IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 23, 2020, accepted July 22, 2020, date of publication August 5, 2020, date of current version August 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014506

Multi-Element Hierarchical Attention Capsule Network for Stock Prediction

JINTAO LIU[1], HONGFEI LIN[1], (Member, IEEE), LIANG YANG[1], BO XU[1,2], (Member, IEEE), AND DONGZHEN WEN[1]

[1]School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[2]State Key Laboratory of Cognitive Intelligence, iFLYTEK, Hefei 230088, China

Corresponding author: Hongfei Lin (hflin@dlut.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61632011, Grant 61702080, Grant 61602079, and Grant 61806038; in part by the Ministry of Education, Humanities, and Social Science Project under Grant 16YJCZH12 and Grant 19YJCZH199; in part by the Fundamental Research Funds for the Central Universities under Grant DUT18ZD102DUT19RC(4)016; in part by the National Key Research Development Program of China under Grant 2018YFC0832101; in part by the China Postdoctoral Science Foundation under Grant 2018M631788 and Grant 2018M641691; and in part by the Foundation of State Key Laboratory of Cognitive Intelligence, iFLYTEK, China, through Intelligent Medical Question Answering Based on User Profiling and Knowledge Graph, under Grant COGOS-20190001.

ABSTRACT Stock prediction is a challenging task concerned by researchers due to its considerable returns. It is difficult because of the high randomness in the stock market. Stock price movement is mainly related to the capital situation and hot events. In recent years, researchers improved prediction accuracy with news and social media. However, the existing methods do not take into account the different influences of events. To solve this problem, we propose a multi-element hierarchical attention capsule network, which consists of two components. The former component, multi-element hierarchical attention, quantifies the importance of valuable information contained in multiple news and social media through its weights assignment process. And the latter component, capsule network, learns more context information from the events through its vector representation in the hidden layer. Moreover, we construct a combined data set to maintain the complementarity between social media and news. Finally, we achieve better results than baselines, and experiments show that our model improves prediction accuracy by quantifying the different influences of events.

INDEX TERMS Stock prediction, hierarchical attention, capsule network, text mining, natural language processing.

I. INTRODUCTION

The stock prediction has already been researched for decades [1], due to its great value in seeking to maximize stock investment profit. According to the Efficient Market Hypothesis (EMH) [2], stock price movement is thought to be related to the news. Each day, a listed company often has more than one piece of relevant news that may have different influences. However, the existing methods ignore the different impacts of events on the stock [3]–[5]. These methods are based on the hypothesis that every news has equal influence, but the fact is not always the case. For example, after the event, ''Playing before a crowd littered with celebrities - from Spike Lee to former President Barack Obama - Williamson was hurt in the opening minute of the game as his Nike PG 2.5, from Oklahoma City Thunder star Paul George's signature sneaker line, tore apart.'',[1] Nike, which manufactured the shoes Williamson was wearing, was feeling the impact of the injury. The company's stock price was down about 1 percent during the second trading day as the sportswear manufacturer became the target of ridicule on social media. This bad news played a decisive role in the stock market, even if there were other news of Nike which should be positive.

For the stock market, public news and social media are two primary resources for stock prediction, and meanwhile, most of the contents on social media are news-related. In the U.S., Twitter spreads information faster and ensures high coverage of important information contained in the news [6]. From

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Afzal.

[1]https://www.apnews.com/65152cf7a3534316b677b1f2744bf7a5

this perspective, Twitter is a good source for stock prediction. We aim to combine the news and tweets to predict the U.S. stock price movements. With the news, we get all the events happened on a trading day, and from the tweets, we obtain the comments based on the events people-focused. The stock market is often driven by hot news. And the tweets related to news reflect the heat of the event, as well as the attitude of investors towards the event. Hence, it is necessary to combine the news and tweets to cover significant information for prediction.

Besides, we mainly research the different influences of events on stock market prediction. In our work, the weight of text information is reassigned through the model, which is in essence to redistribute the weight of events. In the stock market, people mainly obtain event information from news and social media. The stock price data reflects not only the event influence but also non-event information such as the trading operation of the capitals. The later information cannot be obtained through news or social media. If we introduce the stock price data, the different influences of events cannot be verified, and it is not consistent with our research purpose. Meanwhile, the baselines mainly research the impact of events on the stock market through text information. To keep fair with these methods, we choose text data to verify the effectiveness of our model and do not adopt the price data or other indicators.

To increase the weights of critical information from the multiple news and tweets, we propose a multi-element hierarchical attention model to handle the texts. In the multi-element hierarchical attention layer, we ensure the information captured from the news and tweets is complementary with a measurement method [7]. And then we join a capsule network to obtain more context information from the multi-element hierarchical attention layer for improving prediction accuracy. Experiments show that our model is effective, and the main contributions are as follows:

1) We raise the different influences of events in the stock prediction task and prove it is valuable to be solved through the experiment of Virtual Trading.

2) We improve the hierarchical attention mechanism to accommodate multi-element inputs which can be applied to other text tasks.

3) We propose a novel approach for financial text processing and achieve state-of-the-art results, which have a good generalization ability to other financial tasks.

II. RELATED WORK

A. STOCK MARKET PREDICTION

A series of researches have predicted stock trends adopting text information from news and tweets [8]–[12]. Pioneering work extracted textual features from texts, such as bags-of-words, noun phrases, named entities, and structured events. Ding *et al.* [13] showed that structured events from open information extraction [14], [15] achieve better performance compared to conventional features, as they can capture structured relations. However, one disadvantage of the structured representation of the event is that it leads to increased sparsity, which potentially limits the prediction ability. Ding *et al.* [16] proposed to address this issue by representing structured events using event embeddings, which are dense vectors. After that, Ding *et al.* [17] proposed to leverage ground truth from a knowledge graph to enhance event embeddings. These methods improved prediction performance in terms of text representation.

Recently, Xu *et al.* [18] introduced recurrent, continuous latent variables for better treatment of stochasticity, and used neural variational inference to address the intractable posterior inference and also provided a hybrid objective with temporal auxiliary to flexibly capture predictive dependencies. Shah *et al.* [19] retrieved, extracted, and analyzed the effects of news sentiments on the stock market.

With the widespread use of attention mechanism, various models based on attention mechanism begin to appear in financial forecasting tasks due to its effectiveness. Wu *et al.* [20] proposed a novel Cross-modal attention based Hybrid Recurrent Neural Network (CH-RNN) to obtain better results. Liu *et al.* [7] adopted a two-level attention mechanism to quantify the importance of the words and sentences in given news, and designed a measurement for calculating the attention weights to avoid capturing redundant information in the news title and content. Yang *et al.* [21] proposed a dual-layer attention-based neural network and introduced a knowledge-based method to adaptively extract relevant financial news. However, they only considered a single data source, and that is why they can not quantify the impact of events. In this paper, we improve the attention mechanism based on the two-level attention mechanism [7], adapting it to multi-element inputs.

B. CAPSULE NETWORK

Hinton *et al.* [22] firstly introduced the concept of ''capsules'' to address the representational limitations of CNNs and RNNs. Capsules with transformation matrices allowed networks to automatically learn part-whole relationships. Consequently, Sabour *et al.* [23] proposed capsule networks that replaced the scalar-output feature detectors of CNNs with vector-output capsules and max-pooling with routing-by-agreement. The capsule network has shown its potential by achieving a state-of-the-art result on MNIST data. Unlike max-pooling in CNN, the capsule network does not throw away information about the precise position of the entity within the region. For low-level capsules, location information is place coded by which the capsule is active. Xi *et al.* [24] further tested out the application of capsule networks on CIFAR data with higher dimensionality. For achieving higher accuracy on the small NORB data set, Hinton *et al.* [25] proposed a new iterative routing procedure between capsule layers based on the EM algorithm. And Zhang *et al.* [26] generalized existing routing methods within the framework of weighted kernel density estimation.

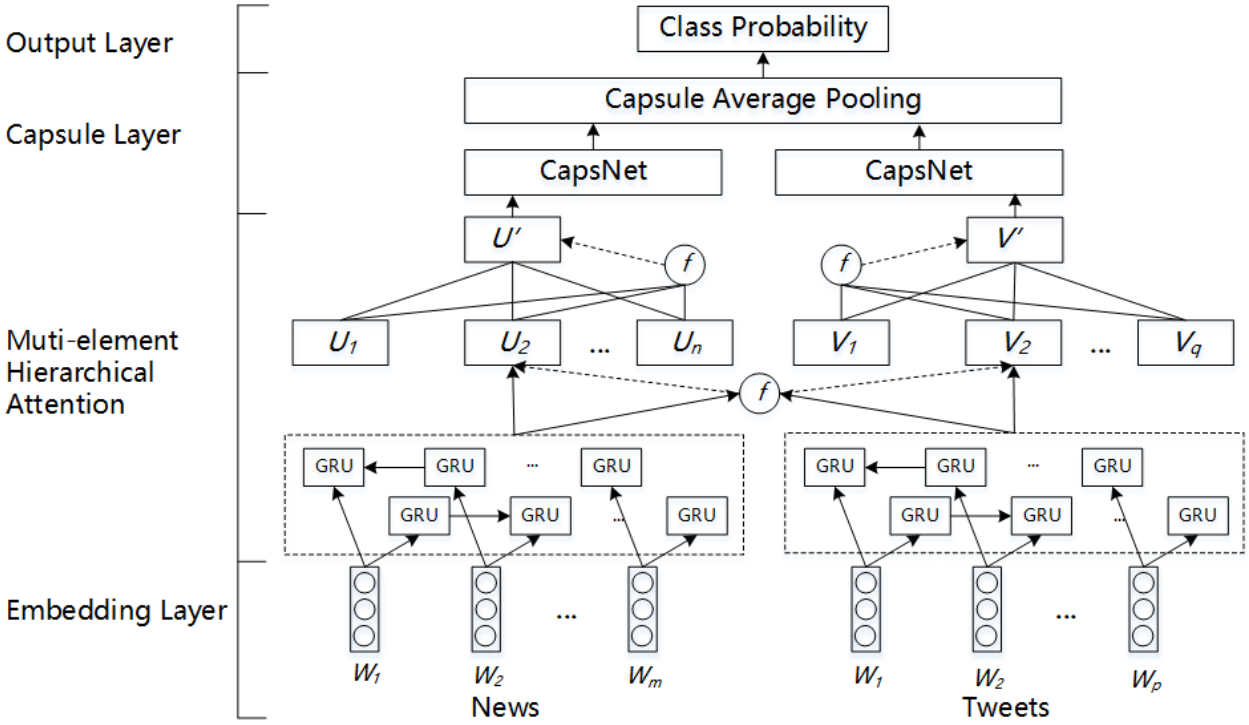


FIGURE 1. Multi-element hierarchical attention capsule network.

The above methods are mainly utilized in the task of image classification. Recently, Zhao *et al.* [27] firstly investigated the performance of capsule networks in NLP tasks. They also demonstrated that the capsule network retains more contextual characteristics. On the stock market, trading date limits the size of the data because there are fewer than 300 trading dates in a year. It means a stock has only fewer than 300 labels in a year. Therefore, information retention is crucial in the stock prediction task. Hence, we introduce the capsule network into our model to reduce the loss of information due to dimensional reduction.

III. METHOD

Through the multi-element hierarchical attention, we quantify the event influences. To capture the decisive information on the stock trends from news and tweets, we enhance the relationship by a measurement called score-inverse similarity [7] (S-IS) to calculate the attention weights. It makes the weight bigger when the correlation is stronger.

To receive more contextual information for prediction, we choose a capsule network [27] after the multi-element hierarchical attention, which adopts vector representation in the intermediate results. A vector representation keeps more characteristics than a scalar one.

Additionally, we find that using the news titles gets a better result than adding the contents in our experiments. Even if we sample the first few words of contents, the result is still worse than the titles only. Hence, we just employ news titles and tweets in this paper. Figure 1 shows the architecture of the proposed model, namely Multi-element Hierarchical Attention Capsule Network (MHACN).

We learn that in a three-categories stock prediction task a common practice is to categorize a series of targets with exceptionally minor movement to another ''preserve'' class by setting upper and lower thresholds on the stock price change. Since we aim at the binary classification of stock changes identifiable from events, inspired by Xu *et al.* [18], we set two particular thresholds, -0.5% and 0.55%. And based on this method, we balance the two categories to a similar proportion. Meanwhile, the targets with exceptionally minor movement suggest less affected by events, which do not make sense to the influence of events. For these reasons, we remove the ''preserve'' class and adopt binary classification as existing methods [18], [28].

The same as most researches in the stock prediction task, we predict the stock price movements on a trading day *td* and calculate the labels from price data for the texts. The labels contain two values, where 1 denotes rise and 0 denotes fall,

$$y = 1(p^c_{td} > p^c_{td-1}) \tag{1}$$

where p^c_{td} denotes the adjusted closing price which is adjusted to actions affecting stock trends, e.g. splits. Before our work, the adjusted closing price has been adopted for stock prediction [16], [18].

A. MULTI-ELEMENT HIERARCHICAL ATTENTION

Based on the Hierarchical Complementary Attention Network (HCAN) model [7], we propose a multi-element hierarchical attention mechanism to quantify the importance of the news and tweets. HCAN adopt a two-level attention mechanism to quantify the importance of the words and sentences in given news but only apply a one-level attention mechanism to quantify the importance of the sentences in given contents. It is essentially the processing method of homologous data. In this paper, we change it into a new structure with multi-element inputs. As Figure 1 showed, that is the two-level attention mechanism structure of multi-element inputs.

Generally, hierarchical attention captures valuable information through the redistribution of weights. In this task, tweets directly reflect people's concerns, so we must consider the interaction between the news and tweets for researching the impact level of events. The measurement called S-IS solves this problem by calculating the shared attention weights.

As Figure 1 showed, we first encode each word of all the texts by Bi-GRU. After getting the word embeddings, we adopt the word-level attention to obtain the weights of each word, and through the sentence-level attention, we obtain the weights of each sentence in the news titles and tweets. Next, we employ the weighted average values of sentence vectors to represent the texts. In this way, we achieve the updated representation of the news titles and tweets respectively.

1) BI-GRU FOR WORD EMBEDDING

On each trading day, suppose one stock has n titles, while each title has m words, and all the titles contain $u = m * n$ words. Similarly, we suppose the stock has q tweets, while each tweet has p words, and all the tweets contain $v = p * q$ words. We use a bidirectional GRU [29] to finetune the pre-trained word embeddings (word2vec). And then, we concatenate the hidden states of the forward-GRU $\overrightarrow{h_w(i)} = \overrightarrow{GRU}(w_i)$ and backward-GRU $\overleftarrow{h_w(i)} = \overleftarrow{GRU}(w_i)$ to obtain the word embedding $h_w(i) = [\overrightarrow{h_w(i)}, \overleftarrow{h_w(i)}]$, where w_i represent the i-th words. In this part, we test some other methods of pre-trained word embeddings, such as Bert [30], and receive similar results. For the sake of algorithm efficiency, we adopt the simple method, word2vec.

2) WORD-LEVEL ATTENTION-BASED S-IS

After getting the word embeddings of the news and tweets from Bi-GRU, we quantify the significance of each word through word-level attention. Here, we calculate the attention matrix $K \in R^{u \times v}$,

$$K_{i,j} = f(h_{news}(i), h_{tweet}(j)) \tag{2}$$

where $h_{news}(i) \in R^{2d} (i \in [1, u])$ and $h_{tweet}(j) \in R^{2d} (j \in [1, v])$ represent the i-th words in the news titles and the j-th words in the tweets, respectively. d is the dimension of the unidirectional GRU. And we calculate $f(h_{news}(i), h_{tweet}(j))$ as follows:

$$f(h_{news}(i), h_{tweet}(j)) = \frac{Corr(h_{news}(i), h_{tweet}(j))}{Sim(h_{news}(i), h_{tweet}(j))} \tag{3}$$

$$Corr(h_{news}(i), h_{tweet}(j)) = h_{news}(i) W_1 h_{tweet}(j)^T \tag{4}$$

$$Sim(h_{news}(i), h_{tweet}(j)) = \frac{h_{news}(i) h_{tweet}(j)^T}{\|h_{news}(i)\| \|h_{tweet}(j)\|} \tag{5}$$

where $W_1 \in R^{2d \times 2d}$ denotes the internal weight matrix learned in the training process. The function *Corr* measures the correlations between different words of the news titles and tweets. And the function *Sim* measures the extent of similarity between any two words, which enhances the complementarity between the news titles and tweets. $\|x\|$ denotes the L2-norm of vector x.

After we get the attention matrix K which is based on S-IS, we compute the attention vectors α and β for the words in the news titles and tweets respectively. And then we use them to derive the sentence representation of the two resources. For the news titles on a trading day, there are n titles, while each title contains m words. We get the word attention vectors α as follows:

$$\alpha_s = softmax(\frac{1}{v}\sum_{i=1}^{v} K^s_{1,i}, \ldots, \frac{1}{v}\sum_{i=1}^{v} K^s_{m,i}) \tag{6}$$

where $\alpha_s \in R^m (s \in [1, n])$ denotes the word attention vector of the s-th title in the news titles. And then, we represent U_s of the s-th title by:

$$U_s = \sum_{j=1}^{m} \alpha_{sj} h_{cs}(j), s \in [1, n] \tag{7}$$

where $h_{cs}(j)$ denotes the j-th word in the s-th title. Different from Liu *et al.* [7], we calculate the word attention for every resource. And the word attention vectors β and the representation V_r of the r-th tweet as follows:

$$\beta_r = softmax(\frac{1}{u}\sum_{i=1}^{u} K^r_{1,i}, \ldots, \frac{1}{u}\sum_{i=1}^{u} K^r_{p,i}) \tag{8}$$

$$V_r = \sum_{j=1}^{p} \beta_{rj} h_{cr}(j), r \in [1, q] \tag{9}$$

where $\beta_r \in R^p (r \in [1, q])$ denotes the word attention vector of the r-th tweet, and $h_{cr}(j)$ denotes the j-th word in the r-th tweet.

3) SENTENCE-LEVEL ATTENTION-BASED S-IS

After getting the representations of the news titles and tweets in word-level attention, we further calculate the significance of each sentence in the two resources through sentence-level attention.

The same processing as the word-level attention, we get the sentence-level attention matrix $K' \in R^{n*q}$ as follows:

$$K'_{s,r} = f(U_s, V_r), \quad s \in [1, n], r \in [1, q] \tag{10}$$

Besides, with the matrix K', we obtain the sentence attention vector α'_s of the title and the sentence attention vector β'_r of the tweet. In the end, we derive the representation U' and V' of the s-th title and the r-th tweet respectively as follows:

$$U' = \alpha'_s U_s, \quad s \in [1, n]; V' = \beta'_r V_r, r \in [1, q] \tag{11}$$

B. CAPSULE NETWORK

For a stock, predicting its trend requires comprehensive information. Generally, hot news often has a lot of relevant comments, and how to retain more comprehensive context information is a critical point.

We utilize the capsule network due to its effectiveness in feature retention. The research by Zhao *et al.* [27] shows that in the NLP tasks, the capsule network can better retain text features compared with CNNs and RNNs and improve classification accuracy. In the text representation, the CNNs methods usually abandon some characteristics in their intermediate results due to its mapping way. For example, the operation of CNN max-pooling only retains the maximum convolution result from the previous layer, and the average pooling calculates the average convolution results from the previous layer. These operations are scalar transfer. However, the capsule network retains the whole intermediate results as the input vector to the next layer which reduces the information loss compared to CNNs, that is, it retains more context characteristics. This is more conducive to a text representation, and related research by Zhao *et al.* [27] indicates the capsule network is more effective. Hence, we choose a capsule network to keep the information obtained from the multi-element hierarchical attention layer more comprehensive.

Different from the method of CNNs and RNNs, the capsule network adopts a vector representation to save more features instead of the scalar representation. It is vectorization improvement based on a conventional convolution neural network (CNN). The nonlinear mapping between hidden layers was finally solved by the dynamic routing, which is proposed by Sabour [23]. It displaces the pooling operation used in the CNN. Thus, the whole process maintains the position information for features, which is beneficial to the text representation. Figure 2 shows the structure of the capsule network.

There are two outputs of the multi-element hierarchical attention, which represent the news and tweets respectively. We put them as two parts into the capsule network, and every part has the same structure, which consists of four layers: n-gram convolutional layer, primary capsule layer, convolutional capsule layer, and fully connected capsule layer.

1) N-GRAM CONVOLUTIONAL LAYER

This layer is a standard convolutional layer that extracts n-gram features at different positions of a sentence. For the news titles, $U' \in R^{n \times d}$ denotes the input representation. Each kernel K_i with a bias b emits a feature map M_i by convolution,

$$M_i = U' * K_i + b \tag{12}$$

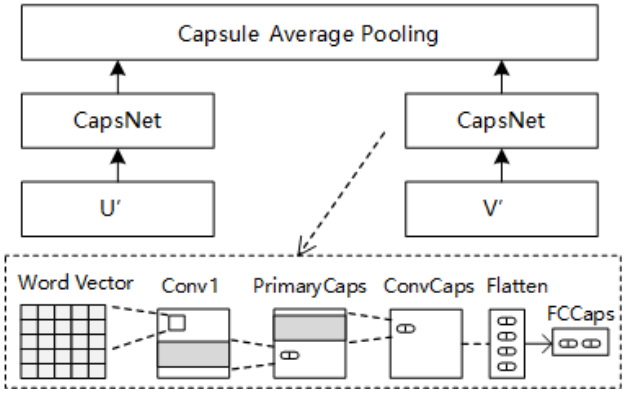


FIGURE 2. Capsule network.

where K is the kernel of convolution operation $*$. By assembling I feature maps, we have a I-channel layer.

$$M = [M_1, M_2, \ldots, M_I] \tag{13}$$

2) PRIMARY CAPSULE LAYER

The feature maps generated from the n-gram convolutional layer are fed into this layer, piecing the instantiated parts together via another convolution. This is the first capsule layer. In this layer, the capsules replace the scalar-output feature detectors of CNNs with vector-output capsules to preserve the instantiated parameters of each feature. It not only represents the intensity of activation but also records some details of the instantiated parts in the input. For images, these details represent the positions and directions and so on. But for texts, these details contain orders and contexts, which are significant in the stock news. In this way, the capsule can be regarded as a short representation of instantiated parts which can be detected by the kernel.

Sliding over the feature map M, each kernel K_j outputs a series of capsules $f_j \in R^{d_c}$ of d_c-dimension. These capsules comprise a channel F_j of the primary capsule layer.

$$F_j = g(K_j * M + b) \tag{14}$$

where g is nonlinear squash function, b is the capsule bias term. For all the J filters, the generated capsule feature maps are rearranged as follows:

$$F = [F_1, F_2, \ldots, F_J] \tag{15}$$

3) CONNECTION BETWEEN CAPSULE LAYERS

In the next layer, the capsule network generates the capsules utilizing "routing-by-agreement" [23]. It takes the place of pooling operation that usually discards the location information, which helps cluster features.

Between two neighbor layers l and $l+1$, a prediction vector $\hat{c_{j|i}} \in R^{d_c}$ is first calculated from the capsule c_i in the lower layer l, by multiplying a weight matrix W^c_{ij}.

$$\hat{c_{j|i}} = W^c_{ij} c_i \tag{16}$$

Next, in the higher layer $l+1$, a capsule g_j is generated by the linear combination of all the prediction vectors with

weights z_{ij}

$$g_j = \sum_i z_{ij} c_{j|i} \tag{17}$$

where z_{ij} are coupling coefficients computed by the a ''routing softmax'' function as follows:

$$z_{ij} = \frac{exp(b_{ij})}{\sum_j exp(b_{ij})} \tag{18}$$

where b_{ij} denotes the original logit. And the coupling coefficients are the log prior probability that capsule i should be coupled to capsule j. This process of ''routing softmax'' guarantees the sum of all the coefficients for capsule j is 1.

The length of the capsule represents the probability that the input sample has the object capsule describes, and it is limited in range from 0 to 1 by a non-linear squashing function.

$$o_j = \frac{\|g_j\|^2}{1 + \|g_j\|^2} \frac{g_j}{\|g_j\|^2} \tag{19}$$

4) CONVOLUTIONAL CAPSULE LAYER

In this layer, each capsule is connected only to a local region $K_2 \times J$ spatially in the layer below. Those capsules in the region multiply transformation matrices to learn child-parent relationships followed by routing-by-agreement [23]. We call the capsule in the layer l child capsule, and the capsule in the layer $l + 1$ is parent capsule. $K_2 \times J$ is the number of the capsules in a local region in the layer l. When the transformation matrices are shared across the child capsules, we get each potential parent capsule $\hat{c_{j|i}}$. And then, we utilize routing-by-agreement to produce parent capsules feature maps totally $(n - K_1 - K_2 + 2) \times D$ d_c-dimensional capsules in this layer, where K_1 is the N-gram size in the n-gram convolutional layer, and K_2 is the convolution kernel size in this layer. D is the number of parent capsules.

5) FULLY CONNECTED CAPSULE LAYER

The capsules in the layer below are flattened into a list of capsules, and fed into fully connected capsule layer in which capsules are multiplied by transformation matrix $W^{d_1} \in R^{E \times d_c \times d_c}$ followed by routing-by-agreement to produce final capsule $o_j \in R^{d_c}$ and its probability $a_j = |o_j|$ for each category. $E = 2$ is the number of categories.

To increase the difference between the lengths of categories, we adopt a separate margin loss $Loss_j$:

$$Loss_j = G_j max(0, m^+ - ||O_j||)^2 + \lambda(1 - G_j) max(0, ||O_j|| - m^-)^2 \tag{20}$$

where O_j is the capsule for class j; $m^+ = 0.9$ and $m^- = 0.1$ is the top and bottom margins respectively, that help to punish false positives and false negative. $G_j = 1$ if and only if class j is the ground truth. λ is the ratio coefficient, used to adjust the proportion of false positives and false negative. In our model, λ is set to 0.5.

6) THE ARCHITECTURE OF CAPSULE NETWORK

Capsule network gets two representation U' and V' from the mutli-element hierarchical attention, followed by a 3-gram ($K_1 = 3$) convolutional layer with 32 filters ($I = 32$) and a stride of 1 with ReLU non-linearity. All the other layers are capsule layers starting with a $I \times d_c$ primary capsule layer with 32 filters ($J = 32$), followed by $K_2 \times J \times d_c \times d_c$ ($K_2 = 3$) convolutional capsule layer with 16 filters ($D = 16$) and a fully connected capsule layer in sequence.

Each capsule has 16-dimensional ($d_c = 16$) instantiated parameters and their length describes the probability of the existence of capsules. The final output with two classes ($E = 2$) of the fully connected capsule layer is fed into the average pooling for producing the final results O_j and its probability $A_j = |O_j|$ for each category of stock trends.

IV. EXPERIMENTS

A. DATASET

We build a real-world data set by crawling stock news from Yahoo Finance according to an open-source data of tweets.[2] It ranges from January 2017 to January 2018 and chooses 47 stocks which have sufficient tweets from the Standard & Poor's 500 lists. We crawl the corresponding news with the stock symbol during the time. And then we calculate the values of the labels by crawling the stock price data from Yahoo Finance. The temporal nature of the stock market determines that it can not randomly shuffle the order of data sets like traditional text classification tasks. Hence, we follow Xu *et al.* [18] to split the data set with a ratio of approximately 8: 1: 1 in chronological order.

Generally, in the stock prediction task based on texts, researchers choose to predict the index trend or individual stock trends. Index trend prediction, such as DJIA prediction, utilizes a longer time data set while the individual stock prediction employs a shorter time data set. It is because they adopt the same method to calculate the labels by the price data, and the calculation is based on timestamp, that is, the texts in one day need to merge in some way and share one label. For index prediction, the time length of most data sets is about 10 years, with less than 3,000 labels. For individual stock prediction, labels are calculated through the individual stock trading data. If each stock is calculated for one year, 47 stocks are equivalent to 47 years in the index prediction. Hence, individual stock prediction usually chooses data with a shorter period. Meanwhile, the news about a stock involves only company-related events such as the company's operation, shareholder change, industry policy, and so on. These events usually occur within a year of the company's operation. therefore, for a stock, data in a year covers most types of events. To sum up, we introduce an open-source data set and add news data to verify the effectiveness of our model. To sum up, inspired by Xu *et al.* [18] and Wu *et al.* [20], we construct a short time data set to predict individual stock trends.

[2]https://github.com/wuhuizhe/CHRNN

Besides, we originally consider to employ multiple data sets to verify the effect of our model, however, with the in-depth research, we found that in the stock prediction task, there are very few open-source data sets. Most researchers construct a new data set only to verify their methods and do not public the data. Hence, unlike other classification tasks that contain a series of classic data sets, there are almost no classical data sets in this task. To verify the effectiveness of our model, we adopt the open-source data set that contains typical 47 stock to reflect the impact of events on the stock market. Also, the same event may have the opposite effect on different stocks, for example, rival companies react differently to negative news from each other. Therefore, different from other classification tasks, a text data of a stock is equivalent to a data set in the stock prediction task. For these reasons, inspired by Xu *et al.* [18], Hu *et al.* [31], and Deng *et al.* [32], we adopt one data set to verify our model as the following methods.

B. EXPERIMENTAL SETUPS

In our experiment, the dimension of the initial word embedding is set to 300, which is obtained by word2vec. We utilize the rise (1) and fall (0) of the stock price as the final output, which is obtained by comparing the values $A_j = |O_j|$ of each category in the output layer. If the $A_j > 0.5$, we get 1 as the final prediction value, otherwise we get 0. The internal weights in our model are initialized by sampling from the uniform distribution and tuned in the training process. Mini-batch is adopted during the training process, with a batch size of 128. We employ Adam optimizer with a learning rate 0.001. In the process of parameter optimization, grid search is utilized to obtain the best parameters.

C. EVALUATION METRICS

Following previous work for stock prediction [16], [18], we adopt the standard measure of accuracy and Matthews Correlation Coefficient (MCC) as evaluation metrics. MCC has a range of -1 to 1 where -1 indicates a completely wrong binary classifier while 1 indicates a completely correct one. MCC is calculated by the confusion matrix which contains the number of samples classified as true positive (tp), false positive (fp), true negative (tn) and false negative (fn):

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \quad (21)$$

D. COMPARISON METHODS

We conducted extensive experiments to compare our model with several baselines. These methods received better results than the other simple model, such as CNN and LSTM. To be fair in the data set, we merged the embeddings of news titles and tweets as the input of these baselines instead of adding weights to each other. After all, adding weights means acknowledging the different influences of events. Meanwhile, the data set is divided into the same proportions as our model in these baselines, and also the optimization process (grid search). Since Deng *et al.* [32] introduced the knowledge graph in their model, and we do not draw on any knowledge, so we do not compare with their model and other models using knowledge. Details of the baselines are described below:

- TSLDA: A generative topic model jointly learning topics and sentiments [4].
- GPC: A classic classifier using Gaussian processes from Scikit-learn.
- WB-CNN: A convolutional neural network prediction model [33] which adopts word embeddings input.
- Stru-Event: This method [16] captures the structured events in the titles, and represents an event tuple as a vector by combining all event elements for prediction.
- StockNet: A deep generative model jointly exploiting text and price signals [18]. To keep consistent with our model, we only put the text data into the baselines.
- HCAN: A hierarchical complementary attention network with S-IS [7].
- MHACN-C: To verify the effectiveness of the multi-element hierarchical attention, we move out the multi-element hierarchical attention and adopt the concatenated vectors of texts as the input to the capsule network.
- MHACN-N: To verify the impacts of tweets, we remove tweets and employ the hierarchical attention [34] with one branch of the capsule network as the model.
- MHACN-M: To verify the performance of the capsule network, we only reserve the multi-element hierarchical attention and put the concatenated representation of the texts into a softmax layer for prediction.

E. EXPERIMENTAL ANALYSIS

We tested our model from different perspectives. And the results are shown in Table 1 and Table 2.

TABLE 1. Performance of baselines and MHACN variations in accuracy and MCC.

Model	Acc.	MCC
TSLDA	54.18%	0.0635
GPC	55.70%	0.0737
WB-CNN	56.91%	0.0829
Stru-Event	57.25%	0.0936
StockNet	56.65%	0.0536
HCAN	58.11%	0.0866
MHACN-C	58.86%	0.0924
MHACN-N	57.97%	0.0627
MHACN-M	59.01%	0.1011
MHACN	**60.56%**	**0.1473**

1) COMPARISON RESULTS

From Table 1, we can see that our model obtains the best results. The scores of Acc. and MCC obtained in MHACN are both better than the baselines. In particular, the accuracy increased by 2.45%, which indicates that our model captures more valuable features for prediction from the texts. MHACN-C, only the capsule network reserved, adopts the same way with the baselines to handle the texts and receives

TABLE 2. Profit comparison between HCAN and MHACN.

Stock	HCAN	MHACN
AAPL	755$	801$
CVS	852$	935$
GE	522$	619$
KO	1,022$	1,588$
CAT	714$	593$
UTX	632$	667$

comparable results of the baselines. This suggests that just merging data is not an effective means to improve prediction performance. That is, in our model, the multi-element hierarchical attention captures more significant information in this task. And MHACN-C achieves better scores than WE-CNN verifies the capsule network keeps more valuable context information through its vector representation in the model. Besides, MHACN-M is not better than MHACN. It indicates that MHACN receives more context information due to its vector representation in the hidden layer of the capsule network. Ablation experiments also prove that the model we constructed is effective.

2) VIRTUAL TRADING

We simulated real stock trading by following the strategy proposed by Lavrenko [8], which mimics the behavior of a daily trader who uses our model.

If our model indicates that an individual stock price will increase, the fictitious trader will invest in $10,000 worth at the opening price. And then the trader will hold the stock for one day. During this time, if the stock makes a profit of 2% or more, the trader sells immediately. Otherwise, the trader sells the stock at the closing price on that day. The same strategy is used for shorting if an individual stock price will decrease. And if the trader can buy the stock at a price 1% lower than short, he/she buys it to cover. Otherwise, the trader buys the stock at the closing price.

Table 2 shows the returns of six randomly selected stocks through HCAN and MHACN with $10,000 in 20 trading days. We can see that the maximum return of KO is over 15%. Overall, the benefits of our model are considerable, especially the trading rules we set are conservative.

3) INFLUENCE EXPERIMENT

To prove the different influence of events, we randomly choose some trading days to test our model. In the results, we find some pieces of evidence to prove our thoughts. For example, on September 12, 2017, Apple released their new products. The main news and tweets that appeared on that day are shown in Figure 3. In our test, we get 1 from MHACN-N with the news while getting 0 from MHACN with the whole data set. And on the next day, the stock of Apple falls. From Figure 3, we can easily conclude that most tweets are not optimistic about the prospect of the new products even if the most news is positive. It makes intuitive sense that tweets reflect the impacts of events. And in the results, MHACN achieves better results than MHACN-N, which indicates tweets complement the news. To sum up, our model is effective in quantifying the impact of different events with the combined data set.

NEWS:
The 5 Coolest Features On Apple's New iPhone X ...
Apple unveils three new iPhones, hails ...
Is the iPhone 8 Waterproof? Apple Reveals New Features.
Apple Celebrates 10-Year Anniversary of 1st iPhone.
Apple shares fall after iPhone X release date was later ...
Lucky 8? $1,000 price tag dampens iPhone enthusiasm ...
TWEETS:
Most Americans cannot cover $500 in unexpected expenses w/o going into debt.
AppleEvent will you pay 1k for the iPhoneX?
Who is willing to spend $999?
$AAPL great features plus the size.

FIGURE 3. News and Tweets about the stock AAPL (Apple).

7 Stocks to Buy to Double Your Money in 2017
Cetera Advisors LLC Reduces Stake in Caterpillar
Bullish analyst action by Wells Fargo on Caterpillar

FIGURE 4. Examples of error analysis.

4) ERROR ANALYSIS

In Table 2, the CAT's return in our model is less than HCAN. We compared the outputs between them and analyzed the results which were wrong in MHACN but right in HCAN. Finally, we summarized two situations: firstly, a tweet was written to confuse the traders. For example in Figure 4, "7 Stocks to Buy to Double Your Money in 2017", but after this message, the stock price fell the next day. For the market makers, encouraging people to buy when they are ready to short and encouraging people to sell when they are ready to bull is the main method to obtain profit. Secondly, the events require professional knowledge to be interpreted, especially when there are two events with different points of view. For example, "Cetera Advisors LLC Reduces Stake in Caterpillar", typically, reducing stake is bad news, meaning that shareholders lack confidence in the company. But on the same day, "Bullish analyst action by Wells Fargo on Caterpillar", normally, it is good news. In this case, without relevant knowledge, it is difficult to calculate the impact of the news. Hence, for our model, it is hard to obtain the correct prediction without introducing the relevant knowledge in these conditions.

V. CONCLUSION

We demonstrate the reliability of our model and prove that quantifying the impact of events is helpful for stock prediction. Through the improved multi-element hierarchical attention mechanism, our model is adapted to the stock prediction task with a multi-element resource. It also helps us quantify the different influences of events by the shared weights. At the same time, the capsule network makes the model

retain abundant context information. Our model combines the advantages of the multi-element hierarchical attention and capsule network and meanwhile improves the prediction accuracy. Moreover, we introduce no stock data except texts in our model. It means that our model has a good generalization capability for other text-based tasks in the financial field, such as prediction based on financial reports.

However, the factors that affect the stock trend are complex, not only corporate events but also related to international environmental, national policy, capital status, etc. These factors are often lack of relevant text description. In future work, we will try to quantify these factors for improving the performance of stock prediction.

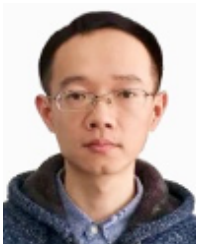
REFERENCES

[1] W. F. M. De Bondt and R. Thaler, "Does the stock market overreact?" *J. Finance*, vol. 40, no. 3, pp. 793–805, Jul. 1985.
[2] E. F. Fama, L. Fisher, M. C. Jensen, and R. Roll, "The adjustment of stock prices to new information," *Int. Econ. Rev.*, vol. 10, no. 1, pp. 1–21, 1969.
[3] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng, "Exploiting topic based Twitter sentiment for stock prediction," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2013, pp. 24–29.
[4] T. H. Nguyen and K. Shirai, "Topic modeling based sentiment analysis on social media for stock market prediction," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 1354–1364.
[5] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *Proc. 11th ACM Int. Conf. Web Search Data Mining-(WSDM)*, 2018, pp. 261–269.
[6] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li, "Comparing Twitter and traditional media using topic models," in *Proc. Eur. Conf. Inf. Retr.* Cham, Switzerland: Springer, 2011, pp. 338–349.
[7] Q. Liu, X. Cheng, S. Su, and S. Zhu, "Hierarchical complementary attention network for predicting stock price movements with news," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1603–1606.
[8] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan, "Mining of concurrent text and time series," in *Proc. KDD-Workshop Text Mining*, 2000, pp. 37–44.
[9] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFin text system," *ACM Trans. Inf. Syst.*, vol. 27, no. 2, p. 12, 2009.
[10] B. Xie, R. Passonneau, L. Wu, and G. G. Creamer, "Semantic frames to predict stock price movement," in *Proc. 51st Annu. Meeting Assoc. Comput. linguistics*, 2013, pp. 873–883.
[11] Y. Peng and H. Jiang, "Leverage financial news to predict stock price movements using word embeddings and deep neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 374–379. [Online]. Available: https://www.aclweb.org/anthology/N16-1041/
[12] Q. Li, J. Wang, F. Wang, P. Li, L. Liu, and Y. Chen, "The role of social sentiment in stock markets: A view from joint effects of multiple information sources," *Multimedia Tools Appl.*, vol. 76, no. 10, pp. 12315–12345, May 2017.
[13] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Using structured events to predict stock price movement: An empirical investigation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1415–1425.
[14] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland, "TextRunner: Open information extraction on the Web," in *Proc. 20th Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Demonstrations (NAACL)*, 2007, pp. 25–26.
[15] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 1535–1545.
[16] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI)*, Buenos Aires, Argentina, Jul. 2015, pp. 2327–2333. [Online]. Available: http://ijcai.org/Abstract/15/329
[17] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Knowledge-driven event embedding for stock prediction," in *Proc. 26th Int. Conf. Comput. Linguistics, Conf., Tech. Papers (COLING)*, N. Calzolari, Y. Matsumoto, and R. Prasad, Eds. Osaka, Japan: ACL, Dec. 2016, pp. 2133–2142. [Online]. Available: https://www.aclweb.org/anthology/C16-1201/
[18] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 1970–1979.
[19] D. Shah, H. Isah, and F. Zulkernine, "Predicting the effects of news sentiments on the stock market," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 4705–4708.
[20] H. Wu, W. Zhang, W. Shen, and J. Wang, "Hybrid deep sequential modeling for social text-driven stock prediction," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1627–1630.
[21] L. Yang, Z. Zhang, S. Xiong, L. Wei, J. Ng, L. Xu, and R. Dong, "Explainable text-driven neural network for stock prediction," in *Proc. 5th IEEE Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Nov. 2018, pp. 441–445, doi: 10.1109/CCIS.2018.8691233.
[22] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, 2011, pp. 44–51.
[23] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3856–3866.
[24] E. Xi, S. Bing, and Y. Jin, "Capsule network performance on complex data," 2017, *arXiv:1712.03480*. [Online]. Available: http://arxiv.org/abs/1712.03480
[25] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.
[26] S. Zhang, W. Zhao, X. Wu, and Q. Zhou, "Fast dynamic routing based on weighted kernel density estimation," 2018, *arXiv:1805.10807*. [Online]. Available: http://arxiv.org/abs/1805.10807
[27] M. Yang, W. Zhao, J. Ye, Z. Lei, Z. Zhao, and S. Zhang, "Investigating capsule networks with dynamic routing for text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3110–3119. [Online]. Available: https://www.aclweb.org/anthology/D18-1350/
[28] F. Feng, H. Chen, X. He, J. Ding, M. Sun, and T.-S. Chua, "Enhancing stock movement prediction with adversarial training," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, S. Kraus, Ed., Macao, China, Aug. 2019, pp. 5843–5849, doi: 10.24963/ijcai.2019/810.
[29] K. Cho, B. van Merriënboer, C. C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734. [Online]. Available: https://www.aclweb.org/anthology/D14-1179/
[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Minneapolis, MN, USA, vol. 1, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423
[31] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *Proc. 11th ACM Int. Conf. Web Search Data Mining - WSDM*, Y. Chang, C. Zhai, Y. Liu, and Y. Maarek, Eds. Marina Del Rey, CA, USA: ACM, Feb. 2018, pp. 261–269, doi: 10.1145/3159652.3159690.
[32] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, and H. Chen, "Knowledge-driven stock trend prediction and explanation via temporal convolutional network," in *Proc. Companion Proc. World Wide Web Conf.*, S. Amer-Yahia, M. Mahdian, A. Goel, G. Houben, K. Lerman, J. J. McAuley, R. Baeza-Yates, and L. Zia, Eds. San Francisco, CA, USA: ACM, May 2019, pp. 678–685, doi: 10.1145/3308560.3317701.
[33] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI)*, Q. Yang and M. J. Wooldridge, Eds. Buenos Aires, Argentina: AAAI Press, Jul. 2015, pp. 2327–2333. [Online]. Available: http://ijcai.org/Abstract/15/329
[34] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1480–1489.

JINTAO LIU is currently pursuing the degree with the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. His current research interests include natural language processing and stock market prediction.

HONGFEI LIN (Member, IEEE) received the B.Sc. degree from Northeast Normal University, in 1983, the M.Sc. degree from the Dalian University of Technology, in 1992, and the Ph.D. degree from Northeastern University, in 2000. He is currently a Professor with the School of Computer Science and Technology, Dalian University of Technology, where he is also the Director of the Information Retrieval Laboratory. He has published more than 100 research papers in various journals, conferences, and books. His research interests include information retrieval, text mining for biomedical literature, biomedical hypothesis generation, information extraction from huge biomedical resources, and learning-to-rank.

LIANG YANG received the B.S. degree in computer science and technology, in 2009, and the Ph.D. degree in computer application, in 2016. He is currently working as a Lecturer with the School of Computer Science and Technology, Dalian University of Technology. His research interests include sentiment analysis and opinion mining.



BO XU (Member, IEEE) received the B.Sc. and Ph.D. degrees from the Dalian University of Technology, China, in 2011 and 2018, respectively. He is currently a Postdoctoral Research Associate with the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology. His current research interests include information retrieval and learning-to-rank.

DONGZHEN WEN is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. His current research interests include information retrieval, software repository mining, and meta-learning.