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

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

October 26, 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 incapabale of inventing novel trading ideas and therefore can only increase trading efficiency by 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 undertanding 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. We propose that by properly enginnering our input text prompts, we can leverage the massive amount of real-world knowledge encoded in the network's weights.

Becuase of the complications inherit in working with OpenAi and licencing 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. Intent

We propose that with adequate fine-tuning and well engineered prompts, GPT-J can learn to trade like a human by reading the news, social media, and other sources such as SEC filings.

We will first implement a trading algorithm based on GPT-J. Then we will evaluate the algorithm's prediction accuracy and compare it to the accuracy of similar algorithms. Optimizing our algorithm will involve multiple iterations of fine-tune network training, prompt engineering, and accuracy analysis. Finally, we will conduct a profitability analysis of our algorithm using backtesting and live data.

C. Problem Statement

In summary, as algorithmic traders seek to employ increasingly competitive and complex trading algorithms, we intend to evaluate the viability of using GPT language models to drive trading decisions. To achieve this goal we must first implement a trading algorithm based on GPT-J, then evaluate the accuracy of the algorithm's predictions, and finally evaluate the real-world profitability of the algorithm.

II. Related Works

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. Mehta et al. (2021) evaluated a sentiment analysis methods and found that LSTMs could properly classify news tweets as indicative of positive or negative price movement with an accuracy of 92% correct [9]. Nan & Zaiane (2020) found that adding sentiment analysis to a Deep-Q learning algorithm could improve the Sharpe ratio of the agent by a factor greater than 2 in their test cases. [15].

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. 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 this new network can be trained to learn to trade like a human.

III. Execution Plan

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 I 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. I will compare the model's performance against market indices like the S&P 500.

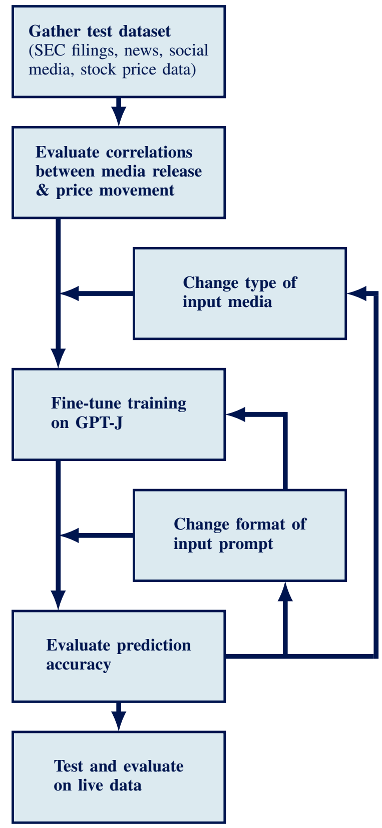

Fig. 1. Execution Flow

B. Dataset Gathering

1) Stock Price Data: We plan to use the publicly available Yahoo Finance API [20] to gather general technical information about a given ticker symbol. For historical price data, we will use the Alpaca Data API v2 [21]. 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: We will scrape financial news websites for a large set of historical articles. We intend to create a backlog of articles from the sources listed in Table I.

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: We will focus on Twitter for our social media data. 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 Methods

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 index. And where $R_{m,t}$ is the return of the actual reference market.

We will 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 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.

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, parse-able outputs to malformed outputs. This first metric we refer to as format-correctness. 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.

IV. Implementation

V. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

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