TKSP: Long-term Stance Prediction for Social Media Users by Fusing Time Series Features and Event Dynamic Evolution Knowledge

Zijian Zhou, Shuoyu Hu, Kai Yang, and Haizhou Wang(✉)

School of Cyber Science and Engineering, Sichuan University, Chengdu, China
{2020141530063, 2020141530110, 2020141530041}@stu.scu.edu.cn
whzh.nc@scu.edu.cn

Abstract. The rise of social media has led to an increasing number of public discussion on hot topics. People always like to express their stances by posting tweets on social media. Mining stance information contained in text is significantly important for researchers to conduct analysis of public opinion, which makes the issue of stance detection as one of the hot research problems. In recent years, user-level stance detection has become a hot research topic in the field of stance detection, while fewer researches have been conducted on predicting users' stances. The current studies are mainly about short-term prediction of users' stances. **To the best of our knowledge, this paper is the first research work to conduct long-term stance prediction for social media users.** Specifically, this paper first builds a large-scale Chinese dataset for long-term prediction of social media users' stances. Subsequently, we propose a method of mining event dynamic evolution knowledge to improve the prediction performance. Finally, we construct a model fusing **T**ime series features and event dynamic evolution **K**nowledge for long-term **S**tance **P**rediction (TKSP). The experimental results demonstrate that our model outperforms other state-of-the-art models on the problem of long-term stance prediction for social media users.

Keywords: Stance detection · Social media · Time series features · Knowledge fusion · Deep learning

1 Introduction

In recent years, with the development of the Internet, social media has become a significant part of people's daily lives. More and more users are expressing their opinions on social media platforms such as Twitter and Weibo. The massive amount of text in social media contain users' stances for various social events. Mining stance information contained in text is significant important for researchers to conduct analysis of public opinion. In such a case, stance detection has become one of the hot research issues in the field of natural language processing. Stance detection is the task of automatically determining whether the text's author is in favor, against, or neutral towards a statement or targeted event, person, organization, government policy, movement, etc. [4].

However, the stances of users are not constant. They may change over time and as social events continuously evolve, which makes long-term stance prediction for users difficult. Conducting long-term stance prediction for users is beneficial for companies or governments to make informed decisions, which may bring immeasurable rewards. In such a case, it is necessary to conduct long-term stance prediction for social media users.

The traditional stance detection problem tends to detect the stance contained text, which has achieved excellent results with various machine learning and deep learning methods [5,7,8,10,19,24]. More and more researchers are now focusing on the user-level stance detection problem [3,6,17].

Stance prediction is a new direction of stance detection research, which aims to predict the future stance of a user on a certain topic. Current user stance prediction studies are mainly about short-term stance prediction for users [12, 13, 15]. They model the user posting behavior to predict the stance attitude contained in one or several future tweets, which are called short-term stance prediction. But there is a lack of work in long-term stance prediction for users.

Compared to traditional stance prediction, the research challenges of long-term stance prediction of social media users are as follows: (1) There is no publicly available dataset for long-term stance prediction for social media users. (2) Existing studies on stance prediction do not consider the influence on users brought by the evolution of events. (3) The existing stance prediction models oriented to users are mainly for short-term predictions, which are not suitable for the problem of long-term stance prediction. Therefore, a model for long-term stance prediction is needed.

In order to address the above challenges, this paper builds and publishes a benchmark dataset, proposes a model fusing time series features and event dynamic evolution knowledge for long-term stance prediction (TKSP). Specifically, we first collect and build a large-scale dataset from Sina Weibo, named SP-Dataset. After that, we propose a method of mining event dynamic evolution knowledge to improve the prediction performance of users' stances. Finally, this paper constructs a model for long-term stance prediction of social media users by fusing time series features and event dynamic evolution knowledge. The main contributions of this paper are summarized as follows:

- We construct the first large-scale Chinese dataset oriented to the problem for long-term stance prediction of social media users.
- We propose an innovative method of mining event dynamic evolution knowledge to improve the prediction performance of users' stances.
- To the best of our knowledge, we are the first to propose a model TKSP oriented to the problem of long-term stance prediction for social media users.

2 Related Work

The current work on stance detection is mainly divided into text-level stance detection and user-level stance detection. The stance prediction problem is completely a new research direction in the field of stance detection.

2.1 Text-level stance detection

So far, researches in the field of stance detection are mainly about text-level stance detection. Machine learning and deep learning methods have been widely used in the research field. Majumder et al. [5] used a memory neural network to model inter-aspect dependencies and the network showed effectiveness in classifying multifaceted sentences. Hardalov et al. [8] proposed a cross-linguistic stance detection method based on sentiment pre-training. Xu et al. [9] used two feature selection strategies (top k-based selection and leave-out k-based selection) to generate the optimal feature set.

At present, more mature results have been achieved in the text-level stance detection research.

2.2 User-level stance detection

User-level stance detection research has evolved rapidly in recent years. Darwish et al. [3] proposed an efficient unsupervised framework for detecting Twitter users' stances on controversial topics. Samih et al. [6] improved user-level stance detection by representing tweets using contextual embeddings that capture the potential meaning of words in context. Williams et al. [17] proposed a highly efficient Twitter Stance Propagation Algorithm (TSPA) for detecting user-level stance on Twitter that leverages the social networks of Twitter users.

Currently, user-level stance detection research is continuing to make progress.

2.3 Stance prediction

Stance prediction is a new direction for stance detection research. Chen et al. [11] use temporally ordered tweets for stance prediction by introducing an attention layer to weight the importance of a user's previous tweets, current tweets, and neighboring tweets. They also used an LSTM layer to capture historical influences from past eras. Fang et al. [13] proposed a multi-task learning model that exploits a large amount of textual information in existing datasets to improve stance prediction. Zhou et al. [15] used recurrent neural networks to model each user's posting behavior on Twitter and used an attention mechanism to merge neighbors' topic-related contexts into attention signals for user-level stance prediction.

Currently, stance prediction works are focused on short-term prediction of users' stances (e.g., predicting the stance of the user's next tweet or next few tweets), but such prediction results do not reflect users' stance attitudes in a certain time period in the future. A small number of existing studies consider the influence of users' historical tweets on their future stances, but these works do not use the time-series features contained in historical tweets to achieve long-term prediction of users' stances. There are also no studies that consider introducing event dynamic evolution knowledge into stance prediction work.

3 Methodology

In this section, we will introduce our method of long-term stance prediction for social media users that fuses time series features and event dynamic evolution knowledge.

3.1 Dataset construction

Currently, Sina Weibo[1] is one of the largest Chinese social platforms in the world, and hundreds of millions of users post their tweets on Sina Weibo every day. In our study, we choose Sina Weibo to collect historical tweets from different users to construct our dataset. We chose three events "Russia's special military operation against Ukraine", "Chatbot ChatGPT" and "The movie *Full River Red*" as the objects of data collection. In the specific labeling process of the dataset, we labeled the tweets with a supportive attitude as "1", the tweets with a neutral attitude as "0", and the tweets with an opposing attitude as "-1". After rigorous manual annotation, our dataset contains a total of 949 users and 58,310 tweets. We name the dataset as SP-Dataset. The dataset is now publicly available[2] and its specific information is shown in TABLE 1.

Table 1: Dataset statistics

Topics	**Number of users**	**Number of tweets**	**Period**	**Examples**
俄罗斯对乌克兰的特别军事行动(Russia's special military operation against Ukraine)	425	44,419	2022.3.1-2023.3.31	战争残酷无情，昔日兄弟还是早日坐下和谈吧。(The war is merciless. Let the old brothers sit down and talk soon.) label=0 见到入侵的俄军就打，绝不要手下留情！(If you see invading Russian troops, fight them. Show no mercy!) label=-1
聊天机器人模型ChatGPT(Chatbot ChatGPT)	241	5,845	2022.12.1-2023.4.7	个性化定制版ChatGPT？这个厉害了。(Personalized customized ChatGPT? This is amazing.) label=1 ChatGPT能让百度少一点竞价广告。(ChatGPT can allow Baidu to offer fewer ads through bidding.) label=0
电影《满江红》(The movie *Full River Red*)	283	8,046	2022.5.17-2023.4.7	满江红真的好厉害呀！！！单日4.6亿。(The movie *Full River Red* is really good! It made 460 million yuan in a single day.) label=1 欢喜传媒上市公司公告，《满江红》盈利12亿。(Huanxi media company announced that *Full River Red* made a profit of 1.2 billion yuan.) label=0

[1] https://weibo.com/

[2] https://github.com/yiyepianzhounc/TKSP

3.2 Prediction Model

As shown in Fig. 1, we propose a model for stance prediction of social media users that incorporates time series features and event dynamic evolution knowledge. This model contains five parts, which are named pre-processing module, stance detection module, calculation of stance scores, the mining of event dynamic evolution knowledge and prediction module. The details of each part are as follows.

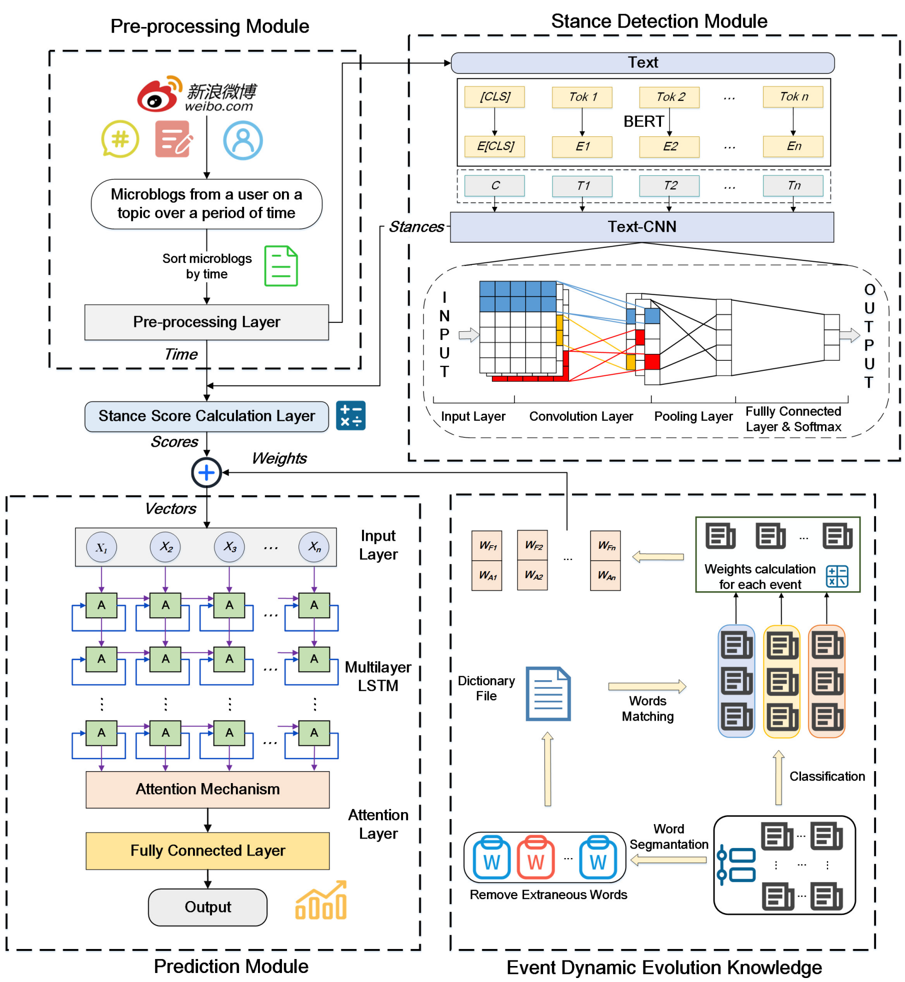


Fig. 1: TKSP model structure

Pre-processing module In the pre-processing module, we pre-process the user's historical tweets. First, we collect a user's historical tweets from Sina

Weibo. Then these tweets will be arranged in chronological order and fed into the pre-processing layer. In the pre-processing layer, the text content is processed through text normalization, text cleaning, word segmentation, word cleaning, and word normalization. The pre-processed tweets will be used as input to the stance detection module.

Stance detection module We first input the pre-processed tweets from a user into the pre-trained Chinese BERT model [1] to get their vector representations. The output vectors go through convolutional neural networks for further feature extraction. As for the CNN module, we choose the Text-CNN model [20], which can help BERT with classification tasks of specific topics. By connecting the CNN structures, the BERT model is fed into the CNN model as an embedding layer for further learning. Based on the original semantic information combined with the text information of the topic, CNN structure is used for convolutional pooling to extract the semantic features of the target topic. And then the classifier is trained and classified according to these features to finally achieve the classification of stances under the target topic.

The loss function for the training process uses the cross-entropy. The parameter optimizer used to train the model uses the Adam optimizer [21]. The loss function is defined as follows:

$$Loss = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_j^i \log p_j^i \tag{1}$$

where y_j^i indicates the true label of the ith sample, C is the number of categories of stance labels and N is the total number of samples. The model is trained iteratively using this loss function until convergence.

Calculation of stance scores The stance of a user in a certain time window cannot be simply characterized as "favor", "neutral", or "against", we propose a stance-scoring mechanism based on the number of tweets and stance of each tweet. For a given user, we will get the stance of each historical tweet according to the stance detection module. And these tweets are arranged in chronological order and divided according to the specific time window (In all the experiments, the size of the time windows is forty-eight hours). The procedure for stance score calculation of a given user is elaborated in Algorithm 1.

After the above process, we can get the user's stance score for each time window. If the score is higher than 0 and the higher the score is, the more supportive to the topic the user is during this time window. If the score is less than 0 and the lower the score is, the more opposed to the topic the user is during this time window. If the score is equal to 0, the user is neutral during this time window. The stance-scoring mechanism can help us reasonably evaluate the degree of users' inclinations, which will be beneficial to our prediction work based on time series.

Algorithm 1 Stance scores calculation

Input :
N_f, The number of tweets with a favorable stance in the each time window
N_n, The number of tweets with a neutral stance in the each time window
N_o, The number of tweets with an opposing stance in the each time window
f_s, Square root of arithmetic calculation function f_t, Inverse tangent function
Output :
S_s, Stance scores of the user

```
// n is the number of time windows
for i in n do
    A_i = π/2 − f_t(n_i/(f_i − o_i)), f_i ∈ N_f, n_i ∈ N_n, o_i ∈ N_o; //Calculate intermediate variables
end for
for i in n do
    P_i = f_s(f_i + n_i + o_i), f_i ∈ N_f, n_i ∈ N_n, o_i ∈ N_o; //Calculate intermediate variables
end for
for i in n do
    s_i = P_i × A_i //Calculate stance score in each time window
end for
return S_s = {s_1; s_2; ...; s_n}
```

The mining of event dynamic evolution knowledge In this part, we consider the influence on users brought by the dynamic evolution of topic events. We collect breaking news that occurred during the dynamic evolution of topic events from news websites[3]. We generate two weights, W_F and W_A for each news. W_F indicates the weight of the news that makes the user's stance favorable. W_A indicates the weight of the news that makes the user's stance unfavorable. Specifically, for a given topic, we first perform word segmentation on the news we collected. After that, we remove the words that are not related to the change of stance. Then we assign three levels of scores (1, 2, 3) to the remaining words. If a word has a higher score, it will exert a greater impact on users. Then we form a dictionary with these words and their corresponding scores. Finally, we manually divide all the news into three categories: news that is favorable to supporters, news that is favorable to opponents, news that is neutral. The procedure for weights calculation of each news is elaborated in Algorithm 2.

After the above process, we get values of W_F and W_A for each news, while each news has its time node of occurrence. We generate three-dimensional vectors by fusing the user's stance score in each time window and weights for the news at the corresponding time node. The three-dimensional vector corresponding to the nth time window is $\boldsymbol{X_n} = (S_n, W_F(n), W_A(n))$. S_n is the stance score in the nth time window; $W_F(n)$ and $W_A(n)$ denote the value of W_F and W_A for news

[3] https://www.sohu.com/a/645326473_121425542, https://zhuanlan.zhihu.com/p/615712600, https://www.sohu.com/a/635606929_120546417

that occurred in the nth time window. If none of the collected news occurred during a time window, we set both the value of the W_F and W_A to 0.

Algorithm 2 Weight calculation

Input :
n_s, News that is favorable to supporters
n_o, News that is favorable to opponents
n_n, News that is neutral
s_d, The score for each word in the dictionary
f_m, Word match function
f_c, Total score calculation function
Output :
W_n, Weights for each news;
// *news* is the news for which we calculate the weights
if $news \in n_n$ **then**
  $W_F = 0$
  $W_A = 0$
end if
if $news \in n_s$ **then**
  $W_F = f_c(f_m(news), s_d)$ //Calculate the value of W_F
  $W_A = 0$
end if
if $news \in n_o$ **then**
  $W_F = 0$
  $W_A = f_c(f_m(news), s_d)$ //Calculate the value of W_A
end if
return $W_n = (W_F, W_A)$

Prediction module In this part, we input three-dimensional vectors that we get from previous modules and time information into the prediction module. The prediction module first consists of a multilayer LSTM. The data are processed in the LSTM [14] structural unit in the following order: Forget gate f_t discard unwanted information:

$$f_t = \sigma(W_f \times [h_{t-1}, X_t] + b_f) \tag{2}$$

Input Gate i_t determines the data that needs to be updated:

$$i_t = \sigma(W_i \times [h_{t-1}, X_t] + b_i) \tag{3}$$

Outputs the updated data through the output gate O_t:

$$O_t = tanh(W_O \times [h_{t-1}, X_t] + b_O) \tag{4}$$

Update cell status C_t :

$$C_t = C_{t-1} \odot f_t + i_t \odot O_t \tag{5}$$

Determine the value of the output at the current moment:

$$h_t = \sigma(W_O \times [h_{t-1}, X_t] + b_O) \odot \tanh(C_t) \tag{6}$$

where X_t is the input at moment t, h_{t-1} and h_t are the outputs at moments $t-1$ and t, C_{t-1} and C_t are the cell states at moments $t-1$ and t. The W_f, W_i, W_O, b_f, b_i and b_O are the weights. The σ denotes the activation function sigmoid.

A multilayer LSTM structure is used in our model, which consists of multiple sets of LSTM cells stacked together. The cell state C_t and the hidden state h_t of each LSTM layer are used as input for the next LSTM layer. The use of multi-layer LSTM structure helps to improve the prediction performance.

The output of the last LSTM hidden layer is fed to the attention layer for further processing. The attention mechanism can improve the effect of important time steps in the LSTM, thus further reducing prediction errors of the model. The output vector of the LSTM hidden layer is used as the input of the attention layer, which is trained by a fully connected layer. And then the output of the full connection layer is normalized using the softmax function to obtain the assigned weight of each hidden layer vector. The weight size indicates the importance of the hidden state at each time step for the prediction result. The weight training process is as follows:

$$S_i = \ tanh(Wh_i + b_i) \tag{7}$$

$$\alpha_i = \ softmax\,(S_i) \tag{8}$$

$$C_i = \sum_{i=0}^{k} \alpha_i h_i \tag{9}$$

where h_i is the output of the last LSTM hidden layer; S_i is the score of the output of each hidden layer; α_i is the weighting factor; C_i is the result after weighted summation, and $softmax$ is the activation function.

The final output specifies a prediction time step of O_t. So the final part outputs O_t steps of the prediction results.

4 Experiments

In this section, we evaluate the performance of our proposed model. First, we describe the experimental setup in our work. Then, we illustrate the superiority of our model compared to the baseline models. After that, the effects of different modules in the model are examined by the ablation experiment. Finally, we tested the performance of our model at different proportions of training data through early detection experiment.

4.1 Experimental Settings

The dataset used in our experiments is SP-Dataset that we collected and constructed from Weibo platform. In our work, all experiments were undertook on a workstation equipped with Intel Xeon Platinum 8255C CPU and NVIDIA Tesla T4 with 32GB of memory.

In following experiments, we divide our tweet data into training, validation and test sets in the ratio of 6: 2: 2. These data are used for training and testing in the stance detection module. In the prediction module, for each user, we use the first 80% of the user's data for training and the last 20% for testing.

4.2 Baseline Model Comparison Experiment

To demonstrate the effectiveness of our TKSP model, we tested our model and seven other baselines on the constructed SP-Dataset. Our study is divided into two parts: stance detection and stance prediction, so we select seven popular model combinations. In the specific experiment, we randomly select 30 users from the total of 949 users to conduct the experiment. The evaluation metrics of the experiment are MAE and RMSE of original data and MAE and RMSE of the first-order difference of original data.

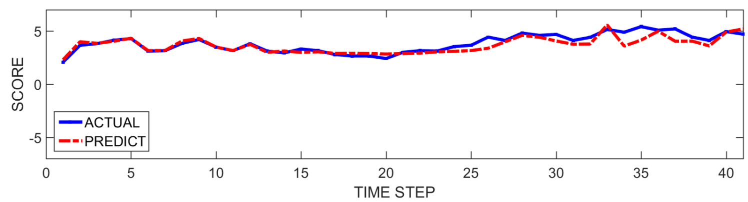
We present the average, maximum and minimum values of the experimental results of the 30 users. The experimental results are shown in TABLE 2. The results show that our model outperforms these advanced models in terms of MAE and RMSE of original data and MAE and RMSE of the first-order difference of original data, which indicates that the prediction performance of our model is better. In the test of our TKSP model, we present the prediction results for six randomly selected users (we call them User 1, User 2, User 3, User 4, User 5 and User 6) as a display, which are shown in Fig. 2. From the prediction results, we can find that User 5 has significantly better prediction results than other users. This is because the stance scores of User 5 change much less than that of other users. The stance scores of User 5 tend to be stable within a smaller interval. Therefore, User 5 can achieve better prediction results.

Table 2: Results of the baseline model comparison experiment

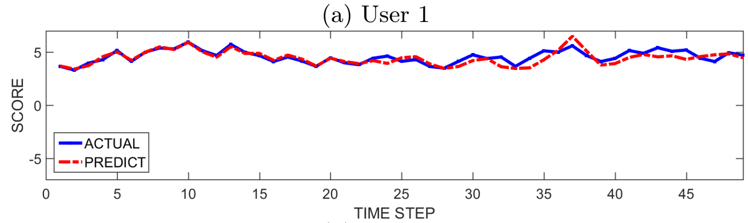
Metrics \ **Models**		ARIMA [16]	RF [25]	ANN [23]	CNN [26]	GRU [16]	BiLSTM [18]	LSTM [18]	**TTSF (ours)**
MAE	AVG	0.4737	0.4115	0.4562	0.4500	0.4484	0.4484	<u>0.3838</u>	**0.3432**
	MAX	0.8171	0.6442	0.7903	0.7935	0.7944	0.7912	<u>0.5658</u>	**0.5141**
	MIN	<u>0.1420</u>	0.1452	0.1483	0.1464	0.1455	0.1458	0.1468	**0.1380**
RMSE	AVG	0.5648	0.4791	0.5382	0.5038	0.5009	0.4998	<u>0.4546</u>	**0.4230**
	MAX	0.9081	0.7235	0.9665	0.8829	0.9044	0.8801	<u>0.6230</u>	**0.5537**
	MIN	0.1873	0.1897	0.1865	0.1862	<u>0.1852</u>	0.1861	0.1864	**0.1784**
MAE of FOD	AVG	0.4559	0.4639	0.4044	0.3931	0.3952	<u>0.3930</u>	0.4032	**0.3819**
	MAX	0.5677	0.5388	0.5337	0.5016	0.5028	0.5024	<u>0.5011</u>	**0.4832**
	MIN	0.2670	0.2838	0.2544	0.2556	<u>0.2541</u>	0.2550	0.2556	**0.2501**
RMSE of FOD	AVG	0.4732	0.5490	0.4696	<u>0.4458</u>	0.4478	0.4570	0.4497	**0.4371**
	MAX	<u>0.5574</u>	0.6489	0.6079	0.5628	0.5636	0.5625	0.5622	**0.5460**
	MIN	0.2852	0.2873	<u>0.2839</u>	0.2856	0.2859	0.2867	0.2854	**0.2793**

[1] Due to the limitation of the table size, we use "MAE of FOD" and "RMSE of FOD" instead of the MAE and RMSE of the first-order difference of the original data in the table.

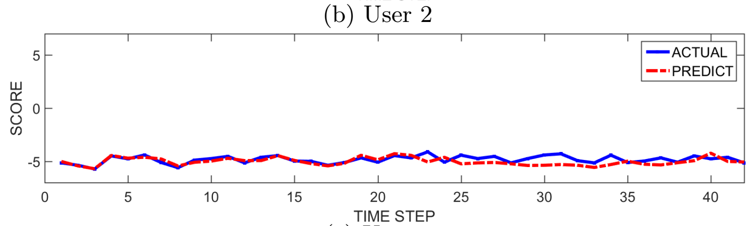
[2] The baseline models we use in the table are combinations with the BERT model. Due to space limitations, we do not show theirs full names.



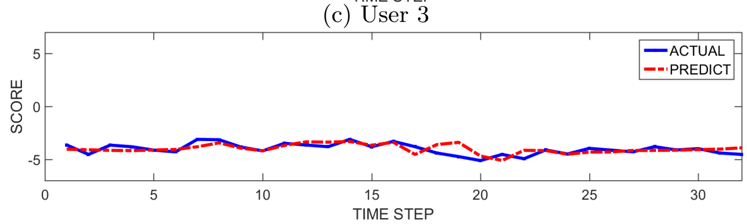
(a) User 1



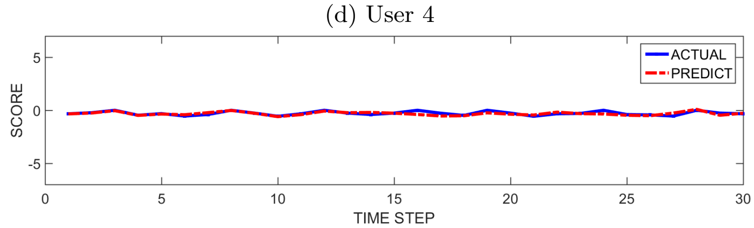
(b) User 2



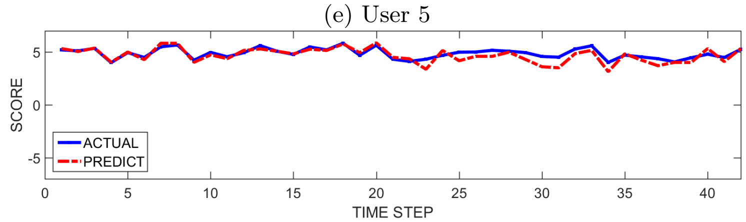
(c) User 3



(d) User 4



(e) User 5



(f) User 6

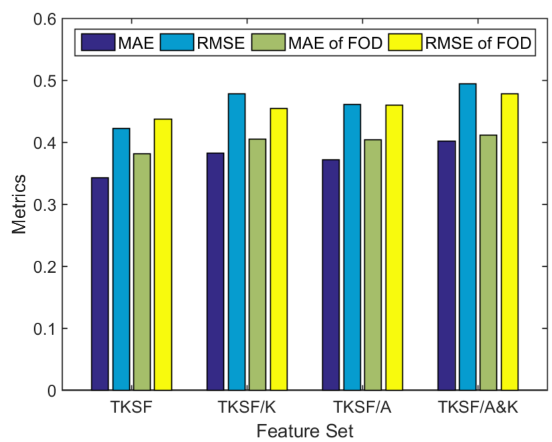
Fig. 2: Prediction results for six randomly selected users

4.3 Ablation Experiment

Our proposed TKSP model combines text features, event dynamic evolution knowledge, and attention mechanism. We examine the contribution of each block to the model through this experiment. Specifically, the use of ablation features is shown in TABLE 3, where **K** denotes the event dynamic evolution knowledge, **A** denotes the attention mechanism and "/" means "without". We use the same metrics in the baseline experiment as the evaluation metrics of the experiments and use the average of the experimental results of 30 users that are randomly selected from 949 users as the demonstration. The experimental results are shown in Fig. 3. The results show that each block plays an important role in our model, and the ablation of any block weakens the prediction effect of the model. Evidently, our proposed blocks can effectively help the model predict stance scores of users.

Table 3: Settings of ablation experiment

Module set	**Categories of modules included**
TKSP	Text features, Event dynamic evolution knowledge, Attention mechanism
TKSP/**K**	Text features, Attention mechanism
TKSP/**A**	Text features, Event dynamic evolution knowledge
TKSP/**K&A**	Text features

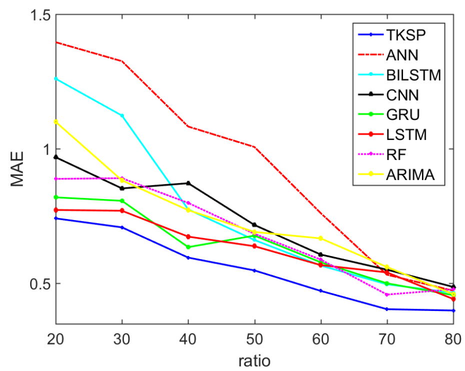


* Due to the limitation of the image size, we use "MAE of FOD" and "RMSE of FOD" instead of the MAE and RMSE of the first-order difference of the original data in the figure.

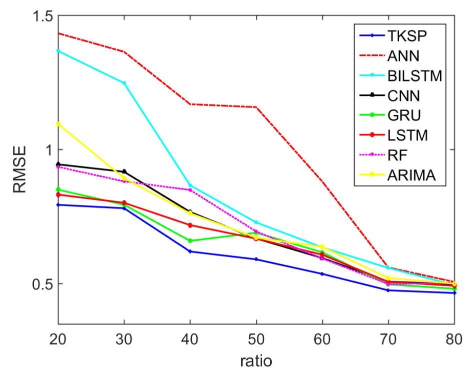
Fig. 3: Results of ablation experiment

4.4 Early Detection Experiment

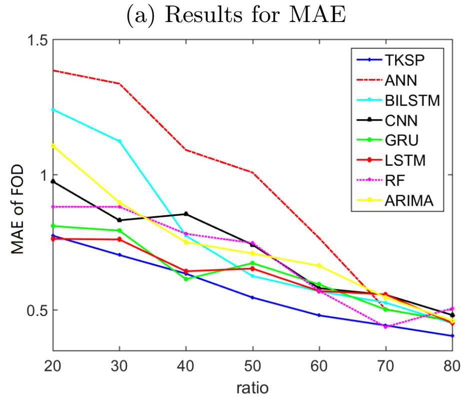
In the above experiments, we use 80% of the user data to train and predict the other 20% of the user data. In the early detection experiment, we reduce the proportion of our training data to varying degrees and compare the prediction performance of our model with other baseline models. We use the same metrics in the baseline experiment as the evaluation metrics of the experiments. We use the average of the experiment results of 30 users that are randomly selected from 949 users as the demonstration. The experimental results are shown in Fig. 4. Experimental results show that our model outperforms other baseline models in the vast majority of cases. In addition to this, the overall decrease in the prediction performance of our model is not as significant as other models.



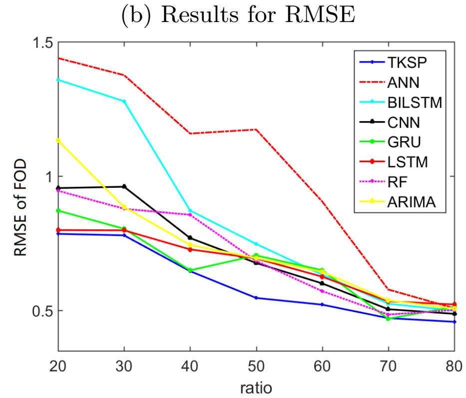
(a) Results for MAE



(b) Results for RMSE



(c) Results for MAE of FOD



(d) Results for RMSE of FOD

* Due to the limitation of the image size, we use "MAE of FOD" and "RMSE of FOD" instead of the MAE and RMSE of the first-order difference of the original data in the figure.

Fig. 4: Results of early detection experiment

5 Conclusion

In this paper, we construct the first SP-Dataset for the long-term stance prediction problem for social media users. In the problem of long-term stance prediction for social media users, we utilize historical tweets of users, time series features and event dynamic evolution knowledge to construct a new model TKSP. Experimental results on our SP-Dataset show that our model can effectively achieve the long-term stance prediction for social media users and our method outperforms other state-of-art models. In the future, we will try to incorporate images and videos from social media users' tweets into our research to achieve a more comprehensive analysis of users' stances.

Acknowledgements. This work is supported by the Key Research and Development Program of Science and Technology Department of Sichuan Province under grant No. 2023YFG0145. In addition, this work is also partially supported by the National Key Research and Development Program of China under grant No. 2022YFC3303101, the Key Research and Development Program of Science and Technology Department of Sichuan Province (nos. 2020YFS0575, 2021YFG0159, 2021KJT0012-2021YFS0067), Sichuan University and Yibin Municipal People's Government University and City Strategic Cooperation Special Fund Project (No. 2020CDYB-29), and Science and Technology Plan Transfer Payment Project of Sichuan Province (No. 2021ZYSF007).

References

1. Cui, Y., Che, W., Liu, T.,et al.: Pre-training with Whole Word Masking for Chinese Bert. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 3504-3514 (2021)
2. ALDayel, A., Magdy, W.: Stance Detection on Social Media: State of the Art and Trends. Information Processing & Management, 58(4), 102597 (2021)
3. Darwish, K., Stefanov, P., Aupetit, M., et al.: Unsupervised User Stance Detection on Twitter. In: Proceedings of the 14th International AAAI Conference on Web and Social Media, pp. 141-152 (2020)
4. Mohammad, S.M., Sobhani, P., Kiritchenko, S.,: Stance and Sentiment in Tweets. ACM Transactions on Internet Technology, 17 (3), 26:1–26:23 (2017)
5. Majumder, N., Poria, S., Gelbukh, A., et al.: IARM: Inter-aspect Relation Modeling with Memory Networks in Aspect-based Sentiment Analysis. In: Proceedings of the 11th Conference on Empirical Methods in Natural Language Processing, pp. 3402-3411 (2018)
6. Samih, Y., Darwish, K.: A few Topical Tweets are Enough for Effective User Stance Detection. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 2637-2646 (2021)
7. Umer, M., Imtiaz, Z., Ullah, S.,et al.: Fake News Stance Detection Using Deep Learning Architecture (CNN-LSTM). IEEE Access, 8, 156695-156706 (2020)
8. Hardalov, M., Arora, A., Nakov, P., et al.: Few-shot Cross-lingual Stance Detection with Sentiment-based Pre-training. In: Proceedings of the 14th AAAI Conference on Artificial Intelligence, pp. 10729-10737 (2022)
9. Xu, J., Zheng, S., Shi, J., et al.: Ensemble of Feature Sets and Classification Methods for Stance Detection. In Proceedings of 5th CCF Conference on Natural Language Processing and Chinese Computing, pp 679-688 (2016)
10. Wang, Z., Huang, Y., He, B., et al.: TDDF: HFMD Outpatients Prediction Based on Time Series Decomposition and Heterogenous Data Fusion. In: Proceedings of the 15th Conference on Advanced Data Mining and Applications, pp. 658-667 (2019)

11. Chen, C., Wang, Z., Li, W.: Tracking Dynamics of Opinion Behaviors with a Content-based Sequential Opinion Influence Model. IEEE Transactions on Affective Computing, 11(4), 627-639 (2018)
12. Dong, R., Sun, Y., Wang, L., et al.: Weakly-guided User Stance Prediction via Joint Modeling of Content and Social Interaction. In: Proceedings of the 26th ACM on Conference on Information and Knowledge Management, pp. 1249-1258 (2017)
13. Fang, W., Nadeem, M., Mohtarami, M., et al.: Neural Multi-task Learning for Stance Prediction. In: Proceedings of the 2nd Workshop on Fact Extraction and Verification, pp. 13-19 (2019)
14. Hochreiter, S., Schmidhuber, J.: Long short-term Memory. Neural computation, 9(8), 1735-1780 (1997)
15. Zhu, L., He, Y., Zhou, D.: Neural Opinion Dynamics Model for the Prediction of User-level Stance Dynamics. Information Processing & Management, 57(2), 102031 (2020)
16. Yamak, P. T., Yujian, L., Gadosey, P. K.: A Comparison between Arima, Lstm, and Gru for Time Series Forecasting. In: Proceedings of the 2nd International Conference on Algorithms, Computing and Artificial Intelligence, pp. 49-55 (2019)
17. Williams, E. M., Carley, K. M.: TSPA: Efficient Target-Stance Detection on Twitter. In: Proceedings of the 13th IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 242-246 (2022)
18. Siami-Namini, S., Tavakoli, N., Namin, A. S.: The Performance of LSTM and BiLSTM in Forecasting Time Series. In: Proceedings of the 7th IEEE International Conference on Big Data, pp. 3285-3292 (2019)
19. Wei, P., Mao, W., Chen, G.: A Topic-aware Reinforced Model for Weakly Supervised Stance Detection. In: Proceedings of the 13th AAAI Conference on Artificial Intelligence, pp. 7249-7256 (2019)
20. Zhang, Y., Wallace, B.: A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. arXiv preprint arXiv:https://arxiv.org/abs/1510.03820 (2015)
21. Kingma, D. P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:https://arxiv.org/abs/1412.6980 (2012)
22. Zhou, H., Zhang, S., Peng, J., et al.: Informer: Beyond Efficient Transformer for Long Sequence Time-series Forecasting. In: Proceedings of the 15th AAAI Conference on Artificial Intelligence, pp. 11106-11115 (2021)
23. Aras, S., Kocakoç, İ. D.: A New Model Selection Strategy in Time Series Forecasting with Artificial Neural Networks: IHTS. Neurocomputing, 174, 974-987 (2016)
24. Kobbe, J., Hulpuș, I., Stuckenschmidt, H.: Unsupervised Stance Detection for Arguments from Consequences. In: Proceedings of the 11th Conference on Empirical Methods in Natural Language Processing, pp. 50-60 (2020)
25. Cutler, A., Cutler, D. R., Stevens, J. R.: Random Forests. Ensemble Machine Learning: Methods and Applications, 157-175 (2012)
26. Wang, K., Li, K., Zhou, L., et al.: Multiple Convolutional Neural Networks for Multivariate Time Series Prediction. Neurocomputing, 360, 107-119 (2019)
27. Cai, Y., Wang, H., Ye, H., et al.: Depression Detection on Online Social Network with Multivariate Time Series Feature of User Depressive Symptoms. Expert Systems with Applications, 119538 (2023)
28. Li, Y., Zhu, Z., Kong, D., Han, H., et al.: EA-LSTM: Evolutionary Attention-based LSTM for Time Series Prediction. Knowledge-Based Systems, 181, 104785 (2019)
29. Wang, W., Liu, W., Chen, H.: Information Granules-based BP neural Network for Long-term Prediction of Time Series. IEEE Transactions on Fuzzy Systems, 29(10), 2975-2987 (2020)