Sentiment Analysis for Meme Stock Price Prediction

Stanford CS224N Custom Project

Zihan Qiang
Department of Computer Science
Stanford University
zqiang@stanford.edu

Jingyu Shen
Department of Computer Science
Stanford University
jingyu.shen@stanford.edu

Yu Dong Zhang
Department of Computer Science
Stanford University
lz927@stanford.edu

Abstract

Market sentiment is one of the most significant driving forces in the financial market nowadays, and such sentiment is mostly displayed via social media platforms that allow people to freely share ideas, even without proper regulations in certain cases. However, discussions and comments from social media are often expressed in a very informal setting where internet slangs, jargons, sarcasm, frequent typos, and even incoherent phrases can often appear. Here we present an NLP model called MemeBERT, built by adjusting configurations of FinBERT [1] and further pre-training using Reddit and Twitter text data to capture sentiment from highly informal text. A neural network, with heavy uses of Long Short-Term Memory networks, is later developed to generate trading decisions based on the outputted sentiment scores along with technical indicators. The generated strategies, compared to the strategy of passively holding stocks, have generally reached higher returns and Sharpe ratios on the hold-out test set.

1 Key Information to include

- TA mentor: Angelica Sun
- External collaborators: No
- External mentor: No
- Sharing project: No

2 Introduction

Stock prices nowadays depend heavily on market sentiments because such sentiments indicate the market players' views of how the financial market will evolve, and thus most of the changes in stock prices can be interpreted as a result of changes in such views. Much of the market research conducted by large financial institutions, such as hedge funds, investment banks, and asset managements, are attempting to infer market sentiments via technical analysis, a trading discipline of analyzing statistical trends gathered from trading activity, such as price movement and volume [2]. Nevertheless, very few such institutions are directly modelling market sentiments via the form of financial corpora (e.g. financial reports, press news, forums and discussions) due to the inherent complexity of human languages.

Inferring market sentiments using text data has not been gaining much popularity until a most recent event of a price surge on GME (GameStop) stock in late January 2021 [3]. Prior to the price surge,

investors from the social media site Reddit had discovered that many institutional investors are betting against GameStop by taking a short position on its stocks. This discovery then led to a mass-buying of GME stocks by Reddit users, causing a fourteen-fold price increase. Many institutional investors have suffered heavy losses on this short squeeze and it is now clear that sentiment analysis is indispensable to price movement predictions.

Natural language processing is therefore a natural choice people would turn to in order to discover sentiments from financial text data. However, language modelling in the financial domain is a rarely touched upon area, and much of the NLP literature are focusing on tasks outside of such domain, making it difficult to use transfer learning to finance-related down-steam tasks. Moreover, even the majority of NLP research conducted on the financial domain are only focusing on formal texts, such as language models on financial reports and news articles, and so far little attention has been paid to informal texts like Reddit and Twitter posts and comments. This project aims to develop an end-to-end deep learning model that incorporates both language models and neural network that is trained on financial corpora to generate trading decisions of four popular meme stocks, GameStop (GME), American Multi-Cinema (AMC), Bitcoin (BTC), and Ethereum (ETH).

3 Related Work

Many research has been done to pre-train word embedding and language model weights. ELMo (Embeddings from Language Models) [4], a deep contextualized word representation model, is one of the first and most successful attempt at pre-training word embedding. ELMo is developed to capture complicated features of language uses such as syntax and semantics. Word vectors of ELMo are learned functions of all of the internal layers of a deep bidirectional language model (biLM), which is pretrained on a large text corpus. ELMo can be added to existing state-of-the-art language models and provide better performance in the tasks of question answering, textual entailment, and sentiment analysis. Furthermore, ULMFiT (Universal Language Model Fine-tuning for Text Classification)[5], compared to ELMo which only provides weights for the first layer, i.e. the word embeddings, is one of the first and most successful attempts at pre-training a whole language model. ULMFiT has adopted novel techniques of discriminative fine-tuning, slanted triangular learning learning rates, and gradual unfreezing.

In the most recent natural language processing literature, the language representation model of BERT (Bidirectional Encoder Representations from Transformers)[6] has become the single most popular off-the-shelf pre-trained model, and can be widely adopted for down-stream tasks like question answering and language inference with minimal further pre-training. Low-resources tasks, including sentiment analysis in the financial domain, can benefit from the pre-trained models of BERT. Based on BERT, FinBERT [1] has made an attempt to further pre-train the BERT model for uses in the financial domain. The FinBERT model is a further pre-trained version of BERT, which uses almost the exact same model structure as BERT, while only adding a dense layer after the last hidden state of the [CLS] token to accommodate the classification task. FinBERT model is further pre-trained on labeled financial text from TRC2-financial, a subset of Reuters' TRC2. With extensive experiments, Araci has concluded that FinBERT model can still benefit from further pre-trained on financial corpus without the issues of catastrophic forgetting, and is able to outperform state-of-the-art models of ULMFit and LSTM with ELMo.

4 Approach

This project is divided into two parts. The first part is devoted to developing a language model to predict sentiment scores of Reddit and Twitter texts, and the second part is for predicting stock price movement with the outputted sentiment scores along with technical indicators.

4.1 Sentiment Analysis

The FinBERT model as presented in [1] is a BERT model further pre-trained on financial sentiment corpora, which uses almost the exact same model structure as BERT, with an addition of a dense layer after the last hidden state as shown in Figure 1 to accommodate the classification task.

We started with the FinBERT model to evaluate sentiments on Reddit data. In this project, we then developed the model MemeBERT, that is a further pre-trained and fine-tuned FinBERT model on labelled Twitter and Reddit corpora (described in 5.1.1) that are characterized by its abundance of internet slangs. Additionally, different ways of changing the configurations of FinBERT are explored to give better sentiment classification performance.

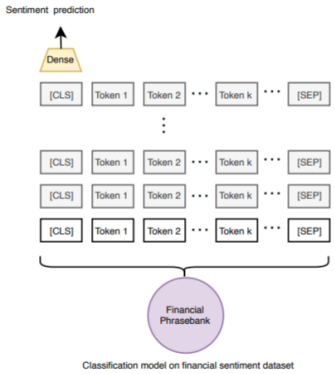


Figure 1: Finetuned FinBERT Structure [1]

4.2 Price Movement Prediction

The sentiment outputs of the MemeBERT model are then used along with price data (described in 5.1.1) as input features to a neural network with LSTM for predicting stock price changes. The output of the neural network is the label of price movement (a label of 1 means the close price is higher than the open price, and vice versa). Therefore, a label of 1 indicates a buy signal on the stock and a label of 0 indicates a sell signal. A simple trading strategy of buying with a positive signal and selling with a negative signal is used for evaluating the neural network performance.

5 Experiments

5.1 Data

5.1.1 Data for Sentiment Analysis

The dataset we use for pre-training and fine-tuning Finbert is a corpus of Twitter texts. Although our downstream task is to analyze sentiments from Reddit, it is not a platform that has abundant sources of data compared to Twitter. On the other hand, Twitter has always been the breeding ground for many sentiment analysis tasks of online forums because of its widely available data and highly diversified pool of users, including celebrities, politicians, and business entities. The labelled Twitter dataset is collected and integrated from Kaggle [7] with 1.6 million examples. Unlike the Financial Phrases Bank Dataset used in [1] for training FinBERT, the datasets we use are filled with internet slangs, and are better suited for our task of Reddit sentiment analysis. Table 1 provides a snippet of the Twitter data.

Text	Sentiment
ooooh.... LOL that leslie.... and ok I won't do it again so leslie won't get mad again	-1
At the mall with the fam. Gonna eat some japanese	1
This own goal sums up Maguire. Smh	-1

Table 1: Twitter Dataset

The input to our resulting MemeBERT model is the unlabelled Reddit data that are scrapped using the PushShift API [8]. Table 2 provides a snippet of the Reddit data. Note here that the sentiments for Reddit data are manually added for illustration purposes only.

Text	Sentiment
Gamestop (GME) is about to skyrocket and I'm disappointed in you fags for not noticing it	1
LMAO DFV bringing the long ladder to GME	1
At one point, I literally could and should have made $1M if I had just YOLOed.	1

Table 2: Reddit Dataset

The Reddit texts data are 30 Reddit posts everyday mentioning the stocks of interest, and are scrapped from a list of subreddits (forums dedicated to specific topics). The posts would need to have a score of greater than 200 to be scrapped. The score for posts serves as a popularity measure, and each Reddit user can like the post to boost up the score and dislike to do the reverse. This step helps filter out posts with unpopular opinions and less influence. Table 3 shows a summary of the collected data.

Word	Subreddit	Dates	Count
GME	GME	Feb. 1, 2021 - Mar. 1, 2022	273
GME	wallstreetbets	Feb. 20, 2020 - Mar. 3, 2022	512
gamestop	GME	Feb. 1, 2021 - Mar. 1, 2022	273
gamestop	wallstreetbets	Feb. 20, 2020 - Mar. 3, 2022	512
AMC	amcstock	Feb. 1, 2021 - Mar. 1, 2022	273
AMC	wallstreetbets	Feb. 20, 2020 - Mar. 3, 2022	512
BTC	CryptoCurrency	Feb. 3, 2020 - Mar. 2, 2022	759
bitcoin	CrytoCurrency	Feb. 3, 2020 - Mar. 2, 2022	759
ETH	CryptoCurrency	Feb. 3, 2020 - Mar. 2, 2022	759
ethereum	CryptoCurrency	Feb. 3, 2020 - Mar. 2, 2022	759

Table 3: Summary of Reddit Data

Note that the data are only collected for each trading day prior to the opening of the market (BTC and ETH are traded everyday whereas GME and AMC are only traded on weekdays).

5.1.2 Data for Price Movement Prediction

The second dataset is used to train the neural network that takes sentiment score and 11 technical indicators as inputs and predicts increase or decrease of the interday prices of given stocks. This dataset contains daily prices of the 4 meme stocks (i.e. GME, AMC, BTC, ETH) that span for 3 years from 2019 to present. The summary of the data is shown in the Table 4 (note here that the samples for BTC and ETH have more data points because they are also traded in weekends and holidays). By inspecting the summary the tables below, the datasets for price movement are relatively balanced in terms of their classes.

GME	Labels	Samples	Percentages
Train	0	250	45.7%
	1	210	54.3%
Test	0	29	55.8%
	1	23	44.2%

AMC	Labels	Samples	Percentages
Train	0	273	59.3%
	1	187	40.7%
Test	0	28	53.8%
	1	24	46.2%

BTC	Labels	Samples	Percentages
Train	0	317	46.4%
	1	366	53.6%
Test	0	44	57.9%
	1	32	42.1%

ETH	Labels	Samples	Percentages
Train	0	305	46.4%
	1	378	53.6%
Test	0	40	52.6%
	1	36	47.4%

Table 4: Summary for Stock Price Data

The 11 technical indicators consist of momentum (MOM), relative strength index (RSI), normalized average true range (NATR), commodity channel index (CCI), moving average convergence divergence (MACD), Chaikin oscillator (ADOSC), Accumulation Distribution (AD), on-balance volume (OBV), average directional index (ADX), Aroondown, and Aroonup. The exact calculations of technical

indicators are shown in Table 5. All the price data (open, high, low, close) are obtained from Yahoo Finance and the technical indicators are calculated using the TA-Lib python package. The input features are normalized before feeding into the neural network.

Indicator	Formula
$MOM(n,t)$	$C_t - C_{t-n}$
$RSI(n,t)$	$100 - (\frac{100}{1+\frac{n_{up}}{n_{down}}})$
$TR(t)$	$Max[(H_t - L_t), \lvert H_t - C_t \rvert, \lvert L_t - C_t \rvert]$
$ATR(n,t)$	$\frac{\sum_{i=1}^{n} TR(t+1-i)}{n}$
$NATR(n,t)$	$\frac{100ATR(n,t)}{C_t}$
$CCI(n,t)$	$\frac{1/3\times(High_n+Low_n+C_t)-SMA(n,t)}{0.015\times MD(n,t)}$
$MACD(t)$	$EWMA^{12}_{close}(t) - EWMA^{26}_{close}(t)$
$ADOSC(t)$	$EWMA^{3}_{V}(t) - EWMA^{10}_{V}(t)$
MFV_t	$\frac{(C_t-L_t)-(H_t-C_t)}{H_t-L_t} \times V_t$
$AD(t)$	$AD(t-1) + MFV_t$
$OBV(t)$	$OBV(t-1) + V_t$
$ADX(t)$	$\frac{ADX(t-1)\times 13+ADX(t)}{14}$
$Aroonup(t)$	$\frac{100(t-days\ since\ t\ days\ high)}{t}$
$Aroondown(t)$	$\frac{100(t-days\ since\ t\ days\ low)}{t}$

n: look-back window size; C_t: close price at time t; n_{up}, n_{down}: averages of n-day up and down closes; $High_n$, Low_n: highest and lowest price from last n periods; The subscript of EWMA denotes data and the superscript denotes the lookback period.

Table 5: Technical Indicators

The sentiment score for a stock on a given day is calculated using the formula below:

$$S_i^t = \frac{1}{|SR_i|} \sum_{x \in SR_i} \frac{\sum_{j=1}^{30} s_{i,j,x}^t \cdot w_{i,j,x}^t}{\sum_{j=1}^{30} w_{i,j,x}^t}$$

where S_i^t is the sentiment score for stock i in day t, SR_i is the set of subreddits relevant to stock i, $s_{i,j,x}^t$ is the sentiment score of stock i's j^{th} post in subreddit x on day t, and $w_{i,j,x}^t$ is its corresponding score (score of the post, not to be confused with sentiment score). Put in simple terms, the sentiment score to be inputted into the neural network is an average sentiment score of all relevant subreddit posts weighted by the score of subreddit posts in a given day.

5.2 Evaluation Methods

Accuracy, loss, precision, recall and F1 score are used for evaluating the performance of MemeBERT model on predicting sentiment scores. Accuracy, net asset value and Sharpe ratio (of the trading strategy described in 4.2) for evaluating the performance of the subsequent neural network for price prediction.

5.3 Experimental Details

5.3.1 Sentiment Analysis

For our implementation of FinBERT, different configurations of the models have been experimented. After extensive experiments, we have finalized the configurations to a dropout probability of p = 0.1 for attention and p = 0.2 for hidden layers. The maximum position embeddings are 512, the number of attention heads is 12 and the number of hidden layers is 12. We trained the model for 100 epochs with mini-batch size of 64 with learning rate of 2e -5.

5.3.2 Price Movement Prediction

The architecture of Long Short-Term Memory networks [9], as shown in Figure 2 is used in our project to utilize its ability to memorize information further back in time in a sequence of data. The LSTM features input gate and forget gate that are able to learn long term relationships among time series data.

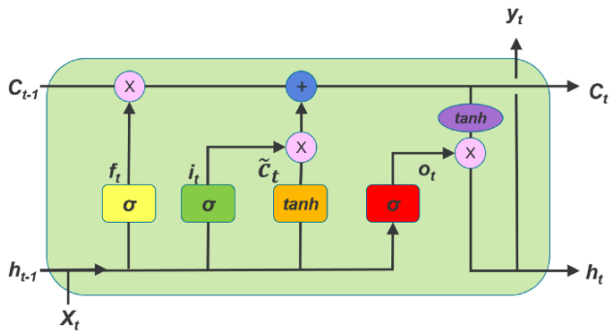


Figure 2: LSTM Architecture [9]

To implement LSTM, the price data are transformed into blocks of data, with a time step of 20 trading days, which correspond to a month of time. After grid search and rounds of hyperparameter tuning, we have achieved the optimal structure of the neural network in terms of both predictive accuracy and net asset value for all the stocks shown in Figure 3

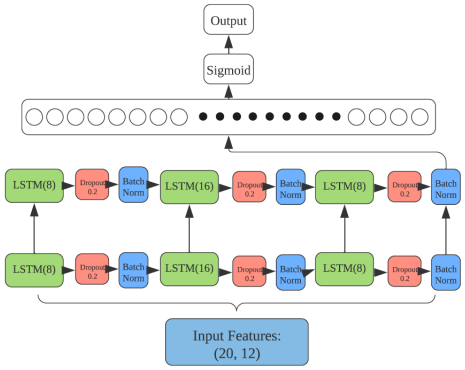


Figure 3: Fine-tuned LSTM Structure

5.4 Results

5.4.1 Sentiment Analysis

We first run FinBERT, without any modification, on Twitter corpus, and have obtained an accuracy around 62%. On the other hand, when we test FinBERT with Financial Phrases Bank for baseline purpose, the accuracy is around 84%.

Table 6 are the evaluation results of FinBERT on baseline Financial Phrases and on Twitter data:

	Loss	Precision	Recall Score	F1 Score
Financial Phrases	0.39	0.83	0.85	0.84
Twitter Data	0.81	0.63	0.60	0.64

Table 6: Evaluation result of FinBERT on Financial Phrases and Twitter Data

According to the results above, FinBERT understands and interprets internet slangs only to a very limited extent and it is evident that FinBERT still needs further pre-training to accurately perform sentiment analysis tasks on informal textss.

Next, we experimented FinBERT on texts extracted from Reddit to get the sentiment score that will be used as the input to the subsequent neural network. Here are two examples of results:

Reddit text	Predicted Sentiment
So you're saying the squeeze hasn't been squoze?! To the moon!	neutral
We all makes mistakes, but holding GME isn't one of them. YOLO	neutral

Table 7: Reddit sentiment prediction with FinBERT

Similarly, we observe here that FinBERT cannot fully understand slangs, such as 'to the moon' and 'yolo', used by Reddit users. It proves that further pre-training and fine-tuning are need for FinBERT to understand internet slangs.

After training the FinBERT using dataset described in 5.1.1, MemeBERT shows improvements on understanding internet slangs with an accuracy of 77%. Below is the result of MemeBERT in comparison with FinBERT above on the same set of test data:

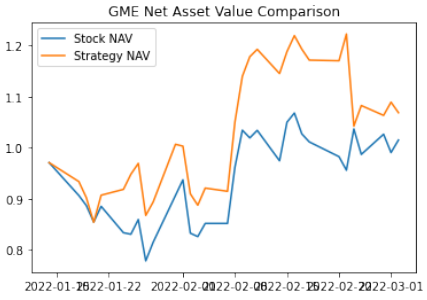
	Loss	Precision	Recall Score	F1 Score
Twitter Data	0.40	0.79	0.74	0.78

Table 8: Evaluation result of MemeBERT on Twitter Data

The trained MemeBERT is then used to predict sentiments on Reddit data, which are then inputted into the neural network.

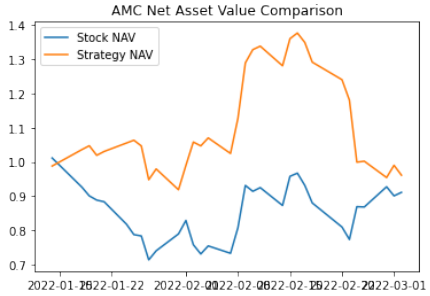
5.4.2 Price Movement Prediction

The same neural network is applied to all the stocks of interest (GME, AMC, BTC, and ETH). The accuracies of predicting price movement on both the training set and test set, as well as the net asset values and Sharpe ratio of the 4 assets are shown in the following figures and tables.



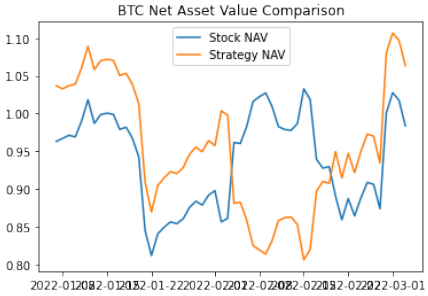
	Accuracy		NAV	SR
Train	57.88%	Strategy	1.07	0.04
Test	52.56%	Stock	1.02	0.14

Figure 4: Performance on GME



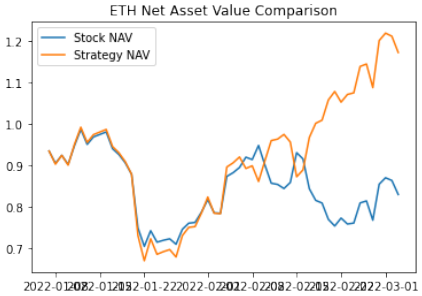
	Accuracy		NAV	SR
Train	62.96%	Strategy	0.96	0.12
Test	54.22%	Stock	0.91	0.07

Figure 5: Performance on AMC



	Accuracy		NAV	SR
Train	62.14%	Strategy	1.06	0.031
Test	53.57%	Stock	0.98	0.01

Figure 6: Performance on BTC



	Accuracy		NAV	SR
Train	63.53%	Strategy	1.17	0.068
Test	54.98%	Stock	0.83	-0.051

Figure 7: Performance on ETH

It can be seen that our developed trading strategy can achieve higher net asset values across the 4 assets, and also higher Sharpe ratio, except for GME, during the test period. Therefore, it is clear that the neural network, while using the sentiment score along with technical indicators, is able to provide useful trading signals.

6 Analysis

6.1 Sentiment Analysis

We observed some interesting results that worth detailed analysis. For example, the prediction of 'to the moon' from MemeBERT is correct, however, the prediction of 'diamond hands' is not correct. The phrase 'to the moon' satisfies linguistic logic, one could get the general idea of this phrase even do not know the exacting meaning due to the fact that it is the internet slang. However, the phrase 'diamond hands' does not satisfies linguistic logic. There is no way to understand this phrase without knowing the background and origin of this phrase. Thus, it is hard for MemeBERT to understand some internet slangs that come without logic. Generally speaking, our model is intended to underfit the data to provide a more general scope due to the ambiguity nature of internet slang.

6.2 Price Movement Prediction

Based on the four plots of the net asset values, we can see that net asset values of the generated strategies generally follows those of passively holding assets, and are able to outperform on occasions. This implies that the developed neural network is able to analyze the historical price and volume data as well as the market sentiment to make price predictions. Furthermore, although the predictive accuracy for the test set are merely above 50% (54.98% in maximum), the net asset values sometimes have substantial differences. For example, in the case of ETH, an accuracy of 4.98% above the 50% baseline is able to achieve a different of around 41% for net asset value. Therefore, our neural network is better at finding trading signals that can have more significant impact the on the returns.

On the other hand, by inspecting the plots, it can also be observed that the net asset value curves for the generated strategies often exhibit more volatility, which led to a lower Sharpe ratio in the case of GME stocks. Because we have only employed a simple trading strategy of buying with a positive label and selling with a negative label using the predicted price movement.

7 Conclusion

Our newly developed MemeBERT model has shown significant improvements to the baseline model of FinBERT. MemeBERT has achieved an accuracy of 73%, whereas FinBERT only has an accuracy of 63% for predicting Twitter sentiment. Using the sentiment score from MemeBERT and 11 technical indicators, the neural network, with the uses of LSTM along with dropout and batch normalization, has achieved higher net asset values and Sharpe ratios across the 4 assets with our defined simple trading strategy of buying with a positive prediction and selling with a negative prediction. In order for the trading strategy to have less volatility and drawdown, the trading problem can be modelled as an optimization problem where not only the predicted return, but also the predicted volatility will be taken into account, and this leaves venues open for future work.

Financial sentiment analysis is not a goal itself, it is important because it supports investment decisions. One thing that can be done in future is to model meme stock volatility instead of returns when the meme stocks experience high publicity on Twitter or Reddit. Another interesting work that has the potential to be meaningful in the future is that MemeBERT could be applied to the question-answering setting. It can be extended to answer Twitter or Reddit questions regarding meme stock with internet slang.

References

[1] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *CoRR*, abs/1908.10063, 2019.

[2] Adam Hayes. Technical analysis. 10 2021.

[3] Adam Bear. What happened to gme stock? 2 2021.

[4] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[5] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[7] Paolo Ripamonti. Twitter sentiment analysis.

[8] Pushshift reddit api documentation.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.