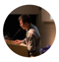
Stock Price Movement Prediction from Financial News with Deep Learning and Knowledge Graph Embedding: 15th Pacific Rim Knowledge Acquisition Workshop, PKAW 2018, Nanjing, China, Au...

Chapter · July 2018

DOI: 10.1007/978-3-319-97289-3_8



Yang Liu
Wuhan University

Adrian Carrio
Universidad Politécnica de Madrid

Stock Price Movement Prediction from Financial News with Deep Learning and Knowledge Graph Embedding

Yang Liu[1], Qingguo Zeng[2], Huanrui Yang[3](✉), and Adrian Carrio[4]

[1] Department of Industrial Engineering, Business Administration and Statistics, Universidad Politécnica de Madrid, 28006 Madrid, Spain
yang.liu00@alumnos.upm.es
[2] South China Normal University, Shipai, Guangzhou, China
domceng@gmail.com
[3] Electrical and Computer Engineering Department, Duke University, Durham 27708, USA
inociencio@gmail.com
[4] Center of Automatic and Robotic, Universidad Politécnica de Madrid, 28006 Madrid, Spain
adrian.carrio@upm.es

Abstract. As the technology applied to economy develops, more and more investors are paying attention to stock prediction. Therefore, research on stock prediction is becoming a hot area. In this paper, we propose to incorporate a joint model using the TransE model for representation learning and a Convolutional Neural Network (CNN), which extracts features from financial news articles. This joint learning can improve the accuracy of text feature extraction while reducing the sparseness of news headlines. On the other hand, we present a joint feature extraction method which extracts feature vectors from both daily trading data and technical indicators. The approach is evaluated using Support Vector Machines (SVM) as a traditional machine learning method and Long Short-term Memory (LSTM) model as a deep learning method. The proposed model is used to predict Apple's stock price movement using the Standard & Poor's 500 index (S&P 500). The experiments show that the accuracy of news sentiment classification for feature selection achieved 97.66% by model of joint learning, the performance of joint learning is better than feature extraction by CNN, the accuracy of stock price movement prediction through deep learning achieved 55.44%, this result is higher than traditional machine learning. This model can give the investors greater decision support.

Keywords: Stock market · Deep learning · Event tuple · Financial news
Knowledge graph embedding

1 Introduction

Stock market predictions are important business activities, today. However, establishing an accurate stock prediction model is still a challenging issue [1]. In addition to historical market movements, the prediction of stock market is also affected by

K. Yoshida and M. Lee (Eds.): PKAW 2018, LNAI 11016, pp. 102–113, 2018.
https://doi.org/10.1007/978-3-319-97289-3_8

financial news. Currently, financial news includes a large number of news events that are presented in the form of a knowledge graph [2]. As a result, both news events and historical market movements can improve the forecasting ability of the models.

Knowledge graphs were formally proposed by Google on May 2012 as an intention to improve the search engine's ability, enhancing the search quality and the users experience [3]. At present, due to the development of Artificial Intelligence technology for knowledge graphs, they have been widely applied in intelligent search, question and answering, and intelligent finance. A knowledge graph in finance aims to find the relationships in entities such as: companies, management, news events and user preferences [4]. These entities enable efficient, financial data-based decision making and provide business advice for investors to predict stock trends. In view of the above-mentioned reasons, this research focuses on investigating how to improve the accuracy of stock price predictions through the use of knowledge graphs.

Knowledge graphs are databases implementing semantic search by preserving the relationships between multiple entities [5]. The typical structure of a knowledge graph is: entity 1, relation, and entity 2. Entities are concrete things in the objective world. For example, "*Tim Cook is the CEO of Apple*" or "Apple is an Internet company", etc. Events are objective activities [6]. For example: the increase or fall of stocks, the release of new products, etc. The relation describes the objective relationship between concepts, entities and events. For example: "*Jobs and his partners founded Apple in 1976*", and "Apple released iPhone X in September 2017", etc. According to the definition of tuples in a knowledge graph, we can introduce the definition of an event tuple as a tuple (A, P, O), where A represents agent, P represents predicate, O represents object [7]. For example, "*Apple says initial quantities of iPhone 7 Plus sold out*". Event tuples group together relevant elements and can be used as an efficient way to improve prediction accuracy.

In the previous works [6, 7], there is a common problem in the integration of event tuple with knowledge graph. These works didn't not capture the structural information in the text, and these information is very important for affecting stock to increase or decrease. The bag of word and event tuple are used in previous work [6, 8], for example, Samsung sues Apple for stealing patents. If this text is represented in the bag of word, it could be represented that "Samsung", "sue", "Apple", "stealing", and "patents". Because there is no structured information in this text, this way is difficult to determine which company's stock price will increase and which company's share price will decrease. If this text represented by a structured event:(Agent: "Samsung"), (Predicate, "sue"), (Object: "Apple"). Although the object of each tuple is clearly known, it lost more information when text converted into a structural vector. Therefore, we propose a joint learning model of tuple and texts to maximize the retention of structured information in event tuple. Moreover, we select Apple's financial news as structure information extraction. We found that the accuracy of the classification reached 97.6% on this data. The structured information in the text was retained, these information more directly predicts the stock market.

2 Related Work

2.1 Deep Learning in Stock Market Prediction

There are three major applications of deep learning in stock market prediction. Firstly, the prediction of the operation in financial markets. Forecasting stock volatility and price of financial assets has attracted a lot of interest in the field of finance, especially in the prediction of price trend and direction [9]. As an example, in the work of [10], deep neural networks analyze the close price to predict the daily fluctuation of the S&P 500 index. Secondly, the application of natural language processing for forecasting Stock Market. In the work of [6], fully-connected neural networks and convolutional neural networks are used to build models that extract event tuples from the news. This method analyzes the impact of long-term events on the stock prices and judges on the future direction of the stock market price. Finally, deep learning can help investors improve their trading strategies. In the work of [11], deep learning is used to encode stock market information to design an effective portfolio based on the results of the decoding obtained. This method can screen market common factors and stock individual factors. None of the above methods make use of event tuples.

2.2 Knowledge Graph Embedding

Knowledge graph embedding is a type of representation learning between entities and relations in a knowledge base. The entities and relations are mapped into a low-dimensional space representing the semantic information between entities and relationships. Currently, there are some algorithms which explore knowledge graph embedding in translation distance models. Both entities and relations can be represented as vectors in the same space. Given a fact (h, r, t), the relation is interpreted as a translation vector r so that the embedded entities h and t can be connected by r with low error, for example, when $h + r \approx t$, the result (h, r, t) is true. Currently, there are different translation distance models. TransE [12] model is a computationally efficient and predictive model, this model that can be modeled well for "one-to-one" relational types. TransH [5] model, the head and tail vectors are mapped onto the hyperplane where the relationship is located, then the translation process is completed on the hyperplane. TransR [13] model is based on TrasnE model, which model's entities and relations are in different dimensions of space, this model defines one matrix for each relation, which is used to transform the entity vectors into the spaces, and this model completes the translation in the relational vector space. The TransD [14] model is an improvement based on TransR model, TransD model considers the transformation matrix based on TransR model, which is dynamically determined by the entity relationship. Comparing in these models, we need a simple mapping of one-to-one entities for extracting the feature, the accuracy of representation between entities and relationships is improved, the TansE model have these features. Therefore, the joint learning model of text from news and events tuple is presented based on TransE model in this work.

2.3 Representation Learning Based on Text and Knowledge

Because of the TransE model includes a one-to-one relation between two entities, in order to build a large-scale knowledge graph embedding, the relationship between many entities has to be constantly added. Recently, several methods have been applied in knowledge graph completion. In the work of [5] a word embedding was extracted from Wikipedia and a knowledge base was trained using the TransE model. This model makes the word representation corresponding to the entity in the text as close as possible to the entity representation in the knowledge base. Moreover, a deep architecture was proposed using both structural and textual information of the entities [15]. Specifically, three neural models were used to encode the valuable information from the entity in the text description. The method proposed by [16] illustrates two encoders for encoding entity descriptions: a continuous bag-of-words and a deep learning model. Previous works didn't make full use of description information in the text and a lot of semantic information was lost during the feature extraction process. In order to retain and fully extract the semantic information in financial news, we propose to extract feature vectors from the news text by means of a CNN model.

3 Task Description

3.1 Research Architecture

Figure 1 shows the research architecture [17]. The steps are as following:

1. Corpus collection and stock data compilation: By looking up keywords such as "Apple" at Thomson Reuters' website and writing a web-crawler to obtain Apple's financial news.
2. Data pre-processing: including corpus analysis, text normalization, word tokenization, label tagging and word to vector transformation.
3. Feature selection: selecting the features from the embedding and stock data layers to compute eigenvalues, to afterwards generate a feature vector using deep learning.
4. Deep Learning: Feature extraction, prediction model built with Tensorflow [18].
5. Results evaluation: analysis of the results and extraction of conclusions.

3.2 Dataset Description

In this work, we build our own financial news corpus with headlines from Apple, published in Thomson Reuters between October 2011 and July 2017. This database consists of 6,423 financial news headlines, each including its title and release date. The title is used for event embedding and feature extraction, while the release date is used to align the corresponding the financial news with trading data from a temporal series. As previous work has shown, using the headline can help reducing noise in text mining [19]. It has been shown that the headline concisely represents the content of the text [20]. We use exclusively the news headlines from Apple for stock price movement prediction.

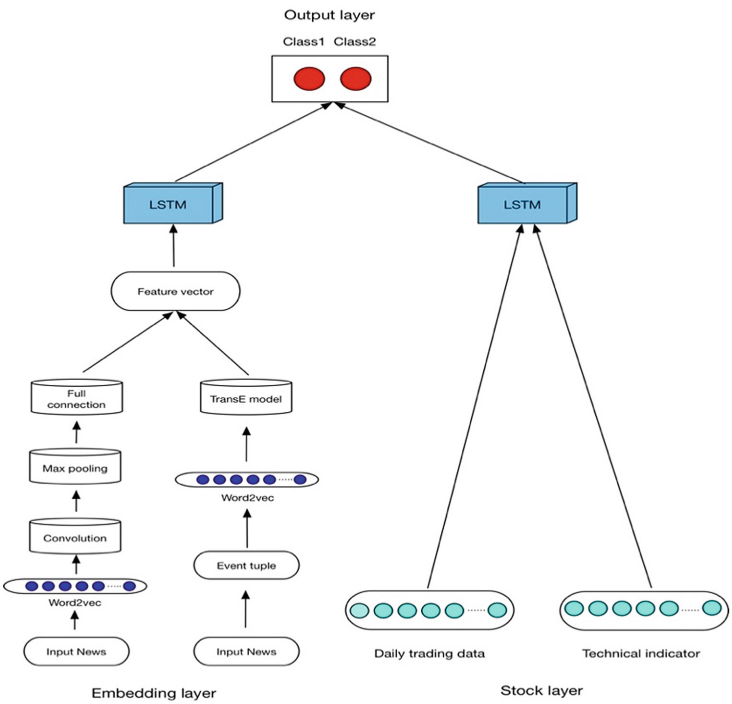


Fig. 1. Proposed learning architecture.

We also collected related daily stock data from the Standard & Poor's 500 (S&P 500) index in Yahoo Finance, in the same period as the stock data and the financial news headline from AAPL stock data. Daily trading data and technical indicator features are used by our model [21, 22]. And in the previous work [23], the technical indicators strongly prove the improvement of the stock forecast. Meanwhile, three technical indicators were used, which are calculated based on the AAPL daily trading data. the stock data of AAPL on S&P 500 from October 30, 2011 to July 8, 2017. In total, there are n = 1467 trading days. The variables are as follows:

1. Opening price; 2. Closing price; 3. High price; 4. Low price; 5. Volume; 6. Stochastic oscillator (%K); 7. Larry William (LW) %R indicator; 8. Relative Strength Index (RSI). The data is therefore arranged in 8-dimensional vectors.

The target output consists of a binary variable. A value of 1 indicates that the close price in the $t + 1$ will be higher than day t, while a value of 0 indicates that the close price drops down as compared to that of day t. Because of each news headline is presented as an event tuple, in Table 2 these headlines need to be transformed to event triplets, which has been done here using Reverb[1], there are 2,799 financial news headlines after filtering. The news event tuples and stock data are aligned creating input-output pairs, and those days without released news are left out. In Table 1,

[1] http://reverb.cs.washington.edu/.

the matches found are 941 pairs of event tuples and stock data. For this dataset, we use 80% of the samples for training and the remaining 20% for testing.

Table 1. Dataset information

Dataset	Total	Training	Test
Time interval	1.10.2011–30.7.2017	1.10.2011–12.30.2016	1.1.2017—7.31.2017
Event tuples	941	780	161

3.3 Data Pre-Processing

A web crawler to capture financial news has been developed. The obtained data consists of unstructured information, so the following procedures were applied to structure it:

1. Remove redundant information in the text, such as: stop words, excessive punctuation and repeated words.
2. Tagging the label for each news headline. The label consists of a categorical value among five possible labels: 0 (extremely negative), 1 (negative), 2 (natural), 3 (positive) and 4 (extremely positive). The label is given by the level of sentiment [24].

In Table 2, label 0 means that the event happened to an Apple's competitor, label 1 means that Apple lost something in this event, label 2 means that this event had nothing to do with Apple or that it did not cause any impact to Apple, label 3 means that this event caused the Apple to obtain something and label 4 means that Apple increased its profit or created more value from this event.

Table 2. Samples extracted from the dataset

Date	News headline	News headline event tuple	Label	Number of labels
25/1/2012	Motorola sues Apple for patent infringement	Motorola, sues, Apple for patent infringement	0	502
3/2/2012	Apple stops selling some devices online in Germany	Apple, stops selling, some devices	1	537
15/2/2012	Exclusive: Proview says any ban of iPad exports hard to impose	Proview, say, any ban of iPad exports	2	661
29/2/2012	How Apple, and everyone, can solve the sweatshop problem	Apple, solve, the sweatshop problem	3	692
2/4/2012	Apple's iPad tops Consumer Reports' list despite heat issue	Apple's iPad, tops, Consumer Reports' list	4	405

3. Word vector transforming. We use the Word2vec[2] model to train the word embedding. Word2vec is an algorithm for learning this distributed word vector, this vector is a real-size vector of fixed size, we determined its size, such as 300 dimensions. The word embedding was trained on 100 billion words from Google News, using the continuous bag-of-words architecture.

4 Methodology

4.1 Feature Selection by the Model of Joint Learning

4.1.1 Feature Extraction from the News Title Using CNN Model

As extracting the tuple from the original text may cause a loss of information, we also encode the original text of each piece of news' headline directly using a CNN model. Given the sequence of words in the title of a financial piece of news, we embed each word using Word2vec [25] as $(x_0, x_1, x_2 \ldots, x_n)$, and then concatenate these vectors as a matrix for the input to the CNN model. In our implementation, the joint learning model is made up of four consecutive layers: the first layer is the input layer, the second layer is a convolutional layer, the third layer is a max-pooling layer and the fourth layer is a fully connected layer for extracting features describing the relations between words. The convolution layer and max-pooling layer are designed according to TextCNN [26], which has been proved effective for emotion classification. Another linear layer followed by a softmax layer is attached to the fourth layer, which classifies the title into 5 previously defined emotional classes. The CNN model is trained using these emotion labels, with the classification loss denoted as E_d.

4.1.2 Feature Extraction from the Event Tuple Using TransE Model

In order to use the knowledge graph information and event tuple in news text at the same time. Given the knowledge graph *KG = (E, R, T)*, for each event tuple *(h, r, t)* $\in T$, h_s and t_s denote the structure vector representations of the head entity and tail entity the joint learning model is defined as the average of each word's word vector in the entity, obtained with the Word2Vec model. The two entity vectors are then mapped into the same relation space using a trained low rank weight matrix, i.e.

$$H = L_r h_s$$

$$T = R_r t_s$$

As assumed in TransE model, the relationship vector R should satisfy $H + R \approx T$, such as $Tim\ Cook + found \approx Apple\, and\, Apple + release \approx Iphone8$. The loss function of this structure model is defined as follow:

$$E_S = \|H + R - T\|_2^2$$

[2] https://code.google.com/archive/p/word2vec/.

4.1.3 Combined Loss Function for Feature Extraction

We combine two types of representation learning together to map the news titles into feature vectors. Here we denote the parameter set as $\emptyset = (L_r, R_r, \theta)$, where L_r, R_r are the mapping matrices for entities in the structure model, and θ are the weights of the CNN. For the structure model, we use a relationship vector R, identical to the result of the feature extraction layer of the CNN, to compute the loss:

$$E_S = \|H + R - T\|_2^2$$

And then combine this loss with the classification loss of the CNN model using L2 regularization to obtain the following overall loss function for feature extraction:

$$E = E_s + \alpha E_d + \beta\|\phi\|^2$$

4.2 Stock Market Prediction Model

4.2.1 Long Short-Term Memory Networks

On top of the feature extraction models, there are two LSTM models used in parallel, one is to interpret the output of the feature vector, and other one is interpreted the stock data features [17]. Long Short-Term Network is a special type of RNN that learns long-term dependencies. LSTM has already been proved successful in predicting stock prices [27].

LSTM model is an important part of deep learning [28]. Figure 2 shows a workflow describing the principles of this model. There are three gates: forget door, input door and output door. The door consists of a sigmoid activation function and a point-by-point multiplication operation. The model holds the hidden state of the previous time step, which has a three-wise connection: first one to the forget gate, f_t, second one to the input gate, i_t, and third one to the output gate, O_t.

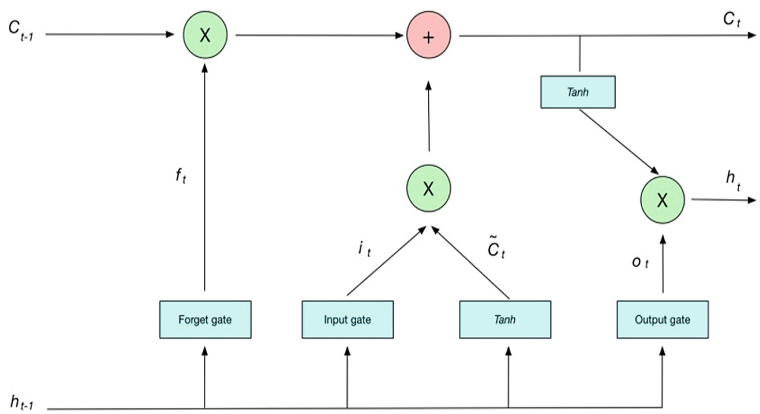


Fig. 2. Proposed LSTM model architecture.

In this case, i_t decides when to pass the activation into the memory cell, and O_t is used to activate the outgoing memory cell. Accordingly, for post-delivery, the O_t decides when to let the error flow into the storage cell,i_t decides when to let it flow out of the storage cell. We define the cell's new state as follow:

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

The advantage is that this method solves the vanishing gradient problem, which is caused by the gradual reduction of gradients during backpropagation. During training, we adjust several parameters in the LSTM. We consider the word embedding, originally sparse and defined over high-dimensional vectors. The word embedding eigenvectors are reduced to 50-dimensional vectors. These feature vectors are entered together with the stock data into the LSTM model.

4.2.2 Output Layer

The output of this model is a traditional fully connected layer with Logistic regression-based classification, that finds the probability distribution of eigenvectors over the labels.

5 Experiment and Results

5.1 Experiment Settings

In the experiments, to evaluate the influence of using both embedding layer and stock layer on stock price movement, a comparison is made between the hybrid model proposed in this work. We compare different models and the following notation identifies each model:

(1) **T-SVM**: Input of Tf-idf algorithm feature extraction and SVM prediction model [29].
(2) **J-SVM**: Input of joint learning feature extraction and SVM prediction model.
(3) **C-SVM:** Input of CNN feature extraction and SVM prediction model.
(4) **C-LSTM**: Input of CNN feature extraction and LSTM prediction model [17].
(5) **J-LSTM**: Input of feature extraction by the model of joint learning and LSTM prediction model.

The time step of above model set is one day, which makes a prediction of the next day. We compare the predictive performance on the test dataset in predicting the movement for the next day and evaluate the performance of the model by means of accuracy, F1-score [30].

5.2 Results and Discussion

We select feature vectors in the embedding layer by means of a joint learning model. When news is tagged with five sentiment labels, the classification accuracy is 97.66% for the five labels. This result demonstrates that the mapping ability of the head entity and tail entity in the same space is improved and the sparseness of each news and event tuple are reduced. In consequence, the joint learning model has a good performance using the feature vectors built from event tuples and financial news, and we consider the short-term interval of one day.

According to the results shown in Table 3, the accuracy of T-SVM, J-SVM and C-SVM is 49.17%, 54.92% and 45.6%, respectively. However, through joint learning using the proposed feature extraction model, the accuracy of C-LSTM reaches 51.32% and the accuracy of J-LSTM reaches 55.44%. Obviously, the accuracy of the joint learning model is higher than the other feature extraction models. In the feature selection by joint learning and CNN, the accuracy of C-SVM is 5.76% lower than the accuracy of C-LSTM, this decrease proves that feature extraction performance by means of joint feature learning is improved with respect to the Tf-idf algorithm and CNN. Comparing between traditional machine learning and deep learning techniques, the prediction accuracy using deep learning is 6.27% higher than traditional machine learning. Furthermore, F1-score increases from 53.48% to 71.33%. Thus, the comparison result provides the evidence that deep learning techniques can outperform traditional machine learning in stock price movement prediction, and the performance of feature selection by the joint learning surpasses the feature selection by CNN in this work.

Table 3. Accuracy and F1-score in the prediction of next day's close price movement

Predictive analytics Method	Accuracy	F1-Score
1. T-SVM	49.17%	44.58%
2. J-SVM	54.92%	53.48%
3. C-SVM	45.60%	33.96%
4. C-LSTM	51.32%	63.04%
5. J-LSTM	**55.44%**	**71.33%**

In the previous work [6, 17], the financial news headline as input that is difficult to investigate the relationship in some companies, the financial news from many competing companies may be noise data, these data reduce the stock's forecasting accuracy. The knowledge graph can provide the attributes and relationships of the entities and obtain more information from the companies, which increases the prediction accuracy of the stock prices [7]. The above accuracy results comply with the reported rates in previous works [20, 27]. Deep learning remains the preferred choice for stock forecasting models. LSTM model requires a lot of data for training. Given the scarcity of data in our problem, Deep learning would have difficulties to capture temporal features and the features of word embedding for long-term predictions. However, these experiments show that the model deep learning and joint learning is valid and provides good performance in predicting the movement of next day's price.

6 Conclusions and Future Works

In this paper, we propose the application of joint learning of event tuples and text for stock prediction, which solves the problem of text sparsity in feature extraction. It has been proved that our deep learning model predicts better than traditional machine learning in the short term. This can give investors feedback to support business decisions and to improve investment planning.

In future research, we will expand the developments in this paper. Firstly, our research will consider the incorporation and comparison of more machine learning models. Secondly, we will include companies from different areas, such as Boeing or Walmart and multiple financial news sources. The objective will be to prove the feasibility and generalization capability of the model. Finally, with respect to individual event tuple, the specific impact on the stock market needs to be analyzed and classification could be done over other time periods.

Acknowledgments. The authors Yang Liu would like to thank all the reviewers for their insightful and valuable suggestions. This work is supported by the China Scholarship Council (CSC).

References

1. Nguyen, T.H., Shirai, K., Velcin, J.: Sentiment analysis on social media for stock movement prediction. Expert Syst. Appl. **42**, 9603–9611 (2015)
2. Liu, K., Zhang, Y.-Z., Ji, G.-L., Lai, S.-W., Zhao, J.: Representation learning for question answering over knowledge base: an overview n ature. Acta Autom. Sin. **42**(6), 807–818 (2016)
3. Qiao, L., Yang, L., Hong, D., Yao, L., Zhiguang, Q.: Knowledge graph construction techniques. J. Comput. Res. Dev. **53**(3), 649–652 (2014)
4. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**, 2724–2743 (2017)
5. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. AAAI Conf. Artif. Intell. **14**, 1112–1119 (2014)
6. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: IJCAI International Joint Conference on Artificial Intelligence, pp. 2327–2333 (2015)
7. Ding, X., Zhang, Y., Liu, T., Duan, J.: Knowledge-driven event embedding for stock prediction. Coling **2016**, 2133–2142 (2016)
8. Filliat, D.: A visual bag of words method for interactive qualitative localization and mapping. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3921–3926 (2007)
9. Harris, G.: A Survey of Deep Learning Techniques Applied to Trading (2016)
10. Xiong, R., Nichols, E.P., Shen, Y.: Deep Learning Stock Volatility with Google Domestic Trends. 2, 0–5 (2015)
11. Heaton, J.B., Polson, N.G., Witte, J.H.: Deep Learning in Finance, pp. 1–20 (2016)
12. Bordes, A., Usunier, N., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Adv. NIPS. **26**, 2787–2795 (2013)
13. Lin, H., Liu, Y., Wang, W., Yue, Y., Lin, Z.: Learning entity and relation embeddings for knowledge resolution. Procedia Comput. Sci. **108**, 345–354 (2017)

14. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol, 1, Long Papers, pp. 687–696 (2015)
15. Xu, J., Qiu, X., Chen, K., Huang, X.: Knowledge graph representation with jointly structural and textual encoding. In: IJCAI International Joint Conference on Artificial Intelligence, pp. 1318–1324 (2017)
16. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: AAAI, pp. 2659–2665 (2016)
17. Vargas, M.R., de Lima, B.S.L.P, Evsukoff, A.G.: Deep learning for stock market prediction from financial news articles. In: IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), pp. 60–65 (2017)
18. Abadi, M., et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2016)
19. Chan, W.C.: Stock price reaction to news and no-news: Drift and reversal after headlines. J. Financ. Econ. **70**, 223–260 (2003)
20. Fehrer, R., Feuerriegel, S.: Improving decision analytics with deep learning: The case of financial disclosures, pp. 1–39 (2015)
21. Prosky, J., Song, X., Tan, A., Zhao, M.: Sentiment Predictability for Stocks (2017)
22. Weng, B., Ahmed, M.A., Megahed, F.M.: Stock market one-day ahead movement prediction using disparate data sources. Expert Syst. Appl. **79**, 153–163 (2017)
23. Lin, X., Yang, Z., Song, Y.: Expert systems with applications intelligent stock trading system based on improved technical analysis and echo state network. Expert Syst. Appl. **38**, 11347–11354 (2011)
24. Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing, pp. 347–354. Association for Computational Linguistics. pp. 347–354 (2005)
25. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space, pp. 1–12 (2013)
26. Kim, Y.: Convolutional Neural Networks for Sentence Classification (2014)
27. Kraus, M., Feuerriegel, S.: Decision support from financial disclosures with deep neural networks and transfer learning. Decis. Support Syst. **104**, 38–48 (2017)
28. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
29. Nikfarjam, A., Emadzadeh, E., Muthaiyah, S.: Text mining approaches for stock market prediction. In: The 2nd International Conference on Computer and Automation Engineering (ICCAE), vol. 4, pp. 256–260 (2010)
30. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, F-Score and ROC: a family of discriminant measures for performance evaluation. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1015–1021. Springer, Heidelberg (2006). https://doi.org/10.1007/11941439_114