Deep learning for stock market prediction from financial news articles

Manuel R. Vargas, Beatriz S. L. P. de Lima and Alexandre G. Evsukoff
COPPE/Universidade Federal do Rio de Janeiro,
P.O. Box 68506, 21941-972, Rio de Janeiro, RJ, Brazil

Abstract—**This work uses deep learning methods for intraday directional movements prediction of Standard & Poor's 500 index using financial news titles and a set of technical indicators as input. Deep learning methods can detect and analyze complex patterns and interactions in the data automatically allowing speed up the trading process. This paper focus on architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), which have had good results in traditional NLP tasks. Results has shown that CNN can be better than RNN on catching semantic from texts and RNN is better on catching the context information and modeling complex temporal characteristics for stock market forecasting. The proposed method shows some improvement when compared with similar previous studies.**

Keywords: Deep learning; Recurrent neural network; Convolutional neural network; intraday stock forecasting

I. Introduction

The aspiration of any investor is to forecast the market behavior with the aim of making the best decision when he comes to buying or selling shares of stocks seeking to maximize his profits. This is a difficult task because market behavior is volatile and influenced by many factors such as global economy, politics, investor expectation and others.

The random walk theory [1] introduces a hypothesis that stocks prices are defined randomly and for these reasons they are impossible to forecast. However, advances in artificial intelligence and the growth of available data have made possible to forecast the stock price behavior with a better performance than a random process [2]-[8].

There are three approaches related to the information required to make a prediction. The first approach, technical analysis, is based on the premise that the future behavior of a financial time series is conditioned to its own past. The second approach, fundamental analysis, is based on external information as political and economic factors. This information is taken from unstructured data as news articles, financial reports or even publishing in microblogs by analysts. Nofsinger [9] shows that in some cases, investors tend to buy after positive news resulting in a stress of buying and higher stocks prices; and after negative news, they sell, resulting in a decrease of prices. Finally the third approach considers as relevant all information coming from both, financial time series and textual data.

Prior works in this area focus on technical analysis. These works use different statistical techniques and artificial intelligence models to make a prediction based only in technical information [10][11]. This approach has a limitation since the market reacts to external information that is not contained in the historical data used to extract the technical information.

Inspired by fundamental analysis, many authors propose the use of text mining techniques and machine learning techniques to analyze textual data and take out information that can be relevant to the forecast process [12]-[15]. The most relevant works in the area are reviewed in [16][17]. Other works [3][18] use hybrid models by combining text mining techniques with the technical information. This approach outperforms other baseline strategies.

Recently, with more computational capabilities and the availability to handle massive databases, it is possible to use more complex machine learning models, such as deep learning models, which presents a superior performance in traditional Natural Language Processing (NLP) tasks. The outstanding deep learning models are: Convolutional Neural Network (CNN) [19]-[22], Recurrent Neural Network [23][24], specifically the Long Short Term Memory architecture (LSTM) [25][26], and Recurrent Convolutional Neural Network (RCNN) [27][28].

Some examples of deep learning models for financial time series forecasting are shown in [29][30]. Those authors apply a deep neural network model that use as input events taken from financial news articles to forecast the direction of prices of a set of stocks and the S&P 500 index. The main characteristics in the work described in [30] is the event representation method and the convolutional neural network which models the influence of these events on stock prices behavior in short-term, middle-term and long-term.

From the works cited above it is possible to identify three key points for the construction of deep learning models. The first one is the definition of the prediction horizon, the second one is the temporal effect of a news document and the third one is the representation type of the information. Regarding the first point, daily prediction (intra-day) is the most used.

The authors in [29] show that the performance of daily prediction is superior that weekly and monthly prediction. The second point refers to the time interval that news or events influence the stock prices behavior. If the objective is a daily prediction, it is indicated to use the news published the day before to the prediction day. However, Ding et al [30] shows that even a combination of weekly events and events from the last month can present relevant information to daily prediction. Finally, in relation to the last point, previous works mainly use simple characteristics such as bags-of-words, noun phrases and named entities. Recently, other representation techniques such as word-embedding [31]-[33] and event-embedding [30] are used. They distinguish from previous methods because can represent complex characteristics of words or events with lower-dimensional dense vectors.

In this paper, a RCNN model is proposed to forecast intraday directional movements of the Standard & Poor's 500 index (S&P500). This model uses as input a set of seven technical indicators extracted from the target series and financial news titles published the day before the prediction day. It is applied a two-step process to represent each news in the data set: first, a word2vec model [32][33] is used to generate a word representation and later an average of all the word vectors of the same title is performed, it addresses sparsity in word-based inputs. The RCNN model aims to obtain advantages from both models: CNN and RNN. CNN has a superior ability to extract semantic information from texts in comparison with RNN and RNN is better to catch the context information and to simulate complex temporal characteristics. The results of this work are compared with those presented by [30].

II. MODEL DESIGN

In this section we introduce the design of the recurrent convolutional neural network model (RCNN) to predict intraday directional-movements in financial time series using financial news articles and technical indicators as input. This model is named as SI-RCNN and is shown in Fig. 1. This design has four stages: Input layer, convolutional layer, recurrent layer and output layer. These stages are described below.

A. *Input layer*

The model uses two types of inputs, the first one is the technical indicator and the second one is the sequence of news titles. With the purpose of differentiating them, the input layer is renamed respectively as technical indicator layer and embedding layer.

The technical indicator layer takes as input a delayed sequence of seven technical indicators in chronological order as described in [3]. This input is defined as a matrix $I \in \mathbb{R}^{7\times n}$, where n is the length of the delay window.

The embedding layer takes a sequence of encoded sentences as input, this sequence corresponds to a set of titles of news articles from the day t arranged in chronological order. The encoding of the sentence is performed in two steps. First, a word2vec model trained on the continuous bag-of-words architecture is used for generate word embedding. This embedding is unique vectors of continuous values with length m for each word in the training corpus. The second step performs an average of all the word vectors in a title, so, a unique vector for the entire title is obtained, called sentence vector. To access each of these sentence vectors, each title in the data set is encoded using a one hot encoding method and finally, it is created an embedding lookup table.

The advantage of using the word2vec model is that the resulting word embedding vectors can capture linguistic regularities such as semantic and syntactic regularities, which is a desirable characteristic in NLP tasks.

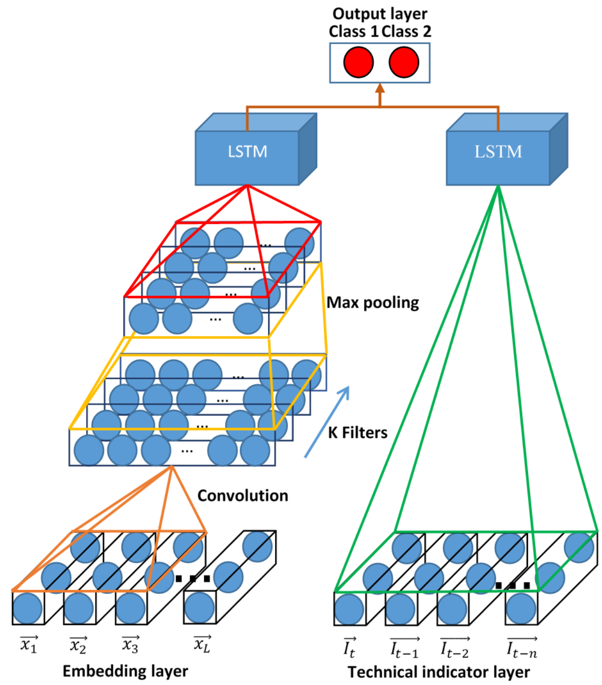

Fig. 1 SI- RCNN Model

B. *Convolutional layer*

The next stage following the embedding layer is the convolutional layer which is composed by four consecutive operations: convolution, subsampling or pooling, activation and dropout. In this work, the convolutional operator is designed to perform one-dimensional convolution also known as temporal convolution. This operator can capture local information through the combinations of sentence vectors in a window.

Formally, given a sequence of news titles as input represented by their embedding vectors, $X = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_L]$, and a filter unit, $W = [\vec{w}_1, \vec{w}_2, \dots, \vec{w}_R]$, where $\vec{x}_l \in \mathbb{R}^m$, $\vec{w}_k \in \mathbb{R}^m$, L is the number of news titles in a day, m is the length of the embedding vector and R is the lenght of the filter window, the one-dimensional convolution is performed as follows:

$$q_j = W \cdot \vec{x}_{j:j+R-1} + \vec{b} \tag{1}$$

where $\vec{b} \in \mathbb{R}^R$ is a bias term. This operation is applied to each possible window in the sentence to produce a feature map Q.

$$Q = [q_1, \dots, q_j, \dots, q_{L-R+1}] \tag{2}$$

Then, a temporal max-pooling is applied. It is similar as a spatial max-pooling module used in computer vision [31] but applied in one-dimensional input. This operation can capture the most important information in the sequence.

Finally, to introduce a non-linearity into the model we use a rectifier linear unit (ReLU) as activation function. This function is defined in (3).

$$h(q_j) = max\{0, q_j\} \tag{3}$$

Due to the high quantity of parameters in this layer, the model can be susceptible to overfitting, thus we use a regularization technique known as Dropout [35] with probability $p = 0.5$.

The previous information describes the process through which one feature is extracted from one filter. The model uses multiple filters with varying window sizes to obtain multiple features, therefore it is possible to explore different combinations of news titles. The output of this layer is defined as $h \in \mathbb{R}^{L-R+1 \times K}$, where K is the number of filters.

C. Recurrent layer

Two separate recurrent layers are used: one following the convolutional layer, to be possible to interpret the output of the convolutional layer as a sequence of $L - R + 1$ time steps; and the second following the technical indicator layer. In both cases the RNN has the same purpose, to model the temporal characteristics in the input sequence.

In this stage, a special RNN architecture named Long Short-Term Memory (LSTM) is used to introduce a new structure called memory cell. The key element of the LSTM is the cell state, C_t, which is controlled by 3 different gates, forget gate, input gate and output gate. The forget gate, f_t, decides which information of the previous cell state is remembered or forgotten. The input gate, i_t, decides which values of the cell state are updated by an input signal. Finally, the output gate, o_t, allows the cell state to have or not an effect on other neurons. One advantage of this structure is that it allows modeling long-term dependences in sequence data and preventing the vanishing gradient problem.

The LTSM equations are shown in equation (4) to (9), where v_t is the input of the recurrent layer, h_t is the output of the recurrent unit and W are the weights matrices. Fig. 2 shows a detailed scheme of this architecture.

$$f_t = \sigma(W_{vf} v_t + W_{hf} h_{t-1} + b_f) \tag{4}$$

$$i_t = \sigma(W_{vi} v_t + W_{hi} h_{t-1} + b_i) \tag{5}$$

$$\tilde{C}_t = tanh(W_{vC} v_t + W_{hC} h_{t-1} + b_C) \tag{6}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{7}$$

$$o_t = \sigma(W_{vo} v_t + W_{ho} h_{t-1} + b_o) \tag{8}$$

$$h_t = o_t * tanh(C_t) \tag{9}$$

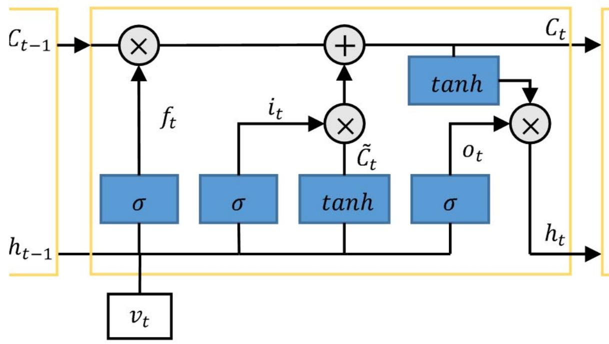

Fig. 2 LSTM architecture

D. Output layer

The last stage of the RCNN model is a traditional fully connected layer with softmax as activation function whose output is the probability distribution over labels. In this work, the objective is to forecast the direction of daily price movements of the S&P 500 index, this direction are used to create a binary class label where a label [1,0] represents that the stock price will increase and label [0,1] represents that the stock price will decrease.

III. Experiments

To evaluate the influence of using both news articles and technical indicators on predicting the movements of financial time series, a comparison is made between the hybrid model proposed in this work, SI-RCNN, and a model using only news articles as input. Additionally, the model is tested with two variations of the sentence representation; the first one consists in stack each word in the sentence and use the word embedding vector learned by the word2vec model, and the second one is described in section II. A, here is performed a mean operator over the sequence of word embedding vectors, in this work this method of representation is named as sentence embedding.

The results are compared with some different models reported in the literature. For the sake of simplicity, the following notation identifies each model:

- W-CNN: word embedding input and CNN prediction model.
- S-CNN: sentence embedding input and CNN prediction model.

- W-RCNN: word embedding input and RCNN prediction model (same model as Fig. 1. ignoring the technical indicator layer).
- S-RCNN: sentence embedding input and RCNN prediction model (same model as Fig. 1. ignoring the technical indicator layer).
- WI-RCNN: word embedding and technical indicators input and RCNN prediction model.
- SI-RCNN: sentence embedding and technical indicators input and RCNN prediction model.
- BW-SVM: bag-of-words and support vector machines (SVMs) prediction model [36].
- E-NN: structure events tuple input and standard neural network prediction model [29].
- WB-NN: sum of each word in a document and standard neural network prediction model [30].
- WB-CNN: sum of each word in a document as input and CNN prediction model [30].
- E-CNN: structured events tuple input and CNN prediction model [30].
- EB-NN: event embedding input and standard neural network [30].
- EB-CNN: event embedding input and CNN prediction model [30].

It is important to note that the CNN architecture using in WB-CNN, E-CNN and EB-CNN, uses as input news from the day, the week and the month before the prediction day, with the aim of modeling short-term, mid-term and long-term dependencies. However, in the cases of the models W-CNN and S-CNN, the CNN architecture correspond to the same as used in the RCNN model, that use only news from the present day to make a prediction of the next day.

A. Data Description

The database used in this work consists of 106.494 news articles from Reuter's website corresponding to the period from October 20, 2006 to November 21, 2013. The main topic of all of these articles is financial news. This dataset was created and released by [29]. Each news article consists of its title, content and publishing date. The publishing date is employed for news alignment with a corresponding financial time series. Reference [29] carried out a set of experiments and showed that news titles are more useful to forecast than news contents. For this reason, the proposed model uses only news titles as input.

In relation to financial time series, the Standard & Poor's 500 (S&P 500) index series are selected. This series was obtained through Yahoo Finance in the same period used to take the news articles. This series was selected with the purpose of having the largest number of news directly related to it. The target output and the seven technical indicators were calculated based on the information of this series and used as input of the model.

The target output consists in a binary variable where a value $[1,0]$ represents that the close price in the day $t+1$ will increase compared with the closing price in the day t and a value $[0,1]$ represents that the close price in the day $t+1$ will decrease compared with the previous day.

Since the approach in this work is a daily prediction, all the news titles of the same day are aligned in a unique instance, therefore each instance represents one single day. Then, it is applied a filter step, that consists in selecting from the news just the titles directly related to a specific stock, for example, in the case of the S&P500 index, news that refers to the first 100 companies that compose this index. The reason of using this filter step is that financial news of the general market can contain much irrelevant information. Authors in [29] confirm this affirmation. Finally, the set of news, technical indicators and outputs are aligned creating the pair input-output. The days without released news are ignored. Detail statistics of this data set are shown in Table I.

TABLE I. STATISTICS OF DATASET

Dataset	Training	Development	Test
Time interval	02/10/2006 18/06/2012	19/06/2012 21/02/2013	22/02/2013 21/11/2013
# Instances	1419	168	187
# Total Documents	13149	1976	2046
News' average per day	10	12	11

B. Details of implementation

To train the word embedding we use the Word2vec model. The selected length of these word vectors is 300 and it is initialized with publicly available vectors that were trained on 100 billion words from Google News using the continuous bag-of-words architecture. Words not present in the set of pre-trained words are initialized randomly. Thus, the input of the model has dimension $[L \times 300]$ when L is the number of news in a day t.

In the case of the technical indicators, our model uses a set of seven indicators arranged in a delayed sequence, the length of the delay window is defined as 5.

Regarding to the model parameters, we used three different filter window: $[3 \times 300]$, $[4 \times 300]$ and $[5 \times 300]$. For each one of these types of filters we used 64 filter units, stride 1 was applied and a padding convolution performed. The window of the temporal pooling layer was set to 2. In this way, the output of the convolutional layer, h, have dimension $[((L-2)/2)+1 \times 192]$. The recurrent layer on top of the output h, have 128 LSTM units. Finally, for modeling the sequence of technical indicators a recurrent network with 50 LSTM units is used.

The algorithm used for training the model is the stochastic gradient descent (SGD), using momentum 0.9 and initial step size 0.1, implemented using Tensorflow[1].

[1] https://www.tensorflow.org/

IV. RESULTS AND DISCUSSION

The focus of these experiments is to test the influence of the text representation method (word embedding or sentence embedding) and the influence of technical indicators as a complementary input of the model. The architecture RCNN proposed in this work was compared with a Convolutional Neural Network. Finally a comparison is made with a set of baseline models. The experimental results on S&P500 index prediction are shown in Table II.

TABLE II. RESULTS OF S&P 500 INDEX PREDICTION

Model	Training	Test
W-CNN	58,93%	57,22%
S-CNN	60,71%	60,96%
W-RCNN	59,76%	60,22%
S-RCNN	61,31%	61,49%
WI-RCNN	62,72%	61,29%
SI-RCNN	63,09%	62,03%
BW-SVM [36]	56,42%	56,38%
E-NN [29]	58,94%	58,83%
WB-NN [30]	60,25%	*
WB-CNN [30]	61,73%	60,57%
E-CNN [30]	61,45%	*
EB-NN [30]	62,84%	*
EB-CNN [30]	**65,08%**	**64,21%**

The results on the comparison between the models W-CNN vs S-CNN, W-RCNN vs S-RCNN and WI-RCNN vs SI-RCNN leads to conclusion that sentence embedding is better than word embedding as input of the model, obtaining better performance in the proposed task. This can be explained because the sentence embedding representation method addresses the problem of word sparsity in the text corpus.

Results obtained by models W-CNN vs W-RCNN and S-CNN vs S-RCNN show that the RCNN is better structure than CNN for the index prediction task. This structure can model the temporal features of the input sequence better than CNN. Additionally, the results of WI-RCNN and SI-RCNN models show that the influence of the sequence of technical indicators as complementary input is relevant, leading to a better performance.

The proposed SI-RCNN model outperforms all the baseline models with the exception of the EB-CNN, this is likely due to event embedding (EB) that is a more powerful method for model the content in news articles than the sequence embedding shown in this paper. But a comparison between the models EB-NN vs SI-RCNN, shown that the architecture used for the prediction model is also an important factor. In this case the SI-RCNN model has a better performance than EB-NN.

V. CONCLUSIONS AND FUTURE WORK

This work has presented a deep learning model that combines a convolutional layer with a recurrent layer for intraday stock price movement prediction and uses as input a combination of technical indicators and news titles. The RCNN architecture can model the local relation of a sequence of news titles and their temporal features. Results presented in Table II show that the RCNN architecture is a better alternative that convolutional neural network in this application. Moreover, it is possible to confirm the positive influence of the use of a hybrid input (news and technical indicators), leading to the conclusion that both sources of information are relevant.

The proposed model uses only news from the day before the forecasting day and outperform a set of models that uses news from the past day, week and month. This result reinforces the hypothesis that the information in the news articles have a short temporal effect in the financial market.

Regarding to the text representation method, it can be concluded that the sentence embedding used in this article is better than the word embedding because the problem of word sparsity in the data set, but is possible to use more powerful representation methods such as the proposed in [30] named as Event Embedding.

This work has been motivated by the successes of Deep learning methods in Natural Language Processing task. The future work will include the use of test methods such as reported in [37] and [38] for making better embedding vectors for the news titles. Other research direction is the use a reinforcement learning algorithms to train the proposed model on market simulation (trading simulation). These algorithms can train a model to create its own trading strategy. The financial time series is known by its volatility, in many cases occurs small changes in the series that can be interpreted as noise. Moreover, in several markets the operations have a cost, which must be considered. The elimination of small variations makes the model to focus only in events with significant variation on prices.

ACKNOWLEDGMENT

The authors are grateful to the Brazilian Research Council - CNPq for the financial support for this research. The autors are also grateful to NVIDIA Research Program that provided the hardware to run the experiments.

REFERENCES

[1] B. G. Malkiel, "A Random Walk Down Wall Street", W. W. Norton, New York. 1973.

[2] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market", *Journal of Computational Science*, vol. 2, no. 1, pp. 1-8, 2011.

[3] Y. Zhai, A. Hsu and S. Halgamuge, "Combining news and technical indicators in Daily stock price trends prediction", *Advances in Neural Networks*, pp. 1087-1096, 2007.

[4] G. Rachlin, M. Last, D. Alberg and A. Kandel, "ADMIRAL: A data mining based financial trading system", *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 720-725, 2007.

[5] J. J. Wang, J. Z. Wang, Z. G. Zhang and S. P. Guo, "Stock index forecasting based on a hybrid model", Omega, vol. 40, no. 6, pp. 758-766, 2012.

[6] R. P. Schumaker and H. Chen, "A quantitative stock prediction system based on financial news", *Information Processing and Management*, vol. 45, no. 5, pp. 571-583, 2009.

[7] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting", *Applied Soft Computing*, vol. 11, no. 2, pp. 2664-2675, 2011.

[8] A. K. Nassirtoussi, S. Aghabozorgi, T. Y Wah and D. C. L. Ngo, "Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment", *Expert Systems with Applications*, vol. 42, no. 1, pp. 306-324, 2015.

[9] J. R. Nofsinger, "The impact of public information on investors". *Journal of Banking & Finance*, vol. 25, no. 7, pp. 1339-1366, 2001.

[10] H. Mizuno, M.Kosaka, H. Yajima and N. Komoda, "Application of neural network to technical analysis of stock market prediction", *Studies in Informatic and control*, vol. 7, no. 3, pp. 111-120, 1998.

[11] W. Leigh, R. Purvis and J. M. Ragusa "Forecasting the NYSE Composite Index with Technical Analysis, Pattern Recognizer, Neural Network, and Genetic Algorithm: a Case Studying Decision Support", *Decision Support Systems*, no. 32, pp. 361–377, 2002.

[12] G. Gidofalvi and C. Elkan, "Using news articles to predict stock price movements". Department of Computer Science and Engineering, University of California, San Diego, 2001.

[13] R. P. Schumaker, H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFin text system", *ACM Transactions on Information Systems (TOIS)*, vol. 27, no. 2, pp. 12, 2009.

[14] B. Wang, H. HUANG and X. WANG, "A novel text mining approach to financial time series forecasting", *Neurocomputing*, vol. 83, pp. 136-145, 2012.

[15] H. Gunduz and Z. Cataltepe, "Borsa Istanbul (BIST) daily prediction using financial news and balanced feature selection", *Expert Systems with Applications*, vol. 42, no. 22, pp. 9001-9011, 2015.

[16] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah and D. C. L. Ngo, "Text mining for market prediction: A systematic review", *Expert Systems with Applications*, vol. 41, no. 16, pp. 7653-7670, 2014.

[17] R. C. Cavalcante, R. C. Brasileiro, V. L. Souzam J. P. Nobrega and A. L. Oliveira, "Computational intelligence and financial markets: A survey and future directions", *Expert Systems with Applications*, vol. 55, pp. 194-211, 2016.

[18] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis and A. Jaimes, "Correlating financial time series with micro-blogging activity", *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 513-522, 2012.

[19] Y. Kim, "Convolutional neural networks for sentence classification", *arXiv preprint arXiv:1408.5882*, 2014.

[20] X. Zhang and Y. LeCun, "Text understanding from scratch", *arXiv preprint arXiv:1502.01710*, 2015.

[21] P. Blunsom, E. Grefenstette and N. Kalchbrenner, "A convolutional neural network for modelling sentences", *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[22] R. Collobert and J. Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning", *Proceedings of the 25th International Conference on Machine Learning, ICML*, New York, NY, USA, pp. 160–167, 2008.

[23] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký and S. Khudanpur, "Recurrent neural network based language model", *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, Makuhari, Chiba, Japan, September 26-30, pp. 1045–1048, 2010.

[24] O. Irsoy and C. Cardie, "Opinion Mining with Deep Recurrent Neural Networks", *EMNLP*, pp. 720-728, 2014.

[25] X. Wang, Y. Liu, C. Sun, B. Wang and X. Wang, "Predicting Polarities of Tweets by Composing Word Embeddings with Long Short-Term Memory". *ACL (1)*, pp. 1343-1353, 2015.

[26] J. Xu, D. Chen, X. Qiu and X. Huang, "Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification", *arXiv preprint arXiv:1610.04989*, 2016.

[27] C. Zhou, C. Sun, Z. Liu and F. Lau, "A C-LSTM neural network for text classification", *arXiv preprint arXiv:1511.08630*, 2015.

[28] S. Lai, L. Xu, K. Liu and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification", *AAAI*, pp. 2267-2273, 2015.

[29] X. Ding, Y. Zhang, T. Liu and J. Duan, "Using Structured Events to Predict Stock Price Movement: An Empirical Investigation", *EMNLP*, pp. 1415-1425, 2014.

[30] X. Ding, Y. Zhang, T. Liu and J. Duan, "Deep Learning for Event-Driven Stock Prediction", *IJCAI*, pp. 2327-2333, 2015.

[31] Y. Bengio, R. Ducharme, P. Vincent and C. Jauvin, "A neural probabilistic language model", *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137-1155, 2003.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space", *Proceedings of ICLR Workshop*, 2013.

[33] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality", *Advances in neural information processing systems*, pp. 3111-3119, 2013.

[34] Y. L. Boureau, F. Bach, Y. LeCun and J. Ponce, "Learning mid-level features for recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2559-2566, 2010.

[35] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[36] R. Luss and A. d'Aspremont, "Predicting abnormal returns from news using text classification", *Quantitative* Finance, vol. 15, no. 6, pp. 999-1012, 2015.

[37] Q. V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents", *ICML,* vol. 14, pp. 1188-1196, 2014.

[38] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval", *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 4, pp. 694-707, 2016.