Bidirectional Encoder Representations from Transformers for Modelling Stock Prices

Parinnay Chaudhry

Shaheed Sukhdev College of Business Studies, University of Delhi

FC 103: Managerial Economics

Dr. Sushmita

11 February, 2022

Abstract

Bidirectional Encoder Representations from Transformers (BERT) is a transformer neural network architecture designed for natural language processing (NLP). The model's architecture allows for an efficient, contextual understanding of words in sentences. Empirical evidence regarding the usage of BERT has proved a high degree of accuracy in NLP tasks such as sentiment analysis and next sentence classification.

This study utilises BERT's sentiment analysis capability, proposes and tests a framework to model a quantitative relation between the news and reportings of a company, and the movement of its stock price.

This study also aims to explore the nature of human psychology in terms of modelling risk and opportunity and gain insight into the subjectivity of the human mind.

Keywords: natural language processing, BERT, sentiment analysis, stock price modelling, transformers, neural networks, self-attention

Bidirectional Encoder Representations from Transformers for Modelling Stock Prices

How to predict stock prices? There have been a number of attempts to treat stock prices as any other time series data and perform trend and pattern analysis on it to try and predict it. Use of concepts like Long Short Term Memory (LSTM) (Pramod & Mallikarjuna, 2020), multiple regression, clustering (Enke et al., 2-11), etc. have been applied to try and accomplish this task. But, qualitatively speaking, past financial trends are not a predictor of future results, and thus, there is an intrinsic flaw in making this a purely numbers game.

A retail investor relies on the news and past performance of a company to decide whether or not the stock is a good pick for him or her. Past financial performance can be analysed using technicality, but what about the news? That part remains qualitatively understood by each individual investor and each of them takes one of three different actions based on that news- buy, sell or hold. A method to quantify the positivity or negativity of the news and a relation which predicts what the expected movement of the price could be based on that news would be useful for understanding the direction and rough magnitude of the price change expected when a new headline regarding the company is released.

This study uses BERT to perform sentiment analysis on the news of a particular company (this could be a single, prevalent headline of the day or many different headlines reported on the day) and using the corresponding close price of that day, finds an average value of the price movement for the kind of news that was published, in the context of the particular company.

Method

Summary of Methodology

The methodology followed in this study is summarised in the following flowchart *(diagram-1)*-

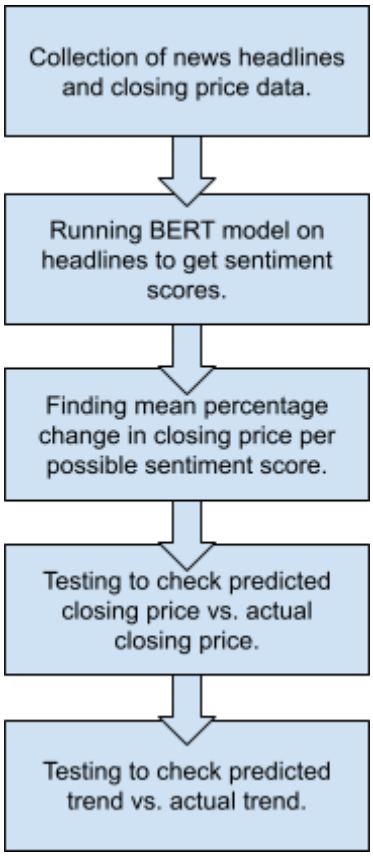


Diagram-1

Data Collection

The data needed for this study is all the prominent headlines of a particular publicly traded company in a certain time period and the stock closing prices of the corresponding dates in that time period. Since many data points here would only benefit the accuracy of the final output, a prominent company which stays in the news would be the most applicable use case. Therefore, news headlines and closing prices of Apple Inc. (ticker- AAPL) were used.

The news headlines were taken from a pre-existing *kaggle dataset* which contained the headlines of several large companies traded on the New York Stock Exchange and had a stock price greater than $10. The time period of this data set was from August, 2012 to January, 2020.

The initial dataset was filtered to get only AAPL news and the headlines obtained were in the date-range- *August, 2012 to January, 2020,* and a total of *20,231 headlines*.

The BERT Model

Architecture of BERT

The architecture of the Bidirectional Encoder Representations from Transformers (BERT) model is summarised in the following diagram *(diagram-2)*-

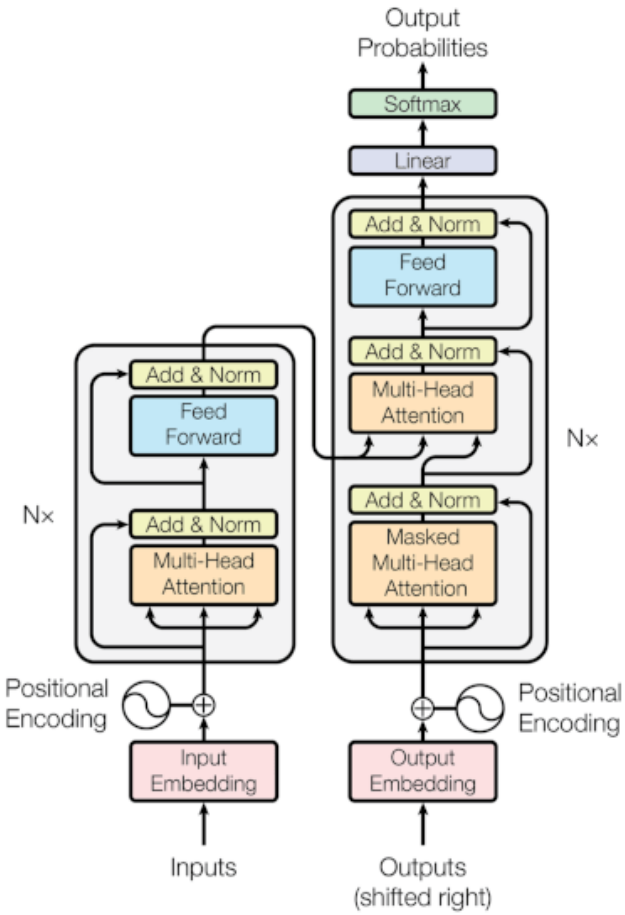


Diagram-2

Overview of the Functioning of BERT

BERT is based on the transformer layers methodology- "transformers" refers to the neural network architecture where each output node is connected to each input node. This allows the encoder to provide all the hidden states of each node to the decoder, not just the last node as in its NLP architectures which preceded BERT. To select inputs, the decoder performs a scoring calculation over all input matrices and does a softmax calculation to find the largest fractional value which depicts the highest contextual correlation. This context vector passes through a trained feedforward NN which indicates the output of the timestep being processed. *(Appendix section- 1 defines the parametrized function classes for the transformer block).* In case of BERT's preceding RNN based models, if there are *n* words, the *nth* token would be vectorized based on its features and viewed along with *n-1* tokens each in their own dimensional space. But, BERT's independence from utilising this vector-representation method allows for faster training and processing times. BERT utilises *WordPiece Tokenization* and processes collections of words with it's capability to view it's current material under-processing both from left to right and right to left. This not only leads to training time reduction but also advances in NLP ability, as elaborated in the later sections.

The final output of BERT is an array of probabilities which can depict the probabilities of sentiment scoring, the prediction of whether or not a selected word is the next word in a sentence or not, etc. based on the use-case.

As mentioned in the above paragraphs and as cited at several places in the architecture diagram, the softmax function for an input vector with values of the kind z_i where $0 \leq i \leq k$ with k being the number of classes in our multi-class classifier system is-

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

The Self-Attention Process

Compared to traditional back-propogation through time algorithms which are used to train neural networks, transformers rely on an additional process known as *self-attention* (Vaswani et al., 2017). The general flow of the attention process is as follows-

- The initial input embedding vector is multiplied with a randomly initialised matrix (original dimension is 64x512) and the value of this matrix is fine tuned by backpropagation in time. The three vectors hence generated are the "key," "query," and "values" vectors.
- Self-attention values are then calculated using the "key" and "query" vectors. They depict how much "attention" is to be paid to different parts of the input sentence during encoding at a certain location. The result is divided by a constant (the square root of the first dimension of the randomly initialised matrix), and a softmax calculation is run over all these scores to get a relevance value of each word to the word at the current encoding position.
- Lastly, the "value" vector is multiplied with the softmax results to get the self-attention value at the current node.

(Note- The above is the flow of "scaled dot-product attention" where value distribution is calculated based on the similarity between the query and key vectors.)

Training of BERT

The design of BERT has allowed it to be trained using a vast corpus of unlabelled data (Devlin et al., 2018). Compared with its predecessors which required a large amount of labelled

data for training, BERT efficiently trained on the Wikipedia corpus of 2.5 billion words and the BookCorpus of 800 million words for two tasks- masked language modelling (MLM) and next sentence prediction (NSP).

MLM is the task of inputting a sentence to the model with a few tokens hidden or masked and having the model output the completed sentence. NSP is the process of training the model for recognising context. In this task, the model is given two sentences and it's task is to output which sentence comes first and which one after it. The "positional encoding" layers are used for this task in finding positional context of the tokens of the sentence. These two training objectives allow BERT to have a high accuracy in recognising and finding words and also in understanding the context of the words of a sentence.

Usage of BERT in the Study

In the case of this study, our goal with BERT is to classify news headlines based on whether or not they are positive, negative or neutral and then correspond percentage changes in price according to those categorizations. In our case, we use BERT to generate an array with five probability values each depicting how positive, negative or neutral the sentence is likely to be. The index (from 1 to 5) of the highest of those probabilities in the array is referred to as sentiment score.

The sentiment scoring system utilised is summarised in the following table *(table-1)*-

Sentiment	Score
Highly Positive	5
Positive	4
Neutral	3
Negative	2
Highly Negative	1

Table-1

Each of the collected headlines was run through the model and the sentiment scores were appended in front of the headline. In case of a particular date where there were several headlines with different sentiments, the mean of the sentiment scores was taken and assigned to that particular date. Since a retail investor would view all of those headlines and then take a particular action related to the stock of Apple Inc., it's only logical that all the headlines be given a weightage in the sentiment score of the day.

A glimpse of assigned scores is shown in the table below *(table-2)*-

Headline	Release Date	BERT Sentiment Score
Time To Short Apple Hardly	24/09/12	2
Apple Has The Flu	17/03/13	1
Stage Is Set For A Major Breakout In Apple	06/09/13	4
APPLE Stabilised For The Short Term	07/08/13	3

Table-2

Utilising the Sentiment Scores for Stock Price Prediction

After using BERT to classify all the headlines of the dataset, each headline and sentiment score set was appended with the percentage change in closing price of that day with respect to the previous day. This helps us map the effect of that day's news on the closing price.

Then, the sentiment scores were segregated based on the value and the percentage change in closing price was averaged for each of the 5 possible sentiment score values to get 5 mean percentage change values.

The output is summarised in the table below *(table-3)*-

Sentiment Score	**Mean Percentage Change in Closing Price w.r.t. Previous Day's Closing Price**
1	0.09599214%
2	-0.04461355%
3	0.13888213%
4	0.04420133%
5	0.09729779%

Table-3

We observe here that there is a large positive value for percentage change in closing price for a sentiment score of 1 (highly negative news). This can be attributed to some of the headlines of the dataset not being entirely related to AAPL, more to the market in general or any other inconsistencies in the dataset.

Testing

Testing the Derived Values Against Actual Closing Price of the Stock

To test the accuracy of the derived mean percentage changes w.r.t previous day per sentiment score, we take the closing price of AAPL on the first day for which we have a headline in our data set, and iterate through each day that we have a headline, changing the closing price based on the sentiment score of the headline(s) of the day. To account for any errors in the sentiment scoring by BERT, the scores were divided into categories of positive, neutral and negative and the ranges and corresponding mean percentage changes in closing price are as follows *(table- 4)*-

Sentiment Score Range	**Mean Percentage Change in Close Price w.r.t Previous Day's Closing Price**
1-2.99	0.02568929%
3-3.99	0.13888212%
4-5	0.07074955%

Table- 4

The final output after utilising the above values on the closing price is summarised as follows *(table-5)*-

Date	Actual Closing Price	Predicted Closing Price
16.07.2012	$21.56	(Actual value used)
28.01.2020	$79.42	$80.45
Difference between Actual Closing Price and Predicted Closing Price		**$1.03**
Accuracy of Prediction		**98.7%**

Table- 5

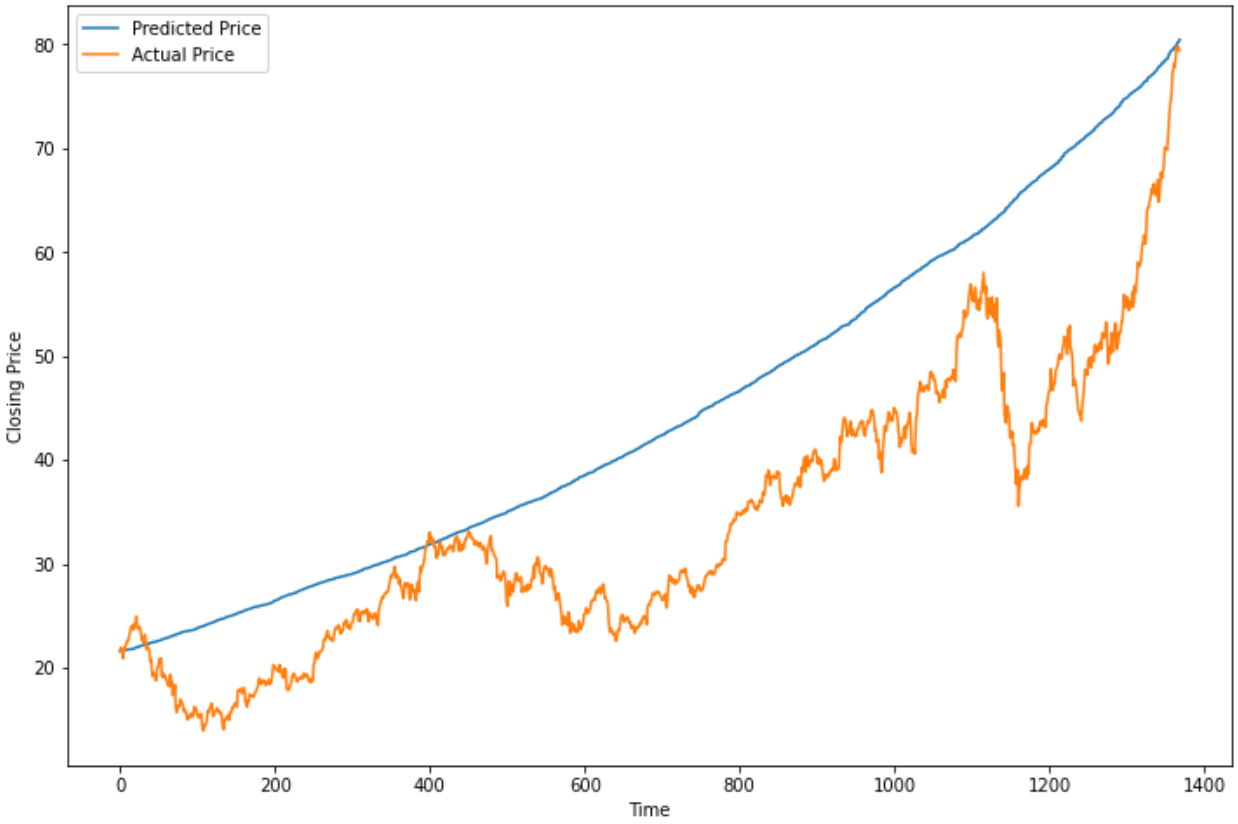
Testing the Predicted Day to Day Price Movement Against Actual Day to Day Movement of the Closing Price

To test the adherence of the predicted prices against the actual movement of the closing price, the root mean squared error (RMSE) was calculated and the actual movements plotted against the predicted movements. The results have been summarised below *(table-6)*

RMSE	11.023

Table- 6

The graph plotting the movement of the predicted prices and actual prices is *(graph-1)*-



Graph-1

Conclusions

Interpretation of Results

Clearly, while the final prediction price is extremely close to the actual final price, the adherence to the actual movement of the stock price is poor thus making the accuracy of the final close price a chance occurrence.

The primary cause of the poor adherence of the predicted prices to the actual movement of the closing price seems to be a lack of high quality data and corruptions in the data set used for the study. Headlines which might not be exactly related to AAPL or unreliable sources could've been the cause of the corruption.

Consistent headlines scraped from reliable sources will not only lead to better results in the "sentiment scoring of the headlines" step but also in the subsequent step of assigning mean percentage change to each sentiment score range.

Conclusion to Study

The suggested framework for modelling and predicting stock prices based on news headlines is heavily reliant on clean, usable headlines, the utilisation of which could potentially produce better results.

Given the current data set used and the methodology applied, however, the average accuracy of a prediction won't be good enough to warrant a trading decision. Long duration scraping of reliable data could produce better results but doesn't guarantee the success of the framework considered in this study.

Appendix

Section-1: Mathematical Description of the transformer architecture

As described by "The Transformer Model in Equations" (Thickstun, 2020),

the transformer block is a parameterized function class $f_\theta : \mathbb{R}^{n\times d} \rightarrow \mathbb{R}^{n\times d}$

If $x \in \mathbb{R}^{n\times d}$, $f_\theta(x) = z$,

Where $Q^{(h)}(x_i) = W^T_{h,q} x_i$, $K^{(h)}(x_i) = W^T_{h,k} x_i$, $V^{(h)}(x_i) = W^T_{h,v} x_i$ (1)

$$\alpha^{(h)}_{i,j} = softmax_j\left(\frac{\left\{\langle Q^{(h)}(x_i), K^{(h)}(x_j)\rangle\right\}}{\sqrt{k}}\right), \quad (2)$$

$$u_i^{'} = \sum_{h=1}^{H} W^T_{c,h} \sum_{j=1}^{n} \alpha^{(h)}_{i,j} V^{(h)}(x_j), \quad (W_{c,h} \in \mathbb{R}^{k\times d}), \quad (3)$$

$$u_i = LayerNorm(x_i + u_i^{'}; \gamma_1, \beta_1), \quad (\gamma_1, \beta_1 \in \mathbb{R}^d), \quad (4)$$

$$z_i^{'} = W_2^T ReLU(W_1^T u_i), \quad (W_1 \in \mathbb{R}^{d\times m}, W_2 \in \mathbb{R}^{m\times d}), \tag{5}$$

$$z_i = LayerNorm(u_i + z_i^{'}; \gamma_2, \beta_2), \quad (\gamma_2, \beta_2 \in \mathbb{R}^d) \tag{6}$$

Here, the LayerNorm function defined for $z \in \mathbb{R}^d$ is-

$$\text{LayerNorm}(z; \gamma, \beta) = \gamma\frac{(z-\mu_z)}{\sigma_z} + \beta, \quad \gamma, \beta \in \mathbb{R}^k \tag{7}$$

$$\text{And, } \mu_z = \frac{1}{k}\sum_{i=1}^{k} z_i, \ \sigma_z = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(z_i-\mu_z)^2} \tag{8}$$

And the softmax function being defined as-

$$\hat{y} = softmax(W_z^T z) = \frac{exp(W_z^T z)}{\sum_{k=1}^{m} exp(W_z^T z)_k}, \quad W_z \in \mathbb{R}^{d\times o} \tag{9}$$

θ consists of the entries in the matrix of weights associated with the NN, W and the LayerNorm parameters γ and β.

The input x ∈ $\mathbb{R}^{n\times d}$ is a collection of *n* objects of *d* features each or a sequence of *d*-dimensional vectors of length *n*.

The hyperparameters of the transformer are *d, k, m, H* and *L.* They are constants with values as follows: *d* = 512, *k* = 64, *m* = 2048, *H* = 8 and *L* = 6. Here, *L* represents the depth of the transformer blocks, *H* represents the index of the attention-head, *m* represents the token limit, and *n* x *d* refers to the output matrix dimensions.

Section- 2: Interpretation of the mathematical description of the transformer architecture

It is imperative to mention that the collection of *n* objects in the *n* x *d* input matrix gets transformed to an output matrix of the same dimensions and it consists of a collection of *n* objects not intrinsically connected in a sequence. If a sequence does exist, the positional encoding process has to be utilised and the sequence has to be encoded in those layers. Also, in

the architecture of the transformer, a complete interconnectivity between the nodes exists but the parameters are independent of the value of n (similar to an RNN).

For viewing the equations defined in section-1 of the appendix in the context of the architecture of the transformer block, the two "layers" are the multi-headed self-attention layers (refer to equations 1, 2 and 3) and a fully connected object specific layer (refer to equation 5). The normalisation layers are expressed in equation 4 and equation 6 defines the residual connections between the attention layer(s) and fully-connected layers are fine-tuned based on empirical data.

Section- 3: Description of the self-attention layers and mechanism

Equations 1 and 2 are sets of equations where h ranges from 1 to H. Each of these equations with the parameters for a value of h is an attention head and the collective set of equations are the multi-headed self-attention. The weights are given by the $\alpha_{i,j}^{h}$ term and is the attention-weight. The qualitative meaning of the weight is the controlling term that determines how much attention x_i gives to the element x_j in the head with index h.

As far as the object x_i and it's relation to the "query," "key," and "value" matrices as mentioned in the "THE BERT MODEL" section, for each object x_i, there is an associated query $Q(x_i)$ used to test the compatibility of x_i with the key of each object x_j defined as $K(x_j)$. "Compatibility" of x_i with x_j is measured by the inner product $\langle Q(x_i),\ K(x_j)\rangle$. If the value of the inner product is high, the query of x_i is a match with the key of x_j. Further, we can look up x_j's value $V(x_j)$ and make u_i a soft lookup of values which are similarly compatible with the key determined for x_i.

Section- 4: Positional encoding layers

Since the transformer architecture doesn't interpret sequential relationships which exist amongst it's inputs, the positional (or contextual in the case of NLP tasks) relationships are expressed as encoded features in the positional encoding layers.

The encodings are based on sinusoidal positional embedding defined by-

$$p \in \mathbb{R}^{n\times d}: p_{k,2i} = sin\left(\frac{k}{10000^{2i/d}}\right), p_{k,2i+1} = cos\left(\frac{k}{10000^{2i/d}}\right)$$

Then, the rectified linear unit function (ReLU) is utilised to build distinct representations of the inputs and positions:

$$z = W_{zx}^{T} ReLU(W_{z}^{T} x_{1}) + p$$

References

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv.org. https://arxiv.org/abs/1810.04805

Enke, D., Grauer, M., & Mehdiyev, N. (2-11). Stock Price Prediction with Multiple Regression, Fuzzy Type-2 Clustering and Neural Networks. *Procedia Computer Science*, *1*(6), 201-206. https://www.researchgate.net/publication/251714399_Stock_Market_Prediction_with_Multiple_Regression_Fuzzy_Type2_Clustering_and_Neural_Networks

Galassi, A., Lippi, M., & Torroni, P. (2021). Attention in Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(10), 4291-4309. https://arxiv.org/pdf/1902.02181.pdf#:~:text=The%20attention%20mechanism%20is%20a,to%20its%20higher%20level%20representation

Pramod, B. S., & Mallikarjuna, P.M. S. (2020). Stock Price Prediction Using LSTM. *Test Engineering and Management*, *83*(May-June 2020), 5246-5251. https://www.researchgate.net/publication/348390803_Stock_Price_Prediction_Using_LSTM

Thickstun, J. (2020). *The Transformer Model in Equations*. johnthickstun.com. https://johnthickstun.com/docs/transformers.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. https://arxiv.org/abs/1706.03762