 7, pp. 108 014–108 022, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2932789

[17] M.-Y. Day and C.-C. Lee, "Deep learning for financial sentiment analysis on finance news providers," in *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, Aug. 2016. [Online]. Available: https://doi.org/10.1109/asonam.2016.7752381

[18] J. Schmitz, "Stock predictions with state-of-the-art transformer and time embeddings," Sep 2020. [Online]. Available: https://towardsdatascience.com/stock-predictions-with-state-of-the-art-transformer-and-time-embeddings-3a4485237de6

[19] B. Wang and A. Komatsuzaki, "GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model," https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[20] [Online]. Available: https://www.yahoofinanceapi.com/

[21] [Online]. Available: https://alpaca.markets/docs/api-documentation/api-v2/market-data/alpaca-data-api-v2/

[22] "Edgar application programming interfaces," Jun 2021. [Online]. Available: https://www.sec.gov/edgar/sec-api-documentation

[23] [Online]. Available: https://developer.twitter.com/en/docs/twitter-api

[24] A. Neuhierl, A. D. Scherbina, and B. Schlusche, "Market reaction to corporate press releases," *SSRN Electronic Journal*, 2010. [Online]. Available: https://doi.org/10.2139/ssrn.1556532