
News-Driven Stock Prediction With Attention-Based Noisy Recurrent State Transition

Xiao Liu¹, Heyan Huang¹, Yue Zhang^{2*}, Changsen Yuan¹

¹School of Computer Science and Technology, Beijing Institute of Technology

²School of Engineering, Westlake University

³Institute of Advanced Technology, Westlake Institute for Advanced Study

{xiaoliu, hhy63, yuanchangsen}@bit.edu.cn

yue.zhang@wias.org.cn

Abstract

We consider direct modeling of underlying stock value movement sequences over time in the news-driven stock movement prediction. A recurrent state transition model is constructed, which better captures a gradual process of stock movement continuously by modeling the correlation between past and future price movements. By separating the effects of news and noise, a noisy random factor is also explicitly fitted based on the recurrent states. Results show that the proposed model outperforms strong baselines. Thanks to the use of attention over news events, our model is also more explainable. To our knowledge, we are the first to explicitly model both events and noise over a fundamental stock value state for news-driven stock movement prediction.

1 Introduction

Stock movement prediction is a central task in computational and quantitative finance. With recent advances in deep learning and natural language processing technology, event-driven stock prediction has received increasing research attention [20, 4]. The goal is to predict the movement of stock prices according to financial news. Existing work has investigated news representation using bag-of-words [8], named entities [15], event structures [3] or deep learning [4, 21].

Most previous work focuses on enhancing news representations, while adopting a relatively simple model on the stock movement process, casting it as a simple response to a set of historical news. The prediction model can therefore be viewed as variations of a classifier that takes news as input and yields stock movement predictions. In contrast, work on time-series based stock prediction [9, 1, 21, 23], aims to capture continuous movements of prices themselves.

We aim to introduce underlying price movement trends into news-driven stock movement prediction by casting the underlying stock value as a recurrent state, integrating the influence of news events and random noise simultaneously into the recurrent state transitions. In particular, we take a LSTM with peephole connections [7] for modeling a stock value state over time, which can reflect the fundamentals of a stock. The influence of news over a time window is captured in each recurrent state transition by using neural attention to aggregate representations of individual news. In addition, all other factors to the stock price are modeled using a random factor component, so that sentiments, expectations and noise can be dealt with explicitly.

Compared with existing work, our method has three salient advantages. First, the process in which the influence of news events are absorbed into stock price changes is explicitly modeled. Though

*Corresponding author.

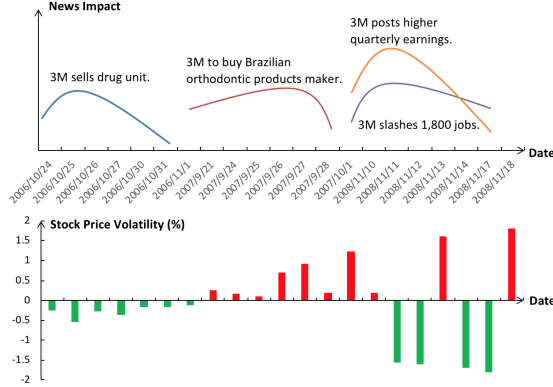


Figure 1: Example of news impacts on *3M Company*. Over the first and the second periods (from Oct. 24 to Nov. 1, 2006 and from Sep. 21 to Oct. 1, 2007), there was only one event. In the third period (from Nov. 10 to Nov. 18, 2008), there were two events affecting the stock price movements simultaneously.

previous work has attempted towards this goal [4], existing models predict each stock movement independently, only modeling the correlation between news in historical news sequences. As shown in Figure 1, our method can better capture a continuous process of stock movement by modeling the correlation between past and future stock values directly. In addition, non-linear compositional effects of multiple events in a time window can be captured.

Second, to our knowledge, our method allows noise to be explicitly addressed in a model, therefore separating the effects of news and other factors. In contrast, existing work trains a stock prediction model by fitting stock movements to events, and therefore can suffer from overfitting due to external factors and noise.

Third, our model is also more explainable thanks to the use of attention over news events, which is similar to the work of [2] and [22]. Due to the use of recurrent states, we can visualize past events over a large time window. In addition, we propose a novel future event prediction module to factor in likely next events according to natural events consequences. The future event module is trained over gold “future” data over historical events. Therefore, it can also deal with insider trading factors to some extent.

Experiments over the benchmark of [4] show that our method outperforms strong baselines, giving the best reported results in the literature. To our knowledge, we are the first to explicitly model both events and noise over a fundamental stock value state for news-driven stock movement prediction. Note that unlike time-series stock prediction models [24, 21], we do not take explicit historical prices as part of model inputs, and therefore our research still focuses on the influence of news information alone, and are directly comparable to existing work on news-driven stock prediction.

2 Related Work

There has been a line of work predicting stock markets using text information from daily news. We compare this paper with previous work from the following two perspectives.

Modeling Price Movements Correlation

Most existing work treats the modeling of each stock movement independently using bag-of-words [8], named entities [15], semantic frames [20], event structures [3], event embeddings [4] or knowledge bases [5]. Differently, we study modeling the correlation between past and future stock value movements.

There are also some work modeling the correlations between samples by sparse matrix factorization [19], hidden Markov model [23] and Bi-RNNs [21, 22] using both news and historical price data. Some work models the correlations among different stocks by pre-defined correlation graph [13] and

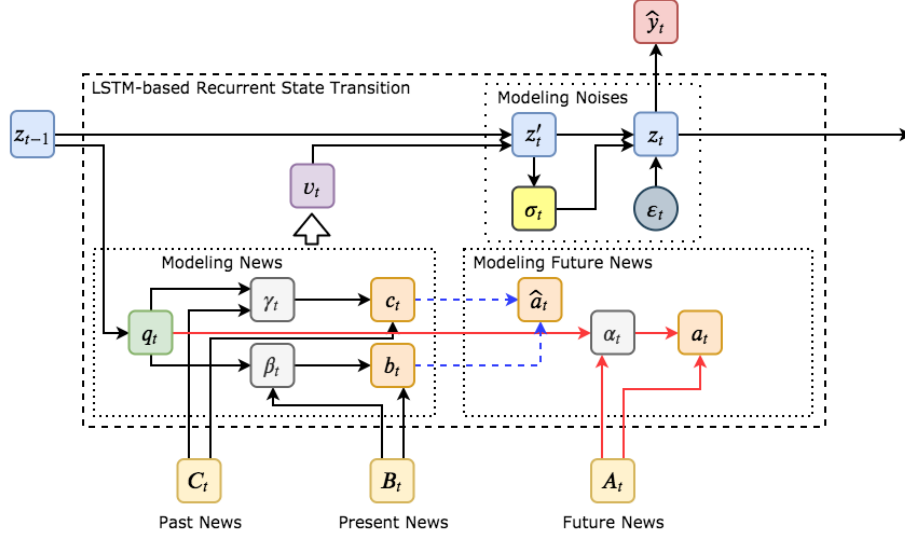


Figure 2: The ANRES model framework for trading day t in a trading sequence. The black solid elbows are used both in the training and the evaluating procedures. The red solid elbows are only used in the training procedure, while the blue dotted elbows in the evaluating procedure.

tensor factorization [24]. Our work is different from this line of work in that we use only news events as inputs, and our recurrent states are combined with impact-related noises.

Explainable Prediction

Rationalization is an important problem for news-driven stock price movement prediction, which is to find the most important news event along with the model’s prediction. Factorization, such as sparse matrix factorization [19] and tensor factorization [24], is a popular method where results can be traced back upon the input features. While this type of method are limited because of the dimension of input feature, our attention-based module has linear time complexity on feature size.

[22] apply dual-layer attention to predict the stock movement by using news published in the previous six days. Each day’s news embeddings and seven days’ embeddings are summed by the layer. Our work is different from [22] in that our news events attention is query-based, which is more strongly related to the noisy recurrent states. In contrast, their attention is not query-based and tends to output the same result for each day even if the previous day’s decision is changed.

3 Task Definition

Following previous work [3, 4], the task is formalized as a binary classification task for each trading day. Formally, given a history news set about a targeted stock or index, the input of the task is a trading day x and the output is a label $y \in \{+1, -1\}$ indicating whether the adjusted closing price p_x will be greater than p_{x-1} ($y = +1$) or not ($y = -1$).

4 Method

The framework of our model is shown in Figure 2. We explicitly model both events and noise over a recurrent stock value state, which is modeled using LSTM. For each trading day, we consider the news events happened in that day as well as the past news events using neural attention [18]. Considering the impacts of insider trading, we also involve future news in the training procedure. To model the high stochasticity of stock market, we sample an additive noise using a neural module. Our model is named attention-based noisy recurrent states transition (ANRES).

Considering the general principle of sample independence, building temporal connections between individual trading days in training set is not suitable for training [21] and we find it easy to overfit.

We notice that a LSTM usually takes several steps to generate a more stable hidden state. As an alternative method, we extended the time span of one sample to T previous continuous trading days $(t - T + 1, t - T + 2, \dots, t - 1, t)$, which we call a trading sequence, is used as the basic training element in this paper.

4.1 LSTM-based Recurrent State Transition

ANRES uses LSTM with peephole connections [7]. The underlying stock value trends are represented as a recurrent state z transited over time, which can reflect the fundamentals of a stock. In each trading day, we consider the impact of corresponding news events and a random noise as:

$$\begin{aligned} z'_t &= \overrightarrow{\text{LSTM}}(v_t, z_{t-1}) \\ z_t &= f(z'_t) \end{aligned}$$

where v_t is the news events impact vector on the trading day t and f is a function in which random noise will be integrated.

By using this basic framework, the non-linear compositional effects of multiple events can also be captured in a time window. Then we use the sequential state z_t to make binary classification as:

$$\begin{aligned} \hat{p}_t &= \text{softmax}(W^y z_t) \\ \hat{y}_t &= \arg \max_{i \in \{+1, -1\}} \hat{p}_t(\hat{y}_t = i | x_t) \end{aligned}$$

where \hat{p}_t is the estimated probabilities, \hat{y}_t is the predicted label and x_t is the input trading day.

4.2 Modeling News Events

For a trading day t in a trading sequence, we model both long-term and short-term impact of news events. For short-term impact, we use the news published after the previous trading day $t - 1$ and before the trading day t as the present news set. Similarly, for long-term impact, we use the news published no more than thirty calendar days ago as the past news set.

For each news event, we extract its headline and use ELMo [14] to transform it to V -dim hidden state by concatenating the output bidirectional hidden states of the last words as the basic representation of a news event. By stacking those vectors accordingly, we obtain two embedding matrices C'_t and B'_t for the present and past news events as:

$$\begin{aligned} ec_t^i &= \overleftarrow{\text{ELMo}}(hc_t^i), i \in \{1, 2, \dots, L_c\} \\ eb_t^j &= \overleftarrow{\text{ELMo}}(hb_t^j), j \in \{1, 2, \dots, L_b\} \\ C'_t &= \text{stack}(\{ec_t^1, ec_t^2, \dots, ec_t^{L_c}\}) \\ B'_t &= \text{stack}(\{eb_t^1, eb_t^2, \dots, eb_t^{L_b}\}) \end{aligned}$$

where hc_t^i is one of the news event headline in the present news set, ec_t^i is the headline representation of hc_t^i , L_c is the size of present news set; while hb_t^j , eb_t^j and L_b are for the past news set.

To make the model more numerically stable and avoiding overfitting, we apply the over-parameterized component of [12] to the news events embedding matrices, where

$$\begin{aligned} C_t &= \sigma(W^f C'_t) \odot \tanh(W^c C'_t) \\ B_t &= \sigma(W^f B'_t) \odot \tanh(W^c B'_t) \end{aligned}$$

\odot is element-wise multiplication and $\sigma(\cdot)$ is the sigmoid function.

Due to the unequal importance news events contribute to the stock price movement in t , we use scaled dot-product attention [18] to capture the influence of news over a period for the recurrent state transition. In practical, we first transform the last trading day's stock value z_{t-1} to a query vector q_t ,

and then calculate two attention score vectors γ_t and β_t for the present and past news events as:

$$\begin{aligned} q_t &= \tanh(W^q z_{t-1}) \\ \gamma_t &= \text{softmax}\left(\frac{C_t^i q_t}{\sqrt{V}}\right) \\ \beta_t &= \text{softmax}\left(\frac{B_t^i q_t}{\sqrt{V}}\right) \end{aligned}$$

We sum the news events embedding matrices to obtain news events impact vectors c_t and b_t on the trading day t according to the weights γ_t and β_t , respectively:

$$\begin{aligned} c_t &= \tanh\left(\sum_{i=1}^{N_t} \gamma_t^i C_t^i\right) \\ b_t &= \tanh\left(\sum_{i=1}^{N_t} \beta_t^i B_t^i\right) \end{aligned}$$

4.3 Modeling Future News

In spite of the long-term and short-term impact, we find that some short-term future news events will exert an influence on the stock price movement before the news release, which can be attributed to news delay or insider trading [16] factors to some extent.

We propose a novel future event prediction module to consider likely next events according to natural consequences. In this paper, we define future news events as those that are published within seven calendar days after the trading day t .

Similarly to the past and present news events, we stack the headline ELMo embeddings of future news events to an embedding matrix A'_t . Then adapting the over-parameterized component and summing the stacked embedding vectors by scaled dot-product attention. We calculate the future news events impact vector a_t on the trading day t as:

$$\begin{aligned} A_t &= \sigma(W^f A'_t) \cdot \tanh(W^c A'_t) \\ \alpha_t &= \text{softmax}\left(\frac{A_t^i q_t}{\sqrt{V}}\right) \\ a_t &= \tanh\left(\sum_{i=1}^{N_t} \alpha_t^i A_t^i\right) \end{aligned}$$

Although the above steps can work in the training procedure, where the future event module is trained over gold “future” data over historical events, at test time, future news events are not accessible. To address this issue, we use a non-linear transformation to estimate a future news events impact vector \hat{a}_t with the past and present news events impact vectors b_t and c_t as:

$$\hat{a}_t = \tanh(W^a [c_t, b_t])$$

where $[,]$ is the vector concatenation operation.

We concatenate the above-mentioned three types of news events impact vectors to obtain the input v_t for LSTM-based recurrent state transition on trading day t as:

$$v_t = \begin{cases} [c_t, b_t, a_t], & \text{when training} \\ [c_t, b_t, \hat{a}_t], & \text{when evaluating} \end{cases}$$

where $[,]$ is the vector concatenation operation.

4.4 Modeling Noise

In this model, all other factors to the stock price such as sentiments, expectations and noise are explicitly modeled as noise using a random factor. We sample a random factor from a normal distribution $\mathcal{N}(\mathbf{0}, \sigma_t)$ parameterized by z'_t as:

$$\sigma_t = \sqrt{\exp(\tanh(W^\sigma z'_t))}$$

	Training	Development	Test
#documents	358,122	96,299	99,030
#samples	1,425	169	191
time span	10/20/2006- 06/18/2012	06/19/2012- 02/21/2013	02/22/2013- 11/21/2013

Table 1: Statistics of the datasets.

However, in practice, the model can face difficulty of back propagating gradients if we directly sample a random factor from $\mathcal{N}(\mathbf{0}, \sigma_t)$. We use re-parameterization [17] for normal distributions to address the problem and enhance the transition result z'_t with sample random factor to obtain the noisy recurrent state z_t as:

$$\begin{aligned}\epsilon_t &\sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \\ z_t &= \tanh(z'_t + \sigma_t \epsilon_t)\end{aligned}$$

4.5 Training Objective

For training, there are two main terms in our loss function. The first term is a cross entropy loss for the predicted probabilities \hat{p}_t and gold labels y_t , and the second term is the mean squared error between the estimated future impact vector \hat{a}_t and the true future impact vector a_t .

The total loss for a trading sequence containing T trading days with standard L_2 regularization is calculated as:

$$\begin{aligned}L_{ce} &= \sum_{t=1}^T -\log(1 - \hat{p}_t(y_t|x_t)) \\ L_{mse} &= \frac{1}{V} \sum_{t=1}^T \sum_{i=1}^V (\hat{a}_t^i - a_t^i)^2 \\ L_{total} &= L_{ce} + \theta L_{mse} + \lambda \|\Phi\|_2^2\end{aligned}$$

where θ is a hyper-parameter which indicates how much important L_{mse} is comparing to L_{ce} , Φ is the set of trainable parameters in the entire ANRES model and λ is the regularization weight.

5 Experiments

We use the public financial news dataset released by [3], which is crawled from Reuters and Bloomberg over the period from October 2006 to November 2013. We conduct our experiments on predicting the Standard & Poor’s 500 stock (S&P 500) index and its selected individual stocks, obtaining indices and prices from Yahoo Finance². Detailed statistics of the training, development and test sets are shown in Table 1. We report the final results on test set after using development set to tune some hyper-parameters.

5.1 Settings

The hyper-parameters of our ANRES model are shown in Table 2. We use mini-batches and stochastic gradient descent (SGD) with momentum to update the parameters. Most of the hyper-parameters are chosen according to development experiments, while others like dropout rate r and SGD momentum μ are set according to common values.

Following previous work [20, 3, 21], we adopt the standard measure of accuracy and Matthews Correlation Coefficient (MCC) to evaluate S&P 500 index prediction and selected individual stock prediction. MCC is applied because it avoids bias due to data skew. Given the confusion matrix which contains true positive, false positive, true negative and false negative values, MCC is calculated as:

$$\text{MCC} = \frac{\text{tp} \times \text{tn} - \text{fp} \times \text{fn}}{\sqrt{(\text{tp} + \text{fp})(\text{tp} + \text{fn})(\text{tn} + \text{fp})(\text{tn} + \text{fn})}}$$

²<https://finance.yahoo.com/>

Name	Value
batch size	16
learning rate lr	0.005
SGD momentum μ	0.9
dropout rate r	0.3
MSE loss weight θ	0.4
regularization weight λ	0.0005
news embedding dimension V	256
recurrent state dimension D	100
trading sequence length T	7

Table 2: Hyper-parameters setting.

	Accuracy	MCC
ANRES_Sing_R	62.91%	0.3704
ANRES_Sing_Z	63.63%	0.3672
ANRES_Seq_R	67.94%	0.5141
ANRES_Seq_Z	68.51%	0.5392

Table 3: Development set results on initializing the noisy recurrent states.

5.2 Initializing Noisy Recurrent States

As the first set of development experiments, we try different ways to initialize the noisy recurrent states of our ANRES model to find a suitable approach. For each trading day, we compare the results whether states transitions are modeled or not. Besides, we also compare the methods of random initialization and zero initialization. Note that the random initialization method we use here returns a tensor filled with random numbers from the standard normal distribution $\mathcal{N}(0, 1)$. In summary, the following four baselines are designed:

- *ANRES_Sing_R*: randomly initializing the states for each single trading day.
- *ANRES_Sing_Z*: initializing the states as zeros for each single trading day.
- *ANRES_Seq_R*: randomly initializing the first states for each trading sequence only.
- *ANRES_Seq_Z*: initializing the first states as zeros for each trading sequence only.

Development set results on predicting S&P 500 index are shown in Table 3. We can see that modeling recurrent value sequences performs better than treating each trading day separately, which shows that modeling trading sequences can capture the correlations between trading days and the non-linear compositional effects of multiple events. From another perspective, the models *ANRES_Sing_R* and *ANRES_Sing_Z* also represent the strengths of our basic representations of news events in isolation. Therefore, we can also see that using only the basic news events representations is not sufficient for index prediction, while combining with our states transition module can achieve strong results.

By comparing the results of *ANRES_Seq_R* and *ANRES_Seq_Z*, we decide to use zero initialization for our ANRES models, including the noisy recurrent states also in the remaining experiments.

5.3 Study on Trading Sequence Length

We use the development set to find a suitable length T for trading sequence, which is searched from $\{1, 3, 5, 7, 9, 11, 13, 15\}$. The S&P 500 index prediction results of accuracy, MCC and consumed minutes per training epoch on the development set are shown in Figure 3.

We can see that the accuracy and MCC are positively correlated with the growth of T , while the change of accuracy is smaller than MCC. When $T \geq 7$, the growth of MCC becomes slower than that when $T < 7$. Also considering the running time per training epoch, which is nearly linear w.r.t. T , we choose the hyper-parameter $T = 7$ and use it in the remaining experiments.

5.4 Predicting S&P 500 Index

We compare our approach with the following strong baselines on predicting the S&P 500 index, which also only use financial news:

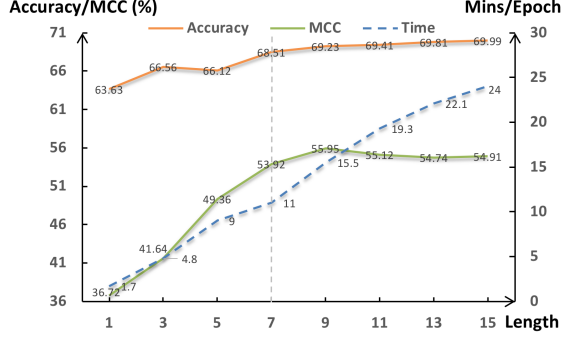


Figure 3: Development set results of different trading sequence length T .

	Accuracy	MCC
[11]	56.38%	0.0711
[4]	64.21%	0.4035
[5]	66.93%	0.5072
[6]	63.34%	-
[10]	64.55%	-
ANRES	67.34%	0.5475

Table 4: Test set results on predicting S&P 500 index.

- [11] uses bags-of-words to represent news documents, and constructs the prediction model by using Support Vector Machines (SVMs).
- [4] uses event embeddings as input and convolutional neural network prediction model.
- [5] empowers event embeddings with knowledge bases like YAGO and also adopts convolutional neural networks as the basic prediction framework.
- [6] uses fully connected model and character-level embedding input with LSTM to encode news texts.
- [10] uses recurrent neural networks with skip-thought vectors to represent news text.

Table 4 shows the test set results on predicting the S&P 500 index. From the table we can see that our ANRES model achieves the best results on the test sets. By comparing with [11], we can find that using news event embeddings and deep learning modules can be better representative and also flexible when dealing with high-dimension features.

When comparing with [4] and the knowledge-enhanced [5], we find that extracting structured events may suffer from error propagation. And more importantly, modeling the correlations between trading days can better capture the compositional effects of multiple news events.

By comparing with [6] and [10], despite that modeling the correlations between trading days can bring better results, we also find that modeling the noise by using a state-related random factor may be effective because of the high market stochasticity.

5.5 Ablation Study on News and Noise

We explore the effects of different types of news events and the introduced random noise factor with ablation on the test set. More specifically, we disable the past news, the present news, future news and the noise factor, respectively. The S&P 500 index prediction results of the ablated models are shown in Table 5. First, without using the past news events, the result becomes the lowest. The reason may be that history news contains the biggest amount of news events. In addition, considering the trading sequence length and the time windows of future news, if we disable the past news, most of them will not be involved in our model at any chance, while the present or the past news will be input on adjacent trading days.

Second, it is worth noticing that using the future news events is more effective than using the present news events. On the one hand, it confirms the importances to involve the future news in our ANRES

	Acc	MCC
w/o Past News	62.17%	0.4421
w/o Present News	64.73%	0.4823
w/o Future News	64.58%	0.4781
w/o Noise	63.90%	0.4608
ANRES	67.34%	0.5475

Table 5: Test set results of ablation study.

Stock	Sector	Company News			Sector News			All News	
		#docs	Accuracy	MCC	#docs	Accuracy	MCC	Accuracy	MCC
Apple	IT	2,398	69.21%	0.5632	12,812	64.35%	0.3861	56.14%	0.2355
Citigroup	Financials	2,058	63.57%	0.5193	117,659	56.29%	0.3021	55.15%	0.1852
Boeing Company	Industrials	1,870	66.25%	0.4423	17,969	61.35%	0.2719	57.23%	0.1824
Google	Communication	1,762	66.13%	0.3717	13,344	60.47%	0.2644	58.41%	0.1387
Wells Fargo	Financials	845	61.64%	0.3944	117,659	57.34%	0.1294	54.64%	0.0823

Table 6: Test set results of individual stock price movement prediction.

model, which can deal with insider trading factors to some extent. On the other hand, the reason may be the news impact redundancy in sequence, as the future news impact on the $t - 1$ -th day should be transited to the t -th day to compensate the absent loss of the present news events.

The effect of modeling the noise factor is lower only to modeling the past news events, but higher than the other ablated models, which demonstrates the effectiveness of the noise factor module. We think the reason may because that modeling such an additive noise can separate the effects of news event impacts from other factors, which makes modeling the stock price movement trends more clearly.

5.6 Predicting Individual Stock Movements

Other than predicting the S&P 500 index, we also investigate the effectiveness of our approach on the problem of individual stock prediction using the test set. We count the amounts of individual company related news events for each company by name matching, and select five well known companies with sufficient news, *Apple*, *Citigroup*, *Boeing Company*, *Google* and *Wells Fargo* from four different sectors, which is classified by the Global Industry Classification Standard. For each company, we prepare not only news events about itself, but also news events about the whole companies in the

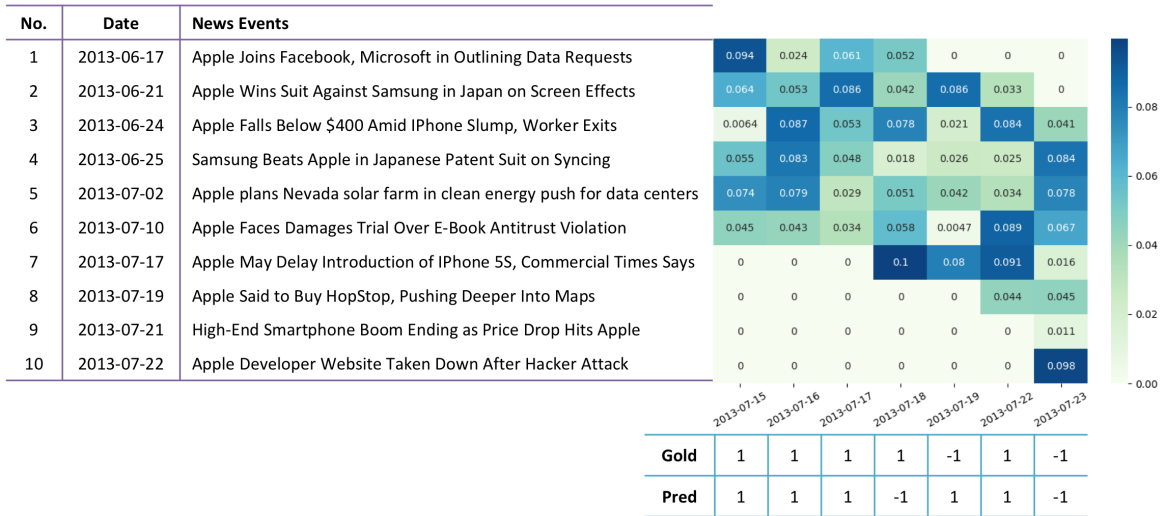


Figure 4: Attention visualization and test set results comparison of the trading sequence [07/15/2013, 07/23/2013] when predicting Apple Inc.'s stock price movements using only company news.

sector. We use company news, sector news and all financial news to predict individual stock price movements, respectively. The experimental results and news statistics are listed in Table 6.

The result of individual stock prediction by only using company news dramatically outperforms that of sector news and all news, which presents a negative correlation between total used amounts of news events and model performance. The main reason maybe that company-related news events can more directly affect the volatility of company shares, while sector news and all news contain many irrelevant news events, which would obstruct our ANRES model’s learning the underlying stock price movement trends.

Note that [4, 5] and [22] also reported results on individual stocks. But we cannot directly compare our results with them because the existing methods used different individual stocks on different data split to report results, and [4, 5] reported only development set results. This is reasonable since the performance of each model can vary from stock to stock over the S&P 500 chart and comparison over the whole index is more indicative.

5.7 Case Study

To look into what news event contributes the most to our prediction result, we further analyze the test set results of predicting *Apple Inc.*’s stock price movements only using company news, which achieves the best results among the five selected companies mentioned before.

As shown in Figure 4, we take the example trading sequence from 07/15/2013 to 07/23/2013 for illustration. The table on the left shows the selected top-ten news events, while attention visualization and results are shown on the right chart. Note that there are almost fifty different past news events in total for the trading sequence, and the news events listed on the left table are selected by ranking attention scores from the past news events, which are the most effective news according to the ablation study. There are some zeros in the attention heat map because these news do not belong to the corresponding trading days.

We can find that the news event No. 1 has been correlated with the stock price rises on 07/15/2013, but for the next two trading days, its impact fades out. On 07/18/2013, the news event No. 7 begins to show its impact. However, our ANRES model pays too much attention in it and makes the incorrect prediction that the stock price decreases. On the next trading day, our model infers that the impact of the news event No. 2 is bigger than that of the news event No. 7, which makes an incorrect prediction again. From these findings, we can see that our ANRES model tends to pay more attention to a new event when it first occurs, which offers us a potential improving direction in the future.

6 Conclusion

We investigated explicit modeling of stock value sequences in news-driven stock prediction by using an LSTM state to model the fundamentals, adding news impact and noise impact by using attention and noise sampling, respectively. Results show that our method is highly effective, giving the best performance on a standard benchmark. To our knowledge, we are the first to explicitly model both events and noise over a fundamental stock value state for news-driven stock movement prediction.

References

- [1] Yakov Amihud. Illiquidity and stock returns: cross-section and time-series effects. *Journal of Financial Markets*, 5(1):31–56, 2002.
- [2] Ching-Yun Chang, Yue Zhang, Zhiyang Teng, Zahn Bozanic, and Bin Ke. Measuring the information content of financial news. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 3216–3225, 2016.
- [3] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1415–1425, 2014.
- [4] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 2327–2333, 2015.

- [5] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. Knowledge-driven event embedding for stock prediction. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 2133–2142, 2016.
- [6] Leonardo dos Santos Pinheiro and Mark Dras. Stock market prediction with deep learning: A character-based neural language model for event-based trading. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 6–15, 2017.
- [7] Felix A. Gers and Jürgen Schmidhuber. Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000, Neural Computing: New Challenges and Perspectives for the New Millennium*, pages 189–194, 2000.
- [8] Shimon Kogan, Dmitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. Predicting risk from financial reports with regression. In *Proceedings of the 2009 Conference of the North American Chapter of the Association of Computational Linguistics: Human Language Technologies*, pages 272–280, 2009.
- [9] Ross Levine and Sara Zervos. Stock market development and long-run growth. *The World Bank Economic Review*, 10(2):323–339, 1996.
- [10] Peikang Lin, Xianjie Mo, Guidong Lin, Liwen Ling, Tingting Wei, and Wei Luo. A news-driven recurrent neural network for market volatility prediction. In *4th IAPR Asian Conference on Pattern Recognition*, pages 776–781, 2017.
- [11] Ronny Luss and Alexandre D’Aspremont. Predicting abnormal returns from news using text classification. *Quantitative Finance*, 15(6):999–1012, 2015.
- [12] Stephen Merity. Single headed attention RNN: stop thinking with your head. *CoRR*, abs/1911.11423, 2019.
- [13] Yangtuo Peng and Hui Jiang. Leverage financial news to predict stock price movements using word embeddings and deep neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 374–379, 2016.
- [14] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237, 2018.
- [15] 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*, 27(2):12:1–12:19, 2009.
- [16] H. Nejat Seyhun. Why does aggregate insider trading predict future stock returns. *The Quarterly Journal of Economics*, 107(4):1303–1331, 1992.
- [17] Akash Srivastava and Charles A. Sutton. Autoencoding variational inference for topic models. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- [18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008, 2017.
- [19] Felix Ming Fai Wong, Zhenming Liu, and Mung Chiang. Stock market prediction from WSJ: text mining via sparse matrix factorization. In *Proceedings of the 2014 IEEE International Conference on Data Mining*, pages 430–439, 2014.
- [20] Boyi Xie, Rebecca J. Passonneau, Leon Wu, and Germán Creamer. Semantic frames to predict stock price movement. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 873–883, 2013.

- [21] 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*, pages 1970–1979, 2018.
- [22] Linyi Yang, Zheng Zhang, Su Xiong, Lirui Wei, James Ng, Lina Xu, and Ruihai Dong. Explainable text-driven neural network for stock prediction. *CoRR*, abs/1902.04994, 2019.
- [23] Xi Zhang, Yixuan Li, Senzhang Wang, Binxing Fang, and Philip S. Yu. Enhancing stock market prediction with extended coupled hidden markov model over multi-sourced data. *Knowledge and Information Systems*, pages 1–20, 2018.
- [24] Xi Zhang, Yunjia Zhang, Senzhang Wang, Yuntao Yao, Binxing Fang, and Philip S. Yu. Improving stock market prediction via heterogeneous information fusion. *Knowledge-Based Systems*, 143:236–247, 2018.