USING NEWS TO PREDICT STOCK PRICE MOVEMENTS

Dmitrii N. Zhavoronkov
Department of Economics
Higher School of Economics
Saint Petersburg, Russia
dnzhavoronkov@edu.hse.ru

July 3, 2019

ABSTRACT

The objective of this research is to build an algorithm which predicts stock movements in the following day by applying modern machine learning approaches and refine state of the art prediction algorithms for the problem. Based on that, the work identifies the importance of different types of news and finds out the patterns between news and stock price movements. The market data is collected from Yahoo! Finance [32]. Reuters [26] and Stockwatch.com [28] are used as the sources of news articles. As a result, news features make a significant contribution to the final prediction of stock price movements, BERT makes an absolute improvement of 2.36%. Visualization of attention layer has been implemented, which can be used in further researches for feature generation.

***K*eywords** Machine learning · Deep learning · Stock market · Sentiment analysis · Bidirectional Encoder Representations from Transformers

1 Introduction

Stock movement prediction is an area of interest for a wide range of industries. In a world with a growing amount of information almost any trader is interested in knowing about a future situation on the market and informational space in order to make a successful investment and make a profit. Precise and effective prediction algorithms indirectly help investors as well by providing additional supportive information such as the future price of a stock or its direction.

It is a well-known fact that a stock price is determined by the behavior of human investors, and the investors determine stock prices by using publicly available information to predict how the market will act and react. Financial news articles can thus play a major role by influencing the movement of stock as humans react to the information. It has been shown that the financial market is "informationally efficient" (Fama, E.F., 1965) [14] - stock prices reflect all known information immediately, and the price movement is in response to news and events going around a company. Also, previous researches have suggested that the relationship between news articles and stock price movement exists, as there is a lag between when a news article is released and when the market reacts to the information [18].

As web information grows, researchers started eagerly applying various machine learning techniques to represent such kind of data from textual to numerical forms. This is done to make an improvement in predictability of the stock market.

2 Related work

The purpose of this chapter is to relate this study to previous academic papers on the academic subject domain. The goal here is to establish the relationship between news and the stock market to which they relate. Additionally, it is worth outlining if the effect from the news is immediate or it has a long-lasting consequence. Another purpose of the literature review is to describe the most commonly used contemporary methods as well as outline their results.

Tetlock [29] showed that the performance of daily prediction is better than the one made on weekly or monthly basis. This idea became popular and Ding [11] developed it further. The author illustrates that the effects of three actual events, happening around Google, have an impact on the company's stock price. He showed that the relationship grew on the first day and reached its highest point on the second day, yet gradually dropped over time. Despite the relatively weaker impact of long-term events on stock price, the volatility of stock markets is still being affected by them. Ultimately, the author concludes that the influence of the events through time is diminishing and target should be chosen depending on the goals a researcher sets.

Xiaodong [22] used the English sentiment dictionary to count words with negative and positive pulses in magazine articles in order to perform sentiment analysis and predict stock returns on the Hong Kong Stock Exchange. The author used the sentiment analysis dictionary instead of the bag-of-words approach and showed that sentiment analysis helps to improve the predictions. Scholar highlights, that models with sentiment analysis outperform the bag-of-words models in both independent test and validation data sets. Nevertheless, the method is highly inaccurate and debatable due to the specificity of financial terminology and low number of words in a corpus.

The work, written by Ding [10], makes significant contribution to the field as financial news from traditional media sources have been largely explored in the past and proved to have high predictive power. Moreover, the author corroborates that text from the article is less valuable for prediction than it's headline, enabling the reduction of the algorithms' training time. Scientist uses deep learning convolutional neural networks to automatically extract features from sequences of words and market features, capturing highly non-linear relationships such as context-dependent meanings. In the same article, the results are examined on both the S&P 500 index and 15 low-, middle-, and high-ranking selected companies. This model achieves better performance compared to the baseline methods of related works on both individual stock and S&P 500 index predictions, reaching 65.48% and 64.21% respectively, compared to 61.47% and 58.83% achieved previously. It's hard to criticize the result for an index, although precision for the selected companies is overestimated as estimation of all 500 companies would lead to lower accuracy scores due to larger volatility and data diversity. Notably, the model achieves relatively higher improvements on those low-ranking companies, for which only a small fraction of news was available.

Some scholars have combined all common methods applied to the task, to present a comprehensive comparison. For instance, Kraus, M., & Feuerriegel, S. [19] examined classical and modern approaches of machine learning to predict stock movement directions in the German stock market. As a source of media, they worked with German ad hoc announcements in English. Ad hoc announcement is information, related to rate changes, which Securities issuers are obligated to publish in a timely manner. These are meant to prevent the scenario of insider trading by providing the same information to all stock market participants at the same time. Traditional methods included Random Forest, Support Vector Machine, Naïve Bayes, AdaBoost, and Gradient Boosting. Naïve Bayes algorithm was taken as a baseline with a score of 54%. Among the methods, the best score was achieved with Random Forest (56.2%). The authors mentioned that deep learning outperforms traditional ML techniques. For instance, the LSTM architecture yields an improvement of 6 percentage points over the naïve baseline, reaching 60.1% of accuracy. Ad hoc materials gave rise to the score of a bit more than 1%, which is a good contribution as no specific features were additionally created. But even an increase of 1% would lead to a monetary value growth of the portfolio considerably. Authors mentioned those accurate predictions based on financial news are hard to evaluate due to the complexity of natural language and financial jargon. Thus, both difficulties underline the necessity to create more complex models, which solve the task better. In conclusion, scholars provide an idea that nowadays neural networks demonstrate the achievements in different tasks, including stock price movement prediction, and research in deep learning is rapidly growing and new and more accurate models are developed regularly.

3 Delimitations of the study

We focus our research only on companies represented in the S&P 500 index, which are the ones with the largest capitalization in the world. As a result, the prediction of movement directions on other markets might lead to completely different precision, feature importance, and, consequently, results. A decision to study these specific companies is related to the fact, that they attract a lot of attention and, therefore, are unlikely to go through individual manipulations and random shocks, compared to relatively small companies from Russian, Indian and other stock exchanges. Moreover, data for analysis is collected for the last 10 years, from January 2009 till January 2019. Also, two large and reliable sources of financial news were used: Reuters [26] and Stockwatch.com [28]. It is possible to enlarge the corpus, but it is unlikely to lead to improvement as daily news are highly correlated.

4 Methodology

This section introduces the methods used in the analysis, as well as the dataset and feature generating process.

4.1 Data

Nowadays, the internet provides a tremendous amount of textual information related to any specific company such as news, posts, which can be used as sound predictors for an algorithm. As was mentioned above, data was collected in the period from January 1, 2009, until January 1, 2019. As news are highly correlated from source to source daily for any company, two large news sources were selected for the work: Reuters [26] and Stockwatch.com [28]. Both sources specialize in financial information and provide a large corpus for analysis. Yahoo! Finance [32] was used as a source of numerical stock quotes. The companies of interest compose S&P 500 index, which are 500 enterprises with the largest capitalization in the world. All three sources don't provide information for download, therefore some techniques were employed to scrape textual news and numerical stock information. The total number of rows after data cleaning (removing NaNs) is 1168477; it contains 487 companies. Tickers of missing ones are WCG, JEF, JKHY, FLT, FTNT, CPRT, EVRG, ROL, MSCI, LIN, KEYS, HFC, ABMD.
Original numerical data consists of the following columns (Table 1).

ticker	Date	Adj.Close	Close	High	Low	Open	Volume
AAL	2008-12-31	7.405228	7.73	7.87	7.48	7.48	4194100.0
AAL	2009-01-02	8.037499	8.39	8.48	7.67	7.73	5167000.0
AAL	2009-01-03	7.980019	8.33	8.39	7.96	8.38	3457100.0

Table 1: Numerical Data. The ticker is a company's short name; Date is a date of observation; Adj. Close is an amended Close price after accounting for any corporate actions; Close, High, Low and Open are Closing, Highest, Lowest and Opening price in a given day; Volume is the number of shares that changed hands during a given day.

Textual corpus has the following structure (Table 2). Stockwatch and Reuters contain 279876 and 887405 relevant observations respectively.

Company Name	Date	headline
American Airlines Group	2009-01-05	Allied Pilots Association Critical..
American Airlines Group	2009-01-12	AAL Starts the New Year...
American Airlines Group	2009-01-15	AAL Launches...

Table 2: Textual Corpus. The first column represents the company's title, the second one date of observation, and the third one represents news headlines for a given company on a specific date.

4.2 Target variable

Our final goal is to predict if the closing price of a stock is higher than the closing price on the previous day. Thus, if the closing price is higher tomorrow than today, it would be profitable to buy a stock, otherwise to sell it. Formally for any company i in moment t:

$$Target_{i,t} = \begin{cases} 1, \text{if} & \frac{Close_{i,t+1} - Close_{i,t}}{Close_{i,t}} \geq 0 \\ 0, \text{otherwise} \end{cases}$$

Just like in previous studies, accuracy was selected as a performance metric. Moreover, it is a reasonable choice since the distribution of the target variable is balanced.

4.3 Preprocessing

Most of the machine learning algorithms perform better on normalized time-series data as it has a consistent scale of distribution. Additionally, as most of the methods use gradient descent as an optimization algorithm for finding optimal model parameters, scaling becomes necessary as it allows gradient descent to converge with more stability. Formally, scaling for any company i and feature k:

$$X_{i,k,scaled} = \frac{X_{i,k} - X_{i,k,min}}{X_{i,k,max} - X_{i,k,min}}$$

4.4 Feature engineering

The subsection consists of two parts: Features derived from text data and features derived from numerical financial data.

4.4.1 Features derived from textual data

We classify the text data into sentiments of three classes: $positive$, $neutral$, $negative$. The output of the classifier is a vector that can be interpreted as probabilities that the article is positive/neutral/negative. Features from the ensemble of logistic regression and Random forest predictions for any company i in period t were generated:

$$EnsembleSentimentPositive_{i,t} = \frac{1}{2} * (PredictedLogitPositive_{i,t} + PredictedRFPositive_{i,t})$$

$$EnsembleSentimentNegative_{i,t} = \frac{1}{2} * (PredictedLogitNegative_{i,t} + PredictedRFPositive_{i,t})$$

Also, the predicted probabilities of Bidirectional Encoder Representations from Transformers [30] were used as a feature.

$$BertFeature_{i,t,p} = Predicted(SentimentPolarity_{i,t,p})$$

where p stands for sentiment class.

For every observation, we created a mean value of positive and negative sentiments of the ensemble for the last q days for every company i in period t.

$$MeanPositiveSentiment_{i,q,t} = \frac{1}{q} \sum_{t-q}^{t} EnsembleSentimentPositive_{i,t-q}$$

$$MeanNegativeSentiment_{i,q,t} = \frac{1}{q} \sum_{t-q}^{t} EnsembleSentimentNegativet_{i,t-q}$$

The chosen q were 2, 3, 5. The intuition is the following: if for the last q days there were mostly positive/negative around company i, the stock price of a company in period $t + 1$ is likely to go up/down, which is expected to make a meaningful contribution to the final prediction. If in any given moment t for a company i there is no article in data, a feature takes a value of 0.

4.4.2 Features derived from numerical data

On the basis of market data, a bunch of technical analysis features was implemented. All of these features help to detect a further direction of a stock price movements. Intuitively, these indicators apply functions to market features for the last k periods of time, which enables to capture market patterns longer, than 1. It is also worth to mention, that for any indicator there default lag parameters exist, offered by authors. As most of these indicators were derived a long time ago and used for different purposes (obtaining information in long-term/short-term), the parameters, or lags, in general, were chosen no more than 14 to catch short term changes. All the indicators, used in the work, are described below.

- Difference between Close Price and Moving Average; Exponential Moving Average.
  ma - moving average; ema - exponential moving average.
  Many authors suggested trading strategies based on moving averages [9], although all these indicators show approximately the same efficiency [21]. Therefore, as the variables we generated as differences between Close Price at time t and moving averages, which can serve as a signal to buy($C_t - ma_t(k) > 0$) or sell($C_t - ma_t(k) <= 0$).
- %k and %d stochastic oscillators According to the author [20], the general idea of oscillators is that Close price is likely to stay around its previous maximum value when it has an expected tendency to grow; around the previous minimum with a tendency to drop. Therefore, we observe the following strategy:

$$\begin{cases} \text{Buy} & \text{, if } \%k - \%d > 0 \\ \text{Sell} & \text{, o/w} \end{cases}$$

- %b derived from Bollinger bands [4]. Bollinger lines, suggested by John Bollinger [5] are widely used by traders on stock markets as they give a large piece of information about market volatility.
- Price Rate of Change (ROC) [23]. It's a simple but useful indicator, which measures the percentage change in price between the current price and the price a certain number of periods ago. The ROC indicator is plotted against zero, with the indicator moving upwards into positive territory if price changes are to the upside, and moving into negative territory if price changes are to the downside.
- Bull and Bear power. These indicators jointly show the relation of buyers' and sellers' power [12].
- Moving Average Convergence/Divergence

  The author derives two general types [2] - linear indicator and histogram one, which is derived by the difference between MACD and moving average with periods l and j.
- Tenkan-sen; Kijun-sen; Senkou-Span A; Senkou Span B; Chikou Span - lines, which jointly form Ichimoku cloud [13].

  Such a complex system of indicators enables to estimate the direction of the trend and it's stability [24].
- Average Directional Index
  It is an indicator of trend strength in a series of prices of a financial instrument. The ADX does not indicate trend direction or momentum, only trend strength. ADX ranges between 0 and 100. Generally, ADX readings below 20 indicate trend weakness, and readings above 40 indicate trend strength. An extremely strong trend is indicated by readings above 50.
- Relative Strength Index (RSI)
  RSI was developed by J. Welles Wilder [31], is a momentum oscillator that measures the speed and change of price movements. The RSI oscillates between 0 and 100. Traditionally the RSI is considered overbought when above 70 and oversold when below 30. RSI can also be used to identify the general trend.
- Volume Weighted Average Price (VWAP)
  It gives the average price security has traded throughout the day, based on both volume and price. It is important because it provides traders with insight into both the trend and value of security.

4.5 Classification Algorithms

In our research, we will use the following algorithms: K Nearest Neighbors, Logistic Regression, Random Forest, Gradient Boosting (LightGBM implementation [17]) for classification. Facebook's implementation of word2vec, called FastText [3], [15], was used in order to map headline articles to vector space. Also, Bidirectional Encoder Representations from Transformers [30] was used as an independent classifier both to articles for sentiment analysis and to prediction of stock price movements only by the news. In our work we have used BERT-Base, Cased: 12-layer, 768-hidden, 12-heads, 110M parameters because of the limitations on GPU. Moreover, better results from BERT can be obtained if to fine-tune the model more precisely.

In many research papers authors use Support Vector Machine algorithm (e.g. [16], [8], [7]) and Recurrent neural networks (e.g. [25], [1], [6]). In our case with a large number of companies, the first algorithm suffers from the curse of dimensionality and cannot be trained. Moreover, there were attempts to implement Long short-term memory (LSTM) algorithm, however, due to the fact that the data contains multiple time-series, we added trainable embedding layer out of companies' tickers. Though obtained results were not significantly better than logistic regression baseline in Table 7, therefore, the algorithm was not described further in the work.

5 Experimental Setup and Results

There were two main setups for testing.

You can see the main pipeline in the Figure 2. Before concatenation and final classification, several additional procedures were executed. Stock market data was cleaned manually as it contained some extreme numbers and missing values. Afterward, normalization and technical feature indications, described in the methodology section, were applied separately for each of the tickers. The financial news were pre-processed by removing punctuation, stop-words and applying stemming, which helped us to reduce the number of words without losing the semantics.

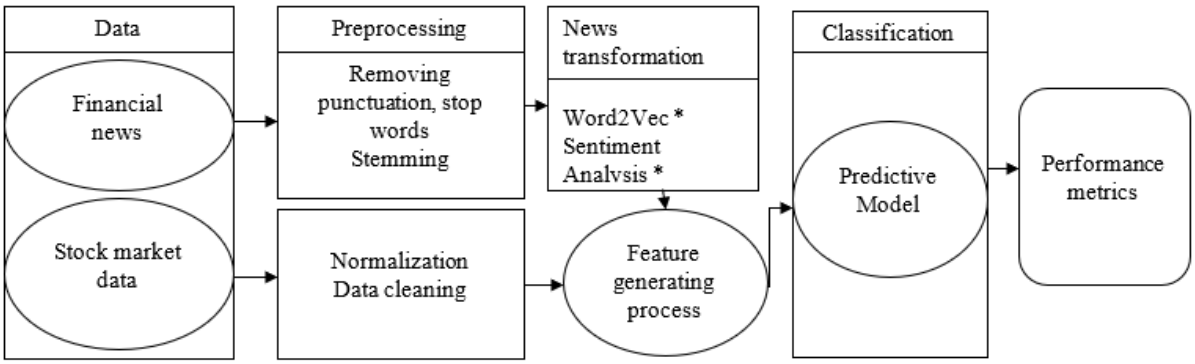

Figure 1: Main pipeline. The first block represents all the collected data we use, which contains Financial news and stock market data; the second block represents preprocessing of different both types of data; the third block shows the transformation of news into vector space and feature generating process, which consists of News vectors, Sentiment analysis, and Technical analysis; Fourth block depicts stock market price movements classification which leads to performance metric comparisons.

5.1 Sentiment analysis

First, two pretrained algorithms were used for classification on corpus: "VADER" from NLTK package and "TextBlob". Both algorithms show approximately the same statistics. The results are represented in Table 3.

TextBlob and VADER results				
Statistic	TextBlob	TextBlob absolute	VADER	VADER absolute
mean	0.043040	0.066282	0.121039	0.207930
std	0.148840	0.140045	0.303241	0.251735
min	-1.000000	0.000000	-0.957100	0.000000
25%	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000
75%	0.027778	0.083333	0.318200	0.401900
max	1.000000	1.000000	0.986500	0.986500

Table 3: Sentiment analysis statistics. The first column represents main statistics of predicted headline polarities; Second and Fourth columns correspond to statistics of TextBlob and VADER algorithms; Third and Fifth columns show respective statistics in absolute values of predicted polarities.

The compound is an aggregated score, derived from positive, negative and neutral polarities and which is normalized between -1 (extreme negative) and +1 (extreme positive). According to the table above, algorithms perform poorly on financial news, mostly marking them as neutral. Meanwhile, VADER algorithm performs better than TextBlob. Undoubtedly, the issue could be in the neutrality of financial headlines, but after taking a closer look into the data manually, the hypothesis was rejected. For example, "exxon mobil eyes multi billion dollar investment" is predicted as neutral by both algorithms, although it is naturally not true. Therefore, it was decided to change the problem solving approach and move to the classification task with 3 classes: $positive, negative, neutral$.

The method, used for text transformation, is called word2vec. It maps each unique word to a vector in n-dimensional space of fixed size and its advantages are the capability of capturing the context of a word in a document, semantic and syntactic similarity and reduction of sparsity. In our case, we have used Facebook's implementation of word2vec, called FastText [3], [15], which maps each word to a vector of size 300.

To obtain a vector of a headline, mean values were taken by all of 300 coordinates. Embedding results are represented in Table 4.

The data for training was collected from the recent competition, organized by Two Sigma [27], which includes financial headlines with sentiment labels.

Embedded instances					
Obs	Coordinate 0	Coordinate 1	Coordinate 2	...	Coordinate 299
1	0.321323	-0.238423	-0.033221	...	0.58912
2	0.541212	0.148840	0.140045	...	0.251735
3	0.939223	-0.32315	0.391485	...	0.4716364
...	...	...	...	...	...

Table 4: News Embeddings. Each headline from Table 2 is represented in a vector space of size 300 by applying Word2Vec transformation. The first column represents observation, the rest contain it's vector representation one coordinate a column. Total number of columns is 301.

5.2 Sentiment analysis evaluation

As there are 3 classes of sentiments, stratified k fold validation was used to get unbiased results. It means that a whole dataset was split into k folds with an approximately equal number of instances of every class. One specific issue to use stratified cross-validation is that even unbiased or balanced algorithms are not able to learn or test a class that isn't represented at all in a fold. Furthermore, even the case where only one of a class is represented in a fold doesn't allow to generalize. By choosing stratified cross-validation we ensure that every fold contains instances with all the classes represented in data. Afterward, folds were given to a model with a set of hyperparameters to perform the sentiment classification task. In our case, we chose the number of folds equal to 5. Generally, this validation scheme is similar to the one represented in Figure 2, except for the method of defining folds.

We have built two uncorrelated algorithms – logistic regression and random forest to classify news' headlines. As logistic regression is a linear classifier and the random forest is a tree-based method, the ensemble of their predictions gives higher precision than both methods separately. This enables us to interpret the news as the probability that the article was positive, negative or neutral. Logistic regression and RF scored 66.1% and 67.24% respectively on the validation set. Also, KNN algorithm was implemented and used as a baseline in order to compare the performance with other algorithms.

The second setup did not use word2vec for news transformation. Instead, pre-trained Bidirectional Encoder Representations from Transformers (BERT) network model, which is a new method of pre-training language representations. It was found to obtain state-of-the-art results on a wide array of Natural Language Processing (NLP) tasks. It is being used to classify news for sentiment analysis and make a prediction only by them for the final classification. Training the model from scratch would result in overfitting as it contains 110 million parameters and our dataset is several times smaller than the number of parameters in the model. As the network was trained on the corpus for the classification task, we needed to fine-tune the weights of the model for our problem by continue training it on the batch of data. An input contained only two columns: headline and Sentiment Class. The results of the classification are shown in Table 6. BERT shows slightly better results than classical machine learning algorithms.

5.3 Final Classification

After all the previous steps of preprocessing and extracting additional features, the classification task is to be accomplished. To make a fair comparison, some of the algorithms were evaluated without a set of features. Note that baseline 0 takes only standard numerical features with scaling; BERT features include both sentiment analysis and final classification features. All the setup explanations are represented in Table 5. The results with different setups are in Table 7.

5.3.1 Validation scheme and evaluation

As we have searched optimal hyperparameters for the models, we needed a cross-validation scheme, which would show a fair estimation of the results on independent data. As, in our case, we had multiple time-series, the following validation scheme was chosen as an estimate of a score.(Figure 2).

First, for a given model random set of hyperparameters was assigned. Then, all the data was split into folds. Each fold consists out of Train and Validation parts. The train was chosen such that it contains data until a certain year, validation contains the rest of the observations. The following decision was made so that we don't predict past, given observations from the future. As all our data was within 10 years (from the start of 2009 till the start of 2019), we obtained the following 8 folds + data from 2018 till 2019 served as a Test set:

$$\begin{aligned}(train, validation) : (2009, 2010), (2009 - 2010, 2011),\\ (2009 - 2011, 2012), \ldots, (2009 - 2016, 2017)\end{aligned} \tag{1}$$

	Features			
Setup	Technical Analysis	News vectors	Sent. Analysis	BERT features
Setup 0				
Setup 1	✓			
Setup 2	✓		✓	
Setup 3	✓	✓	✓	
Setup 4	✓		✓	✓

Table 5: Setup table. Setup 0 contains only market data with scaling; Setup 1 contains both market data and technical analysis features; Setup 2 adds Sentiment analysis features, described in section 4.4.1; Setup 3 contains not only Sentiment analysis features, but also embedding vectors from table 4; Setup 4 contains full set of features, which consists of scaled market data, technical analysis features, sentiment analysis features, and Bidirectional Encoder Representations from Transformers features(predictions of sentiment analysis news polarities and predictions of final classification task by news headlines).

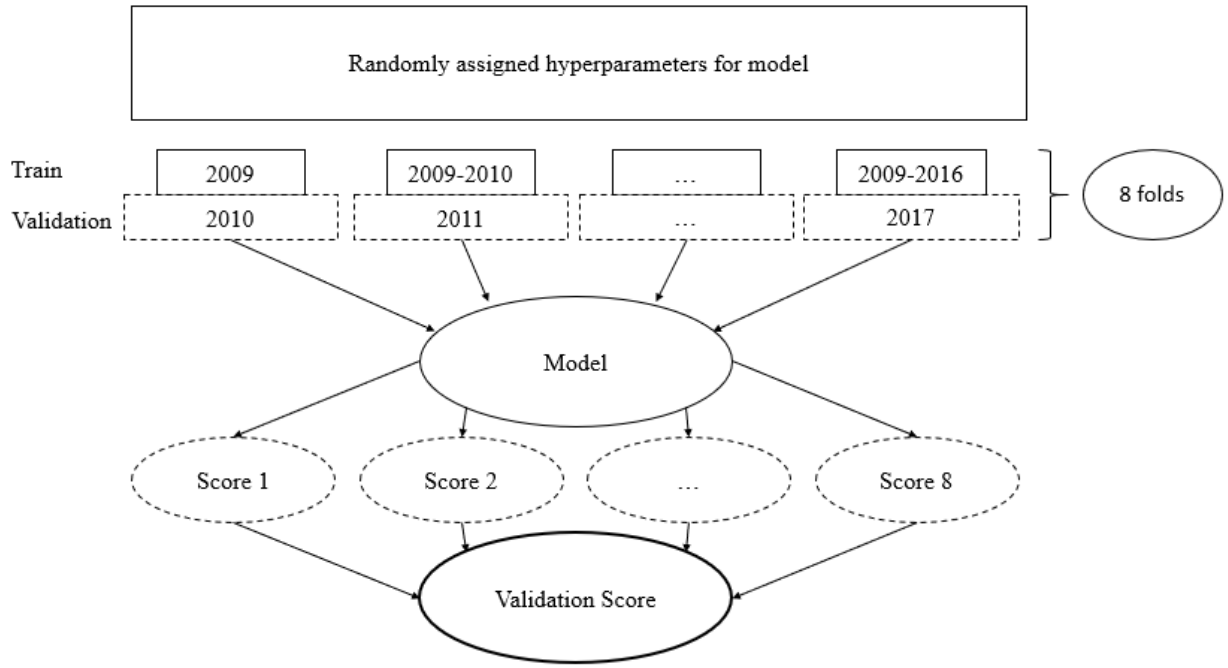

Figure 2: Validation scheme. First, the data split into 8 folds, then each fold sequentially goes as an input to a model with randomly assigned hyperparameters, where it trains on Train part of data and predicts on Validation part; afterward, the mean value of all the scores is computed to generate the final score.

An algorithm takes a set of hyperparameters and trains on Train data, then it generates predictions and score on Validation part. After all scores for 8 folds are computed, final score is calculated as:

$$FinalScore = mean(Score1, \ldots, Score8) \tag{2}$$

The same procedure is repeated N times. In the end, a set of hyperparameters, which corresponds to highest Validation Score, is selected. Then, the model with chosen hyperparameters was trained on data from 2009 till 2017 and tested on data from 2018 till 2019.

6 Visualization of Attention layer

In this chapter visualization of attention layer in transformer was implemented with the aim to understand how neural network handles text classification problem.

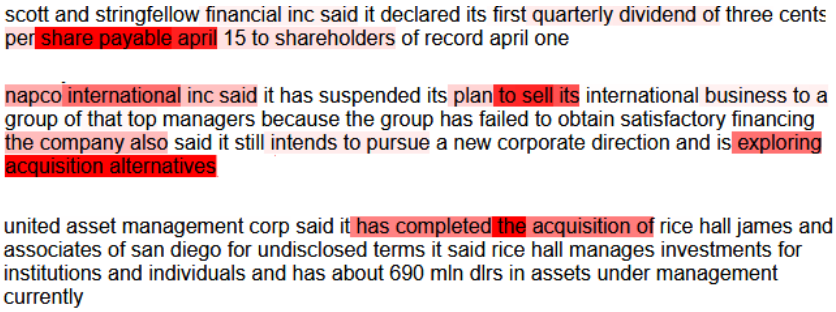

Figure 3: Visualization of attention layers. 3 samples are represented. For each of the samples Bidirectional Encoder Representations from Transformers (BERT) highlighted crucial parts of the sentence, which have the highest importance in text classification.

In Figure 3 we can observe which words have the highest impact on final prediction. In further studies, activated neurons can be pulled out from the Transformer and used to generate additional features for both final classification and sentiment analysis in order to improve the quality.

7 Results

Sentiment analysis classification	
Algorithm	Accuracy
Embedding + KNN	57.94%
Embedding + Logistic Regression	66.1%
Embedding + Random Forest	67.24%
Raw news headlines + BERT	69.54%

Table 6: Sentiment analysis results. Embedding refers to the data from Table 4. Raw news headlines refer to the data from Table 2.

Algorithm results	
Setup and algorithm	Accuracy
Setup 0 + Logistic regression	52%
Setup 1 + Boosting	58.318%
Setup 1 + KNN	54.753%
Setup 2 + Boosting	60.981%
Setup 2 + Logistic Regression	61.15%
Setup 2 + KNN	56.231%
Setup 3 + Boosting	60.213%
Setup 4 + KNN	57.103%
Setup 4 + Logistic Regression	61.743%
Setup 4 + Boosting	63.51%

Table 7: Final classification results. First column represents setup from table 5 and algorithm; Second column shows results of classification task.

The final results show that gradient boosting with transformer neural network outperforms classic algorithms with basic features. As a baseline, logistic regression scores 61% of accuracy on data, while gradient boosting shows 60.981%. Moreover, sentiment analysis plays a significant role in the final prediction. Implementing machine learning sentiment analysis classification features improve final prediction by nearly 3%, comparing the best algorithm, which is Boosting, in setup 1 and setup 2. With the transformer model, it became possible to generate additional sentiment features for final prediction and improve the results. Bidirectional Encoder Representations from Transformers boost final score by 2.36%. We have come to the conclusion that it is not worth to put large vectors of articles to the final prediction as it causes much longer training and worse results due to overfitting. Therefore, vectors were not used as features. As an improvement to the interpretability of neural networks, visualization of Attention layer in Transformer was

implemented. This gives us a possibility to interpret news more clearly and find out which words have a higher weight in text classification.

Acknowledgements. The author wants to thank Alexey Shpilman for his supervision; Nikita Golland and Egor Goriachev for their support, peer-reviewing and overall help with the study.

References

[1] Ryo Akita et al. "Deep learning for stock prediction using numerical and textual information". In: *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE. 2016, pp. 1–6.

[2] Gerald Appel. *Technical analysis: power tools for active investors*. FT Press, 2005.

[3] Piotr Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *arXiv preprint arXiv:1607.04606* (2016).

[4] John Bollinger. *Bollinger on Bollinger bands*. McGraw-Hill New York, 2002.

[5] John Bollinger. "Using bollinger bands". In: *Stocks & Commodities* 10.2 (1992), pp. 47–51.

[6] Kai Chen, Yi Zhou, and Fangyan Dai. "A LSTM-based method for stock returns prediction: A case study of China stock market". In: *2015 IEEE International Conference on Big Data (Big Data)*. IEEE. 2015, pp. 2823–2824.

[7] Raymond Chiong et al. "A sentiment analysis-based machine learning approach for financial market prediction via news disclosures". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM. 2018, pp. 278–279.

[8] Rohit Choudhry and Kumkum Garg. "A hybrid machine learning system for stock market forecasting". In: *World Academy of Science, Engineering and Technology* 39.3 (2008), pp. 315–318.

[9] Stephen V Crowder. "Design of exponentially weighted moving average schemes". In: *Journal of Quality Technology* 21.3 (1989), pp. 155–162.

[10] Xiao Ding et al. "Deep learning for event-driven stock prediction". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

[11] Xiao Ding et al. "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)*. 2014, pp. 1415–1425.

[12] Alexander Elder. *Trading for a living: psychology, trading tactics, money management*. Vol. 31. John Wiley & Sons, 1993.

[13] Nicole Elliott. *Ichimoku Charts: An Introduction to Ichimoku Kinko Clouds*. Harriman House Limited, 2007.

[14] Eugene F Fama. "The behavior of stock-market prices". In: *The journal of Business* 38.1 (1965), pp. 34–105.

[15] Armand Joulin et al. "Bag of Tricks for Efficient Text Classification". In: *arXiv preprint arXiv:1607.01759* (2016).

[16] Joshi Kalyani, Prof Bharathi, Prof Jyothi, et al. "Stock trend prediction using news sentiment analysis". In: *arXiv preprint arXiv:1607.01958* (2016).

[17] Guolin Ke et al. "Lightgbm: A highly efficient gradient boosting decision tree". In: *Advances in Neural Information Processing Systems*. 2017, pp. 3146–3154.

[18] Shimon Kogan et al. "Predicting risk from financial reports with regression". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2009, pp. 272–280.

[19] Mathias Kraus and Stefan Feuerriegel. "Decision support from financial disclosures with deep neural networks and transfer learning". In: *Decision Support Systems* 104 (2017), pp. 38–48.

[20] George C Lane. "Lane's stochastics". In: *Technical Analysis of Stocks and Commodities* 2.3 (1984), p. 80.

[21] Blake LeBaron. "The stability of moving average technical trading rules on the Dow Jones Index". In: *Derivatives Use, Trading and Regulation* 5.4 (2000), pp. 324–338.

[22] Xiaodong Li et al. "News impact on stock price return via sentiment analysis". In: *Knowledge-Based Systems* 69 (2014), pp. 14–23.

[23] Rand Kwong Yew Low and Enoch Tan. "The role of analyst forecasts in the momentum effect". In: *International Review of Financial Analysis* 48 (2016), pp. 67–84.

[24] Manesh Patel. *Trading with Ichimoku clouds: the essential guide to Ichimoku Kinko Hyo technical analysis*. Vol. 473. John Wiley & Sons, 2010.

[25] Yao Qin et al. "A dual-stage attention-based recurrent neural network for time series prediction". In: *arXiv preprint arXiv:1704.02971* (2017).

[26] Reuters. *Reuters.com*. 2019. URL: `https://www.reuters.com/`.

[27] Two sigma. *Two sigma: using news to predict stock movements*.

[28] Stockwatch. *Stockwatch*. 2019. URL: `https://www.stockwatch.com/`.

[29] Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. "More than words: Quantifying language to measure firms' fundamentals". In: *The Journal of Finance* 63.3 (2008), pp. 1437–1467.

[30] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[31] J Wells Wilder Jr. "The relative strength index". In: *J. of Technical Analysis of Stocks and Commodities* 4 (1986), pp. 343–346.

[32] Yahoo. *Finance.Yahoo*. 2019. URL: `https://finance.yahoo.com/`.