Social Media Stance Detection Based on Target-Comment Cross-Attention Mechanism

Sijie Cai[1,2], **Qi Chen**[1,2], **Zexin Lin**[1,2]
[1]Department of Artificial Intelligence, Xiamen University, Xiamen, China
[2]Class of School of Informatics in Deep Learning Course, Xiamen University
sijie@stu.xmu.edu.cn, chenqi@stu.xmu.edu.cn, zexinlin@stu.xmu.edu.cn
31520231154283[Sijie Cai], 31520231154284[Qi Chen], 31520231154300[Zexin Lin]

Abstract

With the development of science and technology, social media has gained unprecedented vitality, and the stance detection task generated on this basis also reflects high research value and application value. In the stance detection task, the existence of the target is crucial, and how to use the target to improve the accuracy of prediction has always been a concern for researchers. This research proposes a model based on BERT and attention mechanism, with fully frozen BERT as the encoder, trying to use the target to assign weights to each token of the comment to improve the accuracy of prediction, and after a series of experiments on the binary classification task and the three-category classification task, it is verified that the new model has a certain degree of performance improvement compared with the baseline models. In addition, this research expands the dataset of NLPCC2016 Weibo stance detection task for the Chinese lack of social media stance detection dataset, and conducts relevant experiments on this dataset, and the proposed model also reflects convergence on this dataset, indicating that this expanded dataset has certain use value.

Introduction

Stance detection is one of the text classification tasks, the main purpose is to train a model that can give the stance of the comment text to the target based on the given target text, which may be a product name, a policy summary, a brief description of an event, etc. Stance detection is usually modeled as a binary or three-category classification task, and the binary classification task is divided into two categories: supporting and opposing. Supporting indicates that comments reflect a supportive attitude towards the target, including directly expressing praise for the target, criticizing the target competitor, and so on. Opposing refers to comments that express opposition to the target, including direct criticism of the target, praise for the target's competitors, and so on. On this basis, the three-category classification task adds the neutral/irrelevant category, which mainly means that the comment does not show obvious emotional tendency towards the target, or has nothing to do with the target.(Sun 2022)

Stance detection has been widely used in many fields, including election analysis, product user research and policy opinion survey. By analyzing a large number of commentary texts, we can obtain the overall attitude of social groups towards targets, including products, events, policies, etc., as well as the specific attitude distribution, which can provide references for relevant individuals, enterprises, groups and even the government, thus assisting in thinking about the direction of adjustment in the next step. Therefore, stance detection has extremely high research and application value.(Li, Sun, and Li 2021)

In fact, stance detection has some correlation with sentiment analysis, which is also a text classification task. However, compared with sentiment analysis that simply analyzes the emotional polarity of a text, stance detection has the target text, and it is the existence of target text that makes stance detection more complicated. Specifically, there will be situations where the emotional polarity of the text itself is positive, but to the target is negative. The typical example of this situation is in the election, for example, if the target is Trump, the commentary text is accusing Biden, then obviously the polarity of the text itself is negative, but the supportive attitude towards the target is indeed, which is to express support for the target by criticizing the target's competing products. At the same time, it is also obvious that if the target is replaced by Biden, the stance of this comment is opposed.(Kawintiranon and Singh 2021)

The main purpose of our study is to explore a suitable method of utilizing targets in stance detection tasks, especially in social media stance detection tasks, aiming to improve the accuracy of classification through the combination of targets and comments. Our study has made the following contributions: Firstly, we proposed a stance detection model based on BERT and attention mechanism which we called it TCCM (Target-Comment Cross-Attention Model) because of using a layer that made the target and the comment to do cross attention. Secondly, we also expands a mainstream Chinese stance detection data set, NLPCC 2016 Weibo stance detection task data set, which makes it more useful for stance detection tasks.

Related Work

The methods of stance detection mainly include machine learning based on feature engineering, deep learning based on neural network and transfer learning.(Li and Yang 2022)

Machine learning methods based on feature engineering mainly use traditional text feature extraction methods and machine learning algorithms for classification. In 2016, Can Liu, Wen Li et al used the bag of words model and unigram to extract features from the original tweets, then used SVM, random forest and gradient enhanced decision tree classifiers for classification, and finally used voting strategies to integrate the results of the classifier.(Liu et al. 2016) In 2017, Yujie Dian, Qin Jin et al. used the method of multi-feature fusion to represent five textual features, including bag of words features, thesaurus based bag of words features, topic subject words, features of stance label co-occurrence relationship, and word vectors extracted by woed2vec. SVM, decision tree and random forest classifiers were used to conduct research on Chinese Weibo stance detection data sets.(Dian, Jin, and Wu 2017) In 2018, Haiyang Zheng, JunBo Gao et al. used the Text Rank algorithm to construct the target word set, then used the continuous bag of Words model (CBOW) for feature extraction, and finally used the SVM classifier for classification, and achieved the first place in the NLPCC2016 Weibo stance detection task.(Zheng, Gao, and Qiu 2018)

Deep learning methods based on neural network is also widely used in stance detection. In 2016, Wan Wei, Xiao Zhang et al. carried out stance detection on the SemEval2016 dataset. This method firstly represented the text sequence as a word vector matrix, and then extracted features from it using convolutional neural networks. Then multiple convolution kernels of different sizes were applied to the input matrix respectively to extract features of different levels. Finally, the features of each layer are spliced together as the final feature representation, and the fully connected layer is used for classification.(Wei et al. 2016) Also in 2016, Isabelle Augenstein and Tim Rocktaschel et al. proposed a method using BiLSTM (Bidirectional Long short-term memory network), taking the feature encoding of the target obtained through LSTM (long short-term memory network) as the initial state input of the comment on the bidirectional long short-term memory network to generate the comment feature code containing target information.(Augenstein et al. 2016) In 2018, Mitra Mohtarami and Ramy Baly used a model called End-to-End Memory Networks (EMN) for stance detection, which uses two LSTM layers to model the semantic information of input text. The first LSTM layer encodes the input text into a fixed-length vector and feeds it into the Memory Network, which uses attention mechanisms to select which information should be stored in the memory bank, and the second LSTM layer, based on the input text and the information in the memory bank, Generate a vector representation to feed into the classifier to predict the stance of the input text.(Mohtarami et al. 2018) In 2019, Anjun Wang and Kaikai Huang et al. proposed using BERT to obtain the sentence vectors of the target and comment, using Condition to calculate the relationship matrix between the two, then using CNN for feature extraction and maximum pooling, and finally using softmax for classification, which achieved a good result in the 2016 NLPCC Stance Detection task.(Wang, Huang, and Lu 2019) In 2023, Ke Chen, Haoxuan Zhou and Guoquan Wang proposed a model that uses LSTM and CNN to extract features in parallel, so as to obtain the global feature (LSTM) and local feature (CNN), and then the two are combined and sent to softmax for prediction.(Chen, Zhou, and Wang 2023)

Transfer learning is also used in stance detection. In their 2016 study, Guido Zarrella and Amy Marsh trained a two-layer feature extraction model using a large number of unlabeled tweets with weak supervision, and then fine-tuning a classifier on five topics, training and cross-validation on five topics.(Zarrella and Marsh 2016) In 2020, Emily Allaway and Kathleen McKeown constructed the target and comment into sentence pairs, and the feature extraction of a sample is completed by using the ability of BERT to connect the next sentence, and then fine-tuning the pre-trained BERT for prediction.(Allaway and McKeown 2020)

In the stance detection task, in the mainstream dataset Semaval2016 and WWT, the TPDG proposed by Bin Liang and Yonghao Fu et al. in 2021 has reached SOTA, which is realized by calculating the pragmatic weights within and between targets, constructing the target pragmatic dependency graph, and combining BiLSTM and CNN.(Liang et al. 2021)In 2023, Tianyu Wang and Jiawei Yuan et al. proposed a method of introducing external knowledge, injecting the external knowledge obtained from the Internet into the comment text to be predicted, so as to carry out multi-type knowledge enhancement, and then send it into the convolutional layer for prediction.This method achieves SOTA on the mainstream Chinese dataset NLPCC2016.(Wang et al. 2023)Of course, with the introduction of ChatGPT, ChatGPT achieved the same results as SOTA on several stance detection datasets by giving it task hints.(MG 2023)

Method

Definition

For a stance test sample, we use a triplet (t, c, s) to represent , where t is the target text, c is the comment text, and s is the numerical value representing the stance. For binary classification, the value for opposing is 0, and the value for supporting is 1. For three categories, the value for opposing is 0, the value for neutral is 1, and the value for supporting is 2. The number of classes that need to be divided for the classification task of modeling is N. n is the actual number of characters in the comment.

Structure of TCCM

The model is based on BERT and attention mechanism, and its structure is shown in Figure 1. It mainly consists of the following parts: (1) Encoder; (2) Target-Comment Cross-attention Layer; (3) Comment Self-attention Layer; (4) Revision Layer; (5) Average Pooling Layer; (6) Output Layer. The structure of attention mechanism includes Target-Comment Cross-attention Layer, Comment Self-attention Layer and Revision layer, which is inspired by the block structure of the coding part of Transformers.

Encoder

BERT uses a self-attention mechanism to extract text features, so the encoding of each word will include attention to each word in the same sentence, overcoming the shortcoming that each word in the RNN network structure can

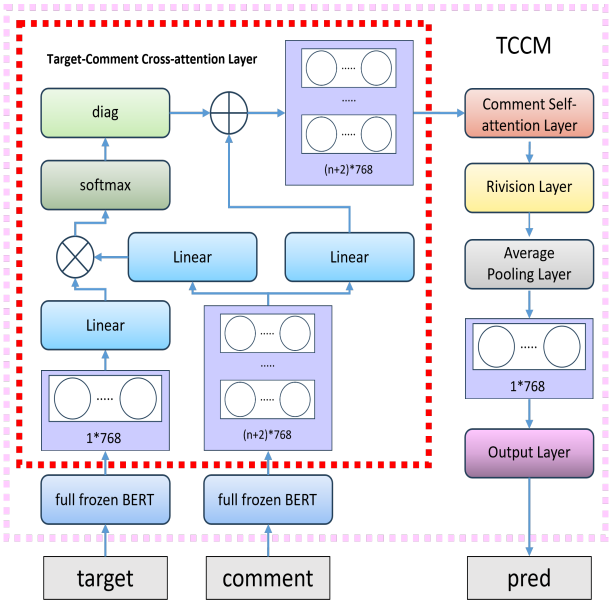

Figure 1: Structure of the TCCM

only pay attention to neighboring words(Devlin et al. 2018). In this model, because of this advantage, fully frozen BERT was used as the encoder to extract features, and the chinese-bert-wwm version of HIT was selected(Cui et al. 2021). The object and comment are fed into the BERT model as "$[CLS]t[SEP]$" and "$[CLS]c[SEP]$", respectively, to obtain the corresponding eigenvectors h_t and h_c. For completeness, h_t selects the feature vector of BERT output representing the whole sentence, while h_c selects the vector group composed of the feature vectors of each word in the sentence output by BERT.

Target-Comment Cross-attention Layer

In this layer, we use the target feature vector h_t and the comment feature vector h_c to compute the attention with trainable parameters, and then we reconstruct the comment feature vector by using the attention weight of each character of the comment. Among the six trainable parameters, the three coefficients are denoted as W_{qtc}, W_{ktc}, W_{vtc} respectively, the dimensions are $768 * 768$, and their intercepts are denoted as b_{qtc}, b_{ktc}, b_{vtc}. In fact, three fully connected layers are used to perform linear operations on h_t, h_c, h_c respectively to obtain three matrices Q, K, V namely:

$$Q = W_{qtc} \cdot h_t + b_{qtc} \tag{1}$$

$$K = W_{ktc} \cdot h_c + b_{ktc} \tag{2}$$

$$V = W_{vtc} \cdot h_c + b_{vtc} \tag{3}$$

Then, the softmax operation is performed after the multiplication of the transpose of Q and K divided by $\sqrt{n+2}$. At this time, the dimension is $1 * (n+2)$, and each value in the vector represents the attention weight assigned to the character at the corresponding position. This vector is converted into a diagonal matrix M of dimension $(n+2) * (n+2)$, namely:

$$M = diag(softmax(\frac{Q \cdot K^T}{\sqrt{n+2}})) \tag{4}$$

Finally, M and V are multiplied to obtain the feature vector h_{tc} of the reconstructed comment, namely:

$$h_{tc} = M \cdot V = diag(softmax(\frac{Q \cdot K^T}{\sqrt{n+2}})) \cdot V \tag{5}$$

It can be seen that the main function of this layer is to use the target to pay attention to each character of the comment, and reduce the feature vector of the corresponding character according to the obtained weight, and the required attention can be reflected according to the degree of reduction.

Comment Self-attention Layer

In this layer, self-attention with parameters is done for the new feature vector of the comment reconstructed by the Target-Comment Cross-attention Layer, expecting that the feature vector of each character can learn the new features reconstructed by the feature vectors of other characters in the same sentence. The method directly uses the formula of the self-attention mechanism of the coding block of Transformers. The output is referred to as h_{tc-cs}.

Revision Layer

This layer is mainly set with reference to the Add&Norm structure of the encoder block of Transformers, the main purpose is to prevent overfitting and accelerate the convergence speed. The input of this layer is the vector obtained by adding the feature vector h_{tc-cs} of the comment that completed the attention reconstruction twice and the original feature vector h_c of the comment. The structure of this layer consists of three parts: a 768 wide fully connected layer, a 0.3 Dropout layer, and a LayerNorm layer. The output of this layer is h_{c-new}.

Average Pooling Layer

This layer mainly performs average pooling on the feature vector group representing the comment, that is, the feature vectors representing each character are averaged to form the sentence feature vector h_{c-pool} representing the whole comment. Obviously, h_{c-pool} is a $1 * 768$ dimensional vector.

Output Layer

This layer is mainly composed of a 128 wide fully connected layer, a 64 wide fully connected layer, a wide number of classified fully connected layer and softmax block, and the GELU function is used to activate between the fully connected layers. Approximate representation is adopted due to the incomputability of the GELU.(Hendrycks and Gimpel 2016)

$$GELU(x) = 0.5x(1 + tanh(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3))) \tag{6}$$

Loss Function

The loss function used in model learning is cross-entropy loss function.

Optimizer

The optimizer uses the AdamW optimizer, a variant of the Adam optimizer proposed by Loshchilov and Hutter in 2017.(Loshchilov and Hutter 2017)

Experiment

Data Sets

In this study, two data sets were used to investigate the performance of the proposed model. The experiment of binary classification is carried out on the Product Review Dataset. On the Expanded Weibo Stance Detection Dataset, a three-classification experiment was carried out.

(1)Product Review Dataset It's a public data set from Chinese IT technology community CSDN(TanXiao&&life 2022). A total of 12 products are taken as targets, which are divided into two categories: opposing and supporting. There are total of 68,636 pieces of data.

(2)Expanded Weibo Stance Detection Dataset Modeled as a three-class classification, the dataset is expanded based on the dataset of the Weibo stance detection task at the 2016 NLPCC(NLPCC-ICCPOL2016 2016). The original dataset contained 8001 pieces of data including seven targets. In this study, on the basis of completing annotated tasks with remaining unannotated data, we added a total of 7,110 pieces of data from eight targets obtained from Chinese social media. The new data set consists of 15,111 annotated data on 15 topics.

Baselines

(1)BERT-Join Inspired by the ideas of Emily Allaway and Kathleen McKeown(Allaway and McKeown 2020). The input is in the form of "$[CLS]t[SEP]c[SEP]$". After the feature extraction of sentence pairs is completed by BERT, it is directly sent to the output layer for prediction.

(2)BERT-Concat Based on a baseline model used in the research of Anjun Wang and KaiKai Huang(Wang, Huang, and Lu 2019), after feature extraction, the target feature vector and comment vector are concatenated and then sent to the output layer for prediction.

(3)BERT-Add A baseline model proposed in this study, after the target and comment are sent to the BERT model to complete feature extraction, the two feature vectors h_t and h_c are added in a weighted way to obtain h_{input}, with the formula as follows:

$$h_{input} = w_{target} \cdot h_t + (1 - w_{target}) \cdot h_c \tag{7}$$

Where w_{target} is a learnable parameter and is initialized to 0 at the beginning of the training, and h_{input} is sent to the output layer for prediction.

(4)BERT-BiLSTM Drawing on the ideas of Isabelle Augenstein, Tim Rocktäschel et al.(Augenstein et al. 2016), the target and the comment are sent to the BERT model for feature extraction. The feature vector of the target is taken as the initial state, and the feature vector of the comment is taken as the input word, and inputted into the BiLSTM, so as to reconstruct the comment feature vector containing the target information. Then concat of the final states of the two directions is sent to the output layer for prediction.

Implements

All experiments in this study adopt the variable learning rate, using the 1e-5 learning rate in the first 10 epochs, the 5e-6 learning rate in the 11-20 epochs, and the 1e-6 learning rate from the 21st epoch. On the binary task, we chose to train TCCM for 30 epochs, and on the three-category we chose to train it for 40 epochs. The evaluation indexes include the loss on the training set and the accuracy on the test set.

Comparative Experiment on Product Review Dataset

The segmentation of the training set and the test set is performed with a ratio of about 8:2. The index curves is shown in Figure 2.

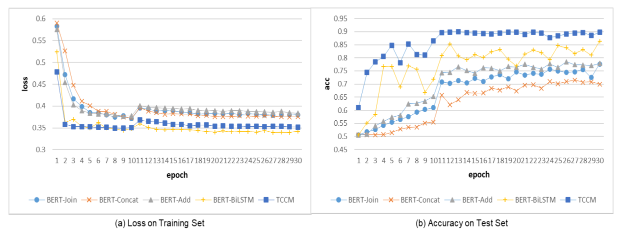

Figure 2: Curves in Binary Classification Task

In terms of training set loss, the TCCM converges faster, and has a certain degree of decline compared with BERT-Join, BERT-Concat and BERT-Add, while slightly higher than BERT-BiLSTM. In terms of test set accuracy, the TCCM has a large improvement compared with BERT-Join, BERT-Concat, BERT-Add and BERT-BiLSTM, and also shows more stable convergence. Select the cases with the highest accuracy of each model on the test set and analyze the prediction distribution, as shown in Table 1.

Table 1: Distribution of the Highest Accuracy of Each Model on the Test Set in the Binary Classification Task

Model	TP	FP	TN	FN
BERT-Join	6,670	2,789	3,985	282
BERT-Concat	6,736	3,677	3,097	216
BERT-Add	6,743	2,744	4,030	209
BERT-BiLSTM	6,703	1,525	5,249	249
TCCM	6,657	1,343	5,431	295

It can be seen that all models are more inclined to predict the supporting stance. TCCM is superior to other baseline models in FP and TN, indicating that TCCM is more likely to make correct predictions for the samples whose stances are opposing. At the same time, FN is higher than all baseline models, which means that the model in this study is more likely to make wrong predictions when predicting the samples which the true stances are supporting, but even then the error rate is only $295/(6657 + 295) = 4.24\%$.

Comparative Experiment on Expanded Weibo Stance Detection Dataset

The segmentation of the training set and the test set is also performed at a ratio of about 8:2. Figure 3 shows the indicator curve.

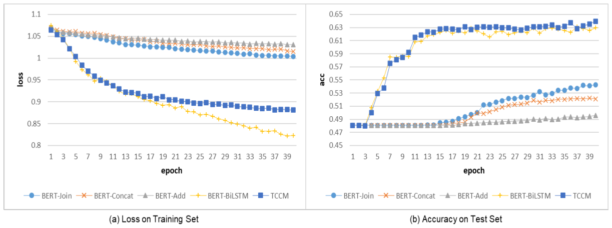

Figure 3: Curves in Three-category Classification Task

As can be seen from the training set loss curve, the final convergence point of the proposed model on the training set is far better than the three baseline models BERT-Join, BERT-Concat and BERT-Add, but slightly worse than BERT-BiLSTM. However, from the accuracy curve of the test set, The final convergence point of the proposed model on the test set is better than that of all the baseline models.Therefore, it can be said that the proposed model has improved to some extent compared with all the baseline models.

Similarly, the version with the highest accuracy of the test set of each model is selected to analyze the prediction distribution, as shown in Table 2. In Table 2, the arrows in front indicate the actual stance of the comment, the arrows behind indicate the predicted stance, S indicates supporting, N indicates neutral/irrelevant, and O indicates opposing.

Table 2: Distribution of the Highest Accuracy of Each Model on the Test Set in the Three-category Classification Task

Model	**S→S**	**S→N**	**S→O**
BERT-Join	18	490	49
BERT-Concat	0	536	21
BERT-Add	0	526	31
BERT-BiLSTM	224	215	118
TCCM	250	250	57
Model	**N→S**	**N→N**	**N→O**
BERT-Join	4	1,009	37
BERT-Concat	0	1,043	7
BERT-Add	0	1,025	25
BERT-BiLSTM	78	828	144
TCCM	85	874	91
Model	**O→S**	**O→N**	**O→O**
BERT-Join	2	469	2
BERT-Concat	0	530	30
BERT-Add	0	496	64
BERT-BiLSTM	26	242	292
TCCM	23	323	214

It can be seen that BERT-Join, BERT-Concat and BERT-Add basically go to the extreme, rarely predicting supporting and opposing, and basically focusing on neutral/irrelevant predictions, especially BERT-Concat and BERT-Add do not even make supporting predictions. The main point is to compare the differences between BERT-BiLSTM and the proposed model.

Abolation Study

We did the three following ablation studies:

(1)TCNotLearing There are no learnable parameters in the Target-Comment Cross-attention layer, and the formula is converted to:

$$h_{tc} = diag(softmax(\frac{h_t \cdot (h_c)^T}{\sqrt{n+2}}) \cdot h_c \quad (8)$$

(2)TCCancel Target-Comment Cross-attention Layer is canceled.

(3)ChooseCLS The feature vector of the whole sentence of a comment is not obtained by means of average pooling, but by using the feature vector of the character "$[CLS]$"

Table 3: The Result of Abolation Study(Test accuracy)

Model	**Binary**	**Three-category**
TCNotLearing	0.897421	0.619580
TCCancel	0.885619	0.624034
ChooseCLS	0.855238	0.622536
TCCM	0.898878	0.639619

Table 3 shows the optimal performance of the three ablation methods and the complete TCCM on the test set of the two data sets, and it can be clearly seen that the three ablation methods show a certain gap compared with the complete TCCM. This shows that the Target-Comment Cross-attention Layer and its learnability can improve the overall performance of the model, and it is more reasonable to use the average pooling as the feature vector of the whole sentence than the feature vector of "$[CLS]$" as the global feature vector.

Conclusion

In this study, a model based on BERT and attention mechanism, specially based on target-comment cross-attention is proposed to solve the stance detection. Two sets of experiments were used to investigate the performance improvement of the TCCM compared to the four baseline models. Finally, according to the results of the experiments, the following points can be drawn: (1) Compared with the four baseline models, the performance of the proposed model is improved, among which the improvement of BERT-Join, BERT-Concat and BERT-Add is more obvious, and that of BERT-BiLSTM is improved to a certain extent; (2)Through ablation study, we demonstrate the value of the core structure,Target-Comment Cross-attention Layer, to the overall performance of the model and the superiority of using average pooling as the feature of the whole sentence;(3)The proposed model shows similar experimental results on both binary classification and three-classification tasks, indicating that the proposed model has certain robustness.

References

Allaway, E.; and McKeown, K. 2020. Zero-shot stance detection: A dataset and model using generalized topic representations. *arXiv preprint arXiv:2010.03640*.

Augenstein, I.; Rocktäschel, T.; Vlachos, A.; and Bontcheva, K. 2016. Stance detection with bidirectional conditional encoding. *arXiv preprint arXiv:1606.05464*.

Chen, K.; Zhou, H.; and Wang, G. 2023. Study of Stance Detection Based on Pre-training Model and Memory Convolution Network. *Computer Technology and Development*, 33(08): 180–185.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; and Yang, Z. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 3504–3514.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dian, Y.; Jin, Q.; and Wu, H. 2017. Computer Engineering and Applications. *Stance detection in Chinese microblogs via fusing multiple text features*, 53(21): 77–84.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Kawintiranon, K.; and Singh, L. 2021. Knowledge enhanced masked language model for stance detection. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, 4725–4735.

Li, J.; and Yang, P. 2022. Survey on Single Target Stance Detection. *Information Computer(Theory Edition)*, 34(04): 70–72.

Li, Y.; Sun, Y.; and Li, W. 2021. Survey of Text Stance Detection. *Journal of Computer Research and Development*, 58(11): 2538–2557.

Liang, B.; Fu, Y.; Gui, L.; Yang, M.; Du, J.; He, Y.; and Xu, R. 2021. Target-adaptive graph for cross-target stance detection. In *Proceedings of the Web Conference 2021*.

Liu, C.; Li, W.; Demarest, B.; Chen, Y.; Couture, S.; Dakota, D.; Haduong, N.; Kaufman, N.; Lamont, A.; Pancholi, M.; et al. 2016. IUCL at SemEval-2016 Task 6: An Ensemble Model for Stance Detection in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

MG, Y. 2023. How will stance detection technology evolve after ChatGPT? https://blog.csdn.net/weixin_41964296/article/details/129811025.

Mohtarami, M.; Baly, R.; Glass, J.; Nakov, P.; Màrquez, L.; and Moschitti, A. 2018. Automatic stance detection using end-to-end memory networks. *arXiv preprint arXiv:1804.07581*.

NLPCC-ICCPOL2016. 2016. Stance Detection in Chinese Microblogs. http://tcci.ccf.org.cn/conference/2016/pages/page05_evadata.html.

Sun, Y. 2022. Research on Key Technologies of Text Stance Detection for Social Media.

TanXiao&&life. 2022. Twelve categories of product reviews dataset. https://download.csdn.net/download/weixin_54626591/85354505?utm_medium=distribute.pc_relevant_download.none-task-download-2~default~BlogCommendFromBaidu~Rate-2-85354505-download-85098073.pc_relevant_download_topnsimilarv2&depth_1-utm_source=distribute.pc_relevant_download.none-task-download-2~default~BlogCommendFromBaidu~Rate-2-85354505-download-85098073.pc_relevant_download_topnsimilarv2&dest=https%3A%2F%2Fdownload.csdn.net%2Fdownload%2Fweixin_54626591%2F85354505&spm=1003.2020.3001.6616.2.

Wang, A.; Huang, K.; and Lu, L. 2019. Stance Detection in Chinese Microblogs via Bert-Condition-CNN Model. *Computer Systems Applications*, 28(11): 45–53.

Wang, T.; Yuan, J.; Qi, R.; and Li, Y. 2023. Multi-type Knowledge-Enhanced Microblog Stance Detection Model. *Journal of Guangxi Normal University (Natural Science Edition)*.

Wei, W.; Zhang, X.; Liu, X.; Chen, W.; and Wang, T. 2016. pkudblab at semeval-2016 task 6: A specific convolutional neural network system for effective stance detection. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, 384–388.

Zarrella, G.; and Marsh, A. 2016. Mitre at semeval-2016 task 6: Transfer learning for stance detection. *arXiv preprint arXiv:1606.03784*.

Zheng, H.; Gao, J.; and Qiu, J. 2018. Computer Systems Applications. *Stance Detection in Chinese Microblog Topic Based on Word Embedding Technology and Thematic Words Feature*, 27(09): 118–123.