Project Title : Deep Learning for Stock Market Prediction Using News Analysis
Name: Ashish Vijay Lakamale
Student Number: 18210227
Email ID: ashish.lakamale2@mail.dcu.ie
Programme: Master's in computing (Data Analytics)
Module Code: CA685
Supervisor : Yvette Graham
Date: 11 Aug 2019

Disclaimer:
A report submitted to Dublin City University, School of Computing for module CA685 Data Analytics Practicum, 2018/2019.
I understand that the University regards breaches of academic integrity and plagiarism as grave and serious.
I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy.
I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the sources cited are identified in the assignment references.
I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.
By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)
Name: Ashish Vijay Lakamale
Date: 11 Aug 2019

Deep Learning for Stock Market Prediction Using News Analysis

Ashish Vijay Lakamale
Dublin City University
Dublin, Ireland
Ashish.lakamale2@mail.dcu.ie

Abstract — **Recently, the performance of stock market prediction has greatly improved relative to a couple of decades ago. This is mainly due to the increased ability to handle huge volumes of information that require extensive computational power. Although the performance of a single business differs in the stock market owing to news headline of its performance, income and purchase reports, external factors influence stocks and the market. The news headlines which makes investors more reluctant to spend money is called 'Headline Effect'. Sentiment analysis of such short texts is complex because they usually contain limited contextual information. In recent years, deep learning models like convolution neural network and recurrent neural network has been applied on text for sentimental analysis achieving improves results. This work aims to improve stock market opening price prediction using deep learning approaches to extract opinions from the text, like (i) articles and news headlines, (ii) historical stock prices. The performance of the proposed approach is demonstrated on real-world data of DJIA. In this experiment GloVe's larger common crawl vectors is used in forecast model to generate the word embedding and further it consists of an architecture of the joint CNN and RNN using coarse-grained local features created by CNN and long-term dependencies learned through RNN for sentiment analysis of brief texts.**

Keywords—Deep Learning; Convolution Neural Network; Neural Network, Stock market prediction

I. INTRODUCTION

Stocks are the hottest investment opportunity to obtain gains faster. The goal of maximizing the returns on the investment is dependent on the investors stock market prediction to buy or sell shares of stock. However, stock markets are highly unpredictable and influenced by multiple external aspects like the global economy, peer growth, government stability, repo rates and investor assumptions. It's difficult to explicitly predict the stock market nature. Nevertheless, recent developments in artificial intelligence and improvements in computing huge information, researchers can predict stock market nature more accurately than ever before. Examples are studied [1] [2] [3] [4].

Lately, there is abundance of user generated short text available on Internet through social networking, blogs, online reviews and more. These tests tend to be semantic and subjective. It's of great value to classify these short texts on the basis of semantic orientation. Existing study on sentiment assessment of brief texts mainly includes techniques based on emotional knowledge and classification based on features. The former focuses primarily on extracting and classifying feelings based on words and opinions. The latter focuses on the classification of feelings based on characteristics.

Similarly, in the traditional approaches stock market are predicted by applying knowledge of artificial intelligence with the technical and fundamental analysis, i.e. Studying historic stock market price patterns and fundamental analysis as a study of technical indicators, past financial articles, the economic business trend to predict future results. These studies used to discover the relationship between historical price trends and textual information with the help of text mining and machine learning approaches.

Modern approaches like [1] for stock market predictions deal with deep learning techniques like CNN and RNN and feed event representation vectors extracted from news sources, articles and blogs. All these techniques had generated improved results as compared to traditional approaches. Therefore, researches are applying deep learning techniques with numerical and textual information for a prediction model.

A. *Paper Structure*

The remainder of this paper is organized as follows. Section II reviews the related literature regarding the general stock market predictions using deep learning approaches. Section III describes the approach used to tackle the problem in detail. Section IV contains tests and results, and Section V lastly provides you the work on the issue in the future.

II. RELATED WORKS

Many studies have attempted to predict stock market behaviour using deep learning techniques. Each has consumed different deep neural network architecture and data sets. Previous studies can be divided into two sections as follows

A. *Deep Learning with Textual Information*

In traditional approaches, textual information was represented using term frequency-inverse document frequency (TF-IDF), bag-of-words, word order, synonyms, co-references, and pronoun resolution. However, these basic approaches were not able to extract the gist of the events from the news headlines and articles.

In [5], Xiao and Zhang introduced a model that uses the Open Information technique to extract events from articles, news sources. It decomposes the stock prices into terms like market stability and the noise of other stock markets around the world. These structured events are then fed into a deep neural network for prediction. In this method structured event was denoted as (O1, P, O2) where O1 represent the first object (company name, ticker name etc.), O2 is the second object and P represents the relationship between them. In this work, an automated trading system is conducted to evaluate the profits under real trading conditions. The stock investment was carried out on the blue-chip stock. They were able to get 55.21% accuracy implementing a feedforward neural network. In comparison with the traditional strategy this improved significantly. Peng et al. implemented the word embedding method with Deep Neural Network to predict the stock market for the S&P 500 index, this method gives an accuracy that was bit improved to 56.87% [3]. Again in 2015, they proposed a framework with an improved method where similar events are encoded into event vectors, calling it an event embedding approach [6].

In [2], Dang Lien proposed a prediction framework using Two-Stream Gated Recurrent Unit Network by feeding word embedding vectors from news headlines. Here they predicted the daily stock prices by analysing the financial statements, articles and historical stock price pattern using deep learning approach. This method has four aspects to it, Firstly Predicting stock price trend by proposing a two-stream gated recurrent unit (TGRU). Next, introducing a Stock2Vec sentiment embedding model trained on sentiment dictionary and stock news. Finally, three technical indicators were applied an add-on for stock price prediction. Furthermore, they analysed if technical indicators and news impacts the stock market immediately or after a short period of time (couple of days or a week). Confusion matrix was used to evaluate the efficiency of this model. The performance of this method was good than the baseline in [5], But it was less when compared to other models on the same dataset [6]. Because of word sparsity in the datasets this method is not appropriate for citing semantics of news headlines, it basically limited the forecasting strength. In addition, only three financial indicators were used, thus other critical indicators like Relative Strength Index (RSI) and P/E ratio were excluded. Other than this the complexity of the TGRU network requires long training time and immense computational resources.

B. Deep Learning with Both Textual and Numerical Information

In [7], Akita et al. introduced a prediction model that uses input like news headlines, articles and historical price patterns. The news events are converted into short-term and long-term event vectors by the paragraph vector approach and combine it with stock price vectors. These vectors are then given as input to Long Short-Term Memory to predict the following day's stock prices. Here stock's closing prices of ten companies were considered and used for prediction. LSTM can memorize previous timesteps due to its architecture. Also, the correlation between companies was studied. For example, an event like "Ford recalls…" might make Ford's stock prices decrease while making the stock price of Jeep's (peer firm) to increase at the same time. The relation among peer companies played a crucial role in predicting stock prices. Paragraph vectors are divided into two different categories: The Distributed Paragraph Vector Memory Model (PV-DM) and the Paragraph Vector Distributed Bag of Words (PVDBOW). These categories were used to predict word(s) in a context (sequence of training words). Evaluations show that the distributed representation of the textual information is effective than only numeric data methods. Furthermore, LSTM could capture the time series influence of input data than other models [6]. Finally, performance is evaluated, and it was found that using both types of information better performance is generated. Bag-of-words vector legs when compared to this model. Also, LSTM efficiency was better compared to traditional approaches like Multi-Layer Perceptron (MLP), Support Vector Machine (SVM) and Recurrent Neural Network (RNN). However, the forecast efficiency would have been improved by inclusion a vector of technical indices such as the moving average (MA) and the moving average convergence (MACD).

In [4], Vargas et al. in proposed a deep model for Standard's & Poor's 500 Index for forecasting intraday directional movements. On the day before the prediction day set of seven technical indicators extracted from the target series and published financial statements. Data set is generated with this information using a two-step process, first: word representation is generated using word2vec model after that all the similar title vectors are grouped, and the average is calculated to remove the sparsity in word-based inputs. A hybrid R-CNN model driving advantages from both models: CNN and RNN. RNN model legs in the ability to extract semantic information from text, CNN has greater capability in extracting it. But RNN is good with catching context information from text and to simulate complex temporal characteristics. This model showed an improvement when used with natural language processing (NLP) task. Also, it gives improved results compared to previous methods. Here, the model used the news as input only a day before the forecasting day and outperforms a set of models that uses inputs from the past day, week and month to predict. This concretes that the information in the news article has a short temporal effect on the stock market prices, but for long-term prediction, the systems lag in predicting accurately. Sentence embedding used in this paper is better than word embedding but lags event embedding features when efficiency is compared. Also, in order to train the proposed model on trade simulation reinforcement learning algorithm could have been used. This could train a model to have its own plan of action for trading.

In [8], Bin Weng introduced a model which uses a data-driven approach, consisting of 3 stages. In the first stage, a dataset from 4 different sources is extracted. These datasets are (i) Market information from Yahoo Finance on stocks movements, including opening /closing prices, volume of trade, NASDAQ and Dow Jones Industrial Average index; (ii) Critical technical indicators that has impact on market over time; (iii) Continuous monitoring of google news on stocks; and (iv) Counting number of visitors on Wikipedia page of relevant stock firm. In the Second stage, Variable selection method is used to select a subset of predictors that can generate accurate predictive power. In the third phase, three Artificial Intelligence techniques are implemented to predict the stock market prices [8]. These techniques are then evaluated against the 10-fold cross-validation sample using the area under curve (AUC) and seven other matrices. This model has tested on Apple Inc. feature-rich dataset for a period of three years and one month (May 1,2012 to June 1,

2015) and it generated a hit ratio of 85%. Even after achieving such a high hit ratio this model has few limitations that are only Apple Inc. stock were analyzed for a short period of time only, it's an assertion on the long period have reservations. Also, if other stock like Facebook, Google are considered than in that case we may require different online news sources. And the question persists, whether this model will work with totally new data sets or not?

The enhanced efficiency of these algorithms is primarily based on: (1) Similar semantic-oriented phrases are very comparable in the high dimensional distributional vectors. (2) CNN is also capable of learning local features from words or sentences at different places of the text, like the translation, rotation and invariance of images in CNN. (3) RNN takes words sequentially in a sentence and can learn text dependence rather than local features for the longer term.

III. Proposed Framework

This experiments aim is to improve the performance of prediction using numerical and textual information as an input to a deep neural network. Our proposed framework process flow is shown in the figure 1.

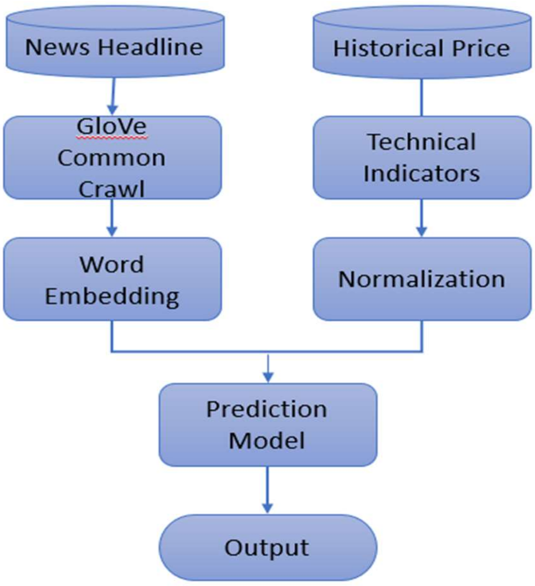

Figure 1. Process flow of the proposed framework

It takes the news headlines and generate word embeddings using Glove's Common Crawl. This prediction model uses CNN followed by RNN, in order to explore long term dependency, next it uses LSTM. Furthermore, to create target values, the difference in opening prices between current day and following day, which is further normalized and consumed in the model to predict the behaviour.

A. Data Understanding

In the dataset, there are two information channels:

1) News Data: Historical news headlines are crawled from Reddit World News Channel. They are classified by Reddit user votes and only the top 25-40 headlines for a single day are considered.

2) Stock Data: The "proof of the idea" is based on the Dow Jones Industrial Average (DJIA), or simply the Dow indicates the value of 30 large US-based publicly owned enterprises and the way they traded on the stock market for various periods of time.

Data	Reddit News Headlines
c	2
l	37.02
h	1,02,527
w_e	2,196,017

Table 1: Summary statistics for the datasets after tokenization.

c: number of target classes, *l*: average news count per day, *h:* total count of headlines, $w_{e:}$ total word embeddings,

B. Data Preprocessing

Data cleaning and pre-processing are the first and most important step in a machine learning project. The process was split into the following phases

1) Clening Text

First, to get the most signal out of the word, data cleaning is must. This will include converting the headlines into a lowercase and others given in [8].

a. Removing Special Characters and Puntuations: Since our embedding method do not have vector values for special characters and punctuations, which will be removed.

b. Cleaning numbers: It do not carry any significance in sentimental analysis.

c. Removing misspells: *[9]* Search for the misspells and replace with correct spelling as misspells n order to ger better good embedding coverage.

d. Removing contractions: *[10]* Words written with the apostrophe are contractions. As standard text is require to for extracting semantics.

Further, reformat the text to maximize the use of pre-trained word vectors for GloVe's. For instance, if every hashtag is separated from its text. Because GloVe lacks pre-trained word vectors for hashtag words, it is must to create the separation if experiment need to use these pre-trained vectors.

1) Uniform headlines

In headline data preparation, create news headlines of the same length for each day. Next, maximize the length of every headline to 16 words (this is the length of the 75th percentile headline) and length of any day's new maximized to 200 words. These values were chosen to balance the number of headlines with the number of headlines to be used. It is expected it would be useful to use more words for news every day (i.e. increase the limit of 200 words).

2) Word Embeddings and Sentence-Level Representaion
In [11] its shown that word embedding has an important role. Generally unsupervised pre-trained vectors and random initialization of word embedding are in use. Here, in this experiment, unsupervised word-level embedding using the GloVe Crawl vector is performed. The GloVe file contains about 2,196,017 word embeddings, each with 300 dimensions.

The principal concept of word embedding is to present words as feature vectors. Every entry in the vector represents a hidden function within the significance of the word, which can reveal the syntactic or semantic dependencies.

[12] presents that when deep learning methods are used for classification task, there is need to convert words into high dimensional vectors in order to take syntactic, morphological and semantic data on words into account. The objective function in Glove model is

$$J = \sum_{i,j=0}^{v} f(X_{ij})(w_i . \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij})^2 \quad (1)$$

Here f is a weighting function and w_i, w_j, bi and b_j are the learned where w_i and w_j are vectors for main and context words, b_i and b_j are biases for the main and context words.

In order to generate the weights to be used for the embedding of the model, generate a matrix of embedding from the words of our vocabulary. Then use pre-trained vector if a word is present in GloVe's vocabulary, otherwise, create a random embedding if a word is not found in GloVe's vocabulary. As the model trains, update the embeddings simultaneously, thus at the end of training our new 'random' embedding will be more precise.

3) Normalization
The input vectors representing prices must be standardized because they have a distinct value variety. Than use the feature scaling (min-max) to normalize the input vectors by using the following formula. Therefore, normalizing target data from 0 to 1.

$$X_{new} = \frac{x - X_{min}}{X_{max} - X_{min}} \quad (2)$$

Where x is an original value, X_{new} is the normalized value, X_{min} is minimum value for the dataset, X_{max} is the maximum value for the dataset.

C. Architecture

The proposed work altered the model suggested by Ding [13] in the following ways,

1) In the convolution operation, weight matrices and windows of multiple lengths have been used. For the generated feature maps, the sequential information present in the sentence context is preserved by the CNN model. It is because the operation pooling operates on the adjacent feature and instead of moving on the entire sentence it moves from left to right, the way human reads and understands the sentencing behavior.

2) CNN provides the encoded local feature whereas RNN provides the model's long-term dependencies which proves to be beneficial for our deep learning architecture.
Following facts allows us to work these models together. Result of the sentence classification has been shown to be better for CNN [14]. Also, CNNs are likely to obtain local and profound characteristics from natural language. There are various types of time-recursive neural network of Recurrent neural networks (RNNs), which can learn sequential data from long-term dependences. Since words can be viewed from left to right in a sentence, we are able to design RNN to match the readings and understanding the behavior of a sentence by people. [15] presented a deep model, convolutional-recursive, for 3D classification where recursive neural networks and convolutional neural network is combined.

The input used for this model is divided into two parts. The first part is where model consumes randomized initialized vectors on the corpora of the news headlines, second part where model consumes pretrained vectors from GloVe. In order to assess the sentiments of brief texts we presented an architecture jointed with CNN and RNN which takes local features obtained by CNN as input for the consumption by RNN. In an attempt to model sentence representation efficiently, we create an end-to-end and bottom-up algorithm. GloVe's crawler is used to generate word embedding which is provided as input to CNN model for generating several feature maps using windows of different sizes and weights matrices. These feature maps are given as input to the RNN model following convolution and pooling operations. Which allows RNN to learn long term dependencies and can be seen as the sentence level representation. Then fully connected network consumes the sentence level representation and finally, the ReLu outputs the classification result. Figure 1 shows the model architecture. This model includes : word embedding and sentence level representations, convolution layers, concatenation layer, pooling layer, RNN layer and fully connected layer with ReLu activation output.

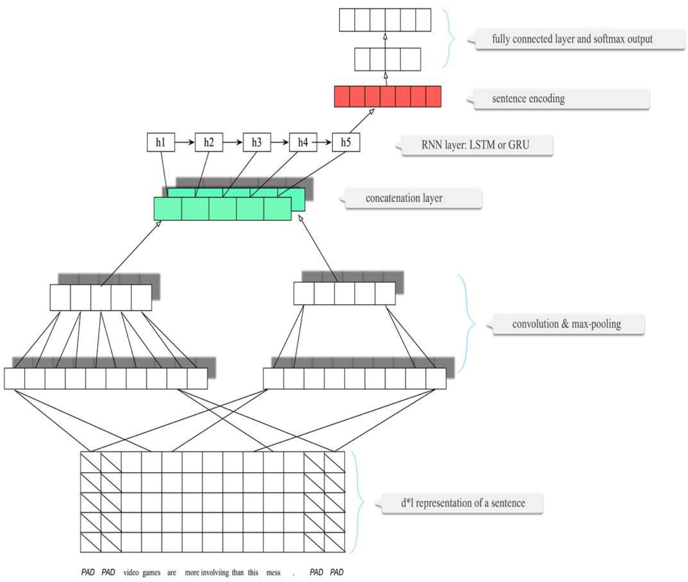

Figure 2. Model architecture for a sentence

1) Convolution and Pooling

In [16], a sentence level representation sequences the convolution layer which applies a matrix vector operation to each window of size w in a successive manner. Let the weight of the matrix be $H \in \mathbb{R}^{dxw}$ for convolution layer, further add a bias term b to the matrix vector operation result for feature mapping generation $c \in \mathbb{R}^{dxw}$. i-th element in feature map is given by,

$$c_i = \sigma \sum (C[*, i: i + w] \circ H + b) \quad (3)$$

Where $C[*, i: i + w]$ is the i-th to the $i + w$-th column vectors of the sentence-level representation.

Likewise, the local features for each specified sentence window are obtained with the same weight. Than extract the feature vector in size $l - w + 1$ with the matrix over all the word windows of the sentence. Both max pooling and average pooling was practiced, finding out that max pooling works better than the average pooling. Therefore, we proceeded with the max pooling operation to the output of CNN layer which transforms the feature map of size $l - w + 1$ to $\lfloor l - w + 1 \rfloor / 2$,

$$p = [p_1, p_2 p_3, \dots \dots . p_{\frac{\lfloor l-w+1 \rfloor}{2}}] \quad (4)$$

Here, n kind of matrix weight is applied to n feature maps.

$$P = [p^1, p^2, p^3 \dots \dots . p^n] \quad (5)$$

Where, $P \in \mathbb{R}^{\frac{\lfloor l-w+1 \rfloor}{2} \times n}$.

2) Recurrent Neural Network

Features generated by convolution and pooling can be considered advanced n-gram in the previous phase. [17] Presents, sequential input and learning long term dependencies can be done by recurrent neural network, than take this an input to RNN and further apply LSTM to it. The output of the model $T \in \mathbb{R}^n$ is considered as the encoding of the sentence.

3) Fully Connected Network with ReLu Output

In the final step features generated from the RNN are passed to fully connected rectified linear unit(ReLu) layer whose output is the probability distribution over all the categories.

IV. Experimental Setup And Result

A. Baseline Models

The following basic models for performance comparison are introduced in this research:

1) CNN-LSTM-rand

A model with randomly initialized a vector from GloVe, max pooling and LSTM recurrent unit.

2) CNN-LSTM-GloVe

A model with a pre-trained vector from GloVe, max pooling, and LSTM recurrent unit.

B. Hyperparametrs and Training

This experiment applies a modern approach taken from [18], namely grid search to our deep neural networks. It is a hyper-parameter tuning method to determine ideal values for a particular model. The two ways that allow altering the model are 'wider' and 'deeper'. 'deeper' adds an additional convolution layer to each branch and adds an additional layer to its end of the model and 'wider' doubles the hyperparameter values. For this model, there are two ' input ' branches because we want to generate CNNs with varying lengths of filters.

To prevent unnecessary training, the early stoppage is really helpful. As each iteration require a different number of epochs to fully train, this provides the flexibility to give adequate epochs to each iteration and properly train each iteration. Thus, we need to set a default number of epochs high enough to avoid stopping of the training session.
If the validation loss stops to decrease, callbacks will reduce your learning rate. This is truly useful because we want to begin at higher learning rates so we can have the model training fast but want it smaller near the end of the training that is essential in order to find the best weights. Further, in order to prevent overfitting dropout technique used.

C. Performance Evaluation

Following metrices are for the performance evaluation:

1) F1 Score: The F1 score is a measure of a model's accuracy. The accuracy p and recall r of the test is taken into account in calculating the score.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F1\ score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (8)$$

where TP is true positive, FP is false positive, FN is false negative.

2) Accuracy: The accuracy of the model is computed as follows:

$$Accuracy = \frac{number\ of\ correct\ predictions}{number\ of\ total\ predictions} \quad (9)$$

3) Mean Absolute error: It refers to the results of measuring the difference between two continuous variables. It is calculated using following,

$$MAE = \frac{1}{n} \sum_{i=0}^{n} |x_i - x| \quad (10)$$

Where n is number of errors, $|x_i - x|$ is absolute error.

D. Experimentl Results

The research is carried out using news headlines and historical price information. Dataset is divided into training and test data in 70-30 ratio. Models are trained using train data and finally, performance is measured using test data.

As shown in Table 2, LSTM and CNN models are compared for individual performance with tf-idf vectorizer. This observation suggests that model #1 and #2 performs nearly same.

Model	Accuracy	Precision	Recall	F1 score
#1 : tf-idf + LSTM	0.5476	0.5474	0.5476	0.5458
#2 : tf-idf + CNN	0.5079	0.5075	0.5079	0.5091

Table 2: F1 score of models on test data

Table 3 lists the results of both models. In this table 'Actual direction' gives how much the percentage of the predicted direction matched the actual direction. Next, MAE giving the opening price error tells that model with GloVe and max pooling pre-trained vectors are particularly successful. Its deduced that deep learning models perform better with pre-trained vectors when compared to randomly initialized vectors on corpora. We can say that semantic sparsity issue can be tackled by pre-trained vectors to some extent. The individual model of CNN and RNN with tf-idf vectorization performs likewise for the classification of short text. This study proves the concept that CNN and RNN model can be combined to get adequately good accuracy than existing models, where CNN extracts the local input features and the sequence input of RNN processes while studying the long term dependency.

Model	MAE	Actual Direction (%)
#1 CNN + LSTM + rand	74.15	51.78
#2 CNN +LSTM+GloVe	59.15	55.52

Table 3: Evaluation on test data

VI CONCLUSION

With the aim to improve the stock market opening price predictions taking the help of word embedding vectors, share market price pattern as input to deep learning algorithm, we presented a deep neural architecture that combines the convolution neural network (CNN) and recurrent neural network (RNN) for sentimental analysis of short texts. In particular, we can keep local features and their sequential relation in a sentence with our pooling procedure in neighboring words. RNN also learns the long-term dependency and sequential relationships between the features in each word and the global features of each word. This model performance is mediocre as a lot of factors, information and even sentiments are important and only Redditt's' 25-40 daily headlines are not enough to take all the complexities into account.

The architecture can be applied to other natural language processing task and sentence modelling task. For future work we can modify our architecture to handle long text classification task, and in order to improve results we can use headlines from all 30 companies individually that make up the DJIA.

V. REFERENCES

[1] Nelson, D. M. Pereira and A. C. Oliveira, "Stock market's price movement prediction with LSTM neural networks," Networks (IJCNN), 2017 International Joint Conference, 2017. [Online].

[2] H. D. Huynh, L. M. Dang and D. Duong, "A New Model for stocks Price Movements Prediction Using Deep Neural Network," Eighth International Symposium on Information and Communication Technology, 2017. [Online].

[3] H. Peng, E. Cambria and A. Hussain, "A review of sentiment analysis research in Chinese language," Springer Link, August 2017. [Online].

[4] Vargas, M. R., Lima, B. S. L. P. De and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2017. [Online].

[5] X. Ding, Y. Zhang, T. Liu and J. Duan, "Using structured events to predict stock price movement: An empirical investigation.," Conference on Empirical Methods in Natural Language Processing, 2014. [Online].

[6] D. X., Z. Y., L. T. and J. Duan, "Deep learning for event driven stock prediction," IJCAI International Joint Conference on Artificial Intelligence, 2015. [Online].

[7] R. Akita, A. Yoshihara, T. Matsubara and K. Uehara, "Deep Learning for stock prediction using numerical and textual information," 15th International Conference on Computer and Information Science (ICIS), 2016. [Online].

[8] Gharatkar, Sandesh, Ingle, Aakash, Naik, T. Save and Ashwini, "Review preprocessing using data cleaning and stemming technique," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 1 February 2018. [Online].

[9] Amorim, R. C. de, Zampieri and Marcos, "Effective Spell Checking Methods Using Clustering Algorithms," Association for Computational Linguistics, 2013. [Online].

[10] L.-C. Yu, J. Wang, K. R. Lai and X. Zhang, "Refining Word Embeddings Using Intensity Scores for Sentiment Analysis," IEEE/ACM Transactions on Audio, Speech, and Language Processing (Volume: 26 , Issue: 3 , March 2018) , 29 December 2017. [Online].

[11] Z. Zhang and M. Lan, "Learning sentiment-inherent word embedding for word-level and sentence-level sentiment analysis," 2015 International Conference on Asian Language Processing (IALP), 14 April 2016 . [Online].

[12] J. Pennington, R. Socher and C. D. Manning, "Glove: Global vectors for word representation," Computer Science Department, Stanford University, Stanford, CA 94305, 2014. [Online].

[13] Y. Z. Wallace and Byron, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," Cornell university, 6 April 2016. [Online].

[14] Y. Kim, "Convolutional neural networks for sentence classification," New York University, 2014. [Online].

[15] R. Socher, B. Huval, B. Bhat, C. D. Manning and A. Y. Ng, "Convolutional-Recursive Deep Learning," Computer Science Department, Stanford University, Stanford, CA 94305, USA, 2012. [Online].

[16] XingyouWang, W. Jiang and Z. Luo, "Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts," Association for Computational Linguistics, 2016. [Online].

[17] Q. Yang, Z. He, F. Ge and Y. Zhang, "Sequence-to-sequence prediction of personal computer software by recurrent neural network," 2017 International Joint Conference on Neural Networks (IJCNN), 03 July 2017. [Online].

[18] B. Qolomany, M. Maabreh, A. Al-Fuqaha, A. Gupta and D. Benhaddou, "Parameters optimization of deep learning models using Particle swarm optimization," 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 20 July 2017. [Online].