



A hybrid approach for stock trend prediction based on tweets embedding and historical prices

Huihui Ni¹ · Shuting Wang² · Peng Cheng¹ 

Received: 4 November 2020 / Revised: 15 March 2021 / Accepted: 29 March 2021 /
Published online: 22 April 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Recently, the development of data mining and natural language processing techniques enable the relationship probe between social media and stock market volatility. The integration of natural language processing, deep learning and the financial field is irresistible. This paper proposes a hybrid approach for stock market prediction based on tweets embedding and historical prices. Different from the traditional text embedding methods, our approach takes the internal semantic features and external structural characteristics of Twitter data into account, such that the generated tweet vectors can contain more effective information. Specifically, we develop a Tweet Node algorithm for describing potential connection in Twitter data through constructing the tweet node network. Further, our model supplements emotional attributes to the Twitter representations, which are input into a deep learning model based on attention mechanism together with historical stock price. In addition, we designed a visual interactive stock prediction tool to display the result of the prediction.

Keywords Tweets embedding · Stock prediction · Deep learning model

1 Introduction

Recently, deep learning and big data researches are booming in various research fields, including image recognition [12], machine translation [3], and text classification [18]. Prediction task is the mainstay of deep learning research, it can be applied to all aspects of our life. For example, deep learning is widely used for automatic driving trajectory prediction [28]. In medical research, deep learning can use the known information to predict the occurrence of diseases such as heart disease to provide diagnostic advice for patients [2].

Huihui Ni and Shuting Wang are joint first author and contribute equally to this work.

✉ Peng Cheng
pcheng@sei.ecnu.edu.cn

¹ School of Software Engineering, East China Normal University, Shanghai, China

² School of Computer and Information Engineering, Zhejiang Gongshang University, Zhejiang, China

In this study, we emphasize the application of deep learning in financial analysis. As a hot research direction in the financial research, researchers scramble for stock movement prediction. Existing studies have demonstrated that it is significant to use public information such as news, social media comments to predict the stock price trend [14].

Various social platforms provide services for users to express their opinions freely. Twitter is a representative one, which has 126 million active users per day from its quarterly report on April 2020. Twitter data contains a lot of valuable information as well as playing a crucial role in enormous amounts of public information. The vital task of using news, Twitter data and other information in stock forecasting is to transform them into text representation with valuable information. For example, the StockNet [38] is proposed to use GRU learning from Twitter data without pre-extracting structured events. Furthermore, Ding et al. (2015) design a deep learning model, use headlines to construct news embedding and apply them to stock prediction [7]. Hu et al. (2018) designed a Hybrid Attention Networks (HAN) to predict the stock trend based on the sequence of recent related news [11]. These works obtain the built-in information detected from news or Twitter data with natural language processing approaches and can generate homologous vectors with similar news or tweets. However, the drawbacks cannot be neglected. Especially for Twitter data, it has unique inter-activities, and thus is different from the independent news data. Users can comment on and reply to the tweets that have been published. This kind of mechanism makes a structural connection to the three types of tweets (i.e., original tweets, comments, and replies), which discriminates the cross-tweet relationship from the semantic knowledge of tweets and reflects the more straightforward relationship of different tweets. Unfortunately, most works are limited to the literal information rather than the linkages crossing tweets. In other words, the interactive tweet relationships cannot be taken into consideration. In this work, we propose a deep learning model to utilize both literal information and connections of tweets to extract useful information for stock market prediction.

DRNews presents the distributed relations of news in the form of network, which enables news articles to embed both the semantic and inter-textual knowledge [20]. Inspired by DRNews, in this paper, we propose a hybrid stock forecasting approach based on tweet embedding and historical stock price and it consists of two parts. Firstly, we design a Tweet Node algorithm that constructs a tweet node network that can embed with the literal message of tweets, and reflect the distributed and structural relations between different tweets. Besides, we utilize the Bert model [6] to add the emotional attribute to the tweet node network. To extract the representation of tweet nodes, we use a variant of Node2vec [8] framework to transform tweets into vectors containing semantic knowledge, different forms of relationships between tweets and emotional bias. Secondly, we present a deep learning framework based on attention mechanism, which uses the results of Tweet Node algorithm and adequately considers the historical stock price. Our framework can learn effective information from the input, and thus predict the stock price trend in the short term.

In conclusion, we make the following contributions.

- In view of the unique attributes of Twitter data, we construct a tweet node network to concatenate each tweet by using textual content and reply relationships. A Tweet Node Model proposed to encode the semantics and reply relationships of tweets into vectors in Section 4.
- We design a deep learning framework on the embedded tweet vectors for short-term stock price trend prediction in Section 5.
- We provide a visual interactive stock prediction tool and apply it to the stock market forecasting scenario, which helps end-users gain intuitions in Section 6. The

experimental results show that our model can obtain the effective information of Twitter data and improve the accuracy of stock price prediction in Section 7.

In order to make the notations in this paper more accurate and easy to understand, Table 1 gives a summary of the major notations for reference, including the notation name, position and definition.

2 Related work

Stock price forecasting is a hot research topic for a long period, and the use of public information to improve the quality of forecasting is also traceable. Previous studies have carried out various extensions and experiments, then proved the feasibility of this idea. For example, Bruce et al. [32] find that the number of public information such as news and tweets is valuable for stock price prediction tasks, they find that the inclusion of predictors based on counts of the number of news and tweets can significantly improve the accuracy of stock price predictions. A novel deep neural network DP-LSTM [15] is proposed for stock price

Table 1 Summary of major notations used

Notation name	Position	Definition
y	Section 3	stock movement
p_d^c	Section 3	adjusted closing price on day d
t	Figure 1	tweets from Twitter
r	Figure 1	edges represent reply relations between tweets
e	Figure 1	elements extracted from tweet texts
k	Figure 1	scores corresponding to elements
$tnidf$	Section 4	the variation of tf-idf
$n_{e,j}$	Section 4	the number of the tweet element e in the tweet text t_j
T	Section 4	the total number of tweets
O	Figure 2	the origin tweet
T	Figure 2	the retweeted tweet
$G = (T, E, W)$	Section 4	a given tweet node network with a tweet node set T , n edge set E and a weight set W
$F(t)$	Section 4	the embedding from the tweet node network for each tweet node $t \in T$
$N(t)$	Section 4	the node t 's network neighborhood
C	Section 4	the sequence generated by random walk
α_{pq}	Section 4	a search bias α with parameter p and q
T	Section 5	an integrated vector composed by the whole tweet vectors
A	Section 5	the weight matrix in attention mechanism
M	Section 5	the final weighting matrix in attention mechanism
f_t	Section 5	the forget gate at t in LSTM
i_t	Section 5	the input gate at t in LSTM
o_t	Section 5	the output gate at t in LSTM
h_t	Section 5	the output at t and the input of next LSTM cell
C_{t-1}	Section 5	the cell state at t

prediction. It incorporates the news articles as hidden information and integrates different news sources through the differential privacy mechanism. Rui et al. integrate sentiment analysis into a machine learning method based on SVM and take the day-of-week effect into consideration for stock trend prediction [26].

The use of public information such as Twitter data for stock price prediction generally involves two necessary steps. One is to convert text into a machine-recognizable language, and the other is to use mathematical statistics or machine learning methods to make predictions.

The representation of tweet texts as vectors actually belongs to the category of word embedding task which is very common in the field of natural language processing. In terms of *Text/Sentence Embedding*, many methods are derived from Word2vec [21], which can be efficiently trained on millions of dictionaries and data sets. The result word embedding can measure the similarity between words. The derived model of word2vec refers to the idea of generating word vectors and learns the distributed sentence level representation regarding the fluency of sentences in texts. There are also some basic models [5, 22] to do numerical operation on the context word vector to generate the sentence vector. SDAE [34] uses an auto-encoder to project text into low-dimensional space. As a substitute for word2vec, BERT [6] uses Transformer [33] as the main framework of the algorithm, which can capture the bidirectional relationship in the statement thoroughly. By running a self-supervised learning model on the basis of massive corpus, BERT can learn predominant representations for words or sentences. The idea of deepwalk [24] is similar to word2vec, which regards an article as a graph containing sentence nodes, and uses the co-occurrence relationship between nodes in the graph to learn the vector representation of nodes. The key problem is how to describe the co-occurrence relationship between nodes. The method given by deepwalk is to use random walk to sample nodes in the graph. DRNews [20] embeddings are trained based on the cross-documental relationship between news stories, which use term frequency-inverse document frequency(tf-idf) to construct text node network.

For the extraction of semantic information in specific data: tweet texts, researchers have also proposed a variety of methods to extract the knowledge and apply it to different fields: A two-stage deep attention neural network(T-DAN) [40] proposed for tweets target-specific stance detection. This model extracts semantic information for stance detection by employing densely connected BI-LSTM and traditional bidirectional LSTM to encode tweet tokens and target tokens respectively. Claire Little et al. [17] presents a new Semantic and Syntactic Similarity Measure (TSSSM) for political tweets. It uses word embeddings to determine semantic similarity and abstracts syntactic features to overcome the limitations of existing measures that may miss identical sequences of words. Nan Xu et al. [37] propose a novel method for modeling cross-modality contrast in the associated context to estimate multimodal sarcastic tweets.

With the development of the field of machine learning, *deep learning frameworks* are widely employed in most stock forecasting tasks recently [14]. Long Short-Term Memory(LSTM) and Convolutional Neural Networks(CNN) are most representative models which has become key technologies in the researches [1, 7, 25]. With the continuous development of deep learning model, attention mechanism has gradually come into our view. Due to the limitation of computing power and optimization algorithm, deep learning models become more complex when it needs to remember a lot of information [4]. At present, the computational ability is still a bottleneck of the neural network development [31]. In addition, although the optimization operations such as weight sharing and pooling can make the neural network simpler [30], which can effectively alleviate the contradiction between model complexity and expression ability, such as the long-distance problem in the cyclic neural network [29], the information memory ability is weak [29]. In this case, the attention

mechanism enables the deep neural network to process information better [4]. Therefore, more researches on stock forecasting introduce this mechanism. For instance, a Hierarchical Attention Networks(HAN) [41] applied on the stock market prediction [11] and Zheng et al. (2019) design a model named Self-attention Networks [42] which employs the idea of Transformer, for the stock volatility forecasting.

3 Problem formulation

In the existing research of stock movement prediction, this task is usually formalized as a binary classification problem [38]. Therefore, our goal is to forecast the binary movement of the target stock s on the target trading day d . We use relevant tweets information and historical price of each stock as the input, and estimate the binary movement y where 1 denotes rise and 0 denotes fall,

$$y = \begin{cases} 0, & p_d^c > p_{d-1}^c \\ 1, & p_d^c \geq p_{d-1}^c \end{cases} \quad (1)$$

where p_d^c represents the adjusted closing price of a given stock on day d . The adjusted closing price is widely used for stock market movement prediction task [27].

4 The tweet node model

In this section, we discuss the formation of Tweet Node Model. With the tweets data as the input, we extracted the key elements for every tweet according to their scores calculated by the modified term frequency-inverse document frequency (tf-idf) algorithm. These elements are then collected together with tweet replies and emotional factors to build tweet node network from which to get the tweet embedding. Figure 1 shows the architecture of tweet node model. We use the improved node2vec [8] to encode the tweet nodes in the tweet node network to generate representations for Twitter data. Node2vec is a graph embedding method that comprehensively considers the DFS and the BFS neighborhood. It can be seen as a deepwalk [24] that combines two walking strategies. In addition, we use BERT [6] to explore the emotional factors in the tweet text to supplement the representation. The network architecture of BERT uses multilayer Transformers [33], which rely on the encoder-decoder structure.

4.1 Tweet node network

A tweet can have multiple parts of speech words including verbs, adjectives, nouns even proper nouns. Actually, the adjectives and nouns contribute most for the content in the classification of tweets [20], thus that we choose them as candidates for keywords. Tweets referring to the same nouns or phrases are often discussing the same thing. In order to build a tweet node network, we use a minor variation of tf-idf algorithm [9] to extract the key elements of tweets and calculate scores for them and establish connections for tweets. Specifically, for any two tweets sharing the same elements, we connect them and assign a weight to the link, which is calculated through our variant tf-idf algorithm. The original tf-idf algorithm considers the frequency of words appearing in the context of the same article and the whole corpus, and prefers to extract the keywords appearing frequently in the current text but less frequently in the whole corpus. However, concerning Twitter data, most of

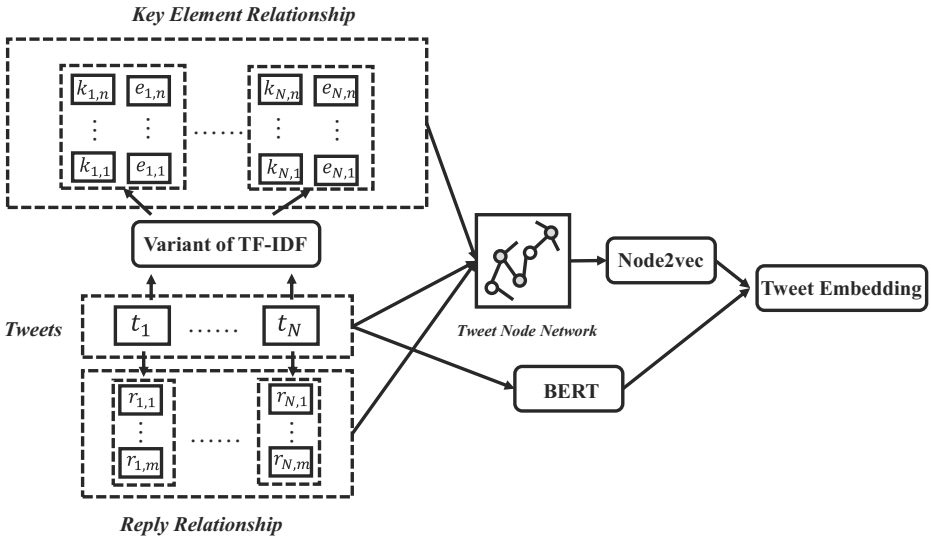


Figure 1 The Tweet Node Model. The model extracts the reply relations r and the key elements e of each tweet, then calculates the weight k of each element in tweets. Then the tweet node network is constructed according to the above three. Finally, the model introduces Node2vec and BERT to extract semantic factors and emotional factors respectively and generates the tweet embeddings

the texts are short and it is hard for the same words to appear in the same tweet. The traditional tf-idf algorithm will affect the score of keywords in different tweets due to the length deviation of the text, which then interferes with the subsequent embedding work. Therefore, we do not consider the length of single tweet text, the variation of tf-idf we called tf-idf is expressed as:

$$tnidf_{e,j} = n_{e,j} \times \log \frac{|T|}{|j : e \in t_j|} \tag{2}$$

where $n_{e,j}$ denotes the number of the tweet element e mentioned in the tweet text t_j . $|T|$ is the total number of tweets whilst the denominator reflects the number of tweets containing element e . Via this way, the elements which could better distinguish the current tweet from others are assigned with higher scores in $[0, 1]$. In addition to the links constructed by key elements, we also supplement the network with tweet responses. Compared with the former links, connection based on reply is the most direct and closest relationship between tweets, so it is rational to assign it a higher weight. Thus, we set the weights of the links between replies and original tweets to 1. According to these rules, we can build the tweet node network. Figure 2 illustrate an example of a tweet node network, which contains three pieces of tweets. Each tweet consists of the user name (shown in the left up corner), the content (shown at bottom) and the type of tweet (shown in the right up corner). Specifically, O indicates an origin tweet while T means a retweeted tweet. With the tags, we can see the relationships between tweets more clearly. For example, two tweets, “Hate how short iPhone chargers are.” and “Apple you make fantastic laptops but how bout you make chargers that can last more than 3 months”, are connected by sharing the same element *chargers*. In addition, keywords such as *iPhone* may be also used as a medium to connect two tweets. In this example, only the connections between the three tweets are displayed. Meanwhile,

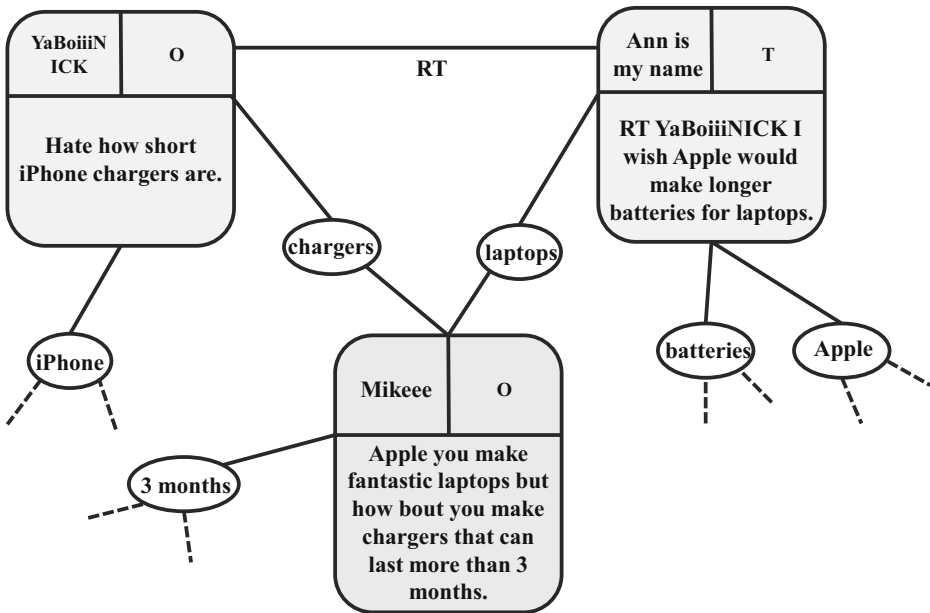


Figure 2 A sample of tweet node network

the link marked with *RT* indicates a reply relationship between two tweets. The remaining tweet elements are linked in the same way to other tweets that are not listed.

4.2 Tweets embedding

4.2.1 Tweets embedding from tweet node network

Let $G = (T, E)$ be a given tweet node network with a tweet node set T and an edge set E . In order to obtain the embedding $F(t)$ from the tweet node network for each tweet node $t \in T$, we adopt a model whose idea is similar to Node2vec [8]. Following the algorithm, the possibility of predicting the node t 's network neighborhood $N(t)$ needs to be maximized, the objective function is written as:

$$\max \sum_{t \in T} \log Pr(N(t) | F(t)) \tag{3}$$

In order to ease the calculation of the embedding, two assumptions are introduced, which are from the skip-gram model [21]. The first is *conditional independence* whose main idea is each neighbor in the sample is independent of each other. Therefore, we can multiply the probability of each neighbor sampled to get the result of all sampled neighbors:

$$Pr(N(t) | F(t)) = \prod_{u \in N(t)} Pr(u | F(t)) \tag{4}$$

The second assumption is that the *symmetry in feature space* is comprehensible. For example, if an edge concatenates two nodes, then these two nodes should have the same influence

on each other when they are mapped to the feature space. A pair of neighbor nodes is represented by a model:

$$Pr(u | F(t)) = \frac{\exp(F(u) \cdot F(t))}{\sum_{v \in T} \exp(F(v) \cdot F(t))} \tag{5}$$

With above two assumptions, the objective in (1) simplifies to:

$$\max \sum_{t \in T} \left[-\log Z_t + \sum_{u \in N(t)} F(u) \cdot F(t) \right] \tag{6}$$

where $Z_t = \sum_{v \in T} \exp(F(v) \cdot F(t))$ which is hard to compute for huge networks so finally negative sampling [13] is used to approximate it.

Consider an edge (t, v) of the sequence C generated by random walk and now reside at node v , then the next step x should be determined. Node2vec sets a search bias α , which makes it adopts different walk strategies when choosing x :

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases} \tag{7}$$

where d_{tx} denotes the distance between the previous node and the next node of the current node. Finally the transition probability of v and x can be defined as:

$$P(C_i = x | C_{i-1} = v) = \frac{\alpha_{pq}(t, x) \cdot \omega_{vx}}{Z} \tag{8}$$

where Z is the normalizing constant. Based on skip-gram model and a group of sequences generated by node2vec, we get the node representations by optimizing the objective function (2) with Stochastic Gradient Descent (SGD).

When applying node2vec in tweet node network (Figure 3), there are two kinds of nodes in the network: tweet nodes and element nodes. Suppose a random walk starting from t_0 resides at tweet node t_1 and the previous node is element node e_0 in order. It can be seen from Figure 3 that the type of the next node may be tweet like t_2 or element like e_1 so that the sequence generated by migration could be $C = \langle t_0, e_0, t_1, t_2, \dots \rangle$ or $C = \langle t_0, e_0, t_1, e_1, \dots \rangle$ or $C = \langle t_0, e_0, t_1, e_2, \dots \rangle$. In view of the fact that what we want to get is the representation

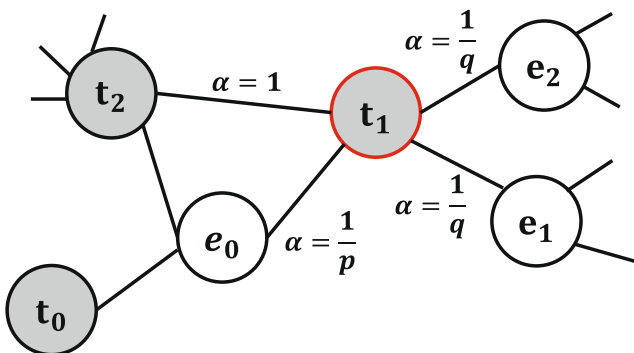


Figure 3 Illustration of the random walk for tweet node network in node2vec

vectors of the tweet nodes, we need to eliminate the unnecessary nodes before entering the final embedding work, the pseudo-code of using node2vec in tweet node network is given in Algorithm 1.

Algorithm 1 Node2vec in tweet node network.

Input: Network $G = (T, E, W)$, Start node u , Length l
Output: tweet node sequences $walk$
 Initialize $walk$ to $[u]$, $temp$ to $[u]$;
for $walk_iter = 1$ to l **do**
 $curr = temp[-1]$;
 $V_{curr} = \text{GetNeighbors}(curr, G)$;
 $s = \text{AliasSample}(V_{curr}, W)$;
 Append s to $temp$;
 if s is a tweet node **then**
 Append s to $walk$;
 end if
end for
return $walk$

There are a variety of ways to generate representations for nodes in graphs, some common graph embedding methods are deepwalk [24], node2vec [8] and SDNE [35]. We choose node2vec to map the tweet nodes into low-dimensional space instead of deepwalk because both of them belong to random walk strategy, but the former is an improvement of the latter. The two parameters p and Q introduced by node2vec can control the discovery of the micro view and larger neighborhood respectively, and the complex dependence relationship between communities can be inferred. SDNE is a structured deep network embedding method that has multi-layer nonlinear functions, and it can learn first-order approximation and second-order proximity to capture highly nonlinear network structure. However, there are element nodes and tweet nodes in our tweet node network. We expect to treat nodes with disparate attributes differently in the process of generating node embedding, so it is not an appropriate strategy to use SDNE directly.

4.2.2 Tweets embedding with emotional attributes

The construction of tweet node network enables extracting valuable information between tweets. However, the emotional leanings of tweet texts should not be neglected. Existing studies have shown that the analysis of emotional factors in news or social comments can contribute a lot to the task of short-term stock price forecasting. For example, Thien et al. propose the topic model Topic Sentiment Latent Dirichlet Allocation (TSLDA) [23], which can capture the topic and sentiment simultaneously to predict stock market movement; Xiao et al. [19] construct recurrent state transition model which can better capture a gradual process of stock movement continuously by modeling the correlation between past and future price movements and then use it to predict stock market trend.

BERT [6] is a mechanism based on attention model and obtaining state-of-the-art results on some natural language processing tasks. The pre-trained BERT model is fine-tuned for specific tasks such as text classification to create the most advanced model for particular scenario. It uses Transformers [33] as the main framework which can more thoroughly capture the bidirectional relationships in the statement and adopt encoder-decoder architecture

like traditional attention model. Specifically, the input of BERT is the normalized sum of following three embedding features:

- WordPiece embeddings [36]. It divides the words into a set of limited common sub-word units, which can achieve a compromise between the validity of word and the flexibility of character.
- Position embeddings. It encodes the position information of words into feature vectors so that the model can distinguish words in different positions.
- Segment embeddings. It is a kind of embedding learned to every token indicating the sentences they belong to.

BERT alleviates the unidirectional constraints by using a pre-training goal of a “masking language model” (MLM). The purpose of the masking language model is to predict the original lexical ID of the masked word according to the context. Furthermore, BERT uses a “next sentence prediction” task to jointly pre-training text pair representations. Transformers in BERT learn a weight for each word of the input vector. Finally, BERT is applied to different natural language tasks with fine-tuning.

Sentiment analysis is included in the fine-tuning task of BERT, so we can use it to explore the emotional factors contained in the tweet texts. The tweet embeddings from tweet node network and emotion elements extracted by BERT are feed into deep learning framework to predict stock market movement.

5 The deep prediction model

In this section, we propose a deep prediction model based on attention mechanism to predict stock trend with price information and the features extracted from tweets. Since both market indicators and social comments in Twitter can be regarded as time-series data, we choose Long Short Term Memory network (LSTM) [10] as the basic framework of the whole deep prediction model. As a variant of recurrent neural network, LSTM can not only keep the memory of past information, but also solve the problem of long-term dependence on traditional neural network. In order to solve the problem that LSTM is difficult to obtain a reasonable vector representation when the input sequence is long, we apply the attention mechanism [16]. Specifically, attention is a model that imitates human attention and can quickly screen high-value message from lots of information. We have reached a consensus that each tweet has a different correlation with a company or the stock market. It is unreasonable to treat each tweet with the same importance. Therefore, we use an effective attention mechanism [16] to assign weights to the data such that the model can treat different tweets adaptively.

According to the above ideas, our deep prediction model based on LSTM and attention mechanism is illustrated in Figure 4. The input of the model can be divided into two parts: tweet-driven section and market-driven section. The flow of the model is to feed tweet embeddings into the attention mechanism and LSTM, and then concatenate with the normalized stock market indicator, finally reach the output layer.

Thanks to that the stock price has the characteristics of continuous fluctuation, financial experts believe that the rise and fall of stock price are closely related to the historical stock price, especially the extreme point [38]. It is not enough to consider the historical stock price of one day only. Therefore, with regard to the market-driven module, the historical stock price information will be reorganized in the form of time windows, and finally combined with the output of tweet-driven module through normalization.

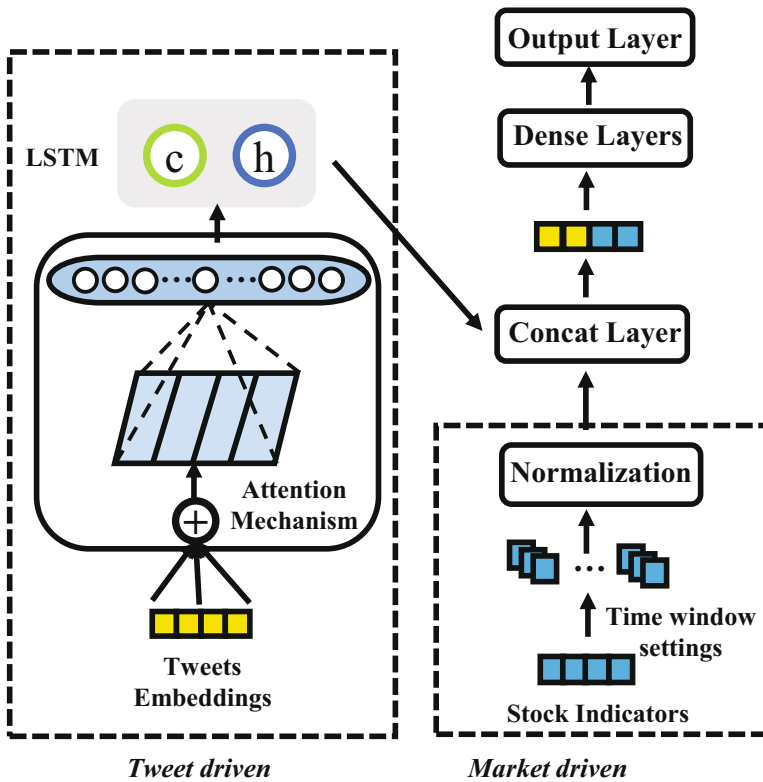


Figure 4 The deep prediction model

This self attention mechanism was first used in sentence embedding. It considers the words and phrases in a single sentence and gives different attention weights to generate the final vector. This paper treats the many single tweets of the day as words, which compose the “sentence” that represents the Twitter situation of a certain stock in one day. Suppose the size of each tweet vector is u , and n is the number of tweets related to a stock on a day. The tweet vectors $t_{s,i}$ of the stock s on the same day are composed of an initial vector T , which has the size n -by- u representing the stock’s situation in Twitter on that day.

$$T = (t_{s,1}, t_{s,2}, \dots, t_{s,n}) \tag{9}$$

The initial vector T first fed into the self-attention layer and then the weight matrix A is generated to transfer the initial vector to M .

$$A = softmax(W_{s2} \tanh(W_{s1}T)), \tag{10}$$

$$M = AT, \tag{11}$$

where W_{s1} is a weight matrix with a shape d_a -by- u , W_{s2} is a matrix extended from a vector of parameters with size d_a and the shape is r -by- d_a . Where d_a is a hyper-parameter that can be set arbitrarily and r is the number of different components extracted from the original vector T such that the mechanism can focus on various parts. The final weighting matrix M can be obtained by multiplying the weight matrix A and the initial tweet vector T .

The main idea of LSTM is to use three gates named input gate i , forgetting gate f and output gate o to determine the information to be retained and discarded and to output the final result. The detailed formula is shown below:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (12)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (13)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (14)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (15)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (16)$$

$$h_t = o_t * \tanh(C_t) \quad (17)$$

where h_t is both the output at the step t and the input of next LSTM cell. Another input is x_t in the above formulas, which is the output of attention layers. Compared with the traditional recurrent neural network, LSTM needs to remember not only the hidden state h_{t-1} and cell state C_{t-1} of the previous unit. The pseudo-code of training model is given in Algorithm 2.

Algorithm 2 Deep prediction model training process.

Input: Training dataset: tweet embeddings E and stock indicators S with n samples

Output: Stock prediction model

Initialize model parameters;

for $i_iter = 1$ to n **do**

 Put E into the attention-based LSTM layer;

 Reorganize S according to w and normalize them;

 Concatenate the output of LSTM layer and normalized S ;

 Put the output of the previous step into the dense layer;

 Update model parameters

end for

return trained model

6 Visualization

We design the visual interface according to two principles as following:

- First of all, the content on the interface can help users to understand and analyze the actual basic characteristics of the stock.
- Over the information of stock, users could obtain reference opinions from interface whose details produced by our model.

Specially, the visual interface integrates the stock forecast model proposed in this paper (Figure 5). Users can browse the actual price of each stock on this interface, click the button to run the model for stock price forecast and check the deviation between the result of prediction and the actual trend.

The sidebar on the left shows the available stock names for switching. After selecting a specific stock, the actual data will be displayed in Figure 5a, including opening price, closing price, high and low price. In addition, the overall trend will be displayed in the form of candle chart where red and green represent uptrend and downtrend respectively whilst

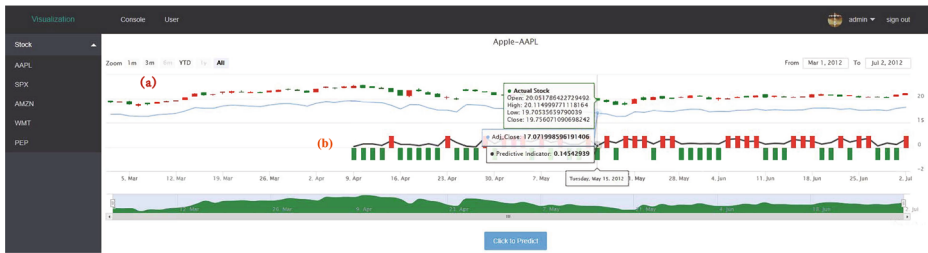


Figure 5 The interface of stock prediction tool: (a) Stock timeline view showing the stock price history with different indicators; (b) Display of trend results with color representation

the adjusted closing price in the style of line chart. Figure 5b shows the area where the prediction results are displayed. Users can click the button to use the model proposed by this paper to predict the trend of the stock market. The model embedded in the interface can give results quickly. The results are displayed in a colored column chart where red represents the upward trend, while green means the stock is falling. There is also a time selection bar at the top of the display area to adjust the time span. Users can view the actual situation and forecast results of the stock on demand.

7 Experiments

In this section, we examine the practicability of our model with real-world tweets data and stock market price data. We confirm the effectiveness of our model and whether the vectors generated by tweet node model with real data enhance the accuracy of stock price forecasting tasks. At last, we summarize the results of market simulation of our approach.

7.1 Data collection

The rise and fall of stocks are according to the trading situation of the day, so most of the previous research work is to regard the stock trend prediction as a binary classification problem, and strive to get more accurate prediction results. We believe that social comments are valuable for stock forecasting tasks, which means that the premise of using this type of data is that a certain stock can have a high degree of discussion and activity. We choose popular stocks or index that are widely mentioned on Twitter as experimental data including \$AAPL (Apple Inc.), \$WMT (Walmart Inc.), \$SPX (S&P 500 Index), \$AMZN (Amazon.com Inc.) and \$PEP (PepsiCo Inc.). The form of data from Twitter is shown in Table 2. In addition to the basic tweet content and posted time, the ID identifies each tweet, the attributes include the sign of retweeted tweet, and the original tweet ID if the sign is “Y”. In accordance with our algorithm, this kind of label plays an important role in constructing the reply relationship edge in the tweet node network. The time span of the data adopted is 01/03/2012 to 01/07/2012, and the amounts of tweets used per day are shown in Table 3. We can see that the average amount of tweet data related to all stocks exceed 1000 such that it is forceful that the experimental data can represent the situation of public opinion of the stock on that day.

For the historical price data of the stock market, we choose the opening & closing & highest & lowest price from Yahoo Finance [39].

Table 2 Two samples of tweet data

Time	Tweet ID	Content	Reply State	Reply Tweet ID
Thu Mar 01	17500927 2244092	I don't know why but the pepsi from this Fosters Freeze taste AMAZING.	N	None
Thu Mar 01	175060627 52036	RT free man You need some diet cherry pepsi.	Y	17504808 666884

7.2 Training setup

As for the tweet embedding, we set the length of walk as 80 and the number of times is 10. The dimension of vectors generated by tweet node network is 128.

We use a 3-day lag window for stock price sample construction and set the LSTM units in a 128 size. The activation functions used in the model are Rectified Linear Unit (ReLU) and Sigmoid. The segmentation ratio of training set and test set is 1 : 1. We train the model with an Adam optimizer with the initial learning rate of 0.001. The stock price movement classification accuracy (ACC) is adopted as the performance metric: $ACC = \frac{\#correct\ predictions}{\#total\ predictions}$.

7.3 Baselines and proposed models

In order to verify the validity of our model, we carry out comparative experiments with our approach, baselines and the variant of our model. Note that to ensure the authenticity of comparative, the performance of all model are measured with the same input data of daily tweets. For the sake of detecting the superiority of our embedding method, we consider two very popular embedding models for comparison: average bag-of-words [21, 22] and doc2vec [13]. The former actually is a kind of algorithm of word2vec, which can measure the similarity of words when the doc2vec is a derivation of word2vec for document representation. As for deep learning model, we adopted LSTM and CNN to cooperate with different embedding methods as the prediction model. In addition, in order to explore the performance gap between the proposed model and the widely used stock price trend prediction model, we also introduce a popular model StockNet [38].

Table 3 Tweets amounts of stocks

Stock	Daily Average Tweet	Total Tweets
\$AAPL	30172	2564653
\$WMT	3643	309681
\$SPX	4702	399715
\$AMZN	4394	373490
\$PEP	1529	130012

We construct the following five baselines in different genres.

- **Average BOW-LSTM**: Tweets are embed into vectors which are the average of words embeddings. Then LSTM is used for prediction task with the input of tweets embeddings and historical price.
- **Average BOW-CNN**: The same tweets embedding method as the previous baseline is adopted, but the vectors are input into CNN.
- **Doc2vec-LSTM**: The content of tweet is represented as dense vector by doc2vec algorithm then fed into LSTM with historical price.
- **Doc2vec-CNN**: Replace LSTM of the previous baseline with CNN.
- **StockNet**: A state-of-the-art variational Autoencoder (VAE) that uses price and text information and could directly learns from data without pre-extracting structured events [38].

To ensure the effect of our approach and primary components, We also construct the following model for experiment,

- **TE+**: Fully-equipped model proposed in this paper which adopts tweets node model.
- **TE-**: The generative model without emotional attributes generated by BERT.

7.4 Results

In this section we analyze the experimental results which include whether the sentimental factors in Twitter data are related to stock price fluctuations and we prove that sentiment analysis of tweets is beneficial to stock price prediction. What's more important is that we compared the traditional baseline model with the approach we proposed, gave the experimental results and analyze the information contained. Through this experiment, we demonstrate that the proposed method can improve the quality of stock price trend prediction to a certain extent.

7.4.1 Stock price movements and tweets sentiment

It is our consensus that the emotional tendency contained in public information can reflect or affect the trend of stock prices. As for enriching the tweets embedding, BERT is used in the Twitter node model to replenish emotional elements. Through BERT, we can categorize tweets as positive or negative according to sentiment, and calculate the proportion of positive tweets every day. In order to exhume the relationship between the overall sentiment of daily tweets and the actual trend of stock prices, Figure 6 shows the proportion of daily positive sentiment and stock trading volume of \$PEP from 01/03/2012 to 30/07/2012.

It can be seen from the Figure 6 that the proportional trend of positive tweets is similar to the trend of stock trading volume at certain moments. For example, the transaction volume from 01/03/2012 to 09/03/2012 turned from a decline to an increase, and the proportion of positive tweets in Twitter also reveals the same trend. A similar phenomenon also appeared in several time periods such as 24/04/2012 to 28/04/2012.

At the same time, it is obvious that the sentiment of public information cannot fully reflect or affect stock price changes. We should take this emotional factor into consideration, but it is unscientific to rely solely on it for stock price prediction. This also proves that we need to introduce more elements explored when using tweets to do stock price forecasting.

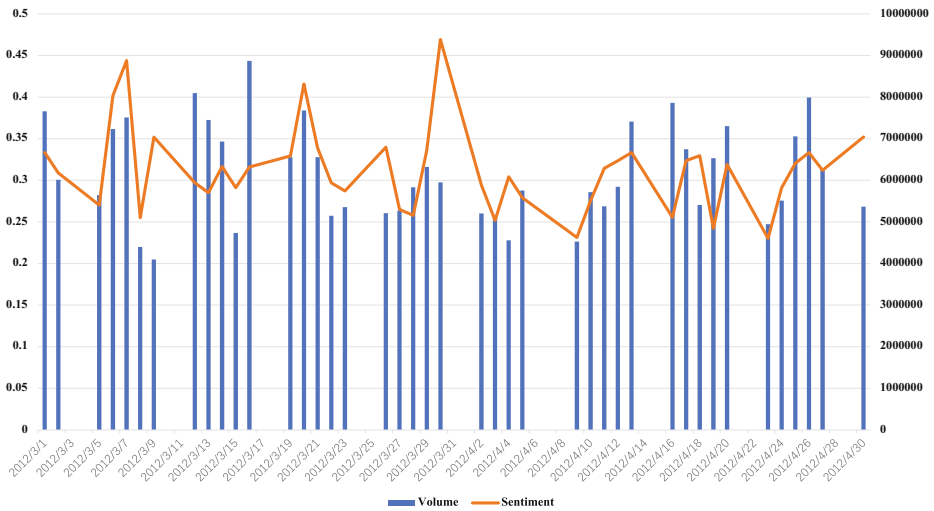


Figure 6 Percentage of daily positive tweets and stock trading volume about \$PEP

7.4.2 Stock prediction

Since stock movement is a tough task with high volatility caused by many factors, previous studies have shown that accuracy of 56% is already a satisfying result for binary stock movement prediction [23, 38]. Table 4 shows the accuracy of different models for stock prediction, which describe the performance of our approach in comparison to other current embedding models.

According to the experimental results, We come to the following conclusions.

First of all, from the experimental results of two embedding models: average bag-of-words and doc2vec which are generally used, the vast majority of stocks can achieve more than 50% accuracy, which means that the combination of tweets text information and historical stock price impact on stock forecasting task positively. Comparing the first two rows in the table with the third row and the fourth row, we can see that the average bag-of-words embedding model can contribute a better benchmark. We speculate that it is better than

Table 4 Performance of baselines and our approach in accuracy. The best score is in bold

Model	AAPL	SPX	PEP	AMZN	WMT	Average
Average BOW-LSTM	53.49	60.47	58.14	53.49	46.51	54.42
Average BOW-CNN	46.51	51.16	62.79	58.14	55.81	54.88
Doc2vec-LSTM	53.49	53.49	53.49	55.81	51.16	53.49
Doc2vec-CNN	53.49	51.16	51.16	53.49	58.14	53.49
StockNet	55.81	55.81	60.46	51.16	55.81	55.81
TE-	60.46	65.12	60.46	65.12	55.81	61.39
TE+	62.79	67.44	67.44	65.12	58.14	64.19

doc2vec because the latter is to use the semantic relationship between sentences to encode, while Twitter data is not like an article that can form a contextual whole, so the average bag-of-words model can better indicate the meaning of each tweet.

Secondly, by comparing the results of our models and baselines, we can know that our model outperforms or in accordance with baselines in each stock, which proves the superiority of our model's performance. Specifically, our fully-equipped model improves by 9.5%, 6.97%, 4.65%, 6.98% in comparison to the best result of baseline on \$AAPL, \$SPX, \$PEP and \$AMZN respectively. In addition, the accuracy of our model is consistent with that of Doc2vec-CNN on \$WMT.

It is worth mentioning that by comparing the TE- and baselines based on LSTM in Table 4, we can see that the former is better than the latter in most cases. It demonstrates that the embeddings from tweet node network can achieve better results with the cooperation of LSTM based on attention mechanism. Our embedding approach is sufficient to strengthen the tweet vector representation with additional message detected from potential connections between tweets. The first four baselines extract semantic information from the text of Twitter data, without considering the extra relationship between tweets produced by user interactions. This experimental result also shows that the interaction between users will produce knowledge that can not be obtained by analyzing text semantics only. Therefore, our embedding method considers different sources of knowledge and can cover more market information.

We can see that StockNet can achieve an average accuracy of 55.81% over all the four baselines based on CNN and LSTM, which proves that StockNet is an advanced technique in the field of stock prediction. However, it is obvious that our model TE+ and TE- outperform this strongest baseline StockNet in all the target stocks. We think that although StockNet uses the market encoder to extract the relevant semantic information from the tweet texts, it does not take into account the unique interactivity of social platform data like tweet data. And the interaction information between two tweets can supplement the message that can not be obtained from a single tweet.

At last, the emotional features brought from BERT can supplement the embeddings from tweet node network, as the ACC of TE+ higher than that of TE-. The accuracy of the experimental results of the fully-equipped model on all stocks is higher than 58%. In the previous section, it was mentioned that the emotional information contained in the tweets can help predict the trend of stock prices to a certain extent, which has also been verified here. To sum up, the tweet embedding method proposed in this paper can capture the valuable information that is conducive to stock price prediction than the current popular text embedding method. The combination of attention mechanism and LSTM can also make good use of tweet embeddings.

8 Conclusions

We design a hybrid approach for stock movement prediction with tweets and historical stock price. Especially for the unique interactive characteristics of Twitter data, instead of using the current common embedding methods based on context, we propose a novel method which can not only extract the semantic information of text, but also pay attention to the relationship of different tweets. This method builds a "social network" for tweet nodes instead of taking each tweet as an isolated node. Therefore tweets with the same semantic elements and interactive relationships are encoded into similar vectors. In addition to the embedding obtained by tweet node network, we also use BERT, which has excellent performance on

natural language processing tasks, to extract emotional factors in the Twitter data to obtain a final representation that contains more effective information.

The attention-based LSTM network is used to predict the trend of short-term stock movement with tweet embeddings and historical price. Further, in the experimental part, we proved that the public sentiment tendency contained in the Twitter data is related to the stock price trend, so it is well-founded to design text representations that can express emotional factors. And our experiments involving a variant of our model and four baselines based on various embedding methods and deep neural network shows that the proposed approach achieves eminent performance in stock trend prediction.

In addition, a visual interface is developed for users to employ our model, and the prediction results displayed can be compared with the actual value for stock analysis.

Acknowledgements This work is partially supported by GuangDong Basic and Applied Basic Research Foundation 2019B1515120048.

References

1. Akita, R., Yoshihara, A., Matsubara, T., Uehara, K.: Deep learning for stock prediction using numerical and textual information. In: 15th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2016, Okayama, Japan, June 26–29, 2016, pp. 1–6. IEEE Computer Society (2016)
2. Ali, S.A., Raza, B., Malik, A.K., Shahid, A.R., Faheem, M., Alquhayz, H., Kumar, Y.J.: An optimally configured and improved deep belief network (OCI-DBN) approach for heart disease prediction based on ruzzo-tompa and stacked genetic algorithm. *IEEE Access* **8**, 65947–65958 (2020)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
5. Conneau, A., Kruszewski, G., Lample, G., Barrault, L., Baroni, M.: What you can cram into a single vector: Probing sentence embeddings for linguistic properties. arXiv:1805.01070 (2018)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics (2019)
7. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: Yang, Q., Wooldridge, M.J. (eds.) Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25–31, 2015, pp. 2327–2333. AAAI Press (2015)
8. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R. (eds.) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13–17, 2016, pp. 855–864. ACM (2016)
9. Hiemstra, D.: A probabilistic justification for using tf x idf term weighting in information retrieval. *Int. J. Digit. Libr.* **3**(2), 131–139 (2000)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
11. Hu, Z., Liu, W., Bian, J., Liu, X., Liu, T.: Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In: Chang, Y., Zhai, C., Liu, Y., Maarek, Y. (eds.) Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5–9, 2018, pp. 261–269. ACM (2018)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)

13. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21–26 June 2014, JMLR Workshop and Conference Proceedings, vol. 32, pp. 1188–1196. JMLR.org (2014)
14. Li, Q., Chen, Y., Wang, J., Chen, Y., Chen, H.: Web media and stock markets : A survey and future directions from a big data perspective. *IEEE Trans. Knowl. Data Eng.* **30**(2), 381–399 (2018)
15. Li, X., Li, Y., Yang, H., Yang, L., Liu, X.: DP-LSTM: differential privacy-inspired LSTM for stock prediction using financial news. arXiv:[1912.10806](https://arxiv.org/abs/1912.10806) (2019)
16. Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings. OpenReview.net (2017)
17. Little, C., Mclean, D., Crockett, K.A., Edmonds, B.: A semantic and syntactic similarity measure for political tweets. *IEEE Access* **8**, 154095–154113 (2020)
18. Liu, P., Qiu, X., Huang, X.: Adversarial multi-task learning for text classification. In: Barzilay, R., Kan, M. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, pp. 1–10. Association for Computational Linguistics (2017)
19. Liu, X., Huang, H., Zhang, Y., Yuan, C.: News-driven stock prediction with attention-based noisy recurrent state transition. arXiv:[2004.01878](https://arxiv.org/abs/2004.01878) (2020)
20. Ma, Y., Zong, L., Wang, P.: A novel distributed representation of news (drnews) for stock market predictions. arXiv:[2005.11706](https://arxiv.org/abs/2005.11706) (2020)
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings (2013)
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States, pp. 3111–3119 (2013)
23. Nguyen, T.H., Shirai, K.: 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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26–31, 2015, Beijing, China, Volume 1: Long Papers, pp. 1354–1364. The Association for Computer Linguistics (2015)
24. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Macskassy, S.A., Perlich, C., Leskovec, J., Wang, W., Ghani, R. (eds.) The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, pp. 701–710. ACM (2014)
25. Rather, A.M., Agarwal, A., Sastry, V.N.: Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Syst. Appl.* **42**(6), 3234–3241 (2015)
26. Ren, R., Wu, D.D., Liu, T.: Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Syst. J.* **13**(1), 760–770 (2019)
27. Sawhney, R., Agarwal, S., Wadhwa, A., Shah, R.R.: Deep attentive learning for stock movement prediction from social media text and company correlations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020 , pp. 8415–8426 (2020)
28. Scheel, O.: Using deep neural networks for scene understanding and behaviour prediction in autonomous driving. Ph.D. thesis, Technical University of Munich, Germany (2020)
29. Staudemeyer, R.C., Morris, E.R.: Understanding LSTM - a tutorial into long short-term memory recurrent neural networks. arXiv:[1909.09586](https://arxiv.org/abs/1909.09586) (2019)
30. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv:[1602.07261](https://arxiv.org/abs/1602.07261) (2016)
31. Thompson, N.C., Greenewald, K., Lee, K., Manso, G.F.: The computational limits of deep learning. arXiv:[2007.05558](https://arxiv.org/abs/2007.05558) (2020)
32. Vanstone, B.J., Gepp, A., Harris, G.: Do news and sentiment play a role in stock price prediction? *Appl. Intell.* **49**(11), 3815–3820 (2019)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, pp 5998–6008 (2017)

34. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**, 3371–3408 (2010)
35. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R. (eds.) *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, August 13–17, 2016, pp. 1225–1234. ACM (2016)
36. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J.: Google’s neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144 (2016)
37. Xu, N., Zeng, Z., Mao, W.: Reasoning with multimodal sarcastic tweets via modeling cross-modality contrast and semantic association. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pp. 3777–3786. Association for Computational Linguistics (2020)
38. Xu, Y., Cohen, S.B.: Stock movement prediction from tweets and historical prices. In: Gurevych, I., Miyao, Y. (eds.) *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers*, pp. 1970–1979. Association for Computational Linguistics (2018)
39. Yahoo finance. <https://finance.yahoo.com/> (2012)
40. Yang, Y., Wu, B., Zhao, K., Guo, W.: Tweet stance detection: A two-stage DC-BILSTM model based on semantic attention. In: *5th IEEE International Conference on Data Science in Cyberspace, DSC 2020, Hong Kong, July 27–30, 2020*, pp. 22–29. IEEE (2020)
41. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: Knight, K., Nenkova, A., Rambow, O. (eds.) *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*, pp. 1480–1489. The Association for Computational Linguistics (2016)
42. Zheng, J., Xia, A., Shao, L., Wan, T., Qin, Z.: Stock volatility prediction based on self-attention networks with social information. In: *IEEE Conference on Computational Intelligence for Financial Engineering & Economics, CIFE 2019, Shenzhen, China, May 4–5, 2019*, pp. 1–7. IEEE (2019)

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.