Transformer on Transformer: A News-Driven Stock Return Prediction Framework

Mingxi Liu
MIDS UCB
mingxi.liu@berkeley.edu

Abstract

This paper investigates the use of BERT on financial news for stock return prediction. While sentiment analysis accuracy has improved with BERT, it alone does not guarantee a better prediction of stock return. Using daily news from Reuters and price data of the S&P 500 index, the paper investigates various methods for extracting and aggregating information from news. The author builds a mini transformer model on the Pooler output from a pre-trained financial BERT model. It incorporates information from both historical news and price movements with an encoder-decoder structure, attention mechanism, and positional encoding, achieving the best performance among the candidates.

1 Introduction

Financial news is among the forces that move the stock market. Investors have been trying to build models based on news text information to predict the movements of the stock market. In this paper, we explore using news from Reuters to predict the daily returns of S&P 500 Index. In the past, such practice usually consisted of two tasks: sentiment analysis of the news and predicting the price movement based on the sentiment scores or classifications. For the first task, with the help of pre-trained BERT models, sentiment analysis of financial news can now be done with higher accuracy and less challenge.

However, better sentiment analysis does not necessarily bring better stock price prediction and investment results. As shown in this paper, during the outbreak of COVID-19 in 2020, while negative sentiments were still dominating the news media, the stock market rebounded with the help of monetary easing and fiscal stimulus. As a result, models based directly on the sentiment classification results of the pre-trained models performed poorly. The models need more information from the news than the sentiment, such as their importance or relevance to the stock market, to make good predictions.

Luckily the BERT models provide flexibility in extracting deeper layers of information from the news, including the Pooler tokens, the CLS tokens, and other hidden states. Our results show that even by fine-tuning the Pooler token, which requires the least data and computing resources, we can acquire significantly better performance than relying on sentiment classifications.

Although some past research treated the prediction of stock price movement as a classification problem, we argue the continuous nature of price data makes it more suitable for a regression problem. The predictions of the models are converted into trading signals, which enables us to simulate investment results based on the predictions and evaluate the models with financial metrics such as Sharpe Ratio and Max Drawdown in addition to statistical metrics.

Our dataset includes around 138,000 pieces of news from Reuters and around 2,400 trading days of the S&P 500 index data, 50 news per day on average. A successful model needs to properly assign the weights when aggregating the news information within a day and fully utilize the information from historical news and price data.

Inspired by the success of transformer models in seq2seq prediction, we propose a mini-transformer structure that takes both news text and past stock returns as inputs, utilizes self-attention and cross-attention to aggregate information, and uses positional encoding to handle the series order. This model outperforms other candidates in both statistical and economic sense.

2 Background

It's no news that news is used to predict stock price movements(e.g. Tetlock (2007)). Since it's an interdisciplinary task involving NLP and financial

engineering, the solution evolves with the progress on either side. On the NLP side, the essential problem is how to extract numeric information from the text of news. Early work used the bag-of-words as text representation to perform sentiment analysis for the news (Schumaker and Chen, 2009). Later work adopted word embeddings such as Word2Vec. For instance,Hu et al. (2018) propose a Hybrid Attention Network to predict the stock trend based on the sequence of recent related news and employ self-paced learning for effective and efficient learning. They use a pre-trained Word2Vec embedding layer to calculate the embedded vector for each word and then average all the words' vectors to construct a news vector. Other techniques include character-based neural language model (dos Santos Pinheiro and Dras, 2017) and topic modeling (Nguyen and Shirai, 2015).

Entering the age of transformers, the work on pre-training BERT models with financial content improved the accuracy of sentiment analysis of financial news. Sawhney et al. (2021) pre-trained a financial domain-specific BERT model, FinBERT, using a large scale of financial communication corpora. Experiments on three financial sentiment classification tasks confirm the advantage of FinBERT over the generic domain BERT model. Similar work and results can be found in Araci (2019).

Some research also explored using BERT models to extract information from related text to predict stock price. Sawhney et al. (2021) used BERT to generate the final hidden states of tweets and financial news and then a reinforcement learning framework to predict the price of individual stocks. They used LSTM to include historical information. Yang et al. (2019) used the BERT model to encode the terms searched on online search engines such as Google to generate a Financial and Economic Attitudes Revealed by Search (FEARS) index for stock return prediction. They adopted the attention mechanism to aggregate the text representations from the original BERT model.

Our research differs from previous research above in at least two aspects. First, inspired by the transformer architecture, we replace LSTM with "Positional Encoding + Attention" to handle the sequence-to-sequence prediction problem. Second, we use a financial domain-specific BERT model instead of the original BERT as the encoder, which should generate better representations of financial content.

3 Methods

3.1 Problem Description

In practice, the ultimate goal for stock prediction problems is to generate a signal S which represents the ratio of stock holding value to the investor's total assets. We constrain $S \in [-1, 1]$, which means short-selling is allowed but leverage is not.

Although the task can be viewed as a classification problem if we only predict whether the stock price is going up or down, the long-term uptrend of S&P 500 index suggests there are more "up" days than "down" days, thus the samples are imbalanced. Some previous research tries to avoid this problem by using other thresholds instead of zero (Xu and Cohen, 2018). However, the choices of these thresholds are arbitrary and are difficult to justify. For example, if we set thresholds at 0 and 1, is it reasonable to label 0.99 and 1.01 as different classes while 0.01 and 0.99 the same class? Essentially, the returns of stocks are continuous rather than discrete, so we prefer to view it as a regression problem.

In our context, the problem is how to use the news as input to generate a continuous signal S, where $S \in [-1, 1]$. This enables us to directly simulate the investment returns by multiplying S with stock returns. Outputs or predictions from different models are first standardized into z-score and transferred with $tanh$ function to generate signal S. In this way the financial metrics from different models are comparable. The benefit of this simulated trading is that we can use investment performance metrics such as the Sharpe Ratio and Max Drawdown in addition to statistical metrics to evaluate the predictions. The Sharpe Ratio is a measure of risk-adjusted return. It compares the average return of an investment to its standard deviation. Max Drawdown is the largest peak-to-trough decline in the value of an investment or trading strategy over a specific period. Both are two key metrics used to evaluate the risk and return of an investment or trading strategy. We skip details such as transaction fees and slippage when calculating the financial metrics.

3.2 Data

This paper experiments with predicting the daily return of the S&P 500 Index based on the financial news from Reuters. To get as many samples as possible, two sources of news data are combined. One is from Ding et al. (2014), which covers the

period from October 2006 to November 2013, total 105,343 pieces. This period witnessed a severe economic downturn in 2007-2010, followed by a modest recovery in 2011-2013. The other is provided by Kaggle, which includes 32,770 Reuters news from March 2018 to July 2020, which also covers a volatile period during COVID-19. Since both sources are from Reuters and are comparable in frequency and length, we think it's reasonable to combine them. The gap between 2013 to 2018 is not a big issue if we use the first source as the training set and the second validation/test set.

We select the lead paragraph (normally the first paragraph) of each news as the text input. It typically contains 30 50 words, providing more complete information than the title which contains less than 10 words, while requiring less computing resources than the full content.

The S&P 500 Index price data comes from Yahoo Finance. It contains a daily series covering both periods above. The size of the labels is 2,393 after aligning the dates with the news data. We use the standardized log returns of the index as the label when training the models. We change back to the original log returns when evaluating the investment results.

Note that the number of news is more than 50 times the number of labels. This is because there are usually 40 to 60 news per day. Therefore, we need to aggregate the information of news of the same day. Different methods will be discussed in the following sections.

3.3 Experiments

In this section, we compare six methods of extracting and processing the information from news text. Their major differences are listed in Table 1

Baseline: Lexicon Senti. We use a lexicon-based sentiment analysis similar to Hao and Chen-Burger (2021) as the baseline. The NLTK package provides a version of VADER (Hutto and Gilbert, 2014) sentiment analysis. It generates respectively the probabilities of Positive, Negative, and Neutral. We take Prob(Positive)-Prob(Negative) and average the values within the same day as the output.

FinBERT Senti. FinBERT is a BERT model pre-trained on financial communication text (Yang et al., 2020). Previous research shows these pre-trained BERT models outperform bag-of-word or rule-based models (Yang et al., 2020; Liu et al., 2020; Araci, 2019). We convert the logits output from FinBERT to probabilities of Positive, Negative, and Neutral. The rest procedures are the same as the baseline. The comparison between these two can provide some evidence of whether better sentiment analysis brings better stock price prediction.

FB Avg. One advantage of the BERT models is that we can fine-tune it with our own data. Constrained by our data size and computing resources, we take the Pooler output from the FinBERT model, instead of applying other deeper fine-tuning methods. The Pooler output of any piece of news has a shape of (768,). With truncating and padding, we take 50 Pooler outputs per day, so the shape of input for each day is (50,768). For this FB Avg model, we simply average the 50 Pooler outputs to get a daily representation with shape (1,768). Then we add a Dense(32) layer and a Dropout(0.3) layer on top of it, followed by the final Dense(1) output layer.

FB Attention. This model replaces the simple average step of FB Avg with a weighted average using the Self-Attention mechanism. The intuition is that some news or some aspects of the news may be more related to the stock market than others. The rest of the model settings are the same with FB Avg.

FB Transformer. The above models only use the information within a day to make the prediction. We can further extend the model to include the information from the historical news. For instance, a neutral sentiment after several positive sentiments may indicate things are getting worse, while a neutral sentiment after several negative sentiments may indicate things a getting better, and they may have different implications for stock price movements. Once we include the historical information, our task becomes a sequence-to-sequence task. Inspired by the success of transformers in sequence-to-sequence task tasks, we build a mini-transformer for our task.

The architecture of our mini-transformer model is shown in Figure 3. Similar to Vaswani et al. (2017), we have two inputs: tokenized text data and lagged outputs. Both inputs contain information for the past 60 days. For example, on day T, we use both the news and the historical returns of S&P 500 Index from day T-59 to day T as input to predict return on day T+1. Then we roll them for 1 day to predict returns on day T+2 and so on. The text input has a shape of (60,50,768), with the first dimension indicating the number of days, the

Table 1: Major settings of the Models

Models	NLP Model	Output from NLP model	Aggregation method
Lexicon Senti	Vader (lexicon)	Prob(P)-Prob(N)	Intraday simple average
FinBERT Senti	FinBERT	Prob(P)-Prob(N)	Intraday simple average
FB Avg	FinBERT	Pooler	Intraday simple average
FB Attention	FinBERT	Pooler	Intraday attention
FB LSTM	FinBERT	Pooler	Intra-&inter-day attention
FB Transformer	FinBERT	Pooler	Intra-&inter-day attention

second the number of news per day and the third the token dimension. The historical return input has a shape of (60,). We also add a Multi-head Attention layer to each of the text and return input. Then we use a Cross Attention layer to connect the two inputs and feed the output to three Dense layers to generate the final output.

Major differences between our mini transformer with the original one include: 1) due to limited data size, we only have 1 attention layer each instead of 6. As a result, the residual connection (Add&Norm) is not needed. 2) Instead of adding positional encodings to the inputs, we include the positional index as a separate feature. The reason is that, unlike Vaswani et al. (2017), which is trained from scratch, our model is based on a pre-trained model and we don't have enough data to perform a deep fine-tuning. Simply adding positional encodings to the tokens will distort the pretrained tokens. Since the length of the sequence is fixed at 60, we can use an integer sequence from 0 to 59 each divided by 60 as the positional index. 3) The original transformer uses causal attention in the decoder to prevent the use of future data. Since we already prevent the use of future data with a rolling window, our model uses global attention on both the encoder and decoder.

FB LSTM. "RNN+Attention" was the state-of-art architecture to build seq2seq models before Transformers. We also build an "LSTM+Attention" model to compare with the mini transformer model. The architecture is shown in Figure 4. It has the same input structure as FB Transformer and the use of attention mechanism is similar. Since the LSTM handles the sequence order, the positional encodings are no longer needed.

3.4 Model training and optimizing

We use KerasTuner[1] to tune hyperparameters including model parameters such as the number of attention heads and training settings such as learning rate. The details are given in Table 3 in the Appendix. KerasTuner helps select the best-performing hyperparameters based on validation MSE. We further train the models, select the best ones based on validation MSE, and generate the evaluation results on the test set.

[1] https://keras.io/keras_tuner/

4 Results and discussion

4.1 Results

The statistical and financial metrics of the models are shown in Table 2. Key observations from the results are:

1. FB Transformer performs the best from both statistical and financial perspectives. It has the highest correlation with the target and the lowest predicting errors. It also brings the highest return in absolute terms as well as in risk-adjusted measures.

2. The two models with encoder-decoder architecture (FB Transformer, FB LSTM) outperform other models, suggesting the inclusion of historical news and stock price changes helps improve the predictive power of the model.

3. FB Attention outperforms FB Avg, suggesting replacing simple average with Self-Attention-based weighted average improves the model.

4. Finally, models based on the final sentiment classifications (Lexicon Senti and FinBERT Senti) underperform those based on Pooler tokens, suggesting the sentiment alone does not provide enough information for predicting the stock movements. FinBERT Senti even underperforms Lexicon Senti, indicating that better sentiment analysis does not guarantee better stock price prediction.

Table 2: Statistical and financial metrics of models

Models	P. Corr	S. Corr	MAE	MSE	Total Return	Sharpe Ratio	Max Drawdown
Target	-	-	-	-	13.53%	0.36	-41.44%
Lexicon Senti	2.24%	-0.26%	-	-	8.98%	0.33	-20.18%
FinBERT Senti	-1.29%	-2.91%	-	-	2.63%	0.09	-22.94%
FB Avg	1.22%	1.29%	0.77	1.17	2.63%	0.10	-27.11%
FB Attention	4.13%	4.35%	**0.74**	1.13	14.15%	1.21	-9.27%
FB LSTM	**14.24%**	6.49%	0.75	**1.12**	41.44%	1.57	-12.32%
FB Transformer	13.01%	**12.60%**	**0.74**	1.13	**58.27%**	**2.15**	**-6.17%**

Note: P.Corr stands for Pearson Correlation and S.Corr stands for Spearman Correlation. The first two models do not use regression so MAE and MSE are not available. The Target in the first row is the series of S&P 500 itself. It can be viewed as the return of the model which always generates a signal 1. We include it here as a benchmark for the model-based investment return.

Figure 1: Simulated total wealth of $1 investment based on different model signals

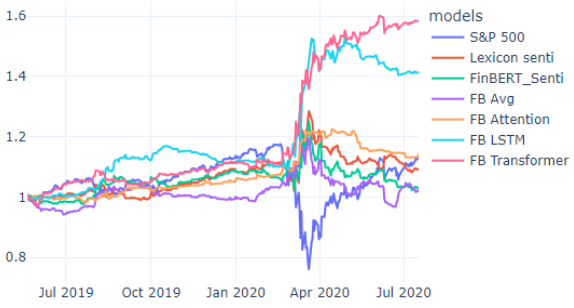
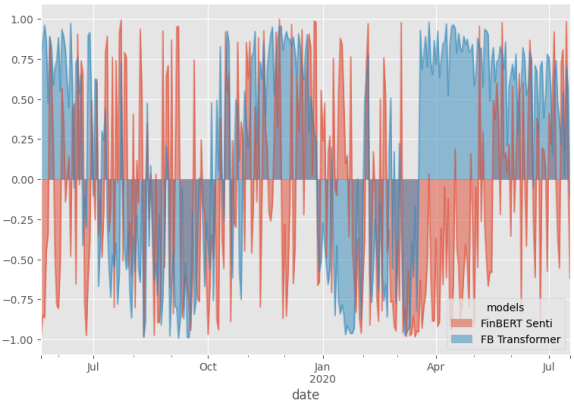


Figure 2: Signals of FB Transformer adjusted much faster than FinBERT Senti in Mar 2020



4.2 Behind the results

A more intuitive illustration of the models' performance is in Figure 1. It shows if we invested 1 dollar based on the signals of different models at the beginning of the test period, how much it would become over time. The performances of the models are close before 2020. When the COVID-19 pandemic broke out, all the models captured the negative sentiment and made a profit by short-selling.

What differentiated the models was the period from March 2020 to Jun 2020. During this time, the S&P 500 Index rebounded strongly with the help of unprecedented monetary easing and fiscal stimulus, but the number of new cases in the US increased to 10,000 per day, which naturally dragged the overall sentiment in financial media (see Figure 5). If a model only relied on the sentiment classification results, most likely it would keep short-selling and give back most of the profit during the market rebound. The best-performing models quickly adjusted their signals and made a profit again during the rebound (see Figure 2).

Based on the inputs and architectures of the models, we can reasonably infer that the best-performing models not only "knew" what was most important to the stock market, powered by the attention mechanism, but also adapted their predictions by comparing their past predictions with the past true values.

5 Conclusion

The major conclusions from this research are:

1. Better sentiment classification of financial news with pre-trained BERT models does not guarantee better stock price prediction, but fine-tuning with the hidden states can significantly improve the prediction.

2. Attention mechanism can help identify and "pay more attention" to the information that is most relevant to the stock market, therefore generating better prediction.

3. We propose a mini-transformer structure that takes both news text and past stock returns

as inputs, utilizes self-attention and cross-attention to aggregate information, and uses positional encoding to handle the series order. This model outperforms other candidates in both statistical and economic sense.

6 Further work

One possible improvement of this research is to use stock price data at higher frequencies, such as at minute level. The publishing time of the news in our dataset is precise to the minute, but we can only collect daily-level stock price data. High frequency not only brings larger data size but can locate the influence of news on stock price more accurately (such as the change in stock price in the five minutes after the news was published).

Another direction worth exploring is to disassemble our mini transformer model and build models with different combinations of the parts. In this way, we can better understand each part's contribution to the overall performance.

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1415–1425, Doha, Qatar. Association for Computational Linguistics.

Leonardo dos Santos Pinheiro and Mark Dras. 2017. Stock market prediction with deep learning: A character-based neural language model for event-based trading. In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 6–15, Brisbane, Australia.

Zhicheng Hao and Yun-Heh Jessica Chen-Burger. 2021. Analysing tweets sentiments for investment decisions in the stock market. In *Agents and Multi-Agent Systems: Technologies and Applications 2021*, pages 129–141, Singapore. Springer Singapore.

Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. 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*, WSDM '18, page 261–269, New York, NY, USA. Association for Computing Machinery.

C. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2020. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4513–4519. International Joint Conferences on Artificial Intelligence Organization. Special Track on AI in FinTech.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1354–1364, Beijing, China. Association for Computational Linguistics.

Ramit Sawhney, Arnav Wadhwa, Shivam Agarwal, and Rajiv Ratn Shah. 2021. Quantitative day trading from natural language using reinforcement learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4018–4030, Online. Association for Computational Linguistics.

Robert P. Schumaker and Hsinchun Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Trans. Inf. Syst.*, 27(2).

Paul C. Tetlock. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yumo Xu and Shay B. Cohen. 2018. 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, Melbourne, Australia. Association for Computational Linguistics.

Linyi Yang, Ruihai Dong, Tin Lok James Ng, and Yang Xu. 2019. Leveraging BERT to improve the FEARS index for stock forecasting. In *Proceedings of the First Workshop on Financial Technology and Natural Language Processing*, pages 54–60, Macao, China.

Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications.

A Appendix

Figure 3: The architecture of FB Transformer

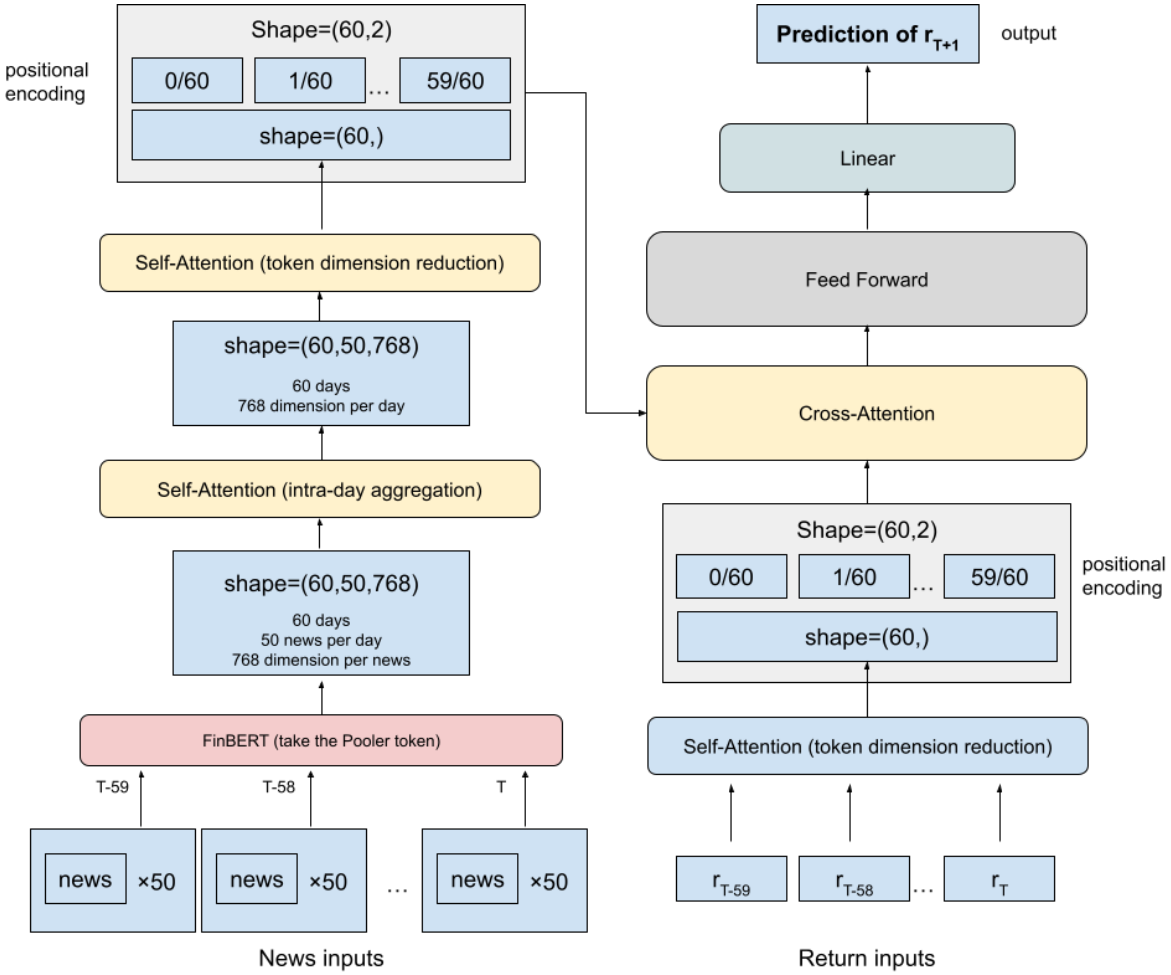


Table 3: Details of hyperparameter tuning

Hyperparameter	**Grid Search Values**
Hidden units	8, 16, 32
Dropout rate	0.3, 0.5
Number of multi-attention heads	1, 2, 4
LSTM units	8, 32
Learning rate	1e-2, 1e-3, 1e-4

Figure 4: The architecture of FB LSTM

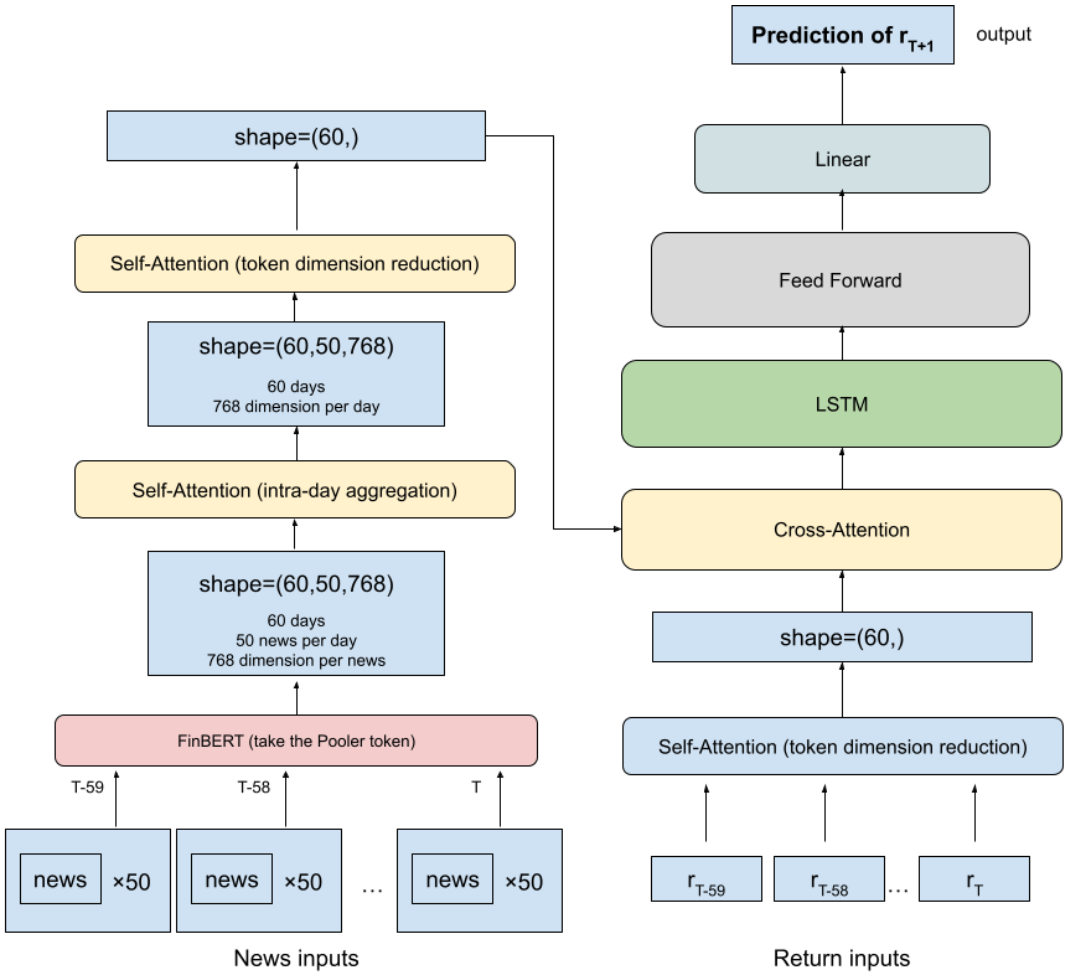


Figure 5: The overall sentiment was dragged by new Cov-19 case numbers while the stock market rebounded strongly during March 2020