# Stock Prediction from Financial News Knowledge Graph

Fu Di, Kelvin Koa Jun Long

April 2021

## 1 Introduction

Stock market prediction is a classical problem in the intersection of finance and computer science. To begin, the famous efficient-market hypothesis (EMH) states that stock prices reflect all currently available information, and any price changes that are not based on newly revealed information thus are inherently unpredictable [12]. As such, stock prediction research commonly uses news sentiment, given that news (especially Financial News) is a common source of new information on stocks, and there are proven technologies to extract sentiment from textual sources (e.g. Natural Language Processing, or NLP).

Early works on this topic used simple NLP models which capture named entities and term-level features (e.g. “Microsoft”, “sues”, “Barnes”, “Noble”) [16] to do sentiment analysis [14], obtaining optimistic results. After which, later works [3] have found that the use of structured events (e.g. Actor = Microsoft, Action = sues, Object = Barnes and Noble) as inputs produced better results for stock prediction, given that it considers the relations between entities. Finally, state-of-the-art methods [2] [6] have seen the use of Knowledge Graph techniques to learn from these structured events, which showed even better performance.

Following this trend, in this project, we proposed to develop a stock prediction model using Knowledge Graphs to predict the stock of several big companies (Apple Inc., Microsoft Corporation, etc.) using financial news scalped from Thomson Reuters and CNN News, as was done in previous works [11]. Given the previous stock data and related news, the model is supposed to classify stock movements (rise or fall) at each timestamp. Eventually, our model will be evaluated based on average prediction accuracy and F1 score.

## 2 Related Work

The most relevant paper that we will be following up on is Liu’s 2019 Paper, *Anticipating Stock Market of the Renowned Companies: A Knowledge Graph Approach* [11]. While most previous works deals with knowledge graphs that capture the relations between different stocks [10], Liu's paper works with event-driven knowledge graphs that capture relations between stocks and events. In this paper, events were captured from financial news headlines, which were then used to build a knowledge graph of stock-event tuples, e.g. *Chinese writers' group, sues, Apple*. TransE embeddings of the knowledge graph are then combined with Convolutional Neural Network embeddings of the input news and the stock data to form a feature vector, which is then fed into a dense layer to predict market prices for a few renowned companies, such as Apple and Samsung (see: Fig 1).

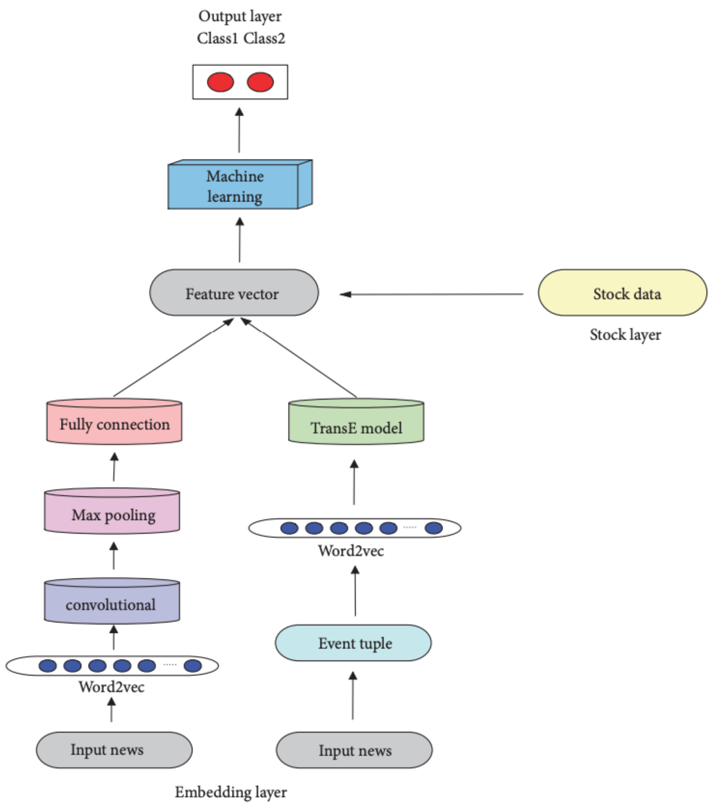


Figure 1: Overall Architecture for Liu's Paper

While the paper produced some positive results, there are still some limitations that could be addressed. Firstly, the highly specific form of the event tuple used means that many news headlines were filtered out because they could not fit in the tuple. For example, headlines such as *"What is next for Apple's board?"* were filtered out. This resulted in many wasted news articles, and a very sparse knowledge graph overall. Secondly, the accuracy of the TransE model alone is unknown, given that only results from the Feature Combine Model were shown (though there was a clear improvement over the Convolutional Neural

Network model). More experiments can be done to see how the TransE model performs alone and whether other knowledge graph embeddings models (TransH [17], TransR [9], etc.) can perform better.

### 2.1 Our Contributions

Based on Liu's work[11], our research aims to address these two limitations, by extracting more general forms of tuples for use in our Knowledge Graph and also comparing results across different combinations and types of embeddings models (LSTM, TransE, TransD, etc.) to look at their overall impact on the Stock Prediction task. Our main contribution in this project can be listed as follows:

- In the data processing, we proposed a different strategy to scalp news with the title that contains keywords related to the company stock from a keyword set (e.g. a proper keyword set for Apple Inc may be {'apple', 'technology', 'smart phone', 'WWDC', 'tablet, 'Tim Cook', }. Comparing with Liu's work [11] in which only news with the title contains 'apple' was selected, our strategy can scalp more useful news related to the stock. Moreover, to avoid costly manual labeling of news sentiment in Liu's work[11], we directly use universal sentence encoder[1] to extract embedding of the news headline.
- We propose alternative models word2vec+transR and word2vec+ transD to further improve the representation of event tuples extracted from the news. Here, we also compare results across different combinations of techniques to explore their impacts on the overall model.

## 3 Proposed Approach

Figure 2 shows the architecture of our proposed model. The steps are as following:

1. Corpus collection and stock data compilation: By looking up a keyword of a company such as Microsoft at Thomson Reuters or CNN website, we write a web-crawler to obtain the financial news and technical indicators of its stock within 10 years.
2. News headline pre-processing: Do the corpus analysis and extract event tuples from the news headline
3. Feature selection: extract structural event embedding features and context features from news headlines.
4. Prediction: selected features of the last five opening days was feed into the LSTM model to predict whether the stock price will increase or decrease the next day.

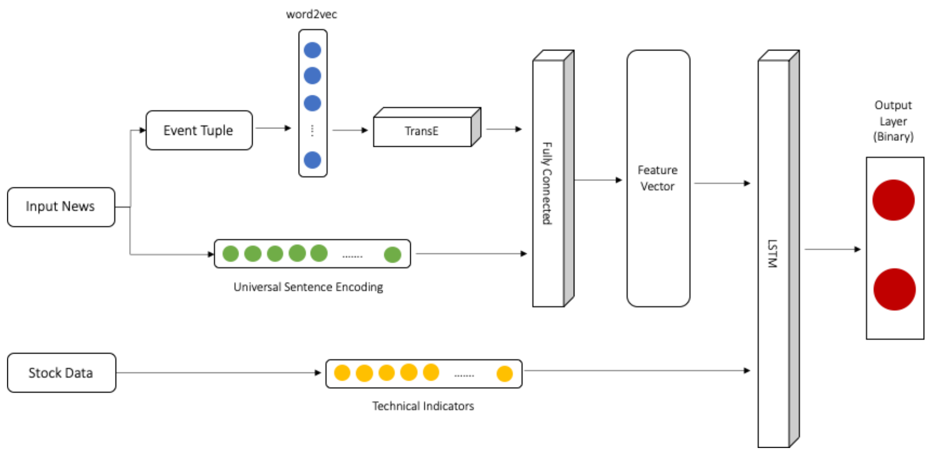


Figure 2: Proposed Architecture

Table 1: Collected News of each Company

| Financial news dataset | Stocks | Company | number of news headlines |
|---|---|---|---|
| Thomson Reuters | AAAL | Apple | 5816 |
| | MSFT | Microsoft | 5992 |
| CNN | BA | Boeing | 3886 |
| | GOOG | Google | 10701 |
| | WMT | Walmart | 3729 |

## 3.1 Dataset Description

### 3.1.1 News Data

Table 2 shows the custom financial news corpus built with headlines from Thomson Reuters or CNN website. It consists of financial news headlines published from 2019/09/01 to 2021/04/02, each news contains an English title and corresponding release date. Following Liu's work[10], we only use news title for text mining and event tuple extraction as the title usually contains the most important information in the news.

### 3.1.2 Stock Data

Stock data were collected from Yahoo Finance, within the same period as the financial news headlines, from 2019/09/01 to 2021/04/02. Daily trading data, consisting of Opening price, Closing price, High price, Low price, Volume were collected. Furthermore, some technical indicators were calculated from the data, as was done in the reference paper [11], and also proven in a previous paper [8] to improve stock forecasting results. These technical indicators are the Stochastic oscillator (%K), Larry William indicator (%R), and the Relative Strength Index

Table 2: Sample Event Tuples

| date | news headline | extracted event tuples |
|---|---|---|
| 2020-12-08 | Apple launches AirPods Max priced at $549 | ['Apple','launches','AirPods Max'] |
| 2018-10-02 | How Walmart is taking on Amazon | ['Walmart','is taking on','Amazon'] |
| 2016-07-26 | GM and Boeing push stocks into reverse | ['Boeng','push',"stock] |
| 2016-05-11 | Google bans all ads for payday loans | ['Google','bans','all ads for payday loans'] |
| 2012-01-10 | Microsoft sues China's Gome for alleged copyright infringement | ['Microsoft','sues','China's Gome'] |

(RSI). In total, there are 2411 trading days and 8 variables used as stock data for each company.

#### 3.1.3 Event Tuple Extraction

We define an event tuple as (A, P, O), where A represents an agent, P denotes a predicate, and O represents an object. We attempted both Reverb [5] and OpenIE [4] for event tuple extraction, the result shows OpenIE can extract more accurate event tuple and are used in the experiment. Table 2 shows example tuples extracted from the news titles. For example, 5816 headlines regarding Apple Inc. were extracted and then reduced to 4370 headlines after filtering via OpenIE[4]. Furthermore, the extracted event tuple is aligned with daily stock data based on the date to create input-output pairs.

### 3.2 Feature Extraction

#### 3.2.1 Feature Extraction from News text

As extracting the tuple from the original text may cause a loss of information, we also encode the text of the news headline. In Liu's work, textCNN[15] was trained with manually annotated sentiment tags to extract embedding of news headline sentences, however, manually annotation for thousands of news titles is time-consuming. Alternatively, we use the pre-trained universal sentence encoder[1] to extract meaningful embedding of news headline text. In the experiment, each news headline was embedded as a 512 dimension vector from the universal sentence encoder.

#### 3.2.2 Feature Extraction from the Event Tuple

To use the knowledge graph information from extracted event tuples, Given the event tuple $[H, R, T]$, we firstly extract vector representations of entity $h, t$ and relation $r$ by using pre-trained word2vec[13] model. Then we developed two different strategies to map the two entity vectors into the same relation space, which are shown in equation (1) and equation (2), $h_r$ and $t_r$ represent head entity embedding and tail entity embedding in the relation space after mapping. Here equation (1) was originally proposed in Liu's work[11] and inspired by TransR[9], while equation (2) is inspired by TransD [7] which deployed dynamically mapping matrix to map entity into relation space.

$$\begin{aligned} h_r &= \mathbf{W_r} h \\ t_r &= \mathbf{W_t} t \end{aligned} \tag{1}$$

$$\begin{aligned} h_r &= (\mathbf{r}_p \mathbf{h}_p^\top + \mathbf{I}) h \\ t_r &= (\mathbf{r}_p \mathbf{t}_p^\top + \mathbf{I}) t \end{aligned} \tag{2}$$

where $\mathbf{W_r}, \mathbf{W_t}$ and $\mathbf{r}_p, \mathbf{h}_p, \mathbf{t}_p$ are learnable parameter for two embedding model respectively.

As assumed by the translation based embedding model, the relationship vector r should satisfy $h_r + r \approx t_r$. Hence the loss of this structure model can be defined as:

$$E_s = \|h_r + r - t_r\|_2^2 \tag{3}$$

For the training of the model, we generate negative samples $\xi'$ by randomly replace head or tail entity in positive samples $\xi$ and use the margin-based loss as the objective for training to separate positive and negative samples:

$$L = \sum_{\xi} \sum_{\xi''} [\gamma + E_s(\xi') - E_s(\xi)]_+ \tag{4}$$

where $\gamma$ is margin and set to be 1 in the experiment, the process of training model takes minimization of the above objective with gradient descent approach.

The final structural embedding of the event tuple takes concatenation of triples in relation space, in our experiment we use pre-trained Google News corpus as word2vec model (`https://code.google.com/archive/p/word2vec/`), the total dimension of event tuple embedding is 900 (300 for each element in tuples). Further, if there exists multiple news headlines this day, we stack the embedding of all the news and use average pooling to get global embedding of all the news and reduce the dimension back to 900. On the other hand, if no news exists this day, the embedding is padded with zero vectors.

Finally, a learnable fully connected layer was built on top of the final event tuple embedding to reduce dimension before it is sent to Stock Prediction Model.

## 3.3 Stock Market Prediction Model

### 3.3.1 LSTM Layer

In addition to the improved features, a change that was made to the model was the inclusion of LSTM in the final layer. While the original model from Liu [10] uses a fully connected layer to predict the next day's stock movements using the previous day's data, our model uses an LSTM layer to learn from the past five days' data (both news + stock data), which allows us to capture temporal information of the past week.

### 3.3.2 Model Outputs

Given that this is a binary classification task, the cross-entropy loss was used to classify stock movements into binary classes: up or down. Additionally, accuracy and F1 scores were also used as metrics to compare performance across different models.

# 4 Experiment

## 4.1 Experiment Setting

In the experiments, to evaluate the influence of using both KG 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:

- LSTM: Only Stock Data is used for prediction
- KG_LSTM: both KG embedding and stock Data are used for the prediction
- Hybrid_LSTM: KG embedding, News headline embedding, and stock data are combined for prediction

Further, we extract two different embedding

The time step of the above models is one day, which predicts the next day using the data of the last 5 open days. Dataset is split into Training/Testing/Validation Data with a ratio of 60%/20%/20%. For the implementation of the model, the prediction model is written in Tensorflow, we set hidden state size and batch size of LSTM are set to be 16 and 32 respectively, the model was trained using Adam Optimizer with a learning rate of 0.005. The predictive performance on the test dataset was evaluated by the average prediction accuracy and F1-score in multiple experiments to reduce the influence of the variance.

## 4.2 Result and Discussion

We test both embedding methods and compare them with the baseline LSTM model, the result is shown in table 3 and table 4.

Table 3 shows the result of word2vec+TransR embedding method proposed by Liu[11], however, the result shows that adding Knowledge graph embedding of the news events and text embedding of news headline didn't significantly increase the performance of the LSTM model(see the result of hybrid_LSTM), which is contrary to Liu's result[11]. A possible explanation is that Liu[11] tags the label for each news headline (for example, whether this news has a positive/negative/neutral effect to the company) and use this tag to train its text embedding model in sentiment classification tasks so that after training, the representation of positive and negative news are well separated and benefit the training of LSTM. However, in our model, we use the pre-trained universal sentence encoder so that representations of positive and negative news are mixed and not separated from each other.

To prove this assumption, we randomly annotated 100 tags(positive or negative) and train a linear classifier to classify the tag given input of the sentence embedding from the universal sentence encoder. The prediction accuracy is low(0.59), this is as expected since universal sentence encoder are trained based on the similarity between sentence instead of difference of sentiment, two news headlines that are close to each other may embed entirely different sentiment.

Furthermore, we also find that event graph embedding along (see the result of KG_LSTM) cannot improve the performance of the stock prediction model, which is not mentioned in Liu's work[11](no related ablation study found in the paper). This is partially due to the sparsity of event triple extracted from the news, which is caused by two reasons:

1. the extracted event triple [agent, predicate, object] from OpenIE 5.1 is not clean enough since open information extraction model such as OpenIE 5.1 often cannot deal with complex sentence, for example, event tuple generated for news 'Google Maps adding new features, including augmented reality for (eventually) getting around airports and malls' is ['Google Maps', 'adding', 'new features including augmented reality for'], in which tail entity is too redundant and the predicate is not in Present tense. A good triple for this might be ['Google Maps', 'adding','new features'] or ['Google Maps', 'adding','augmented reality]. These redundancies in both head entity and head entity lead to the sparsity of entity.

2. the relation is also sparse, the total number of 4845 relations are extracted from 17480 tuples, with an average relation frequency is 3.6. This can be mitigated if all relations can be converted to the same tense (for example, 'is cutting' changed to 'cut', 'will pay' changed to 'pay', 'are' changed to 'is')

Currently, there is no large knowledge graph particularly built for financial news data. It would be interesting if such a knowledge graph could be built by scalping all related financial news and extract triple from the news headlines, and we can use this knowledge graph to reliably extract structure event embedding to facilitate prediction model training. Alternatively, when we do not have a large finical news knowledge graph, it might be beneficial if we could link the Small knowledge graph built from news data to a large database such as freebase to mitigate the sparsity issue.

Table 3: Prediction Accuarcy using word2vec+TransR embedding

| Model | accuarcy | F1 score |
|---|---|---|
| LSTM | 0.515569273 | 0.616648475 |
| KG_LSTM | 0.512208505 | 0.576920839 |
| Hybrid_LSTM | 0.527434842 | 0.660167846 |

Table 4: Prediction Accuarcy using word2vec+TransD embedding

| Model | accuarcy | F1 score |
|---|---|---|
| LSTM | 0.515569273 | 0.616648475 |
| KG_LSTM | 0.527572016 | 0.661617582 |
| Hybrid_LSTM | 0.527846365 | 0.627846365 |

# 5 Conclusion

In this project, we focused on the application of joint learning structural event embedding and sentence embedding for stock prediction. motivated by Liu's work[11], we developed new methods to capture more event tuples and attempted a new strategy of extracting corresponding knowledge graph embedding. However, in the experiment, we found event embedding alone cannot improve the performance of the stock prediction model likely due to the sparsity of relation and entity extracted from news embedding. Further, we show that extracted event embedding combined with pre-trained sentence embedding failed to capture sentiment information in news and unable to improve the performance of the stock prediction model as well, which indicates that in Liu's work [11] manually annotated sentiment tag is essential for capturing useful sentiment information that leads to the improvement of the prediction model. However, manually annotating sentiment tags for thousands of news is time-consuming, hence a semi-supervised/unsupervised sentiment analysis method should be explored in future work. On the other hand, we found knowledge graph (event triples) extracted from news headline is sparse, in the future, it would be beneficial if large financial news knowledge graph could be built or the extracted event knowledge graph be linked to a large database such as freebase to address the sparsity issue.

# References

[1] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[2] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1655–1658, 2018.

[3] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[4] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and

Mausam Mausam. Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10, 2011.

[5] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 1535–1545, 2011.

[6] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–30, 2019.

[7] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 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 (volume 1: Long papers)*, pages 687–696, 2015.

[8] Xiaowei Lin, Zehong Yang, and Yixu Song. Intelligent stock trading system based on improved technical analysis and echo state network. *Expert systems with Applications*, 38(9):11347–11354, 2011.

[9] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[10] Jue Liu, Zhuocheng Lu, and Wei Du. Combining enterprise knowledge graph and news sentiment analysis for stock price prediction. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[11] Yang Liu, Qingguo Zeng, Joaquín Ordieres Meré, and Huanrui Yang. Anticipating stock market of the renowned companies: A knowledge graph approach. *Complexity*, 2019, 2019.

[12] Burton G Malkiel. Efficient market hypothesis. In *Finance*, pages 127–134. Springer, 1989.

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[14] Anurag Nagar and Michael Hahsler. Using text and data mining techniques to extract stock market sentiment from live news streams. In *2012 International Conference on Computer Technology and Science*, volume 47, pages 91–95, 2012.

[15] A Rakhlin. Convolutional neural networks for sentence classification. *GitHub*, 2016.

[16] Robert P Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):1–19, 2009.

[17] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.