Incorporating Expert-Based Investment Opinion Signals in Stock Prediction: A Deep Learning Framework

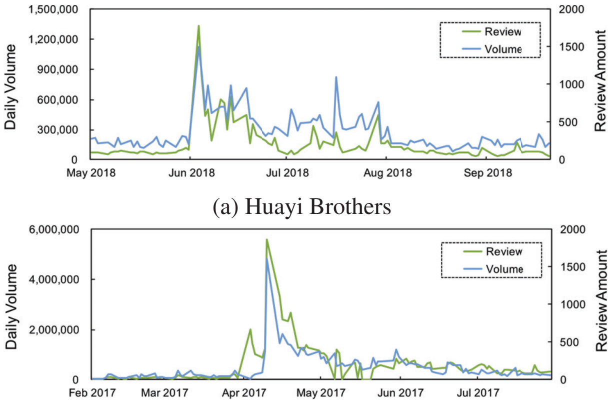
Heyuan Wang,[1] **Tengjiao Wang,**[1]* **Yi Li**[1]
[1]Key Lab of High Confidence Software Technologies (MOE), School of EECS, Peking University
{wangheyuan, tjwang, liyix}@pku.edu.cn

Abstract

Investment messages published on social media platforms are highly valuable for stock prediction. Most previous work regards overall message sentiments as forecast indicators and relies on shallow features (bag-of-words, noun phrases, etc.) to determine the investment opinion signals. These methods neither capture the time-sensitive and target-aware characteristics of stock investment reviews, nor consider the impact of investor's reliability. In this study, we provide an in-depth analysis of public stock reviews and their application in stock movement prediction. Specifically, we propose a novel framework which includes the following three key components: *time-sensitive and target-aware investment stance detection*, *expert-based dynamic stance aggregation*, and *stock movement prediction*. We first introduce our stance detection model named MFN, which learns the representation of each review by integrating multi-view textual features and extended knowledge in financial domain to distill bullish/bearish investment opinions. Then we show how to identify the validity of each review, and enhance stock movement prediction by incorporating expert-based aggregated opinion signals. Experiments on real datasets show our framework can effectively improve the performance of both investment opinion mining and individual stock forecasting.

Introduction

Online forums and stock investment websites such as Stocktwits, Twitter and Weibo allow millions of investors to share transactions, opinions, news events and so on in real time. The numerous public comments are of great value in reflecting market conditions and making trading decisions, which have been well studied in recent years. Figure 1 shows the trading volumes of *Huayi Brothers* and *China Fortune* which are two stocks in Chinese A-share market, and the amounts of their relevant reviews from xueqiu[1] and guba[2] (two popular Chinese Twitter-like investment forums). It can be observed that for each stock, the fluctuations of the two curves are great synchronous, especially when a mutation occurs. In Figure 1a, both the review amount and trading volume of *Huayi Brothers* peaked on June 4, 2018, as a result of the "celebrity twin-contracts" incident. The sudden increase in Figure 1b of *China Fortune* indicates an important policy announcement on April 1, 2017: establishing Xiongan New Area. Shortly after the news release, a lot of comments were generated from the public and related shares witnessed a rapid rise. Therefore, it is reasonable to leverage public opinion signals for stock prediction.



(a) Huayi Brothers

(b) China Fortune

Figure 1: Dynamics of trading volumes and review amounts.

Most existing work on sentiment analysis based stock prediction aligns bullishness with the overall message sentiment, and heavily relies on word-level statistics and traditional classification methods (Tetlock 2007; Bollen, Mao, and Zeng 2011; Nguyen and Shirai 2015). They have a number of disadvantages. Firstly, they are limited in identifying the matching relationship between multiple opinions and entities in a sentence. However, the author is likely to be in favor of one object while against another. Consider the sentence "*Out SANY HEAVY at 8.95, bot back some CMBC this morning*", predictive indicators are words "Out" and "bot back", which represent opposite views (bearish and bullish, respectively) toward different objects (words in uppercase). Through the statistic analysis of real-world online stock reviews (collected from guba and xueqiu), we find that nearly half of reviews contain more than one stock entity. Thus the overall sentiment of a review text cannot be simply applied when it comes to a specific target

*Corresponding Author

[1] https://www.xueqiu.com

[2] http://guba.eastmoney.com

stock, the key is to match different opinion positions with the corresponding objects. Secondly, traditional methods ignore the time-sensitive characteristic of stock reviews that is important for future prediction. For example, in the sentence " *I sold out that stock at 15.2 yesterday, but it may rise sharply over the next month.*", the opinion "sold out" (bearish) refers to "yesterday" (past) and the opinion "rise" (bullish) refers to "over the next month" (future), while the latter is more valuable for stock trend prediction. In another case, the author may present gaining profits in the past few days with a positive emotion, but without any forecast about the future. Such a post should not be viewed as a bullish signal. Thirdly, pioneering work mainly regards each review equally and aggregates opinions from all users into a global sentiment score. Although a few work has taken the user's reliability into account, they simply assume that it is fixed and time-independent (Bar-Haim et al. 2011; Ou et al. 2014). However, on the one hand, the market is unstable and users' reliabilities are changing. On the other hand, investors usually perform different expertise toward different stocks. Intuitively, it is more practical and effective to build a dynamically updated expert mining system for each stock.

In this paper, we develop a framework of detecting and integrating expert-based investment opinion signals for stock prediction. To determine the user's bullish/bearish position on a target stock more accurately, we build a Multi-view Fusion Network (MFN) which contains two major components, i.e., a word level encoder to jointly learn the lexical and multi-view knowledge embeddings including time, polarity and negation attributes, a structured sequence encoder to model the matching relationship of multiple stock entities and opinions in a text. In order to judge the quality of a review, we propose to measure users' reliabilities by dynamically examining the consistency of their published review opinions and the stock price changes over a period of time. In this way, we can identify and follow experts so as to reduce emotion spams and misleading investment information. The final stock forecast score is obtained through the gated recurrent units (GRU) (Chung et al. 2014) with temporal-aware attention mechanism. We construct a human-labeled Chinese stock review dataset and evaluate the effectiveness of MFN module on it. Results show that MFN significantly outperforms the baselines over 2.5% on three-class stance detection test and 3.5% on two-class for weighted F1 metric. Then we conduct compared experiments to investigate the ability of our method for individual stock forecasts. Experimental results prove the importance of the expert mining module for screening high-quality public opinions for stock movement prediction.

To sum up, our contributions are three-folds: (1) the deep analysis of stock reviews and proposal of a novel stance detection model for distilling bullish/bearish investment opinions for a target stock; (2) the introduce of a dynamic expert-based procedure to distinguish and aggregate high quality opinion indicators; (3) empirical verification of the effectiveness of the proposed stock prediction framework and each module on real-world datasets.

Related Work

Stance Detection

Stance detection is to classify the attitude expressed in a text toward a target. It has been exploited to address different issues. Walker et al. (2012) used agreement relations between speakers to represent dialogic structures, and determined stance classification of debates. Faulkner (2014) leveraged target information and Wikipedia link-based measures to classify stances in student essays. Zhang, Zhang, and Vo (2015) extended CRF baseline using neural networks with automatic feature combinations for open domain targeted sentiments. For stance prediction in tweets toward different targets (*Atheism, Feminist Movement, Legalization of Abortion, etc.*), Augenstein et al. (2016) used a bidirectional conditional LSTM encoding to build a target-aware representation of the tweet. On Fake News Challenge task, Ferreira and Vlachos (2016) estimated the position of news articles relative to headline claims, Mohtarami et al. (2018) used end-to-end memory networks to extract the snippets of evidence, which can facilitate the process of rumor debunking. Chen et al. (2017) identified individual's political ideology from speeches and texts using an opinion-aware knowledge graph. Our MFN module is used to detect investment positions on specific target stocks in online public reviews.

Stock Prediction

The literature of stock prediction was early based on the Efficient Market Hypothesis (EMH) and random walk theory (Fama 1965). Some techniques leveraged historical data of market with linear methods like ARIMA and Markov chain, or nonlinear models like SVM, AutoEncoders, Neural networks, etc. (Wang et al. 2012; Bao, Yue, and Rao 2017).

As web information grows, various studies have found that trading decisions can be significantly driven by public opinions. Early work (Tetlock 2007) examined statistic information of qualitative terms included in aggregate market valuations. Bollen, Mao, and Zeng (2011) used large-scale tweet based public mood to predict the movement of Dow Jones Industrial Average (DJIA). Si et al. (2013) regressed topic based sentiments and stock indexs to predict the S&P100 Index. Si et al. (2014) built a network from con-occurrences of ticker symbols in tweets and found the stock relationships. Zhao et al. (2016) first used LDA to filter financial microblogs from Weibo, then used a financial lexicon to get the sentiments for market index prediction.

Apart from sentiment, financial news has been used as another major source to predict stock movements (Xie et al. 2013; Li et al. 2014; Ding et al. 2014; 2015; Hu et al. 2018; Liu et al. 2018). Different features such as bag-of-words, frame semantic parsing and extracted structured events were used in these work. In this paper, we provide a thorough study of the time-sensitive and target-aware characteristics of stock reviews, and emphasis on how to improve the effectiveness of detecting and integrating public investment opinions for stock forecasts.

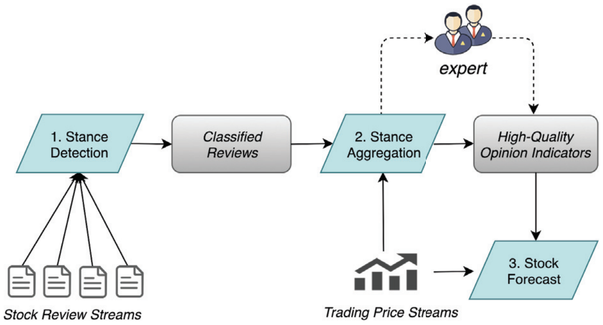


Figure 2: Architecture of the overall framework.

Our Approach

In this section, we describe our framework of exploiting expert investment opinions for stock prediction in detail. There are three major components in our approach, i.e., a *stance detection* network named MFN to distill bullish/bearish investment opinion signals form public reviews, a *stance aggregation* module to identify and aggregate high-quality opinions based on a dynamic expert mining procedure, and a *stock forecasting* module which combines expert opinion indicators to predict the future movement of individual stocks. Figure 2 illustrates the overview of our framework.

Stance Detection Module

Formalization The annotated stock review data set can be described as $\mathcal{D} = \{(r, s, u, t, y)_i\}_{i=1}^{N}$. For each sample, u represents the user who posted the review and is used for expert mining, t is the publish time that is used for time-sensitive judgment. r denotes the review content and s is the target stock. y is the label indicating whether the author holds a bullish, bearish or neutral view toward s in r. The aim is to build a model $g(\cdot, \cdot)$ with $\mathcal{D}$. For any review-stock pair (r, s) with the user id and time stamp, $g(r, s)$ measures the positive score of r to s.

MFN Overview We propose the Multi-view Fusion Network (MFN) to model $g(\cdot, \cdot)$. Figure 3 shows the architecture. MFN distills multi-view semantic features from words, n-grams and structured matching sequence with two encoders, then integrates different representations into a fused vector to calculate the predictive probability the author holds for s. All lexical features and extended knowledge are jointly learned in an end-to-end way. To better describe the construction of multi-view representations, we first detail the process of time reference and stock mention extraction, as well as the extraction and generalization methods of structured opinion triplet sequence in the following sections.

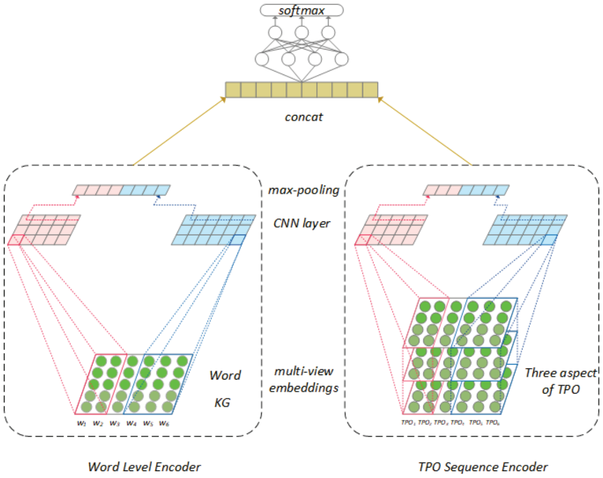


Figure 3: The MFN stance detection model.

Table 1: Conditions used in stock mention extraction.

Type	*Condition*	*Sample Pattern*
lexical	industry suffix	**Shahe** Industry **Gree** Electric Appliances **Lepu** Pharmaceuticals **Inspur** software
	location prefix	Shenzhen **Huaqiang** China **Baoan**
	special market tagging	**Wanke** A *ST **Santai**
syntactic	connection symbol	**F(A) ; A(F) ; A-F ; F-A**
	conjunction phrase	**F**, or **A** ; **A**, or **F**; **A**, namely **F**

Time Reference Extraction To capture the time-sensitive characteristics of stock reviews, we identify and classify ***time references*** in review texts from three aspects, which is inspired by (Tu, Cheung, and Mamoulis 2015): (1) terms in date-time formats are first identified and classified by comparing with the review's publish time t. (2) a temporal dictionary is built to identify time-related words and phrases, such as "last year", "today", "tomorrow", etc. (3) since summative semantics usually refer to the past while predictive semantics refer to the future, a summary/prediction dictionary is built to identify terms such as "review", "retrospect", "expect", "recommend", etc.

Stock Mention Extraction A stock entity may have variant mentions in review texts. To enhance the target-aware ability of our model, we identify ***stock mention pairs***, which are pairs of entity mentions in texts that refer to the same stock objects. Our approach is rule-based with lexical and syntactic matching types. Conditions of rules and sample patterns are shown in Table 1. For lexical-based patterns, the bold substrings are extracted as non-canonical mentions of the corresponding stocks. For the syntactic type, "F" stands for the full name with longer mention, while "A" denotes the abbreviation with shorter mention. Words and phrases that satisfy the conditional patterns are distilled as different mentions of the appropriate stock entities.

Opinion Triplet Extraction and Generalization To better model the matching relationship of multiple opinions and stock entities in a text, we design a structured representation scheme, where the format of each matching pair is $(Time, Predict, Object)$ (TPO), indicating the predictive position P toward the relevant stock object O, as well as its reference time T. Consider a review (translated from the original Chinese post) with target stock ***LeTV***,

@Lee: do not chase ***LeTV*** *at present, this is excellent time to buy* ***Gree****. By the way,* ***SAIC Motor*** *will pick up soon according to the theoretical calculation ...*

Words "chase", "buy" and "pick up" are identified as predictive terms. Time indicators are underlined and the stock entities are shown in bold. The TPO sequence is extracted as { *(at present, do not chase, LeTV), (this is excellent time, buy, Gree), (will, pick up, SAIC Motor)* }. Given a review and its target stock (in general, a review is posted on the homepage of a stock, which we regard as the target stock), the following procedure is used to extract the TPO sequence:

- We first identify all opinion terms P, time indicators T and stock entities O in the text based on financial lexicons and the process of time reference and stock mention extraction. P will be converted to a specific reverse token if there is a negative term at the front.
- For each P, we find the nearest T to the left of it as the reference time of this tuple. If there is no T, the time part will be replaced with the token NO_TIME.
- To determine the object each P expresses toward, we look both left and right from P to find the nearest stock entity. If an object O is the nearest entity of more than one opinion terms, it will be assigned to the TPO tuple whose predictive term P is closest to O. Then we search the second nearest stock entity as the object candidate of other TPO tuples. Analogously, if no entity is found, the object part will be tagged as NO_TAR.

The extracted TPO tuples are extremely sparse due to the large number of stock entities as well as opinion and temporal synonyms, which will affect the fitting ability of our model. To address this problem, we endow each term in TPO with its generalized class. For example, the above sequence will be generalized as { *(at present|T_now, TRANS_NEG|P_bearish, LeTV|CST), (this is excellent time|T_now, buy|P_bullish, Gree|OST), (will|T_future, pick up|P_bullish, SAIC Motor|OST)* }, where *CST* and *OST* refer to the concerned stock and other stock, separately.

Multi-view Representation Let $w_{1:n}$ be the raw input of a stock review of length n, MFN consists of two encoders to harvest multi-view representations of the review text.

- ***Word Level Encoder***

The word level encoder first transforms the input sequence into a matrix $\mathbf{w}_{1:n} = [\mathbf{w}_1\mathbf{w}_2...\mathbf{w}_n] \in \mathbb{R}^{d_w \times n}$, where $\mathbf{w}_k \in \mathbb{R}^{d_w}$ is the d_w dimensional embedding of the k-th word. Then the word embedding is enriched with associated knowledge attributes including time, polarity and negation through matching the created lexicons in financial domain. Specifically, each word w_k is represented by concatenating $\mathbf{w}_k$ with embeddings of the m extended features $\mathbf{e}_{f_{1:m}}$, denoted as:

$$\mathbf{e}_k = \mathbf{w}_k \oplus \mathbf{e}_{f_{1_k}} \oplus \ldots \oplus \mathbf{e}_{f_{m_k}} \tag{1}$$

Given the whole sentence $\mathbf{W} \in \mathbb{R}^{d \times n}$, where $d = d_w + \sum_{i=1}^{m} d_{f_i}$ and d_{f_i} is the dimension of the i-th knowledge embedding, we apply multiple filters $\mathbf{k} \in \mathbb{R}^{d \times l}$ with varying window sizes l to distill n-gram local patterns in the text. A convolution operator toward sub-matrix $\mathbf{W}_{i:i+l-1}$ with respect to $\mathbf{k}$ is denoted as:

$$c_i^k = RELU(\mathbf{k} * \mathbf{W}_{i:i+l-1} + b) \tag{2}$$

where b is the bias, $*$ is the convolution operation and $RELU$ (Nair and Hinton 2010) is a nonlinear activation function. A max-pooling operation is then adopted to extract the most salient features:

$$\tilde{c}^k = max\{c_1^k, c_2^k, ..., c_{n-l+1}^k\} \tag{3}$$

Outputs of each filter $\tilde{c}^{k_j}$ are flatted and concatenated to form the fused word-level multi-view representation vector:

$$s_{WE} = [\tilde{c}^{k_1}\tilde{c}^{k_2}...\tilde{c}^{k_z}] \in \mathbb{R}^z \tag{4}$$

where z is the number of filters.

- ***TPO Sequence Encoder***

Take the generalized TPO sequence as input, this encoder is used to better model the hidden dependency and matching relationship of multiple opinions and stock entities in a review. We propose a *word-class-aligned* and *multi-channel* representation approach to jointly learn embeddings of each feature space. Given a generalized TPO triplet, we first obtain the annotation of each aspect by concatenating the word and class embeddings:

$$\mathbf{M}_{TPO} = [\mathbf{e}_{t\_w} \oplus \mathbf{e}_{t\_c}, \mathbf{e}_{p\_w} \oplus \mathbf{e}_{p\_c}, \mathbf{e}_{o\_w} \oplus \mathbf{e}_{o\_c}] \in \mathbb{R}^{d_s * 3} \tag{5}$$

where d_s denotes the concatenated dimension.

For h triplets extracted from a review, we align and stack the embedding matrices to construct the TPO sequence like a $2D$ image, where each aspect serves as a feature channel:

$$\mathbf{M}_{TPO_s} = [[\mathbf{e}_{t_1}\mathbf{e}_{p_1}\mathbf{e}_{o_1}]...[\mathbf{e}_{t_h}\mathbf{e}_{p_h}\mathbf{e}_{o_h}]] \in \mathbb{R}^{d_s \times h \times 3} \tag{6}$$

After getting the multi-channel input matrix $\mathbf{M}_{TPO_s}$, similar to the word level encoder, we apply v convolutional filters $\mathbf{q} \in \mathbb{R}^{d_s \times l \times 3}$ with varying window sizes l and max-pooling operations to model the interactive relationship of the multiple matching tuples. Outputs are concatenated to obtain the fused vector which is denoted as $s_{TE} \in \mathbb{R}^v$.

Stance Prediction With the output vectors that are jointly learned from the two encoders, we formulate the final representation of a review text as:

$$s_o = s_{WE} \oplus s_{TE} \in \mathbb{R}^o \tag{7}$$

where $o = z + v$. Then the probability $g(r, s)$ is calculated via a single-layer perceptron with softmax function:

$$g(r, s) = softmax(\mathbf{W}_o s_o + \mathbf{b}_o) \tag{8}$$

where $\mathbf{W}_o$, $\mathbf{b}_o$ are parameters. We learn $g(\cdot, \cdot)$ by minimizing cross entropy with $\mathcal{D}$. Let Θ denotes the parameters of MFN, the objective function $\mathcal{L}(\mathcal{D}, \Theta)$ is:

$$-\sum_{i=1}^{N}[y_i log(g(r_i, s_i)) + (1 - y_i) log(1 - g(r_i, s_i))] \tag{9}$$

Expert-based Stance Aggregation Module

Investors have different expertise toward different stocks, and the expertise is changing over time. In order to distinguish high quality investment opinions and reduce misleading noises from non-experts, we design a dynamic expert mining system and apply it to each stock. The procedure is summarized in Algorithm 1.

Algorithm 1 Expert mining system of stock s on date d

```
Require:
    current date $d$, target stock $s$;
    review set $R_s$ toward $s$;
    assessment window size $e$;
Ensure:
    expert ranking of stock $s$ on date $d$;
$R_{s,d^*} \leftarrow filter(R_{s,d-e:d})$;
judge bullish/bearish/neutral$\leftarrow MFN(R_{s,d^*}, s)$;
$U \leftarrow$ all the authors of $R_{s,d^*}$;
for each $u \in U$ do
    $C_{u,s} \leftarrow 0; I_{u,s} \leftarrow 0; R_{u,s} \leftarrow$reviews of $R_{s,d^*}$ posted by $u$;
    for each $r \in R_{u,s}$ do
        check stock price change $\rho$ on the next trading day;
        if $(r \in bullish \bigcap \rho \in rise) or (r \in bearish \bigcap \rho \in fall)$ then
            $C_{u,s}++$;
        else
            $I_{u,s}++$;
        end if
    end for
    $A_{u,s} = \frac{C_{u,s}}{C_{u,s}+I_{u,s}}$;
end for
$T_{u,s} \leftarrow$ rank($A_{u,s}$ for each $u \in U$);
return $(id_{expert}, weight_{expert}) \leftarrow$50 highest of $T_{u,s}$;
```

Intuitively, we judge whether a user is an expert on a stock based on the predictive accuracy of the stock movement. We first aggregate the reviews written by user u on stock s over a past period of time e, and leverage the proposed MFN model to determine the user's bullish/bearish predictive position for each review. Then, we align and compare the opinion indicators with the practical stock price variations. The forecasting opinion is correct if a bullish review is followed by a stock price rise, or a bearish review is followed by a fall. Let $C_{u,s}$, $I_{u,s}$ be the number of correct and incorrect predictive opinions, the accuracy of u toward s is defined as:

$$A_{u,s} = \frac{C_{u,s}}{C_{u,s} + I_{u,s}} \tag{10}$$

In this work, the evaluation interval e is set to 90 days[3]. Within this period, the investor's forecast reliability is relatively stable, and the amount of published reviews is sufficient for investigating the expertise.

For stock prediction, we estimate the quality of each review according to the expert ranking of its author, and only utilize review opinions of the fifty highest-ranked investors. Note that in our framework, the expert ranking system is dynamically updated on each trading day, so as to ensure the effectiveness of the expert-based aggregation mechanism.

Individual Stock Prediction Module

Formalization With historical time-series features $[f_{t-bw}, \dots, f_{t-1}]$ for stock s, where bw is the window size of backward trading days, our goal is to learn $p(s,t)$ which measures the probability that the close price of s will rise or fall on transaction date t.

Method To model the sequential relationship of historical trading indicators, we first employ the gated recurrent units (GRU) (Chung et al. 2014) to encode the feature sequence $[f_{t-bw}, \dots, f_{t-1}]$ in the chronological order. The structure of GRU is described as:

$$\begin{aligned} z_k &= \sigma(\mathbf{W}_z x_k + \mathbf{V}_z h_{k-1}) \\ r_k &= \sigma(\mathbf{W}_r x_k + \mathbf{V}_r h_{k-1}) \\ \tilde{h}_k &= tanh(\mathbf{W}_h x_k + \mathbf{V}_h (r_k \odot h_{k-1})) \\ h_k &= (1 - z_k) \odot h_{k-1} + z_k \odot \tilde{h}_k \end{aligned} \tag{11}$$

where σ is the sigmoid function, z_k and r_k are the update and reset gates, $\odot$ denotes the element-wise multiplication, and $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{W}_h$, $\mathbf{V}_z$, $\mathbf{V}_r$, $\mathbf{V}_h$ are parameters. The annotation for the k-th day is represented by the hidden state h_k. With the hidden vectors $[h_{t-bw}, \dots, h_{t-1}]$, we define $p(s,t)$ as:

$$p(s,t) = softmax(\mathbf{W}_p L[h_{t-bw}, \dots, h_{t-1}] + \mathbf{b}_p) \tag{12}$$

where $\mathbf{W}_p$ and $\mathbf{b}_p$ are parameters. We consider three parameterizations for $L[h_{t-bw}, \dots, h_{t-1}]$: (1) take the last hidden state for prediction, then $L[h_{t-bw}, \dots, h_{t-1}] = h_{t-1}$. (2) the hidden states are linearly combined, then $L[h_{t-bw}, \dots, h_{t-1}] = \sum_{i=t-bw}^{t-1} w_i h_i$. (3) a temporal-aware attention mechanism is employed to dynamically combine the hidden states. Then, $L[h_{t-bw}, \dots, h_{t-1}]$ is calculated as:

$$\begin{aligned} u_i &= tanh(\mathbf{W}_s h_i + \mathbf{b}_s) \\ \alpha_i &= \frac{exp(u_i^T u_s)}{\sum_k exp(u_i^T u_s)} \\ L[h_{t-bw}, \dots, h_{t-1}] &= \sum_i \alpha_i h_i \end{aligned} \tag{13}$$

where $\mathbf{W}_s$, $\mathbf{b}_s$ are parameters. Each annotation is first fed through a one-layer perceptron to get a hidden representation u_i. u_s is a virtual context vector which is randomly initialized and jointly learned during training.

Both the parameterizations (2) and (3) aim to learn weights for $\{h_{t-bw}, \dots, h_{t-1}\}$ and highlight important historical indicators for the final prediction. The difference is that weights in (2) are static which are entirely determined by the temporal order of trading days, while weights in (3) are dynamically computed by the daily feature vectors under the supervision of a global virtual vector u_s. We denote our stock prediction model with the three parameterizations of $L[h_{t-bw}, \dots, h_{t-1}]$ as $Pred_{last}$, $Pred_{static}$, $Pred_{dynamic}$, and empirically compare them in experiments.

[3] we tested various experiments and found that the 90-day interval performed the best.

Table 2: Statistics of stock review dataset.

	Training	*Validation*	*Test*
# bullish	3275	500	600
# bearish	3066	500	600
# neutral	8659	1000	1800
total	15000	2000	3000

Experiments

In this section, we perform two groups of experiments for a better insight into how each module affects the performance of the proposed framework. We first evaluate MFN for stance detection on real-world investment reviews. Then we perform individual stock prediction to investigate the effectiveness of the expert-based stance aggregation module and the forecasting module.

Stance Detection Evaluation

Datasets We crawled real-world stock review dataset which contains 165,823,986 reviews (1/1/2010∼6/10/2018) written by 6,136,128 users from guba, and 6,052,765 reviews (9/11/2017∼6/10/2018) by 218,049 users from xueqiu, involving 3501 stocks. Then we randomly selected 20,000 reviews from the large corpus and recruited three labelers to annotate each review into three classes (bullish / bearish / neutral). Besides the review text, labelers were also provided with the associated information including *publish time*, *author_id* and *stock_id* (we judge the target stock based on which homepage the review is published on). The majority of labels were taken as the final decision. The Fleiss' kappa (Fleiss 1971) of the labeling is 0.74, indicating a relatively high agreement. Table 2 gives the statistics.

Compared Methods We evaluate and compare MFN with the following baseline methods and variants:

- *TextCNN* (Kim 2014): applying CNN to the sequence of word embeddings to distill multi-grained n-gram features.
- *RCNN* (Lai et al. 2015): learning representation of each word with left and right contexts by a recurrent structure.
- *HAN* (Yang et al. 2016): employing a hierarchical attention network to highlight important words and sentences.
- *Transformer* (Vaswani et al. 2017): a seq2seq model entirely based on multi-head self attention and position-wise feed-forward networks. We use the encoder structure to build the representation of review text.
- *MFN w/o TPO*: a MFN variant without the TPO encoder.
- *MFN w/o KE*: a MFN variant without incorporating multi-view knowledge embeddings for the word level encoder.
- *MFN w/o* $KE_{neg|polar|time}$: MFN variants without the knowledge embedding in different aspects: negation, polarity and time, in order to learn the importance of each type of extended knowledge.

Table 3: Evaluation results of MFN and compared methods.

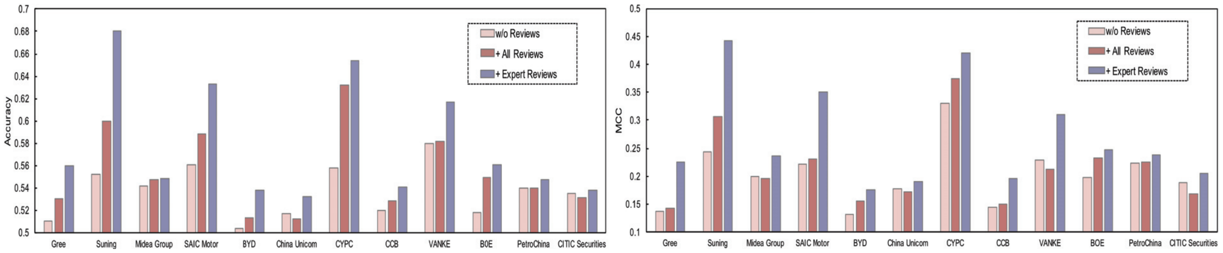
	Three-class F1	*Two-class F1*
TextCNN	69.55	62.53
RCNN	73.10	63.67
HAN	72.54	64.65
Transformer	70.54	64.23
MFN	**75.61**	**68.22**
MFN w/o TPO	72.35	65.89
MFN w/o KE	71.49	65.11
WE w/o KE_{neg}	71.84	65.07
WE w/o KE_{polar}	70.31	63.44
WE w/o KE_{time}	72.17	65.19

Experimental Details We kept at most 120 words for each review with truncating and zero-padding. The validation set was used for tuning hyper-parameters and the final comparison was conducted on the test set. Parameters for baseline algorithms were initialized as in the corresponding papers, and then were tuned to achieve optimal performance. For MFN, the number of neurons is 60 for each window size [2,3,4,5] in both encoders. Word embeddings were trained by word2vec (Mikolov et al. 2013) on the crawled large-scale review corpus and the dimension is 100. Each aspect of knowledge embeddings was randomly initialized with a dimension of 50. We use Adam (Kingma and Ba 2015) as the optimizer, the mini-batch size is 32, the learning rate is initialized as 1e-3 and gradually decreased during training. To prevent overfitting, early-stopping and dropout (Srivastava et al. 2014) with a rate of 0.5 are applied.

Experimental Results The dataset is unbalance that the neutral non-stance category accounts for the majority. To avoid this bias, we use weighted F1 value on both three-class and two-class (bullish or bearish) tests as evaluation metrics. The results are shown in Table 3. As demonstrated, MFN outperforms the baseline methods more than 2.5% on three-class evaluation and 3.5% on two-class evaluation. Comparing with the ablation variants, we can see that even without both the two encoders, MFN can still perform better than baselines on the two-class evaluation. We argue that the word level encoder has the ability to model different n-gram features based on multi-view local interactions, while the TPO sequence encoder is more helpful to enhance global interactions and understand the matching relationship. The two encoders are complementary and can jointly contribute to the performance of the entire model. In terms of extended knowledge, all the three aspects are useful and can exceed TextCNN that simply performs convolution operations on the word embedding sequence, indicating the importance of incorporating domain specific knowledge embeddings to enhance the representation of stock review texts.

Stock Prediction Evaluation

Data Collection and Experiment Setup To evaluate the effectiveness of each module in our prediction framework, three types of features are designed as shown in Table 4. Following most traditional methods, we use historical price



(a) Accuracy (b) MCC

Figure 4: Results of individual stock prediction.

features including *open, high, low, close, volume* and *CSI-300 index*[4] as the baseline, i.e., the set of FS_{pc}. Note that, it is easy to extend our model with other technical features. The other two methods, FS_{allr} and FS_{expr} incorporate the derived investment opinions. The difference is that the former uses all public reviews equally, while the later involves only high quality opinions screened through the expert mining procedure.

We retrieved price data from the publicly available API tushare[5] and aligned it with the crawled daily reviews. Data of period 1/1/2010∼31/12/2015 with 1456 transaction dates was for training, and the period 1/1/2016∼30/7/2018 with 628 transaction dates was for validation and testing with 1:1 ratio. We look back 30 days of historical information to construct the input vector for each sample. Experiments are performed on twelve well-known individual stocks in China A-share market (*Gree, CCB, VANKE, SAIC Motor*, etc.) which are from different sectors (Financial, Oil, Automobile and Electronic Information Technology, etc.). The hidden state of GRU has a dimension of 128, and the dimension of the virtual context vector is 50. Adam Optimizer is used to train the prediction model, the mini-batch size is 32, the learning rate is 1e-3, and a dropout with ratio 0.2 is applied.

Experimental Results We use two metrics, the standard measure of accuracy (Acc) and the Matthews Correlation Cofficient (MCC) to evaluate individual stock prediction, results are shown in Figure 4. We can find that incorporating expert investment opinions (FS_{expr}) can consistently achieve the best performance on the two metrics. For some stocks, the accuracy is quite high, such as 67.78% for *Suning*, and 65.38% for *CYPC*. Comparing FS_{allr} with FS_{pc}, it demonstrates that in most cases the former performs better. However, sometimes using all public investment reviews without screening does not contribute to the results (eg., *China Unicom* and *CITIC Securities*). We argue that it happens because some low quality reviews generated by non-experts can introduce lots of noise and misleading transaction signals to the prediction model in both training and testing phrases. Moreover, it indicates that both the stance detection module and the expert-based aggregation module are significantly important for the final stock forecasting.

Table 4: Input feature types for stock prediction.

Method	*+ numeric price*	*+ all reviews*	*+ expert reviews*
FS_{pc}	✓		
FS_{allr}	✓	✓	
FS_{expr}	✓		✓

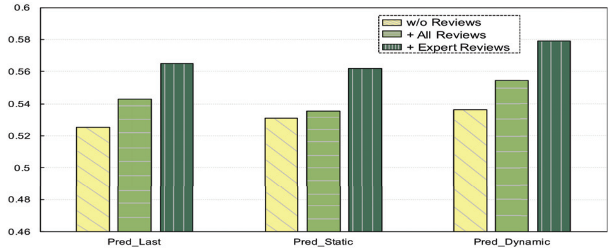


Figure 5: The effectiveness of different aggregation schemes for modeling time-series features with GRU.

We then validate the three aggregation schemes for modeling the sequence of GRU hidden vectors $\{h_{t-bw}, \ldots, h_{t-1}\}$, average accuracy of the twelve individual stocks are shown in Figure 5. We can see that $Pred_{dynamic}$ is better than $Pred_{last}$ and $Pred_{static}$ in terms of all the three different input feature types. In other words, even though the GRU has the ability to select useful information from the input sequence and accumulate them in the final state with its gate mechanism, the temporal-aware attention design can still contribute to improving the performance for this task.

Conclusion and Future Work

In this paper, we present a framework of exploiting expert-based high quality investment opinions for stock prediction. We propose MFN, a multi-view fusion network for detecting investment opinions from public reviews. Then we show how to identify and follow expert investors to selectively aggregate high-quality opinion indicators for stock forecasts. Experiments on real data show the effectiveness of each module in our framework. In the future, we shall study how to distill other investment indicators from multi-source heterogeneous information such as financial news, company networks to improve the neural stock prediction method.

[4] A weighted stock market index to replicate the performance of top 300 stocks on the Shanghai & Shenzhen stock exchanges.

[5] http://tushare.org

Acknowledgments

We are grateful to Shun Li (University of International Relations) and Wei Chen (Peking University) for their invaluable guidance, support and contribution in regard to this research and resulting paper.

References

Augenstein, I.; Rocktäschel, T.; Vlachos, A.; and Bontcheva, K. 2016. Stance detection with bidirectional conditional encoding. In *EMNLP*, 876–885.

Bao, W.; Yue, J.; and Rao, Y. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one* 12(7):e0180944.

Bar-Haim, R.; Dinur, E.; Feldman, R.; Fresko, M.; and Goldstein, G. 2011. Identifying and following expert investors in stock microblogs. In *EMNLP*, 1310–1319.

Bollen, J.; Mao, H.; and Zeng, X. 2011. Twitter mood predicts the stock market. *Journal of computational science* 2(1):1–8.

Chen, W.; Zhang, X.; Wang, T.; Yang, B.; and Li, Y. 2017. Opinion-aware knowledge graph for political ideology detection. In *IJCAI*, 3647–3653.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*.

Ding, X.; Zhang, Y.; Liu, T.; and Duan, J. 2014. Using structured events to predict stock price movement: An empirical investigation. In *EMNLP*, 1415–1425.

Ding, X.; Zhang, Y.; Liu, T.; and Duan, J. 2015. Deep learning for event-driven stock prediction. In *IJCAI*, 2327–2333.

Fama, E. F. 1965. The behavior of stock-market prices. *The journal of Business* 38(1):34–105.

Faulkner, A. 2014. Automated classification of stance in student essays: An approach using stance target information and the wikipedia link-based measure. *Science* 376(12):86.

Ferreira, W., and Vlachos, A. 2016. Emergent: a novel dataset for stance classification. In *NAACL: Human language technologies*, 1163–1168.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76(5):378.

Hu, Z.; Liu, W.; Bian, J.; Liu, X.; and Liu, T.-Y. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *WSDM*, 261–269.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, 1746–1751.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Lai, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, 2267–2273.

Li, Q.; Wang, T.; Li, P.; Liu, L.; Gong, Q.; and Chen, Y. 2014. The effect of news and public mood on stock movements. *Information Sciences* 278:826–840.

Liu, Q.; Cheng, X.; Su, S.; and Zhu, S. 2018. Hierarchical complementary attention network for predicting stock price movements with news. In *CIKM*, 1603–1606.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, 3111–3119.

Mohtarami, M.; Baly, R.; Glass, J. R.; Nakov, P.; Màrquez, L.; and Moschitti, A. 2018. Automatic stance detection using end-to-end memory networks. In *NAACL*, 767–776.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*, 807–814.

Nguyen, T. H., and Shirai, K. 2015. Topic modeling based sentiment analysis on social media for stock market prediction. In *ACL*, 1354–1364.

Ou, G.; Chen, W.; Wang, T.; Wei, Z.; Binyang, L.; Yang, D.; and Wong, K.-F. 2014. Exploiting community emotion for microblog event detection. In *EMNLP*, 1159–1168.

Si, J.; Mukherjee, A.; Liu, B.; Li, Q.; Li, H.; and Deng, X. 2013. Exploiting topic based twitter sentiment for stock prediction. In *ACL*, 24–29.

Si, J.; Mukherjee, A.; Liu, B.; Pan, S. J.; Li, Q.; and Li, H. 2014. Exploiting social relations and sentiment for stock prediction. In *EMNLP*, 1139–1145.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Tetlock, P. C. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance* 62(3):1139–1168.

Tu, W.; Cheung, D. W.-L.; and Mamoulis, N. 2015. Time-sensitive opinion mining for prediction. In *AAAI*, 4214–4215.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 6000–6010.

Walker, M. A.; Anand, P.; Abbott, R.; and Grant, R. 2012. Stance classification using dialogic properties of persuasion. In *NAACL: Human language technologies*, 592–596.

Wang, J.-J.; Wang, J.-Z.; Zhang, Z.-G.; and Guo, S.-P. 2012. Stock index forecasting based on a hybrid model. *Omega* 40(6):758–766.

Xie, B.; Passonneau, R. J.; Wu, L.; and Creamer, G. G. 2013. Semantic frames to predict stock price movement. In *ACL*, 873–883.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *NAACL*, 1480–1489.

Zhang, M.; Zhang, Y.; and Vo, D. T. 2015. Neural networks for open domain targeted sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 612–621.

Zhao, B.; He, Y.; Yuan, C.; and Huang, Y. 2016. Stock market prediction exploiting microblog sentiment analysis. In *IJCNN*, 4482–4488.