Predicting Stock Movement using Transformer-Based Models

Sudhandar Balakrishnan, Md Mahfujul Kadir, Vidhi Kokel (Group-13),
Department of Electrical and Computer Engineering,
Queen's University.
ON, CA.

Abstract

The transformer-based technologies, e.g., BERT, GPT, emerge as a cornerstone technology in processing the sequential textual data. Specifically, the transformer-based models can learn the underlying relation between tokens in a large text, e.g., financial, legal datasets. One of the largest sources of financial text is Twitter where market trends of various companies can be identified from users tweets. Therefore, an efficient prediction mechanism is necessary to realize the overall relationship between the financial texts and the market. In this report, we investigate the Twitter texts related to the stock market to predict the relevant stock movements. We utilize the transformer-based sequence classification to understand the textual input and utilize the models to predict the stock movements. We performed extensive experiments and further compare our results with state-of-art technologies. Our experiments show that the BERT/GPT-2 based models can improve the stock prediction due to the innate ability of the models to understand the context of the given data.

1 Introduction

The stock movement can be seen as a highly dynamic system. The system dynamics cannot be realized by a single stochastic model. Therefore, the researchers rely on generative models to learn the dynamics of the system in terms of historical movements. Traditionally, the market observation is performed by learning from dense corpora which contain financial conversations, e.g., tweets. The co-relations are challenging to predict by traditional neural networks, Li et al. [16]. To address the issues of traditional mechanisms in learning the sequential data, mechanisms such as bi-directional GRU and LSTM, (Xu and Cohen [27], McCann et al. [18], Peters et al. [21]) are proposed.

The recent advancements of transformer-based Natural language processing (NLP) technologies such as Bidirectional Encoder Representations from Transformers (Bert) and Generative Pre-trained Model (GPT), are proven mechanisms to learn from textual data and underlying dependencies. The stock price movements in a given time frame with textual twitter data can successfully realize the temporal dependency. For example, any event on a day d_1 can have its effects until a subsequent time-frame $[d_1, d_t]$. Therefore, it is necessary to combine the above information to predict stock movements on a target day Xu and Cohen [27]. The research Xu and Cohen [27] addresses the problems of market stochasticity, information gap, and temporal dependencies together in stock analysis. On the contrary, we investigate textual sentiment mining leveraging the robust pre-trained BERT and GPT-2 models to realize the textual information. We demonstrate that a significant improvement can be achieved by leveraging generative pre-training of a language model on a corpus of unlabeled text. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. Specifically, we focus on several key research challenges given in the following,

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

1. We performed pre-processing on the Twitter dataset. Further, to correctly determine the random change of market information, we consider the tweets posted between the closure of the stock market on $d - t$ days to the current opening on target day d.
2. We investigate the performance of financial twitter data and predict the movement of prices of relevant companies using BERT, GPT-2 models. Further, we study fine-tuning the proposed models.
3. We also investigate the use of an intermediate larger labeled dataset to train BERT and GPT-2 to perform sentiment analysis and use the trained model to predict the sentiment of the financial tweets. We take the average of the sentiments of the financial tweets and compare it to the stock movement and investigate if we can get better results.
4. We extensively performed our experiments on the proposed models using several experimental setups. Afterward, we present our empirical results in the evaluation section. Our experiments show that the BERT/GPT-2 based models can improve the MCC score due to the innate ability of the models to understand the context of the given data.

The rest of the report is organized as follows, **Section** 2 briefly reviews the theoretical background related to the stock movement prediction using sentiment classification. Afterword, in **Section** 3, provides a clear understanding on datasets. Furthermore, the problem formulation is discussed, which is followed by the performance evaluations of the system through extensive experiments in **Section** 4. Finally, the conclusive analysis on the experimental results and future improvements are presented in **Section** 6.

2 Background

Stock movement prediction has long attracted both in industry and academia (Frankel and Frankel [11]). The movement prediction can be characterized as an NLP task, comprising news articles and/or social media as primary resources. However, the traditional prediction models that use these sources are mostly discriminative. For example, these models mostly rely on feature engineering (Frankel and Frankel [11], Oliveira et al. [20]) with pervasive neural network-based mechanisms. Due to the high stochastic behavior of the market, stock movement prediction is widely considered a challenging task. The traditional approaches model the stock market behavior as a random-walk pattern (Malkiel [17]), in terms of updated market information. The deterministic features of the traditional model fail to reflect the market dynamics due to the fixed set of model parameters. Compared to the deterministic fixed set models, a generative process can reproduce stock signals from the market information and therefore reflects the randomness in the stock market. However, in the early works, the asymmetric social media texts are analyzed with simple models that often fail to understand the underlying semantic information.

Natural language understanding comprises a wide range of diverse tasks, e.g., likelihood assessment and classification of legal texts. Although the large unlabeled text corpora are abundant, labeled data dedicated to some specific tasks is rare. Therefore, it is challenging for the discriminatively trained models to perform adequately. The standard language models are unidirectional, therefore it is challenging to learn the contextual dependencies. Moreover, the unidirectional methods limit the choice of architectures that can be used during pre-training. Such methodologies are not optimal for sentence-level context learning. Furthermore, it is not practical to fine-tune the trained model in considering the token-level tasks, e.g., question answering. Thus it is evident that we need to incorporate the contextual information from both directions.

BERT alleviates the unidirectional constraint by incorporating a masked language model (MLM) pre-training objective. The MLM has the ability to randomly mask a set of tokens from the input texts. The main objective is to predict the original vocabulary from the masked word in terms of context. On contrary to the unidirectional language model, the MLM has the capability of fusing contexts from both previous and future tokens. Therefore it provides a brand new capability to pre-train a deep bidirectional transformer. Additionally, the work Devlin et al. [10] also integrates the next sentence prediction task which can jointly pre-train from multiple text-pair representations. The pre-trained representations can alleviate the burden of task-specific feature engineering. Therefore, BERT-based models are suitable for general text learning tasks without any major change. The fine-tuning-based model is capable of performing sentence-level and token-level tasks while outperforming traditional task-specific architectures.

Transformers (Vaswani et al. [24]), can perform strongly on various tasks, e.g., machine translation, document generation, and syntactic parsing. Compared to recurrent neural networks, it can provide a structured memory in handling the long-term semantic dependency, and therefore, can guarantee robust performance across diverse tasks by simply transferring the model, i.e., BERT, GPT. The transformation can significantly enable the system with fine-tuning capability. On the other hand, the research work ([22]) explores semi-supervised learning to perform language understanding tasks. Specifically, a combination of unsupervised pre-training and supervised fine-tuning is studied. Similar to BERT, the aim is to learn a universal model and further with minor evolution so that it can tackle a wide range of tasks. The generative model works with a large corpus of unlabeled texts and several datasets with manually annotated training examples (target tasks). It is not required that the task-specific dataset is in the same domain as the unlabeled corpus. Essentially, the objective is to learn from the unlabeled data. First, the model learns the initial parameters of a neural network by traversing the unlabeled text. Subsequently, the parameters are adapted to perform a target task leveraging the supervised learning objectives.

3 Dataset and Pre-processing

The stock data can be categorized into the following 9 groups, basic materials, consumer goods, healthcare, services, utilities, conglomerates, financial, industrial goods, and technology. We adopt *StockNet* dataset presented in the paper, Xu and Cohen [27]. Specifically, the dataset consists of two major components, i.e., tweets and historical stock price data, for 88 stocks between the period $01/01/2014 \rightarrow 01/01/2016$. The above-mentioned 88 stocks comprise all 8 stocks in conglomerates and the top 10 stocks in capital size in each of the other 8 industries, the details can be found in Xu and Cohen [26].

Our study focuses on the binary classification of the stock movement, (i.e., high or low), for a given stock on a particular day. To generate the target variable, the movement percentage for each day is derived. To address the issue of stocks with extremely minor movement percentages, the work Xu and Cohen [27] suggests a setting with two thresholds, (i.e., 0.5% and 0.55%) and ignoring stocks having movement percentages within this threshold since the change in the minor. Based on the suggestion, we classify movement percentages $\leq -0.5\%$ as 0 and movement percentages $> 0.5\%$ as 1, hereby ignoring movement percentages between $[-0.5\%, 0.5\%]$. Using above setting, we identify 26623 targets after the classification with 13368 targets with positive (i.e.,1) labels and 13255 targets with negative (i.e.,0) labels respectively. Furthermore, the dataset has been split into train, validation, and test set temporally. Specifically, the movements between $01/01/2014 \rightarrow 01/08/2015$, for training, $01/08/2015 \rightarrow 01/10/2015$, for validation and $01/10/2015 \rightarrow 01/01/2016$, for testing purposes.

The tweets from the *StockNet* dataset have been linked to their target stocks based on their timestamp. To predict the stock movement for stock S on a particular target day d. Especially, the tweets posted between the closure of the stock market on $d - t$ days to the current opening on target day d, are considered. The reason is to prevent future information from entering the prediction on a target day d. In particular, a lag of $t = 5$ days has been selected after experimenting with $t = 5$ day lag, $t = 3$ day lag, and $t = 1$ day lag and we found that the $t = 3$ day lag can capture the stock movement in a better way. The identified tweets are then combined based on the target day d and further truncated to a maximum length of 512 as per input fitting requirements of BERT and GPT-2. The tweets are then pre-processed to mask user mentions, hashtags, and hyperlinks using regex. The dataset is further filtered to ensure that at least one tweet is present for a particular stock on a target day d.

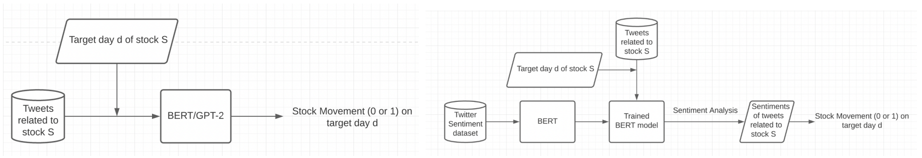
We also experiment on an intermediate twitter dataset, *Sentiment140* dataset, consisting of 1.6 million labelled tweets (i.e., $800,000$ positives and $800,000$ negatives) for sentiment analysis has been used from Kaggle dataset,(Go et al. [13]).

4 Model Overview

4.1 Problem Formulation

The system objective is to predict the stock movement, (i.e., increase/decrease) of a target stock S on the desired day d. Our system relies on tweets data related to individual stocks to learn the stock movement prediction. First, the tweets corresponding to the target enterprise stock type S are selected considering a time-frame, (i.e., $d - t$). The $t = 5$ day lag is taken into account since any random information change in a market generally affects the price of stock S for a few days. In particular, our proposed system considers the tweets in between the specific time, $4:30$ p.m. (stock market closing) from $d-3$-th day and at $9:00$ a.m. to d-th day (market opening). The time frame is chosen specifically to properly filter out the information from the future days entering into the prediction. As discussed in section 3 dataset, we classify movement percentages $\leq -0.5\%$ as 0 and movement percentages $> 0.5\%$ as 1, hereby ignoring movement percentages between $[-0.5\%, 0.5\%]$.

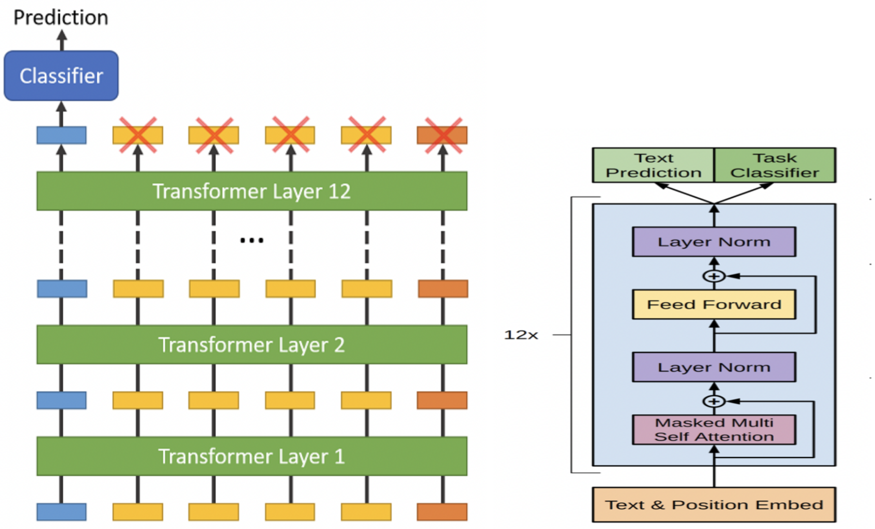
To predict the stock movement, our system follows two disparate methods. First, the tweets related to stock S are fed into BERT/GPT-2 to train our models to predict the stock movements. The models can directly predict the stock movement based on the information learned from the textual corpora, (i.e., $0/1$). In the second method, the system utilizes a larger twitter dataset with approximately 1.6 million samples. The BERT model is trained on the 1.6 million tweet samples to predict the sentiments. This trained model is used to predict the sentiments on the StockNet Twitter dataset. It utilizes the set of target stock S, on a target trading day d in a similar fashion that we have presented in Section 3. The average of the sentiments is calculated for a target stock S, on a target trading day d and compared with the stock movements for the target stock S, on a target trading day d. By extensive experiments, our proposed system illustrates that with sufficient fine-tuning, the system can achieve comparable results in terms of MCC score and average. The details can be found in Section 5. An illustration of the above-mentioned workflows is presented in Fig. 1.


Target day d of stock S
Tweets related to stock S
BERT/GPT-2
Stock Movement (0 or 1) on target day d
Tweets related to stock S
Target day d of stock S
Twitter Sentiment dataset
BERT
Trained BERT model
Sentiment Analysis
Sentiments of tweets related to stock S
Stock Movement (0 or 1) on target day d

(a) Approach 1 (b) Approach 2

Figure 1: Training methods.

4.2 Model Architecture



(a) BERT architecture ber [2] (b) GPT-2 architecture Radford et al. [22]

Figure 2: BERT and GPT-2 based architectures

BERT stands for Bidirectional Encoder Representation from Transformers. BERT is based on the encoder architecture of the transformers. BERT is a bidirectional model which can learn context from both left and right sides and achieves start of the art results in many NLP-related tasks. With the help of transfer learning, pre-trained BERT can be used to fine-tune by adding one linear layer on top of BERT and achieve good results on many downstream tasks. We have used the BertforSequenceClassification model from transformers which consist of the bert-base-model (12 layers of encoders and 12 attention heads with 110 Million parameters) and one linear layer for classification. GPT-2 is a decoder transformer that implies that the last token of an input sequence is used to predict the most probable subsequent token. Therefore, the last token of the sequence contains the information required for prediction. Specifically, we can use that information to make a prediction in a classification task. In other words, we can use the last token embedding as opposed to the BERT model. As a result, it is necessary to pad to the left since the previous token information is important instead of padding to the right which we do in BERT. We leverage the *HuggingFace Transformers* which naturally supports such configuration for GPT-2 tokenizer. The transformer is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data. As shown in Fig. 3, the tweets in the input data are first vectorized. Then the vectorized data is provided to the model for classification. The classification model consists of various layers like token and position embedding, transformer, average pooling, dropout, fully connected layer followed by one more dropout and fully connected layer. On the whole, this architecture of the classification model consists of a total of 87,158 trainable hyper-parameters. Along with this, we use Adam as the optimizer for this model. Here, we have used the transformer as part of one of our classification models to leverage the benefits of both the encoder and decoder components.

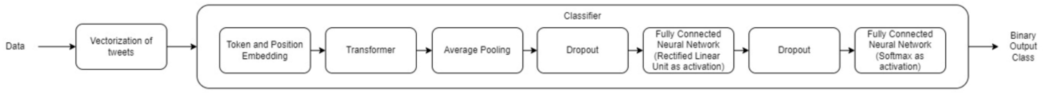
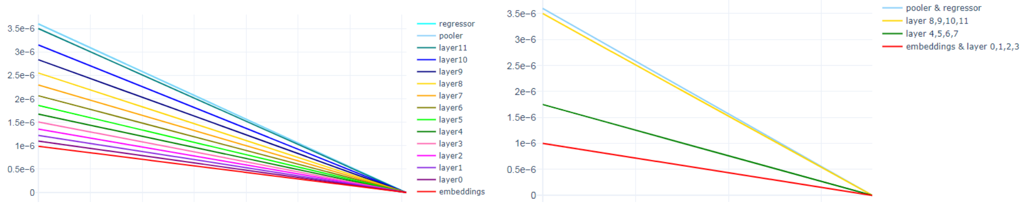

Data
Vectorization of tweets
Classifier
Token and Position Embedding
Transformer
Average Pooling
Dropout
Fully Connected Neural Network (Rectified Linear Unit as activation)
Dropout
Fully Connected Neural Network (Softmax as activation)
Binary Output Class

Figure 3: Transformer based architecture.

4.3 Fine Tuning of BERT

Our system also utilizes fine-tuning techniques to achieve better accuracy. Especially, we investigate on *AdamW* optimizer and the effects of each parameters (Sun et al. [23]). Specifically, instead of using a single learning rate, our system experiments on several variations of layer-wise learning rate decays to improve the performance of BERT/GPT-2. The final results show that the technique significantly enhances the performance. Furthermore, we incorporate another type of layer-wise learning rate decay where 12 layers of BERT are grouped into sets and different learning rates are applied to each set known as grouped layer-wise Learning Rate Decay. The method resulted in even better performance of our BERT model and we have used the following learning rates to train BERT for our stock movement prediction (Chang [8]), i.e.,

- Set 1: Embeddings + Layer $0, 1, 2, 3$, (learning rate: $1e^{-6}$)
- Set 2: Embeddings + Layer $4, 5, 6, 7$, (learning rate: $1.75e^{-6}$)
- Set 3: Embeddings + Layer $8, 9, 10, 11$, (learning rate: $3.5e^{-6}$)



(a) Layer-wise learning rate decay

(b) Grouped layer-wise learning rate decay

Figure 4: Types of layer-wise learning rate decays. ste [6]

4.4 Re-initializing pre-trained layers of BERT

BERT has 12 layers, and each layer of the BERT captures various kinds of information. The lower layers contain low level representations and stores generic information. The task related information is stored on the top layers of the BERT closer to the output. (Zhang et al. [28]) in their paper, suggested that re initializing these top layers will increase in the performance of BERT on several downstream tasks. Based on their work, we have tried to reinitialize the top 3 layers of the BERT for the direct stock movement prediction model.

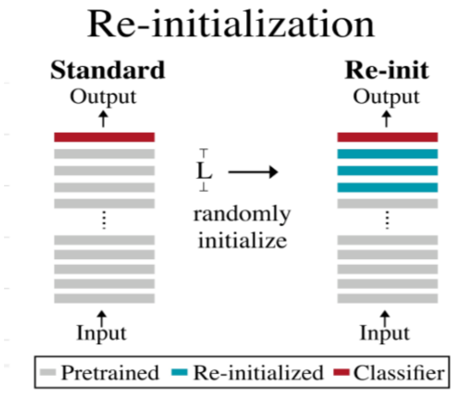


Figure 5: BERT re-initializing pretrained layers. Zhang et al. [28]

Furthermore, we also experimented with several other fine-tuning techniques for BERT. We tried to change the dropout of the attention and hidden layers of BERT, we tried experimenting with Stochastic Weight Averaging (SWA) based on the results of the paper, (Izmailov et al. [15]). But these fine-tuning methods did not result in increasing the performance of the model.

We reinitialized the pre-training layers of the BERT direct stock movement prediction model and we did not use this technique for BERT intermediary sentiment-based stock movement prediction model since it did not improve the performance.

We also introduce 50 warm steps to train BERT/GPT-2 where the learning rate increases linearly from 0 to the initialized *AdamW* optimizer. where the learning rates during the first 50 steps starts from a positive real number and further linearly decrease to 0, (Zhang et al. [28]).

5 Performance Evaluation

5.1 Evaluation Metrics

Based on the previous work for stock movement prediction (Xu and Cohen [27]) and the paper ([9]) we have selected accuracy and MCC as our evaluation metrics. The work in (Chicco and Jurman [9]) clearly states that MCC is a better metric for binary classification since it requires good results in true positives, false negatives, true negatives and false positives to obtain high scores. The following is the formula for the MCC score, i.e.,

$$MCC = \frac{tp * tn - fp * fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \tag{1}$$

5.2 Effects of Fine Tuning on BERT

As it is mentioned in subsection 4.3, we experimented with several fine tuning techniques to improve the performance of BERT. The following table shows the results of the initial BERT model without any fine-tuning and the final fine-tuned BERT model, It is evident from the above results that fine-tuning plays a significant role in increasing the accuracy of the BERT model and the way it learns the context from the textual data. The following are the hyperparameters are used for the final version.

Table 1: Initial BERT models

Models	Acuracy	MCC
Base BERT model	46.7	-0.03
BERT + Layer wise learning rate decay	47.8	0.005
BERT + Grouped Layer wise learning rate decay	51	0.007
BERT + Grouped Layer wise learning rate decay + Reinitializing top 3 layers of BERT + 50 warmup steps	53	0.0344

Table 2: Hyperparameters of BERT

Hyperparameters	Value
Epochs	5
Optimizer	AdamW
Learning rate for optimizer	Set 1: Embeddings + Layer $0, 1, 2, 3$, (learning rate: $1 * e^{-6}$) Set 2: Embeddings + Layer $4, 5, 6, 7$, (learning rate: $1.75 * e^{-6}$) Set 3: Embeddings + Layer $8, 9, 10, 11$, (learning rate: $3.5 * e^{-6}$)
Number of reinitialized layers	3
Batch size	16
Input Sequence Length	512

5.3 Comparison between direct prediction and prediction based on intermediate sentiment analysis model

Table 3: BERT prediction accuracy

Models	Acuracy	MCC
BERT direct prediction	53	0.034
BERT prediction based on sentiments analysis	52.3	0.025

Based on the above results, we can find out that training the BERT model on larger dataset results in achieving comparable performance with the direct prediction method. But it is still not able to achieve better results as we had expected. These results helped us realize that stock movements are not only dependent on the sentiment of the tweets but also on a multitude of other external factors as well.

5.4 Initial Results of GPT-2 and Transformers

We performed extensive experiments on GPT-2 based model. Our baseline model without fine-tuning gives us good accuracy, i.e., $0.51 \pm 0.01\%$. After fine-tuning our model shows better accuracy which is presented in Table 4. We experimented on a $d = 5$ day-lag twitter data, which can capture the market information changes in the dataset. The Table 5 shows the results of GPT-2 and transformer models utilized to predict the stock movements considering a enlarged time-frame. Specifically, we studied the dataset wnd trained our model with different learning parameters, we set-up 50 warmup steps. Due to resource limitation, we only experimented with a maximum text-sequence length of 512, and reduced the batch-size to 10. The detail is presented in Table 6.

Table 4: Performance evaluation of GPT-2.

Models	Acuracy	MCC
GPT-2 Base	51 ± 0.01	0.01129
GPT-2 with fine-tuning parameters	54	0.02433

Table 5: Performance evaluation of GPT-2.

Models	Acuracy	MCC
GPT-2 with 5-day lag	54	0.02433
Transformers	52	0

Table 6: Hyperparameters for GPT-2

Hyperparameters	Value
Epochs	$2, 3, 4, 8$
Optimizer	$AdamW$
Learning rate for optimizer	$1 * 10^{-5},\ 2 * 10^{-5},\ 3 * 10^{-5}$
Batch size	10
Input Sequence Length	512

5.5 Comparison with baseline models and StockNet variations from StockNet :

In (Xu and Cohen [27]), the authors have implemented four variations of StockNet and compared it with five baselines models. In this section, we compare our results with the *StockNet* variations and baseline models. The following are the baseline models used,

- RAND: a naive predictor making random guesses up or down.
- ARIMA: Autoregressive Integrated Moving Average, an advanced technical analysis method using only price signals (Brown [7]).
- RANDFOREST: a discriminative Random Forest classifier using Word2vec text representations (Wu et al. [25]).
- TSLDA: a generative topic model jointly learning topics and sentiments (Nguyen et al. [19]).
- HAN: a state-of-the-art discriminative deep neural network with hierarchical attention (Hu et al. [14]).

Table 7: Performance evaluation of our models with baseline models

Models	Acuracy	MCC	Our Models	Acuracy	MCC
RAND	50.89	-0.002266	BERT	53	0.034
ARIMA	51.31	-0.020588	GPT-2	54	0.02433
RANDFOREST	50.08	0.012929	BERT (Sentiment based)	52.33	0.025
TSLDA	54.07	0.065382	Transformers	52	0
HAN	57.64	0.051800			

The following are presented as the *StockNet* variations introduced in (Xu and Cohen [27]).

- TECHNICALANALYST: the generative StockNet using only historical prices.
- FUNDAMENTALANALYST: the generative StockNet using only tweet information.
- INDEPENDENTANALYST: the generative StockNet without temporal auxiliary targets.
- DISCRIMINATIVEANALYST: the discriminative StockNet directly optimizing the likelihood objective.

Our models surpass the performance of baseline models like RAND, ARIMA, and RANDFOREST in terms of both accuracy and MCC. Our models also surpass the performance of the StockNet variations, TECHNICAL ANALYST, in terms of MCC. Our fine-tuned BERT model achieves MCC score almost equal to the INDEPENDENT ANALYST.

Table 8: Comparison with Stocknet Variations.

Stocknet Variations	Acuracy	MCC	Our Models	Acuracy	MCC
TECHNICAL ANALYST	54.96	0.016456	BERT	53	0.034
FUNDAMENTAL ANALYST	58.23	0.071704	GPT-2	54	0.02433
INDEPENDENT ANALYST	57.54	0.036610	BERT (Sentiment based)	52.33	0.025
DISCRIMINATIVE ANALYST	56.15	0.056493	Transformers	52	0
HEDGEFUND ANALYST	58.23	0.080796			

6 Conclusion

Even though BERT and GPT-2 are trained on large text corpora and surpass state of the art results in many language tasks it is still not able to achieve higher accuracy and MCC than certain StockNet Variations like Fundamental Analyst and Hedgefund analyst. We further analyzed the reason behind this and found out that the main reason is the use of temporal auxiliary attention mechanism in StockNet. It acts as a denoising regularizer which helps the model to filter out noises like temporary rise in a positive movement when the market has an upward trend and helps the model to focus on the main target and generalize well by denoising. This task-specific attention mechanism is not found in models like BERT and GPT-2 even though they learn context by masked self-attention.

Furthermore, while comparing the performance of direct prediction of stock movement from tweets and sentiment-based prediction of stock movement using an intermediary dataset we found that sentiment-based prediction of stock movements achieved comparable performance but still could not achieve a significant increase in performance. This reason might be attributed to the heterogeneous nature of the tweets where the sentiment of the tweets might be negative due to certain events of a particular company and might turn positive after a timely fix within the considered lag of 5 days. In addition to the sentiments, there might be several external socio-economic factors that might affect the stock movement of a target stock S which affects the ability to build a robust stock movement prediction model using just tweets.

In the future, we would like to train a text-based model using financial news of target stocks and use fiscal reports to build a numerical plus text-based model to predict the stock movements of a company.

7 Contributions

Sudhandar Balakrishnan (20296065) initiated the project discussions, designed the framework for the project, preprocessed and analyzed the data. Sudhandar also owned the training and fine tuning of both the BERT based direct prediction model and BERT sentiment based (intermediary data) stock movement prediction model. He experimented with various fine tuning methods suggested from different papers and evaluated their impact on the performance of both the BERT models. Sudhandar also handled the dataset, model architecture, results and conclusion section of the project report. He also helped with the implementation of GPT-2 and other minor code fixes. **Md Mahfujul Kadir (20253855)** owned the implementation and fine tuning of GPT-2. He also owned the abstract, introduction, background work and problem formulation section of the project report and handled the editing and formatting of the entire project report.**Vidhi Kokel (20241891)** owned the implementation and fine tuning of the transformers and contributed to the transformer architecture section of the project report.

8 Code

Our current work (python scripts and colab notebooks) can be found on: `https://github.com/Sudhandar/ELEC825-Project`

Basic ideas of our codes are inspired from the following sources: fin [3, 5, 4], ber [1], Gmihaila [12].

References

[1] *BERT*, 2021. https://colab.research.google.com/drive/1Y4o3jh3ZH70tl6mCd76vz_IxX23biCPP [Accessed: online].

[2] *BERT architectures*, 2021. https://mccormickml.com/2019/07/22/BERT-fine-tuning/ [Accessed: online].

[3] *Fine-Tuning*, 2021. https://www.kaggle.com/rhtsingh/on-stability-of-few-sample-transformer-fine-tuning?scriptVersionId=67176591&cellId=9 [Accessed: online].

[4] *Fine-Tuning*, 2021. https://towardsdatascience.com/advanced-techniques-for-fine-tuning-transformers-82e4e61e16e#1fbc [Accessed: online].

[5] *Fine-Tuning*, 2021. https://www.kaggle.com/rhtsingh/on-stability-of-few-sample-transformer-fine-tuning?scriptVersionId=67176591&cellId=9 [Accessed: online].

[6] *Warm Up Steps*, 2021. https://openreview.net/pdf?id=c01IH43yUF [Accessed: online].

[7] Robert Goodell Brown. *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation, 2004.

[8] Peggy Chang. *Fine Tuning Bert*, 2019. https://towardsdatascience.com/advanced-techniques-for-fine-tuning-transformers-82e4e61e16e [Accessed: online].

[9] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1): 1–13, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Jeffrey A Frankel and Benjamin Frankel. *Financial markets and monetary policy*. MIT Press, 1995.

[12] Gmihaila. *GPT-2*, 2021. https://gmihaila.github.io/ [Accessed: online].

[13] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[14] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 261–269, 2018.

[15] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[16] Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883*, 2019.

[17] Burton Gordon Malkiel. *A random walk down Wall Street: including a life-cycle guide to personal investing*. WW Norton & Company, 1999.

[18] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. *arXiv preprint arXiv:1708.00107*, 2017.

[19] Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications*, 42(24):9603–9611, 2015.

[20] Nuno Oliveira, Paulo Cortez, and Nelson Areal. Some experiments on modeling stock market behavior using investor sentiment analysis and posting volume from twitter. In *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*, pages 1–8, 2013.

[21] Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*, 2018.

[22] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[23] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[25] Qiong Wu, Zheng Zhang, A Pizzoferroto, Mihai Cucuringu, and Zhenming Liu. A deep learning framework for pricing financial instruments. *ArXivorg*, 2019.

[26] Yumo Xu and Shay B Cohen. *StockNet Dataset*, 2018. `https://github.com/yumoxu/stocknet-dataset` [Accessed: online].

[27] Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979, 2018.

[28] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*, 2020.