Incorporating Fine-grained Events in Stock Movement Prediction

Deli Chen[1]*, Yanyan Zou[2], Keiko Harimoto[3], Ruihan Bao[3], Xuancheng Ren[1], Xu Sun[1]

[1]MOE Key Lab of Computational Linguistics, School of EECS, Peking University
[2]StatNLP Research Group, Singapore University of Technology and Design
[3]Mizuho Securities Co., Ltd.
{chendeli,renxc,xusun}@pku.edu.cn, yanyan_zou@mymail.sutd.edu.sg,
{keiko.harimoto,ruihan.bao}@mizuho-sc.com

Abstract

Considering event structure information has proven helpful in text-based stock movement prediction. However, existing works mainly adopt the coarse-grained events, which loses the specific semantic information of diverse event types. In this work, we propose to incorporate the fine-grained events in stock movement prediction. Firstly, we propose a professional finance event dictionary built by domain experts and use it to extract fine-grained events automatically from finance news. Then we design a neural model to combine finance news with fine-grained event structure and stock trade data to predict the stock movement. Besides, in order to improve the generalizability of the proposed method, we design an advanced model that uses the extracted fine-grained events as the distant supervised label to train a multi-task framework of event extraction and stock prediction. The experimental results show that our method outperforms all the baselines and has good generalizability.

1 Introduction

Stock movement plays an important role in economic activities, so the prediction of stock movement has caught a lot of attention of researchers. In recent years, employing the stock related text (such as finance news or tweets) has become the mainstream (Si et al., 2014; Ding et al., 2015; Li et al., 2015; Alostad and Davulcu, 2017; Zhong et al., 2017; Zhang et al., 2018a) of stock movement prediction task. In these text-based stock prediction works, various methods are proposed to extract semantic information from stock related text to help the prediction of stock movement. There are mainly two methods of applying text: employing raw text (Hu et al., 2018; Xu and Cohen, 2018) or coarse-grained <S,P,O> structure (subject, predicate and object) extracted from

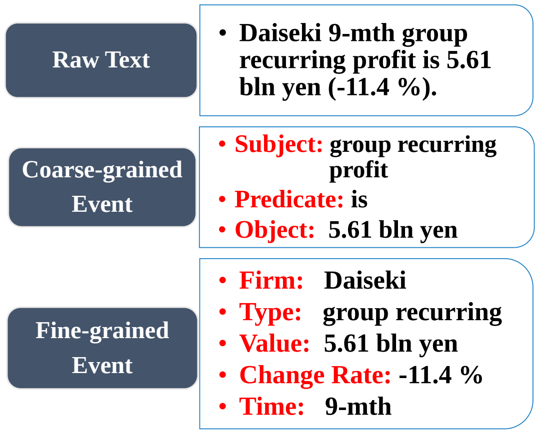

Figure 1: The same news of *Earnings Profit* event in different forms. The event structure consists of event roles (red words) which are the key point of the semantic information.

text (Ding et al., 2016; Zhang et al., 2018b). In the previous studies, the latter method has proven more powerful than the former one, which demonstrates that the event structure containing semantic information is helpful for stock movement prediction. Figure 1 shows a piece of news of *Earnings Profit* event in different forms: raw text, coarse-grained event (<S,P,O>) and fine-grained event (Yang et al., 2018; Liu et al., 2018). We observe that there are still some issues with the <S,P,O> method. Firstly, the <S,P,O> method only extracts subject, predicate and object, which misses some important event roles, such as the earnings *Time* and *Change Rate*, which are included in the fine-grained event. Besides, applying <S,P,O> structure for all event types loses the specific semantic structure in different types of finance events. In Figure 1, the fine-grained event employs *Type* instead of *Subject* used in the coarse-grained event and employs *Value* instead of *Object*, which can describe the event roles in a more detailed way. In this work, we propose to incorporate the fine-grained events in one-day-ahead stock movement prediction. The fine-grained

*This work is done when Deli Chen is a intern at Mizuho Securities.

Proceedings of the Second Workshop on Economics and Natural Language Processing, pages 31–40
Hong Kong, November 4.

event structure describes the specific framework and key points of various finance events. Applying fine-grained events is beneficial for learning a better text representation because the finance knowledge contained in event structure is helpful for understanding the semantic information.

Inspired by the automatic event data generation method (Chen et al., 2017; Zeng et al., 2018; Yang et al., 2018), we propose the TOPIX[1] Finance Event Dictionary (**TFED**) built by domain experts with professional finance knowledge and adopt it to extract fine-grained events automatically for most of finance news. Then we design two different neural models: Structured Stock Prediction Model (**SSPM**) and Multi-task Structured Stock Prediction Model (**MSSPM**). SSPM fuses the extracted fine-grained event and news text firstly, and then conduct interaction between text data and stock trade data to make prediction. SSPM outperforms all the baselines but it can hardly handle the news that can not be recognized by TFED, which we call uncovered news, so MSSPM is designed to learn event extraction using the fine-grained events as the distant supervised label. Besides, we propose to learn event extraction and stock prediction jointly in MSSPM because these two tasks are highly related. The improvement of event extraction result can boost news understanding and promote the stock prediction. And the output of stock prediction can give feedback to event extraction. So the joint learning can share valuable information between tasks. Result shows that MSSPM outperforms SSPM on the uncovered news and increases the method's generalizability. The contributions of this work are summarized as follows:

- We propose to incorporate the fine-grained events in stock movement prediction and this method outperforms all the baselines.
- We propose to learn event extraction and stock prediction jointly, which improves the method generalizability for uncovered news.
- We propose TFED and a pipeline method which can extract fine-grained events from finance news automatically.
- We propose the embedding method for minute-level stock trade data, and adopt time-series models to learn its representation.

[1]Tokyo Stock Price Index, commonly known as TOPIX, is an important stock market index for the Tokyo Stock Exchange (TSE) in Japan.

2 Related Work

2.1 Automatically Event Data Labeling

According to (Chen et al., 2015; Liu et al., 2018; Huang et al., 2018), the fine-grained event structure contains event types, event trigger words and event roles. Zhou et al. (2015) propose a framework to extract events from twitter automatically. Yang et al. (2018) employ a predefined dictionary to label events and then extract document-level events from Chinese finance news. However, they only conduct experiments on 4 event types. While we employ a widely-covered dictionary with 32 different event types. Chen et al. (2017) adopt world and linguistic knowledge to detect event roles and trigger words from text. Zeng et al. (2018) use the Freebase CVT structure to label data and extract event. Araki and Mitamura (2018) adopt distant supervision to extract event from open domain. There are some works using either manual rules (Arendarenko and Kakkonen, 2012) or machine learning methods (Jacobs et al., 2018) for finance event detection, while our event labeling method is stock specific with professional domain knowledge.

2.2 Stock Movement Prediction

Many works using related text for stock movement prediction take the raw text as model input directly. Xu and Cohen (2018) adopt a variational generation model to combine tweets and stock history data to make the prediction. Si et al. (2014) employ the sentiment analysis to help the decision. Li et al. (2015) adopt the tensor decompose method to get the interaction information of different inputs. Duan et al. (2018) use the summary of news body instead of news headline to predict the stock returns. Some other works try to employ structure information to predict the stock movement. Ding et al. (2014) extract <S,P,O> (subject, predicate and object) structure from news to predict the stock movement. Then they propose two improved method based on <S,P,O> structure by applying the weighted fusion of event roles (Ding et al., 2015) and introducing the entity relation knowledge (Ding et al., 2016). Besides, Zhang et al. (2018b) employ a RBM to process <S,P,O> to get the event representation.

3 Fine-grained Event Extraction

3.1 TOPIX Finance Event Dictionary

As shown in (Yang et al., 2018; Zeng et al., 2018) automatic fine-grained event extraction needs an event dictionary to define the event types. Each event type consists of event trigger words and event roles. News containing trigger words perhaps belongs to this event type. The event roles are the key points of semantic structure of this event type. However, there is no specific event dictionary for stock related finance events. So we hired three domain experts to summarize the high-frequency finance events which have a significant impact on stock trading and determine the event trigger words and event roles. With help of domain experts, we also annotated some auxiliary information for the following event extraction process: the POS label of the event roles, the dependency relation pattern of the event types and the necessary/unnecessary label of event roles. Not all event roles will appear in every instance of this event type. Take the *Earnings Profit* event in Figure 1 for example, the *Firm*, *Type* and *Value* will appear in every *Earnings Profit* instance. But the *Change Rate* and *Time* may not appear in some *Earnings Profit* instances. We regard the news containing all the necessary roles as an instance of related event.

TFED contains 32 types of finance events in 8 categories and covers all the main types of finance events that are highly related to stock movement, such as *Earnings Profit*, *M&A* and *Credit Ratings*. All the 32 event types of TFED are displayed in supplementary material A, as well as their trigger words and event roles.

3.2 Event Extraction Process

There are 4 steps in the event extraction process, in which we extract the fine-grained event structures from finance news.

1. Extract Auxiliary Information. In this step, we extract the auxiliary information of news: **POS Tagging** (lexical information) and **Dependency Relation** (syntactic information) by the popular Standford CoreNLP[2] (Manning et al., 2014).

2. Filter Event Candidates. We filter the news that may be an event instance by the TFED. News that contains any trigger word(s) in the dictionary will be regarded as a candidate of the related event. For example, the news in the Figure 1 is a candidate for the *Earnings Profit* event because it contains the trigger word *profit*.

3. Locate Event Roles. We regard news containing all the necessary event roles as an event instance. For event candidates driven by trigger words, we adopt matching rules set by domain experts to check the dependency relation and POS information. Firstly, we match the dependency relation of the candidate news with predefined dependency relation pattern of this event type in TFED to locate the event roles and check if all the necessary event roles are recalled. Then we check if all the event roles' POS labels are consistent with predefined labels. Only if these two conditions are satisfied, this news will be regarded as an event instance and the event roles are determined.

4. BIO Post-process. The result of Step 3 is the label for event roles. Since we want to get the event label for each word in news, we use the BIO label standard to normalize the labeling result. After all these 4 steps, we access the fine-grained event of news. And the extraction result shows that our method covers 71% news in the 210k samples, which proves that the TFED and the pipeline method work well on our experiment data. And for uncovered news, adding more event types is of high cost and low efficiency, so we extract the <S,P,O> structure as replacement following the approach in (Zhang et al., 2018b).

4 Proposed Method

4.1 Problem Formulation

Given N samples in the dataset, and the i-th sample (x^i, y^i, e^i, s^i) contains the news text x^i, the stock trade data y^i in the day before news happens, the event role label e^i generated in Section 3.2 and stock movement label s^i. $x^i = \{x_1^i, x_2^i, ..., x_L^i\}$ is a sequence of words with length of L. $e^i = \{e_1^i, e_2^i, ..., e_L^i\}$ is a sequence of labels indicating the event role of each word in x^i. $y^i = \{y_1^i, y_2^i, ..., y_M^i\}$ is a sequence of trade record vectors for each trade minute with length of M. $s^i \in \{0, 1\}$ is the stock movement label telling whether the stock trade price is up or down at prediction time. The stock movement prediction task can be defined as assigning movement label for the news input and trade data input.

[2]https://stanfordnlp.github.io/CoreNLP/

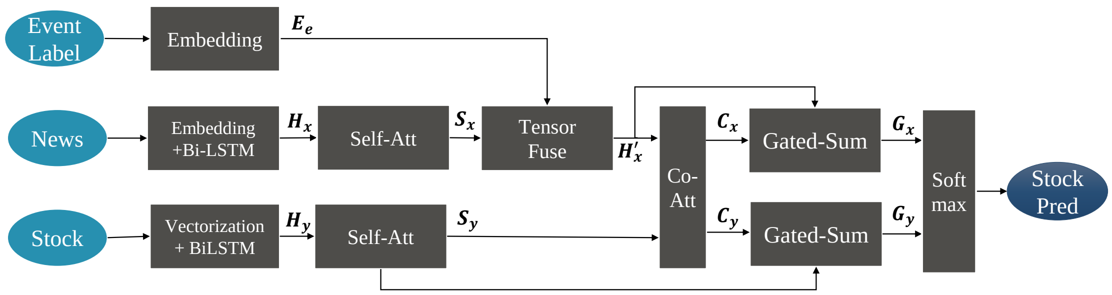

Figure 2: The overview of the proposed SSPM model.

4.2 Trade Data Embedding

Different from works of (Xu and Cohen, 2018; Zhang et al., 2018b) who use limited daily-level stock trade data (stock close price and daily trade volume, for example), we adopt the minute-level stock data to describe the stock movement in a more detailed way. For each minute when at least one trade happens, we collect the following items: (1) First/last/highest/lowest trade price of the minute; (2) Total trade volume/value of the minute; (3) Volume-weighted average trade price. The stock trade data is of time series data, so in order to apply the powerful time series neural models, we transfer the raw trade features into trade data embedding $\boldsymbol{E_y}$. The following combination performs best on the develop set:

- Raw Number: first/last/highest/lowest trade price, total trade volume and volume-weighted average trade price
- Change Rate: change rates of all the raw number items compared to last minute

Now we get 12 feature numbers for each trade minute. We set the length of time step to 10 minutes. Then we get the trade data embedding $\boldsymbol{E_y} \in \mathbb{R}^{T \times D_s}$. $T = \mathrm{M}/10$ and $\mathrm{D_s} = 120$. M is the length of the trade minutes. Finally, we adopt the min-max scale method for each stock's samples and pad the time steps less than 10 minutes with last trade minute's data.

4.3 Base Model: Structured Stock Prediction Model (SSPM)

Figure 2 shows the overview of SSPM. We first transfer various sources of input (x, y, e) into dense vectors. Then we get the representations of text and stock data through bi-directional Long Short Term Memory (BiLSTM) and self-attention. Then we fuse text and event structure to access the structure-aware text representation. Finally, we interact text and stock data by co-attention to predict stock movement. There are 4 modules in SSPM: input embedding, single-modal information representation, bi-modal information interaction and prediction. Experiment results show that SSPM outperforms all the baselines.

4.3.1 Input Embedding

The purpose of this module is to transfer various sources of input (x, y, e) into dense vectors. For words in finance news x, we use both word-level pretrained embedding Glove (Pennington et al., 2014) and character-level pretrained embedding ELMo (Peters et al., 2018) for the purpose of representing words better from different levels. Then we concatenate them to get the final word representation $\boldsymbol{E_x} \in \mathbb{R}^{L \times D_w}$. We use the method proposed in Section 4.2 to get the stock trade data embedding $\boldsymbol{E_y} \in \mathbb{R}^{T \times D_s}$. Besides, we embed event role labels e into dense vectors $\boldsymbol{E_e} \in \mathbb{R}^{L \times D_e}$ using a parameter matrix initialized with random values. D_w, D_s, D_e are the embedding dimensions of word, stock and event role, respectively. T is the length of stock time-steps.

4.3.2 Single-modal Information Representation

The purpose of this module is to get the representations for both news and stock trade data independently. After accessing the input embedding, we employ BiLSTM to encode the $\boldsymbol{E_x}$ and $\boldsymbol{E_y}$:

$$\boldsymbol{H_x} = \mathrm{BiLSTM_x}(\boldsymbol{E_x})$$
$$\boldsymbol{H_y} = \mathrm{BiLSTM_y}(\boldsymbol{E_y})$$

Now we access the sentence representation $\boldsymbol{H_x} \in \mathbb{R}^{L \times 2h}$ and daily stock trade representation $\boldsymbol{H_y} \in \mathbb{R}^{T \times 2h}$. h is the hidden size of BiLSTM. In order to enhance the learning ability, we use the self-attention to allow the $\boldsymbol{H_x}$ and $\boldsymbol{H_y}$ to have a look at themselves and make adjustment. We apply the bilinear attention method which have proven (Wang et al., 2018; Deng et al., 2018) more powerful in learning ability. Here are the formulas for H_x:

$$W_{SA}^{x} = \text{softmax}(\boldsymbol{H_x} \cdot W_1 \cdot \boldsymbol{H_x}^{\top})$$
$$\boldsymbol{S_x} = W_{SA}^{x} \cdot \boldsymbol{H_x}$$

W_1 is a trainable weight matrix and $\boldsymbol{S_x} \in \mathbb{R}^{L\times 2h}$. In the same way we get the self-attention result of the stock data: $\boldsymbol{S_y} \in \mathbb{R}^{T\times 2h}$.

In the <S,P,O> method, event roles are extracted as separated phrases where some words are ignored and the word order information is missing. Instead, we fuse the text representation S_x with the event role embedding E_e to capture the structure information and remain the word order at the same time. E_e contains both word-level (event role) and sentence-level (BIO label) information, which is similar with S_x, so we select to fuse E_e with S_x instead of E_x. Here we adopt the fusion function used in (Wang et al., 2018; Mou et al., 2016) to fuse the event structure and text effectively:

$$\boldsymbol{H'_x} = \sigma(W_f[\boldsymbol{S_x}; \boldsymbol{E_e}; \boldsymbol{S_x} - \boldsymbol{E_e}; \boldsymbol{S_x} \circ \boldsymbol{E_e}] + b_f)$$

; means tensor connection. We ensure $D_e = 2h$ so that E_e has the same dimension with S_x. $\circ$ means element-wise multiplication and σ is the activation function. $\boldsymbol{H'_x} \in \mathbb{R}^{L\times 2h}$ is the structure-aware text representation.

4.3.3 Bi-modal Information Interaction

In this part we conduct the interaction between the two modal information: finance news of text mode and stock trade data of number mode. These two different modal information are highly relevant: the finance news represents the environment variable and the stock trade data represents history movement. The interaction between them can lead to a better understanding of stock movement. We use the co-attention to interact the bi-modal information: $\boldsymbol{H'_x} = \left\{h_x^{\prime 1}, h_x^{\prime 2}, \cdots, h_x^{\prime L}\right\}$ and $\boldsymbol{S_y} = \left\{s_y^1, s_y^2, \cdots, s_y^T\right\}$. The attention weight is computed by the following function:

$$f_{att}(i,j) = \text{Relu}(h_x^{\prime i\top}\, W_2\, s_y^j)$$

W_2 is a trainable weight matrix. We use the softmax function to normalize the attention weight:

$$\alpha_{ij} = \frac{e^{f_{att}(i,j)}}{\sum_{k=1}^{T} e^{f_{att}(i,k)}}; \quad \beta_{ij} = \frac{e^{f_{att}(i,j)}}{\sum_{t=1}^{L} e^{f_{att}(t,j)}}$$

Finally we get the reconstructed representations:

$$c_x^i = \sum_{j=1}^{T} \alpha_{ij}\, s_y^j; \qquad c_y^j = \sum_{i=1}^{L} \beta_{ij}\, h_x^{\prime i}$$

Now we access the reconstructed representations $\boldsymbol{C_x} = \left\{c_x^1, c_x^2, \cdots, c_x^L\right\}$ and $\boldsymbol{C_y} = \left\{c_y^1, c_y^2, \cdots, c_y^T\right\}$ based on the attention to another modal information. Then we use the gating mechanism to incorporate the original representation and the corresponding attention result:

$$\boldsymbol{G_x} = \text{g}(\boldsymbol{H'_x}, \boldsymbol{C_x}) \cdot \boldsymbol{C_x} + (1 - \text{g}(\boldsymbol{H'_x}, \boldsymbol{C_x})) \cdot \boldsymbol{H'_x}$$
$$\boldsymbol{G_y} = \text{g}(\boldsymbol{S_y}, \boldsymbol{C_y}) \cdot \boldsymbol{C_y} + (1 - \text{g}(\boldsymbol{S_y}, \boldsymbol{C_y})) \cdot \boldsymbol{S_y}$$

where the $g(,)$ is the gating function and we use the non-linear transformation with $sigmod$ activation function in experiment.

4.3.4 Prediction

In this module, we concatenate the $\boldsymbol{G_x}$ and $\boldsymbol{G_y}$ and predict the stock movement label $\widehat{p}$:

$$\widehat{p}(s|x,y,e) = \text{softmax}(W_p[\boldsymbol{G_x}; \boldsymbol{G_y}] + b_p)$$

4.4 Advanced Model: Multi-task Structured Stock Prediction Model (MSSPM)

SSPM can hardly process the uncovered news that can not be recognized by the TFED since the fine-grained event structure information is not provided. MSSPM is designed to handle this issue by employing the generated e in Section 3.2 as the distant supervised label to train an event extractor. Furthermore, we design a multi-task framework to jointly learn event extraction and stock prediction because these two tasks are highly related. The quality of event extraction result has a direct influence on the downstream stock prediction task. At the same time, the results of stock prediction can give valuable feedback to event extraction. The multi-task framework can share useful information and make effective interaction between tasks. The overview of MSSPM is shown in Figure 3. The upper half of the dotted line represents the event extraction part. We regard the event extraction task as a sequence labeling task and adopt the self-attended BiLSTM-CRF (conditional random fields) method to make labeling decisions. The lower half stands for the stock movement prediction part which works in a similar way as SSPM.

4.4.1 Event Extraction

After accessing the word embedding $\boldsymbol{E_x}$, we employ the BiLSTM to get the sentence representation $\boldsymbol{H_x}$. Then we employ self-attention to learn a better representation $\boldsymbol{S_x}$. Finally we predict the

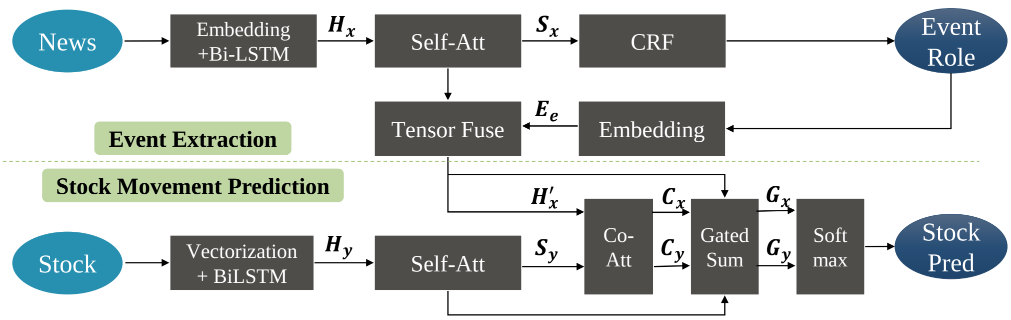

Figure 3: The overview of the proposed MSSPM model.

event label and employ CRF to optimize output:

$$\widehat{e} = \mathrm{softmax}(W_l \boldsymbol{S_x} + b_l)$$
$$\widehat{e'} = \mathrm{CRF}(\widehat{e})$$

$\widehat{e'}$ is the estimation of the event role. Then we adopt the method introduced in Section 4.3.1 to get the event role embedding $\boldsymbol{E_e}$ from $\widehat{e'}$ and adopt the tensor fusion function used in SSPM to get the structure-aware text representation $\boldsymbol{H'_x}$.

4.4.2 Stock Movement Prediction

The stock movement prediction process is similar with SSPM, and the main difference is the event input $\boldsymbol{E_e}$ is predicted from the event extractor in Section 4.4.1. The stock trade representation $\boldsymbol{S_y}$ is accessed in the same way with SSPM. Then we employ the co-attention to interact $\boldsymbol{H'_x}$ and $\boldsymbol{S_y}$, followed by the gated sum and $softmax$ function to predict the stock movement label $\widehat{s}$.

4.4.3 Multi-task Learning Object

The loss function of MSSPM consists of two parts, which are the negative logarithm loss of event extraction and that of stock prediction:

$$LS_e = -\sum_t e_t log\, p(e_t|x),\ t = [1, \cdots, L]$$
$$LS_s = -s \log p(s|x, y)$$

We select the weighted sum of these two losses as the final loss of MSSPM:

$$LS = \lambda LS_e/L + (1-\lambda)LS_s$$

The λ is a hyper-parameter to balance two losses. LS_e is divided by the number of words to ensure it is comparable with LS_s. The experiment result shows that model performs best on develop set when $\lambda = 0.43$.

5 Experiment

The experiment data is from the professional finance news providers Reuters[3]. We collect finance news related to TOPIX top 1000 stocks[4] from 2011 to 2017. The raw data contains both news headline and body, and we use headline only since the headline contains the most valuable information and has less noise than news body. We collect stock trade data for news happens in/out of trade time (9:00 AM - 15:00 PM in trade day) differently. For those news happens in trade time, we collect the trade data from 9:00 AM to the last minute before news happens. And for those news happens out of trade time, we collect the trade data of last trade day. We want to ensure no trade data after news happens are included in the input in which situation the market reactions are leaked to the model. We get about 210k data samples finally. Following (Ding et al., 2015; Xu and Cohen, 2018), the stock movement is divided into two categories: stock up/down. The stock up and down rates are 45% and 55% in our dataset, respectively. We adopt TOPIX Sector Index to correct the stock movement in order to eliminate the influence of macro news and the details are introduced in supplementary materials C. In experiment, we reserve 10k samples for developing and 10k samples for testing. The samples in train set are previous to samples in valid set and test set to avoid the possible information leakage. All the rest 190k samples are applied for training SSPM while only the dictionary covered part (about 70%) in the 190k samples are applied for training MSSPM to acquire a high-quality event extractor. We tune the hyper-parameters on the development set and

[3]Source Reuters News cThomson Reuters cREFINITIV, https://www.thomsonreuters.com/en.html

[4]Each news in Reuters has a manual field indicating its related stock(s) and we use it to filter the stock related news.

test model on the test set. The evaluation metrics are accuracy and Matthews Correlation Coefficient (MCC). MCC is often used in stock forecast (Xu and Cohen, 2018; Ding et al., 2016) because it can overcome the data imbalance issue. More experiment details are listed in the supplementary material D.

6 Results and Analysis

We analyze the results of experiments in this section. Firstly, we compare the proposed methods with the baselines in Section 6.1. Then we analyze the effect of event structure input in SSPM in Section 6.2 and analyze the MSSPM method in Section 6.3. Lastly, we conduct the ablation study in Section 6.4. We also conduct error analysis over different times' news, which is shown in the supplementary material B.

6.1 Comparison With Baselines

The following baselines are used in this work.

- **Bagging Decision Tree:** This method adopt bagging ensemble algorithm to combine 20 Decision Tree classifiers to make the prediction. It outperforms all the other traditional machine learning methods we tried.
- **Sentiment Analysis:** This method (Si et al., 2014) conducts sentiment analysis on news headlines to predict stock movement.
- **Target Specific Representation:** This method (Duan et al., 2018) employs the news headline as the target to summarize the news body in order to utilize the abundant information of news body.
- **Triple Structure:** This method (Ding et al., 2014) adopts the <S,P,O> triple to represent the event structure.
- **Weighted Triple Structure:** This method (Ding et al., 2015) adds trainable weight matrices in <S,P,O> to enhance fitting ability.
- **Triple Structure with RBM:** This method (Zhang et al., 2018b) uses Restricted Boltzmann Machine to handle the <S,P,O> and then adopts multi-instance learning to model the latent consistencies of different data sources. Because tweet data are contained in news data[5] in our dataset, our implementation uses news and stock data instead of news, tweet and stock data.

[5]Some related tweets about stocks are also provided by Reuters mixed with news.

Method	Event	Acc(%)	MCC
Bagging Decision Tree	No	54.9	0.096
Sentiment Analysis (Si et al., 2014)	No	62.8	0.253
Target Specific Rep. (Duan et al., 2018)	No	63.7	0.275
Triple Structure (Ding et al., 2014)	Coarse-grained	63.2	0.270
Weighted Triple Structure (Ding et al., 2015)	Coarse-grained	63.5	0.269
Triple Structure with RBM (Zhang et al., 2018b)	Coarse-grained	64.0	0.278
MSSPM(proposed)	Fine-grained	65.7	0.315
SSPM(proposed)	Fine-grained	66.4	0.330
Ensemble	Fine-grained	**67.2**	**0.348**

Table 1: Results on test set compared with baselines; the results in this table and following tables have proven significant with $p < 0.05$ by student t-test.

Table 1 is divided into three parts. The three baselines in the top part employ the text directly as model input. These methods totally ignore the structure information of text. The three baselines in the middle part take structure information into account. These methods consider <S,P,O> event roles in all event types though, they miss some important event roles and describe the event roles in a very rough way. Moreover, the word order information is missing under such settings. Both SSPM and MSSPM outperform all the baselines. These proposed method incorporate the fine-grained event structure in stock movement prediction. It can extract specific fine-grained event structures for different types of finance events. At the same time, this method remains the original word order through the tensor fusion. Another advantage of our method is that it applies the stock data embedding method for the minute-level stock trade data and conducts interaction between stock data and news data. SSPM performs a little better than MSSPM because SSPM adopts more data for training and the learning of event extraction in MSSPM is not perfect. The Ensemble method follows a simple rule to combine the SSPM and MSSPM: The TFED covered news is processed by SSPM and the uncovered news is processed by MSSPM. It achieves the best result among all the baselines and proposed methods.

6.2 Effect of Event Structure

In this section, we analyze the effect of event structure. Although there have been some comparisons of different event structures in Section 6.1,

Input Form	Acc(%)	MCC
No Text	58.1	0.161
No Event (Raw Text)	62.2	0.246
Coarse-grained Event	64.6	0.291
Fine-grained Event	**66.4**	**0.330**

Table 2: Different Text Input Forms in SSPM.

the models are different. In this section, we conduct an experiment based on the SSPM model and change different text input forms to check the impact of the event structure. We design 4 different text input forms for SSPM: (1) No Text method takes no text information as input and relies entirely on trade data to predict stock movement; (2) No Event takes the raw news text as model input and removes the event input from SSPM; (3) Coarse-grained Event employs the coarse-grained event structure <S, P, O> as event input of SSPM; (4) Fine-grained Event is the proposed method to utilize the category-specific fine-grained event as model input. The results are shown in Table 2. We can find that all the there methods adding text input outperform the No Text method, which proves the effect of finance news. Both the Coarse-grained and Fine-grained Event methods bring improvement to the prediction result, which shows that the event structure is very useful. Moreover, the Fine-grained Event method brings larger improvement than the Coarse-grained Event method, which demonstrates that utilizing fine-grained events is more helpful to help model understanding the semantic information of news text.

6.3 Analysis of MSSPM

Although SSPM performs well in stock prediction, there are two important issues with it. Firstly, 29% news in our dataset can not be recognized by TFED, and the Table 3 shows that the result of uncovered data is obvious lower than the covered data. Secondly, TFED is domain specific, so the generalizability of SSPM may be restricted. MSSPM is designed to handle these two issues.

As shown in Table 3, although the performance of MSSPM is lower than SSPM on the covered test set, its performance is higher than SSPM on the uncovered test set. The performance decrease of MSSPM after transferring from covered set to uncovered set is much smaller than SSPM's, which proves MSSPM has higher transferability. The uncovered news can be regarded as events of new types, and MSSPM performs better on it by learning event extraction, which improves the generalizability of the structured stock prediction method. As shown in Table 4, the performance of multi-task learning is clearly better than the pipeline method, which confirms our assumption that these two tasks are highly related and the joint learning improves both of their results.

Data	Covered		Uncovered	
Metric	Acc(%)	MCC	Acc(%)	MCC
SSPM	67.6	0.351	63.4	0.267
MSSPM	65.9	0.318	**65.2**	**0.305**

Table 3: Result on different sets of test data. The covered set means samples recognized by the TFED and the uncovered set means the samples out of the dictionary. These two sets account for around 30% and 70% of the test set, respectively.

Task	Event Extraction	Stock Prediction	
Metric	Micro-F1(%)	Acc(%)	MCC
Pipeline	79.2	64.8	0.297
Multi-Task	**84.3**	**65.7**	**0.315**

Table 4: Comparison of pipeline method and multi-task learning method (MSSPM). The pipeline method trains the event extractor first and then predicts the stock. We report the micro-F1 score for the event extraction task.

6.4 Ablation Study

In this section, we report and analyze the results of ablation study. We remove different components of both SSPM and MSSPM to check their effect. As shown in Table 5 and Table 6, we found that the model performance drops in all the ablation experiments as expected. The fusion function, attention mechanism (both self-attention and co-attention) and the gating mechanism are all helpful for both SSPM and MSSPM. We can observe an obvious decrease after removing fusion function (adopt adding method instead) both in SSPM ($\downarrow$ 1.5 of Acc) and MSSPM ($\downarrow$ 1.1 of Acc), which demonstrates that the fusion function combines the event structure and the news text effectively. Besides, the co-attention between news and stock trade data also plays an important role in both models.

7 Conclusion

In this work, we propose to incorporate the fine-grained events in stock movement prediction task. We propose the TOPIX Finance Event Dictionary with domain experts' knowledge and extract fine-grained events automatically. We propose SSPM

Metric	Acc(%)	MCC
SSPM	**66.4**	**0.330**
w/o Fusion Function	64.9(↓ 1.5)	0.298(↓ 0.032)
w/o Self-Attention	66.0(↓ 0.4)	0.319(↓ 0.011)
w/o Co-Attention	65.1(↓ 1.3)	0.303(↓ 0.027)
w/o Gated Sum	65.6(↓ 0.8)	0.314(↓ 0.016)

Table 5: Ablation Study of SSPM.

Metric	Acc(%)	MCC
MSSPM	**65.7**	**0.315**
w/o Fusion Function	64.6(↓ 1.1)	0.292(↓ 0.023)
w/o Self-Attention	65.5(↓ 0.2)	0.310(↓ 0.005)
w/o Co-Attention	64.9(↓ 0.8)	0.298(↓ 0.017)
w/o Gated Sum	65.2(↓ 0.5)	0.304(↓ 0.011)
w/o CRF	64.2(↓ 1.5)	0.285(↓ 0.030)

Table 6: Ablation Study of MSSPM.

to incorporate fine-grained events in stock movement prediction which outperforms all the baselines. Besides, to handle the uncovered news, we use the event data as the distant supervised label to train a multi-task framework MSSPM. The results show that MSSPM performs better on uncovered news and improves the generalizability of the structured stock prediction method.

8 Acknowledgement

This work is supported by a Research Grant from Mizuho Securities Co., Ltd. Three experienced stock traders from Mizuho Securities provide the professional support for the TFED dictionary. Mizuho Securities also provide experiment data.

References

Hana Alostad and Hasan Davulcu. 2017. Directional prediction of stock prices using breaking news on twitter. *Web Intelligence*, 15(1):1–17.

Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 878–891.

Ernest Arendarenko and Tuomo Kakkonen. 2012. Ontology-based information and event extraction for business intelligence. In *Artificial Intelligence: Methodology, Systems, and Applications - 15th International Conference, AIMSA 2012, Varna, Bulgaria, September 12-15, 2012. Proceedings*, pages 89–102.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 409–419.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 167–176.

Yang Deng, Yuexiang Xie, Yaliang Li, Min Yang, Nan Du, Wei Fan, Kai Lei, and Ying Shen. 2018. Multi-task learning with multi-view attention for answer selection and knowledge base question answering. *CoRR*, abs/1812.02354.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1415–1425.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. 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.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2016. Knowledge-driven event embedding for stock prediction. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2133–2142.

Junwen Duan, Yue Zhang, Xiao Ding, Ching-Yun Chang, and Ting Liu. 2018. Learning target-specific representations of financial news documents for cumulative abnormal return prediction. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2823–2833.

Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. 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.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Associa-*

tion for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, pages 2160–2170.

Gilles Jacobs, Els Lefever, and Véronique Hoste. 2018. Economic event detection in company-specific news text. In *The 56th Annual Meeting of the Association for Computational Linguistics*, pages 1–10. Association for Computational Linguistics.

Qing Li, LiLing Jiang, Ping Li, and Hsinchun Chen. 2015. Tensor-based learning for predicting stock movements. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 1784–1790.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1247–1256.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Jianfeng Si, Arjun Mukherjee, Bing Liu, Sinno Jialin Pan, Qing Li, and Huayi Li. 2014. Exploiting social relations and sentiment for stock prediction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1139–1145.

Wei Wang, Chen Wu, and Ming Yan. 2018. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. 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 1705–1714.

Yumo Xu and Shay B. Cohen. 2018. 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.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, pages 50–55.

Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2018. Scale up event extraction learning via automatic training data generation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6045–6052.

Lingling Zhang, Saiji Fu, and Bochen Li. 2018a. Research on stock price forecast based on news sentiment analysis - A case study of alibaba. In *Computational Science - ICCS 2018 - 18th International Conference, Wuxi, China, June 11-13, 2018, Proceedings, Part II*, pages 429–442.

Xi Zhang, Siyu Qu, Jieyun Huang, Binxing Fang, and Philip S. Yu. 2018b. Stock market prediction via multi-source multiple instance learning. *IEEE Access*, 6:50720–50728.

Yunsong Zhong, Qinpei Zhao, and Weixiong Rao. 2017. Predicting stock market indexes with world news. In *4th International Conference on Systems and Informatics, ICSAI 2017, Hangzhou, China, November 11-13, 2017*, pages 1535–1540.

Deyu Zhou, Liangyu Chen, and Yulan He. 2015. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2468–2475.