

Modeling the Stock Relation with Graph Network for Overnight Stock Movement Prediction

Wei Li¹, Ruihan Bao^{2*}, Keiko Harimoto², Deli Chen¹, Jingjing Xu¹ and Qi Su¹

¹MOE Key Lab of Computational Linguistics, School of EECS, Peking University

²Mizuho Securities Co.,Ltd

liweitj47@pku.edu.cn, {ruihan.bao, keiko.harimoto}@mizuho-sc.com, {chendeli, jingjingxu, sukia}@pku.edu.cn

Abstract

Stock movement prediction is a hot topic in the *Fin-tech* area. Previous works usually predict the price movement in a daily basis, although the market impact of news can be absorbed much shorter, and the exact time is hard to estimate. **In this work, we propose a more practical objective to predict the overnight stock movement between the previous close price and the open price.** As no trading operation occurs after market close, the market impact of overnight news will be reflected by the overnight movement. One big obstacle for such task is the lacking of data, in this work we collect and publish the overnight stock price movement dataset of Reuters Financial News. Another challenge is that the **stocks in the market are not independent**,* which is omitted by previous works. To make use of the connection among stocks, we propose a LSTM Relational Graph Convolutional Network (LSTM-RGCN) model, which models the connection among stocks with their correlation matrix. Extensive experiment results show that our model outperforms the baseline models. Further analysis shows that the introduction of the graph enables our model to predict the movement of stocks that are not directly associated with news as well as the whole market, which is not available in most previous methods.¹

1 Introduction

Stock movement prediction is one of the most attractive topics in the *Fintech* area [Bollen *et al.*, 2011]. Many researches are devoted to predicting the movement trend of stocks based on news or historic market information. Researchers try to predict the stock price based on historic market data [Feng *et al.*, 2019], the stock related news [Hu *et al.*, 2018] or the combination of both [Xu and Cohen, 2018]. These researches all focus on predicting on the level of a trading day. However, it is a widely accepted fact that the stock movement is

highly stochastic and can be influenced by complicated factors [Malkiel, 1999]. Experts in the financial area agree that the time for the market to absorb the impact of news is uncertain, which ranges from a few minutes to hours, but usually less than a day. Therefore, using the news signal to predict the stock movement for the next day is not very reliable.

In this paper, we explore the prediction of the stock movement in a more practicable way. We propose to predict the **overnight stock movement** based on the overnight financial news. Overnight movement means the movement between the closing price of the previous day and the opening price of the next day. Only the news happened after the market being closed is considered. This way, the reaction of the market on the news can be more precisely reflected because there is no trading operation during the closing hours of the market.

When predicting the stock movement of a company, previous works only consider the news and market data of a single company. This omits the connection among related stocks. It is a common knowledge for market participants that the stock price of a company is often related to others that have business connection. For example, the stock of Toyota is related to the stock of Honda, because they are both in the automobile industry. Therefore, in this work, we propose to consider the information of related stocks when predicting the stock movement instead of treating them as isolated ones.

To represent the connection between two companies, we propose to adopt the correlation matrix among companies, which market participants often refer to. This correlation matrix is calculated based on the correlation of historic market data, which introduces very valuable information. Inspired by the success of graph neural networks, we propose a Long Short Term Memory Relational Graph Convolution Networks model (LSTM-RGCN) to represent the correlation among stocks. In the graph, each stock is a node, and the stock nodes are connected by the correlation between the two stocks filtered by a threshold.

To test the effectiveness of our proposed model, we collect and publish an overnight stock movement prediction dataset of Reuters Financial News, which is widely used in the financial industry. In the dataset, there are the financial news and market data from Reuters during 01-01-2013 to 09-28-2018 for Tokyo Stock Exchange (TSE). The experiment results show that our model outperforms various strong baseline models. Moreover, the introduction of the graph structure in-

*Contact Author

¹The code and dataset will be available in <https://github.com/liweitj47/overnight-stock-movement-prediction>

# News	Avg Len	Max Len	Min Len	Movement
363,929	72.8	262	4	376,414

Table 1: Statistic of our collected data. Movement means the number of price movements exceeding 0.5 hourly standard deviation.

deed helps predict the stock movement. Further analysis suggest that our model can infer the price movement of stocks which are not associated with any news and the whole market because of the graph representation.

We conclude our contributions as follows:

- We propose a more practical objective that aims to predict the overnight movement. One big obstacle in stock prediction is the lack of data, in this work we publish the corresponding dataset from a professional content provider Reuters Financial News.
- We propose to consider the connection among stocks when predicting the stock movement and propose a LSTM-RGCN model to represent the connection.
- Extensive experiment results show that our model outperforms all the baseline models. Further analysis suggests that the introduction of the graph makes our model able to infer the price movement of related stocks that do not have news as well as the whole market.

2 Task Formulation and Dataset

In this section, we describe the task formulation and the dataset. This task aims to predict the overnight stock movement as positive or negative given the overnight news. By overnight movement, we mean the movement between the opening price of the current trading day p_o^t and the closing price of the previous trading day p_c^{t-1} :

$$Movement = (p_o^t - p_c^{t-1}) / p_c^{t-1}$$

Because stock price is volatile in normal cases, we consider the movement as positive or negative only when it exceeds 0.5 times of hourly standard deviation of the stock movement.

By overnight news we mean the news that take place after the trading market being closed. We choose overnight news because the effect of normal news tends to be absorbed by the market within an hour or even few minutes during the trading hours of the day. On the contrary, the effect of the overnight news would be reflected on the overnight movement.

The dataset consists of the headline of the news and the target stock overnight movement. The news are associated with the stocks based on the “RIC” labels provided by Reuters. We choose the data during 2013-01-01 to 2018-09-28. Some statistics of the dataset is shown in Table 1.

3 Approach

In this section, we describe our Long Short Term Memory Relational Graph Convolutional Networks. Given the overnight news text of the stocks at one day, we want to predict the overnight stock price movement of the stocks that are attached with news. Our model first encodes news headline with a text encoder. Then we merge the news vector and the

node embedding as the node vector. After that, we feed the node vectors to the LSTM-RGCN to get the final representation of the node. Finally, we predict the stock movement based on the node representation in the graph.

3.1 Stock Correlation Graph

To model the correlation among stocks, we build a stock correlation graph. In the graph, each node represents a stock. Each node is attached with some news text data. The nodes are connected in reference to a correlation matrix, which is calculated based on the historic market price. The correlation matrix will be published with the dataset. The historic market price considers the market movement information. Therefore, this correlation matrix provides very valuable information about the inter-stock relation. The correlation values can be either positive (including 0) or negative. Therefore, we define two kinds of relationships between nodes depending on the polarity of the value, positively correlated (correlation \geq threshold) or negatively correlated (correlation $< -$ threshold). To reduce the noise of the correlation matrix, we connect the two nodes only when the absolute value of the correlation score is above a threshold in the matrix.

3.2 Node News Encoder

LSTM has been successfully applied in encoding the context information of text data. Therefore, we propose to encode the news headline of a node with LSTM:

$$h_w^t = LSTM(h_w^{t-1}, x_w^t) \quad (1)$$

where x_w is the word embedding, d^t is the t -th word token in the news, h_w^t is the hidden state of word d^t .

Since different words are not equally important in the news, we propose to represent the sentence with attention mechanism. We choose the stock embedding as the query and do attention on the hidden vectors of the news words:

$$s^t = softmax(W_s([x_s; h_w^t])) \quad (2)$$

$$h_n = sum(s^t \times h_w^t) \quad (3)$$

where x_s is the stock embedding of the node. W_s is a learnable parameter matrix, $[:,:]$ means concatenation of vectors.

To represent the node feature, we combine the news text vector h_n and the company embedding together:

$$v_v = W_v([h_n; x_s]) \quad (4)$$

where W_v is a learnable parameter matrix.

3.3 Graph Encoder

In this section, we describe our proposed LSTM RGCN based graph encoder.

GCN [Kipf and Welling, 2017] is able to model the graph structure, which is the correlation among stocks in this case. In our correlation matrix, there are two kinds of relationships representing positive and negative correlation relations. The original GCN is designed for the case where there is only one kind of relation. Therefore, we propose to adopt Relational Graph Convolutional Networks (RGCN) [Schlichtkrull *et al.*, 2018] to encode the graph structure:

$$N^{l+1} = \sigma(\sum_r D_r^{-\frac{1}{2}} A_r D_r^{-\frac{1}{2}} H^l W_r^l + W_h H^l) \quad (5)$$

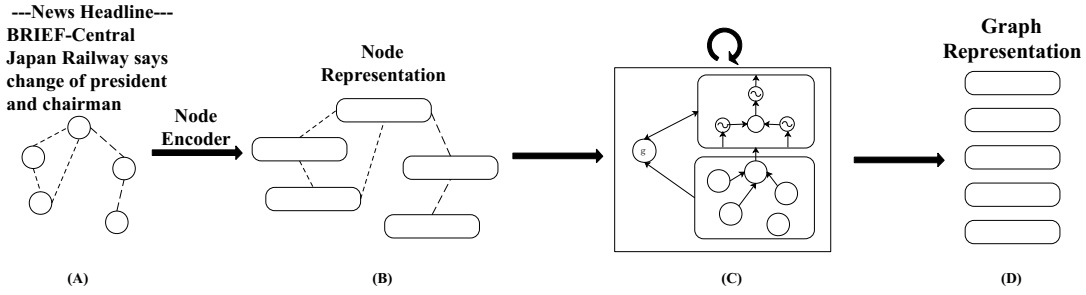


Figure 1: A brief description of our proposed LSTM-RGCN model. Each node in the graph represents one stock. A node can be attached with none or several overnight news text. The dashed lines indicate the two relations that connects stocks (A). The news is first encoded with the node feature encoder (B). Then the node embedding is fed into our proposed LSTM-RGCN model to make use of the correlation graph structure (C). Note that LSTM-RGCN can have multiple layers. Finally, the node vectors are used to predict the overnight stock price movement (D).

where A_r is the adjacency matrix of relation r , $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix. W_r^l is the learnable parameter matrix of the l -th layer for relation r . W_h is the learnable parameter matrix for the node vector. In our model, the parameter matrices are shared across layers. H^l represents the hidden representations of all the nodes in the l -th layer. N^{l+1} is the aggregated neighbor information for the $(l+1)$ -th layer.

Li et al. [2018] claim that GCN is vulnerable to the over-smoothing problem, which means that the value of different nodes would be very close after multiple layers of propagation. To alleviate this over-smoothing problem, we propose to add LSTM mechanism between RGCN layers so that the gate mechanism can dynamically select which part of the information should be transmitted to upper layers. Furthermore, we argue that the movement of one stock is related to the movement trend of the whole market. To model the movement trend of the whole market, we propose to add a global node to the graph, which can interact with each stock node. The LSTM process is calculated as follows:

$$i_i^l, f_i^l, o_i^l = f_{\theta^i}, f_{\theta^f}, f_{\theta^o}(h_i^{l-1}; x_i; g^{l-1}; N_i^l) \quad (6)$$

$$u = \tanh(W_u[h_i^{l-1}; x_i; g^{l-1}; N_i^l] + b_u) \quad (7)$$

$$c_i^l = f_i^l \odot c_i^{l-1} + i_i^{l-1} \odot u \quad (8)$$

$$h_i^l = o_i^l \odot \tanh(c_i^l) \quad (9)$$

where h_v is the aggregated vector calculated with the RGCN, f_{θ} is a one-layer feed forward network with sigmoid activation function and parameters θ . i, f, o indicate input, forget and output gates respectively. Different from the original design of LSTM, we also take the node embedding and v_v and global node vector g into consideration. V_v serves as the role similar to residual connection, while g can provide the information of the whole market.

3.4 Global Node

To calculate the hidden state of the global node, we first aggregate the hidden information of all the nodes with attentive

pooling:

$$\alpha_i = u(W_a h_i) \quad (10)$$

$$score_i = \frac{\exp(\alpha_i)}{\sum_j \exp(\alpha_j)} \quad (11)$$

$$\bar{h} = \sum_j score_j h_j \quad (12)$$

where W_a, u are learnable parameters.

Then, we use LSTM mechanism to filter the aggregated global information based on the hidden state of the global node in the previous layer and the updated node representations of the current layer:

$$\hat{f}_g^l, \hat{f}_i^l, o_g^l = f_{\theta^g}, f_{\theta^i}, f_{\theta^o}(g^{l-1}; h_i^{l-1}) \quad (13)$$

$$f_0^l, \dots, f_m^l, f_g^l = softmax(\hat{f}_0^l, \dots, \hat{f}_m^l, \hat{f}_g^l) \quad (14)$$

$$c_g^l = f_g^l \odot c_g^{l-1} + \sum_i f_i^l \odot c_i^{l-1} \quad (15)$$

$$g^l = o_g^l \odot \tanh(c_g^l) \quad (16)$$

where f_g, f_i, o_g are the forget gate, input gate and the output gate of the global node, respectively.

3.5 Objective

After we get the hidden state of each node in the graph, we can predict the movement label:

$$P = softmax(Wh) \quad (17)$$

$$loss = - \sum q \log(P) \quad (18)$$

where W is a learnable matrix, q is the gold label.

The task is modeled as a two-class classification problem. We use the standard cross entropy as the objective function.

4 Experiment

In this section, we describe the experiment setting, results and give detailed analysis.

4.1 Data

We choose the stocks within the *TPX500* and *TPX100* index.² Because the news data contains noisy news that do not influ-

²Tokyo Stock Price Index, commonly known as TOPIX or TPX, is an important stock market index for the Tokyo Stock Exchange. TPX500 and TPX100 are the indexes for the top 500 and 100 stocks in TPX.

Dataset	Node #	Valid Movement #		
		Train	Dev	Test
TPX500	498	16,190	1,055	1,171
TPX100	95	7,527	461	526

Table 2: Statistics of the dataset used in the experiment. Valid Movement # here means the number of movement that exceeds 0.5 hourly standard deviation and is attached with at least one piece of news.

ence the stock market, we first filter the news with the “RIC” label provided in the data by Reuters, which are the stock codes that the news may influence. Then we filter the news with some financial keywords described in the paper of Chen et al. [2019b]. Because much of the news is not related to the market price, we choose the keywords in the category of earnings, affairs, business, ratings and corporate. News that do not contain these keywords are filtered out. In the task, we only predict the movement when the both news is available and the price movement exceeds 0.5 times of hourly standard deviation. As the results, there are 10,367 positive movements and 8,050 negative ones in TPX500. There are 4,867 positive movements and 3,647 negative ones in TPX100. We choose the data in the period of 01-01-2018 ~ 04-30-2018 as the development set and the data in the period of 05-01-2018 ~ 09-30-2018 as the test set. Some details of the data are described in Table 2.

4.2 Baseline Models

In this part, we describe the baseline models.

- **Random**: this model is the random guess that randomly predict the movement to be improve or decline.
- **Random Forest** [Pagolu et al., 2016]: this model takes word embedding of the news headline as the input feature and applies Random Forest classifier³ to predict the movement label. The word embedding is learned with GloVe⁴ on the bloomberg news data.
- **Naive Bayes**: this model also takes the word embedding of the news headline as the input features, but applies Naive Bayes classifier⁵ to predict the movement label.
- **Linear Regression**: this model also takes the word embedding of the news headline as the input features, but applies linear logistic regression⁶ to predict the movement label.
- **Hierarchical Attention Networks (HAN)** [Yang et al., 2016]: a state of the art text classification model using hierarchical bidirectional LSTM structure with attentive pooling to encode the word and sentence. In our task, each headline is treated as a sentence in the HAN model.
- **S-LSTM** [Zhang et al., 2018]: a state of the art text representation model using LSTM to encode text. A global node is inserted to interact with each word.

³<https://scikit-learn.org/stable/modules/ensemble.html#forest>

⁴<https://nlp.stanford.edu/software/GloVe-1.2.zip>

⁵https://scikit-learn.org/stable/modules/naive_bayes.html#gaussian-naive-bayes

⁶https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

- **Transformer** [Vaswani et al., 2017]: a self-attention based model uses attention to encode context information of each word. A special “CLS” token is inserted in the front of the text, the hidden vector of which represents the whole text.

We use two kinds of word embeddings (GloVe and BERT) as the input features for the Random Forest, Naive Bayes and Linear regression three classifiers. For GloVe, we use the sum of the word embeddings. For BERT, we use the sentence vector. We do not use the sentence vector of BERT in our model because the vocabulary in the financial news headlines is very different from the vocabulary of the pre-trained BERT.

4.3 Setting

In the experiment, we set the layer number of S-LSTM and the proposed LSTM RGCN to be 3. The layer number of Transformer (baseline model) is 6. The headline length is truncated to 50. The maximum sentence number in hierarchical attention networks is truncated to 10. We set the threshold of correlation edge to 0.6, that is, only when the weight of the edge exceeds 0.6, there is an edge built between the two nodes. The embedding size of GloVe [Pennington et al., 2014] is 50. We use BERT (base) model to get the sentence vector, whose dimension is 768. We use Adam optimizer to train the model parameters. The learning rate is initially set to 0.001 and decayed by half after each iteration. The hidden size is 300.

4.4 Results

In Table 3 we show the experiment results. From the results we can see that our proposed model outperforms all other baseline models. The random guess generally results in an accuracy of around 50. Simple models can produce similar results compared with deep learning based baseline models. We assume that this is because the expression form of the financial news is relatively simple, which makes the deep learning based text classifiers do not have big advantage over the simple models.

Both the simple models and deep learning based baseline models do not perform as well as our proposed one. We argue that this is because the news in the market is still not enough to infer the movement of a stock. Even filtered with the topic keywords, there is still much noisy news that do not influence the price of the stock. Therefore, by introducing the information of relevant companies, our model can figure out the trend of the stock from the neighboring nodes and further validate the effect of the attached news.

4.5 The Effect of Graph

In Figure 2, we show the experiment result with and without the graph structure. From the figure we can see that the accuracy on both TPX500 and TPX100 increase by a big margin when adding the graph. We assume that this is because that the information from the related companies can supplement the news information of the current stock. Without the neighboring news, the model would suffer from information deficiency problem. Furthermore, by introducing the information of the related companies, our model can cross validate the effect of the news on the stock price.

Model	TPX500	TPX100
Random	50.34	50.55
Naive Bayes (G)	54.44	50.85
Naive Bayes (B)	44.66	41.63
Linear Regression (G)	54.86	49.91
Linear Regression (B)	52.35	52.09
Random Forest (G) [Pagolu <i>et al.</i> , 2016]	49.66	54.06
Random Forest (B)	51.75	50.19
HAN [Yang <i>et al.</i> , 2016]	54.35	54.63
Transformer [Vaswani <i>et al.</i> , 2017]	55.38	53.50
S-LSTM [Zhang <i>et al.</i> , 2018]	52.17	53.69
Proposal	56.14	58.71

Table 3: Experiment results (accuracy) on TPX500 and TPX100. Naive Bayes, Linear regression and random forest are traditional classification models using word embeddings as the features. “G” means using the sum of the GloVe word vectors, “B” means using the BERT sentence vectors. HAN, Transformer and S-LSTM are deep learning based models. Results show that our proposed model outperforms all the baseline models.

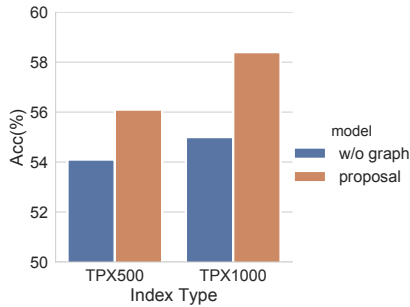


Figure 2: The effect of the graph structure in the model. “w/o graph” means our proposed model without graph structure. The results show that adding the graph structure can improve the accuracy of the model by a big margin.

Model	TPX500	TPX100
Random	50.34	50.55
Proposal	52.72	57.53

Table 4: Associative stock movement inference result. The price movement in this experiment do not have directly attached news. Other models cannot infer the movement of these stocks, because there is no available information.

4.6 Associative Inference

Because of the graph structure, our model is aware of the information of related companies. This makes our model able to learn the representation of a stock even though there is no directly attached news, which is realized by the information propagation via the graph edges. We call this ability the associative inference ability, which is the ability to predict the price movement of a stock where there is no attached news signal. In Table 4 we show the accuracy of associative inference. From the results we can observe that on both TPX500 and TPX100, our model can yield accuracy better than random guess (50%). The accuracy is especially high on TPX100, we assume that this is because the correlation among the big stocks provides more useful information. Other models cannot infer the movement of these stocks, be-

Topic	TPX500	TPX100
Full data	56.14	58.71
-ratings	54.40	57.70
-affairs	53.50	55.95
-corporate	53.05	57.05
-business	51.21	56.91
-earnings	55.21	56.90

Table 5: Experiment results on data eliminating news of each topic. For instance, “-ratings” means we do not use the keywords from the topic of “ratings”.

cause there is no available information.

4.7 Whole Index Inference

To test whether our model can capture the price movement of the whole market, we design an experiment that predicts the price movement of the whole TPX index. The training process remains the same, while during evaluation, we predict the index price movement P_{index} based on the global graph-level representation g (depicted in section 3.4).

$$P_{index} = softmax(Wg) \quad (19)$$

The prediction process is the same as ordinary stocks. The parameter W is shared with the ordinary prediction in Eqn. 17. We use the news of TPX500 stocks. The prediction accuracy is 55.74, which is rather satisfactory.

Actually, the prediction of the index price movement is also an attractive objective. However, it is hard to infer the index price because there is no directly attached news and the data is quite limited compared with ordinary stock. In this paper, we provide a view that predicts the market level price movement based on the global node in the graph. The global node is calculated with attentive pooling on the stock nodes, which gives the model the ability to dynamically select the important information from the stock nodes.

4.8 Effect of Different News Topics

In Table 5 we show the results of data eliminating news of one specific topic. In the experiment, we iteratively eliminate

the filtering keywords from one topic out of the five topics we use, which are “ratings, affairs, corporate, business and earnings”. From the results we can see that the topic of “affairs” generally has the biggest influence on the stock especially for the TPX100. We assume that this is because the content of news in the “affairs” topic is generally negative and the pattern is easier to catch. As long as the model can detect a negative affair happening to a company, its stock is likely to be predicted to decline. When consulting to the domain experts, we expect that news of “earnings” topic would have the biggest influence on the price. However, the results show that news of “earnings” only has medium influence on the price. We assume that this is because the pattern behind the “earnings” topic is more complicated, which involves the digit numbers, the comparison between the published earnings and the earning expectation from the market and so on. By now, this kind of problem is still very difficult to solve especially with very limited data.

5 Related Work

5.1 Stock Prediction

Traditional researches use human engineered features to predict stock price movement. Schumaker and Chen [2009] propose to use features like noun phrases and entities to predict the stock price. Oliveira et al. [2013] propose to use several sentiment indicators to predict stock market variables. Ding et al. [2015] further propose to use the result of information retrieval systems to predict the event-based stock price. Qin et al. [2017] and Feng et al. [2019] propose to predict the stock price trend based solely on the market data sequence. However, the stock market is highly stochastic, it is almost impossible to predict the stock movement with only the historic market price. Hu et al. [2018] propose to predict stock price movement based on sequential news. Xu and Cohen [2018] propose to use sequential tweets and market data to predict the stock movement. Apart from stock movement prediction, Chen et al. [2019a] propose a hierarchical framework to predict the Forex movement by grouping and summarizing a large amount of finance text.

5.2 Graph Neural Networks on Text Representation

Graph neural networks are designed for graph structured data, which have been widely applied in various NLP tasks. Yao et al. [2019] propose to organize documents and words into a unified heterogeneous graph and apply GCN to learn the document node representation, which are later used to predict the label of the documents. Peng et al. [2018] propose to organize the text into local text windows that can be easily modeled by convolution operations. These works organize the text with word co-occurrence information and focus on building the graph based on the local intra-sentence information. Yasunaga et al. [2017] and Sahu et al. [2019] propose to capture the local and non-local dependencies the graph constructed out of the inter- and intra-sentence dependencies. Yin et al. [2019] and Li et al. [2019] propose to model the inter-sentence dependency with the shared entities between sentences.

6 Conclusion

In this paper, we propose the objective of overnight stock price movement prediction, which aims to predict the overnight stock price movement based on the overnight news that take place during the stock market closing time. Because the stocks are not independent, we propose to consider the information of related companies by introducing the stock correlation matrix, which is built based on the historic market information. To make better use of the graph structure, we propose the LSTM-RGCN model that can handle both the positive and negative correlation. Furthermore, the LSTM module makes the proposed model less vulnerable to the over-smoothing problem, which is faced by many graph based neural networks. Experiment results show that our model outperforms the strong baselines. Moreover, our model can infer the price movement of stocks that are not attached with any news as well as the whole market.

Acknowledgements

This work is supported by a Research Grant from Mizuho Securities Co., Ltd. Mizuho Securities also provide very valuable domain experts suggestions. The experiment data is provided by Mizuho Securities and Reuters.

References

- [Bollen et al., 2011] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *J. Comput. Science*, 2(1):1–8, 2011.
- [Chen et al., 2019a] Deli Chen, Shuming Ma, Keiko Harimoto, Ruihan Bao, Qi Su, and Xu Sun. Group, extract and aggregate: Summarizing a large amount of finance news for forex movement prediction. In *2-th ECONLP*, 2019.
- [Chen et al., 2019b] Deli Chen, Yanyan Zou, Keiko Harimoto, Ruihan Bao, Xuancheng Ren, and Xu Sun. Incorporating fine-grained events in stock movement prediction. In *2-th ECONLP 2019*, 2019.
- [Ding et al., 2015] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2327–2333, 2015.
- [Feng et al., 2019] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. Enhancing stock movement prediction with adversarial training. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5843–5849, 2019.
- [Hu et al., 2018] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 261–269, 2018.

- [Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.
- [Li et al., 2018] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [Li et al., 2019] Wei Li, Jingjing Xu, Yancheng He, Shengli Yan, Yunfang Wu, and Xu Sun. Coherent comments generation for chinese articles with a graph-to-sequence model. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4843–4852. Association for Computational Linguistics, 2019.
- [Malkiel, 1999] Burton Gordon Malkiel. *A random walk down Wall Street: including a life-cycle guide to personal investing*. WW Norton & Company, 1999.
- [Oliveira et al., 2013] Nuno Oliveira, Paulo Cortez, and Nelson Areal. Some experiments on modeling stock market behavior using investor sentiment analysis and posting volume from twitter. In *3rd International Conference on Web Intelligence, Mining and Semantics, WIMS '13, Madrid, Spain, June 12-14, 2013*, page 31, 2013.
- [Pagolu et al., 2016] V. S. Pagolu, K. N. Reddy, G. Panda, and B. Majhi. Sentiment analysis of twitter data for predicting stock market movements. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pages 1345–1350, Oct 2016.
- [Peng et al., 2018] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference*, pages 1063–1072. International World Wide Web Conferences Steering Committee, 2018.
- [Pennington et al., 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [Qin et al., 2017] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2627–2633, 2017.
- [Sahu et al., 2019] Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. Inter-sentence relation extraction with document-level graph convolutional neural network. *arXiv preprint arXiv:1906.04684*, 2019.
- [Schlichtkrull et al., 2018] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 593–607, 2018.
- [Schumaker and Chen, 2009] Robert P. Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Trans. Inf. Syst.*, 27(2):12:1–12:19, 2009.
- [Vaswani et al., 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 6000–6010, 2017.
- [Xu and Cohen, 2018] Yumo Xu and Shay B. Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1970–1979, 2018.
- [Yang et al., 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.
- [Yao et al., 2019] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.
- [Yasunaga et al., 2017] Michihiro Yasunaga, Rui Zhang, Kshitij Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, 2017.
- [Yin et al., 2019] Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. Graph-based neural sentence ordering. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5387–5393, 2019.
- [Zhang et al., 2018] Yue Zhang, Qi Liu, and Linfeng Song. Sentence-state LSTM for text representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 317–327, Melbourne, Australia, July 2018. Association for Computational Linguistics.