Neural Computing and Applications

Stance Classification Model with Knowledge-aware Multi-feature Attention Network

--Manuscript Draft--

Manuscript Number:	NCAA-D-23-05679
Full Title:	Stance Classification Model with Knowledge-aware Multi-feature Attention Network
Article Type:	Original Article
Keywords:	Social network analysis, Stance classification, BERT, Knowledge-aware attention network
Abstract:	Stance classification aims to identify the stance conveyed in tweets towards a specific target. Recent works have been devoted to leveraging target word embedding to incorporate target information into the stance classification model. However, it is difficult to capture implicit target information solely through target word embedding. In addition, stance knowledge is often ignored in previous work. To address these issues, this paper proposes a novel Stance Classification model with Knowledge-aware Multi-feature Attention Network (SC-KMAN). Firstly, we introduce richer target information into the model through the target information extractor T-BERT proposed in this paper. Meanwhile, we introduce a sentiment feature extractor S-BERT by transfer learning. Then, we propose a Knowledge-based Multi-feature Attention Network (KMAN) to introduce stance knowledge into the stance detection model. Under the guidance of stance knowledge, KMAN comprehensively analyzes the clues provided by stance, sentiment, and target features to obtain accurate stance detection results. The experimental results on Twitter datasets demonstrate that SC-KMAN achieves state-of-the-art performance (avg F1=74.53%).

Stance Classification Model with Knowledge-aware Multi-feature Attention Network

Chao Meng[1*], Binxing Fang[1], Hongli Zhang[1], Yuchen Yang[1], Gongzhu Yin[1], Kun Lu[1]

[1]School of Cyberspace Science, Harbin Institute of Technology, Harbin, 150001, Heilongjiang, China.

*Corresponding author(s). E-mail(s): mengchao@hit.edu.cn;
Contributing authors: bxfang@pact518.hit.edu.cn; zhanghongli@hit.edu.cn; yangyc@hit.edu.cn; yingz@hit.edu.cn; lukun@hit.edu.cn;

Abstract

Stance classification aims to identify the stance conveyed in tweets towards a specific target. Recent works have been devoted to leveraging target word embedding to incorporate target information into the stance classification model. However, it is difficult to capture implicit target information solely through target word embedding. In addition, stance knowledge is often ignored in previous work. To address these issues, this paper proposes a novel Stance Classification model with Knowledge-aware Multi-feature Attention Network (SC-KMAN). Firstly, we introduce richer target information into the model through the target information extractor T-BERT designed in this paper. Meanwhile, we introduce a sentiment feature extractor S-BERT by transfer learning. Then, we propose a Knowledge-based Multi-feature Attention Network (KMAN) to introduce stance knowledge into the stance detection model. Under the guidance of stance knowledge, KMAN comprehensively analyzes the clues provided by stance, sentiment, and target features to obtain accurate stance detection results. The experimental results on Twitter datasets demonstrate that SC-KMAN achieves state-of-the-art performance (avg F1=74.53%).

Keywords: Social network analysis, Stance classification, BERT, Knowledge-aware attention network

1 Introduction

Social network has become an essential component of people's lives. An increasing number of individuals are accustomed to expressing their opinions on social networks. Stance classification, which aims to identify the stance conveyed in tweets towards a specific target as "FAVOR," "AGAINST," or "NONE," is a significant task in opinion mining. It is also helpful for various other tasks, including rumor detection [17, 21, 37, 38], community detection [6, 9], and election prediction [16, 20, 35], among others.

In the field of stance classification, previous studies primarily focused on congressional debates [30] and debate forums [19]. Recently, the study on stance classification in social networks has rapidly grown [7, 14, 15, 28], and the mainstream data platform for stance classification has gradually become Twitter. Since authors often obscure their stance expressions, it is difficult to detect the stance conveyed in tweets effectively. As depicted in Table 1, these tweets do not explicitly mention the target but express opinions on the target "Legalization of Abortion".

At present, various stance classification models in Twitter have been proposed, mainly including feature engineering based models [1, 15] and neural network based models [23, 31, 44, 45]. In feature engineering based methods, efficient features have a significant impact on performance. However, substantial human labor is needed to extract efficient features. In addition, the efficient features required by the stance classification model toward different targets are generally different. Therefore, weak adaptability and high migration cost limit the development of feature engineering based models.

Neural network based methods, which can automatically learn text features during training, have been proposed for stance classification in Twitter. However, the performance of the primary convolutional neural network (CNN) [34] and recurrent neural network (RNN) [40] models for stance classification in Twitter is unsatisfactory. The attention mechanism, which can effectively identify the importance of words in sentences, has been widely employed in stance classification models and gained excellent performance. A hierarchical attention model [29] was proposed to capture the weights within and among different text feature sets. Although the model's performance has improved significantly, the target information is ignored as in the previous models. To alleviate this problem, various models [22, 36, 39, 41] incorporating target word embedding are proposed, and the performance is further improved. However, the performance improvement remains unsatisfactory for sentences without target word, indicating that the target information is not fully utilized.

In summary, although the existing stance classification models have achieved considerable performance, there is still plenty of room for improvement. In this paper, we propose a hypothesis based on the current study results and experience: stance classification strongly correlates with target features, sentiment features, and stance features. Target features encompass valuable target-related information that aids the model in identifying the relationship between the text and the target. Sentiment features assist in analyzing the author's sentiment toward the target. Stance features facilitate the analysis of stance polarity towards the target. According to this hypothesis, we present a novel stance classification model with knowledge-aware multi-feature attention network. Under the guidance of stance knowledge, our model comprehensively analyzes the clues provided by stance, sentiment, and target features to achieve accurate stance detection results. Our main contributions can be summarized as follows:

- We propose a stance knowledge extractor to automatically extract stance knowledge from the training dataset. Additionally, a target feature extractor T-BERT is designed to extract richer target information from text. Furthermore, transfer learning is utilized to design a sentiment feature extractor S-BERT.
- We propose a novel stance classification model with knowledge-aware multi-feature attention network(SC-KMAN).[1] Guided by stance knowledge extracted automatically from training dataset, the model achieves more accurate stance classification results by comprehensively capturing and analyzing the stance features, sentiment features and target features with rich target information.
- Extensive experiments are conducted to analyze the impacts of various components on the performance of our proposed model. In addition, the experimental results on Twitter datasets demonstrate that SC-KMAN achieves state-of-the-art performance.

2 Related work

Recently, the study of stance classification on social platform has rapidly grown [2, 11, 15, 26]. In particular, SemEval-2016 [24] introduced a sharing task designed to detect the stance of tweets and provided a publicly available dataset, establishing Twitter as a mainstream platform for stance classification.

At present, several stance classification models have been proposed for twitter, including feature engineering based models [1, 15, 25] and neural network based models [4, 13, 39]. In the feature engineering based models, Elfardy and Diab [12] extracted word features, latent semantic features, sentiment features, psychological dictionaries and other features. Then, they utilized the SVM

[1] https://github.com/MengChaoHIT/SC-KMAN

Table 1 Examples for stance classification

Tweet	Target	Stance
I will fight for the unborn !	Legalization of Abortion	AGAINST
The government has given no explanation of why the law was changed macedonia hrctte	Legalization of Abortion	AGAINST
It very simple let women choose repealthe8th notacriminal	Legalization of Abortion	FAVOR

model to detect the stance of texts. Their study demonstrates that sentiment features are significant for stance classification. Mohammad et al [25] described the detailed construction process of the stance classification dataset and analyzed the stance polarity distribution, the sentiment polarity distribution, the label distribution whether tweets contain the target word and the joint distribution between these labels. Their study highlights that the word feature and whether the target word is contained in the text significantly impact the stance classification. Al-Ghadir et al [1] created word and stem dictionaries, using them to extract the top-k words and top-k stems of sentences as feature vectors. In addition, sentiment dictionaries are employed to extract the sentiment features of sentences. Finally, a weighted KNN is utilized as a classifier to achieve sentence stance classification. Based on the Semeval-2016 Task6.A dataset, Aldayel and Magdy [2] expanded the interactive data among social users and realized the stance detection by analyzing social texts and interactive data. Gómez-Suta et al [15] divided stance detection into two stages: the first stage detects whether the text has a stance, and the second stage detects the stance polarity of the text. According to this opinion, a two-stage classification model is designed to achieve interpretable tweet stance detection. In summary, in the feature engineering based models, researchers have designed various text features, including syntactic dependencies, n-grams, discourse markers, and frame-semantic, to capture the syntactic, literal, pragmatic, and semantic information of the text. However, extracting efficient features always requires substantial human resources. In addition, the high-efficiency features required for different targets are generally different, resulting in weaker adaptability of the model based on feature engineering and higher migration costs.

To liberate researchers from feature engineering, the neural network method that can automatically learn text features during the training process is proposed. Wei et al [34] employed CNN for stance classification through cross-validation and voting. Zarrella and Marsh [40] utilized additional datasets related to the target to train word vectors and marked the data of a specific hashtag as the corresponding pseudo-label. Then, the dataset containing the pseudo-label is used to train RNN. Finally, the word vector and RNN model obtained by pre-training were fine-tuned to complete the stance classification of various data. Their study demonstrates that pre-training models using large datasets can be improved by transferring to the target dataset. Attention mechanism has attracted widespread attention since it was proposed. Sun et al [29] utilized long short-term memory (LSTM) network to extract multiple text language features, and then employed hierarchical attention to capture the attention of multiple text linguistic features and the attention of different feature sets. The attention of the text's linguistic features can allow the model to acquire the weight of each feature in the same feature set. The attention of the feature sets can enable the model to obtain the weight of each feature set. Siddiqua et al [27] used a multi-kernel convolution network to represent text and employed two variants of LSTM with attention to obtaining higher-level representations of sentences to improve text stance classification. Kawintiranon and Singh [20] utilized BERT with an enhanced stance vocabulary to achieve stance detection and verified the performance of their model on a Twitter dataset related to the US presidential election. Despite these methods extract richer text features, they ignore the target information. A multi-task attention tree neural network (MATNN) [4] was proposed to jointly classify stances and detect rumor veracity. A tree self-attention mechanism is utilized to extract local features for stance classification. The effectiveness of MATNN is verified on two social network datasets.

Du et al [10] utilized the attention mechanism to introduce target information in stance detection model, effectively improving the preference. Zhou et al [43] also integrated target information into the attention model and utilized CNN to extract text representation, which contains richer feature information. The effect of stance detection was further improved. Wei et al [32] employed external memory to utilize the previous target attention vector as part of the target attention input to obtain more information about the target and text interaction, significantly improving the effect of stance classification. Zhou et al [42] not only embedded the target word into the model, but also used the CNN model with self-attention to extract target-relevant text features while filtering out irrelevant features. Their method obtained a considerable effect of stance classification. Zhao and Yang [41] proposed a multi-dynamic routing capsule network architecture simulating hierarchical clustering to aggregate the word features and transferred the aggregated feature vectors to the category capsules as the feature representation of the text in various categories to achieve stance classification. Yang et al [36] utilized BERT to learn text representation with target word embedding and employed three convolution networks to learn various stance representations of text. Alturayeif et al [3] summarized the study related to stance classification. Hardalov et al [18] proposed a cross-domain label adaptive stance detection model and verified the model's effectiveness on 16 datasets. Li and Caragea [22] employed Auxiliary Sentence based Data Augmentation (ASDA) and Conditional BERT (CBERT) for data augmentation, enhancing the training of the stance detection model. Chen et al [5] combined the feature representation of N-grams and BERT to realize stance detection using specific target attention. Conforti et al [7] incorporated tweets, stock information, and financial information into a multi-tasking model for stance detection in the financial field. Liang et al [23] expanded the dataset using data augmentation and employed contrast learning to achieve zero-shot stance detection, demonstrating the effectiveness of their model on three datasets. Zhu et al [45] utilized target background information from Wikipedia to enhance the performance of the zero-shot stance detection model and verified its effectiveness on three datasets. Yuan et al [39] realized the stance detection by analyzing whether the text contains the target word, whether the text has a stance, and the specific stance on the target. They constructed multiple datasets to verify the effectiveness of their model. Fu et al [13] utilized two additional artificial labels, sentiment label and opinion-towards label, and employed multi-task learning for text stance detection. Dramatically improves the performance of stance classification.

In summary, the mainstream stance classification method integrates target information into the model to enhance performance. However, the methods utilizing target word embedding struggle to effectively detect the conveyed stance in texts without the target word, resulting in underutilization of target information. Although the literature [13] utilized two additional artificial labels to achieve significant performance, the artificial opinion-towards label is not readily available. To address these challenges, this paper proposes a novel stance classification model with knowledge-aware multi-feature attention network (SC-KMAN), which not only extracts richer target information but also leverages the stance knowledge automatically extracted from the training datasets to guide the inference process of SC-KMAN.

3 Methodology

In this section, we provide the problem definition and present the novel Stance Classification model with Knowledge-aware Multi-feature Attention Network (SC-KMAN). Under the guidance of stance knowledge extracted automatically from the training dataset, SC-KMAN achieves more accurate stance classification results by comprehensively capturing and analyzing the stance features, sentiment features, and target features with rich target information. The architecture of SC-KMAN is illustrated in Fig. 1, which primarily comprises two components: multi-feature representation and knowledge-aware multi-feature attention network. The problem definition and details of each component are as follows.

3.1 Problem definition

Stance classification can be formulated as follows. Given a text $Text = [w_1, w_2, \ldots, w_n]$ and a specified target $Target = [t_1, t_2, \ldots, t_m]$, the task is to predict the stance of $Text$ towards the $Target$, where w_i and t_j represent the i-th and j-th word in $Text$ and $Target$, respectively, n and m indicate the length of $Text$ and $Target$, respectively.

3.2 Multi-feature representation

The multi-feature representation layer is utilized to extract original stance features T^O, target features T^T, and sentiment features T^S in the text $Text$. The target features indicate whether the text is related to the target $Target$. The transformer-based BERT model with more prominent language expression capabilities has acquired significant performance in various natural language processing tasks compared with Word2Vector, Glove, and ELMo. Therefore, we utilize BERT model as the primary model for various feature extraction.

In order to extract the target features, firstly, we use the pseudo label to generate the target classification dataset according to the stance dataset. If the stance label of the text is not "None" or the text contains the target word, then the target classification label is set to 1, which indicates that it is related to the target. Otherwise, it is set to 0. Then, employ BERT model to fine-tune the target text classification dataset and obtain the T-BERT for extracting target features.

Although the stance dataset lacks sentiment labels, there are numerous public datasets available for sentiment analysis tasks. In order to extract more accurate sentiment features, we select a suitable sentiment dataset, SST-2, from the public sentiment analysis datasets. Subsequently, BERT is utilized to fine-tune the sentiment dataset to obtain the S-BERT for extracting sentiment features.

Finally, BERT, S-BERT, and T-BERT are employed to extract the original stance features T^O, sentiment features T^S, and target features T^T in the text $Text$, respectively. The calculation process is as follows:

$$Text = [w_0, w_1, \cdots, w_n] \tag{1}$$

$$\begin{aligned} T^O &= BERT(Text) \\ &= [T_0^O, T_1^O, \cdots, T_n^O] \end{aligned} \tag{2}$$

$$\begin{aligned} T^S &= S\_BERT(Text) \\ &= [T_0^S, T_1^S, \cdots, T_n^S] \end{aligned} \tag{3}$$

$$\begin{aligned} T^T &= T\_BERT(Text) \\ &= [T_0^T, T_1^T, \cdots, T_n^T] \end{aligned} \tag{4}$$

where w_i represent the i-th word in $Text$, n indicate the length of $Text$. T_i^0, T_i^S, and T_i^T are, respectively, the last hidden states of $BERT$, S_BERT, and T_BERT corresponding to the word w_i.

3.3 Knowledge-aware multi-feature attention network

To ensure that the model can learn stance information more accurately, we designed a stance knowledge-aware multi-feature attention network. Firstly, we design a knowledge extractor to extract stance knowledge from the training dataset. Then employ this stance knowledge to guide the knowledge-aware attention mechanism to learn the stance feature weight. Meanwhile, the general attention mechanism is utilized to learn the target feature weight and sentiment feature weight between words. Ultimately, the text representation is obtained through the comprehensive analysis of various features and stance knowledge.

3.3.1 Stance knowledge extractor

The knowledge extractor is exploited to extract words strongly related to the stance from the training dataset. Initially, TF_IDF was employed to extract the vocabulary from the training dataset.

$$\begin{aligned} Voc &= \mathrm{TF\_IDF}\left(\{Text_i \mid 0 \le i \le N_{train}\}\right) \\ &= \{w_j \mid 0 \le j \le L\} \end{aligned} \tag{5}$$

where N_{train} is the number of train data, L is the number of words in the vocabulary Voc.

To obtain the potential stance vocabulary, filter out words from the vocabulary based on the following conditions: either the number of samples containing the word is less than N, or the word's maximum probability of stance polarity is not less

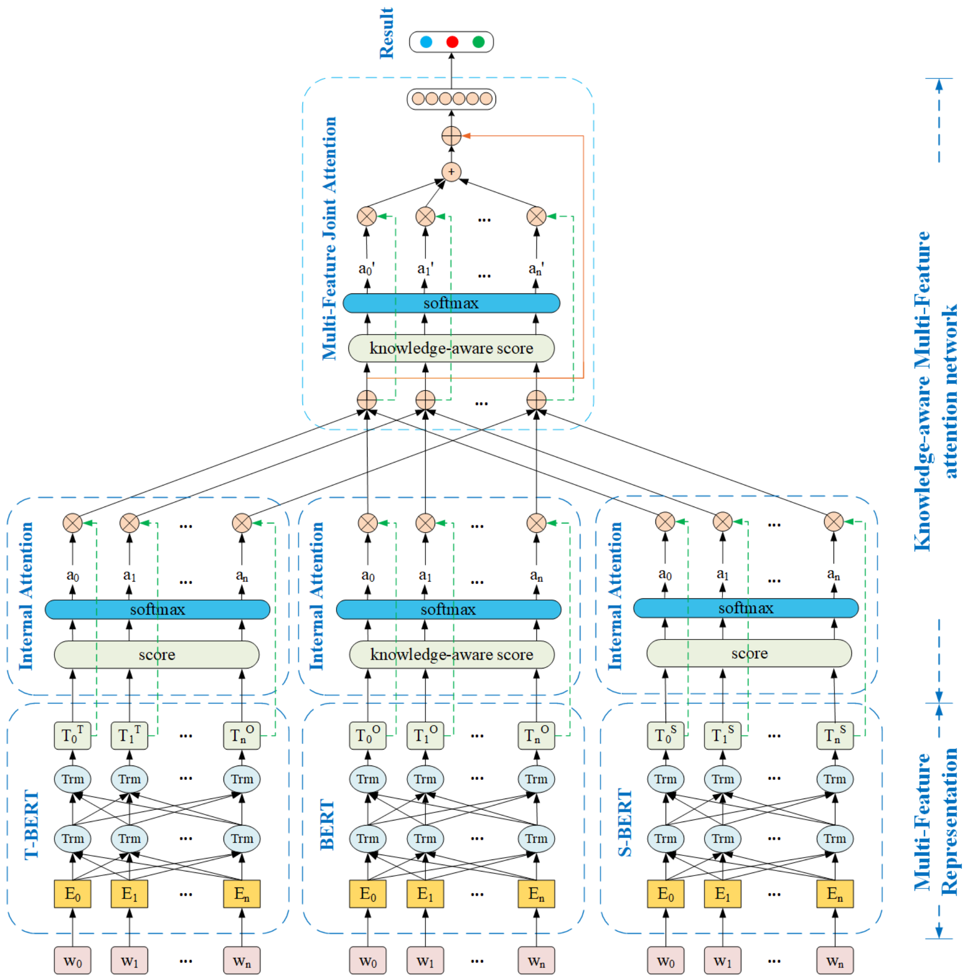

Fig. 1 The architecture of our model SC-KMAN.

than P.

$$Voc_{SK} = \{w_i \mid 0 \leq i \leq L_{SK}, n_{i,0} + n_{i,1} + n_{i,2} \geq N, \\ max\,(p_{i,0}, p_{i,1}, p_{i,2}) \geq P\} \tag{6}$$

where L_{SK} represents the number of words in Voc_{SK}; $n_{(i,0)}$, $n_{(i,1)}$, $n_{(i,2)}$ represent the number of samples in which the stance polarity is "Against", "None", and "Support", respectively, and the word w_i appears in these samples; $p_{i,0}$, $p_{i,1}$, $p_{i,2}$ represent the probability of w_i in different stance polarities.

The stance knowledge extractor SKE generates the stance knowledge vector skv of the text according to the potential stance vocabulary Voc_{SK}.

$$\begin{aligned} skv &= SKE(Text) \\ &= [skv_0, skv_1, \cdots, skv_n] \end{aligned} \tag{7}$$

If the i-th word w_i in $Text$ appears in Voc_{SK}, the corresponding i-th element skv_i in the skv is set to 1; otherwise, it is set to 0.

3.3.2 Knowledge-aware attention

Unlike the conventional attention mechanism, our knowledge-aware attention mechanism considers stance knowledge when calculating the scoring function. Under the guidance of stance knowledge vector skv, the score of stance words is increased. The calculation process of our knowledge-aware scoring function ka_s is as follows:

$$s(T_i, Q) = \frac{(T_i)^T Q}{\sqrt{d}} \tag{8}$$

$$\begin{aligned} max\_score = & Max([s(T_0, Q), s(T_1, Q), \cdots, \\ & s(T_n, Q)]) \end{aligned} \tag{9}$$

$$\begin{aligned} ka\_s(T_i, Q, skv_i) = & s(T_i, Q) + [max\_score - 1] \cdot \\ & [skv_i \odot s(T_i, Q)] \end{aligned} \tag{10}$$

where $\odot$ represents the elementwise multiplication operation, T_i is the i-th element in the key vector, d indicates the dimension of the key vector, and Q represents the query vector.

3.3.3 Multi-feature internal attention

Multi-feature internal attention is utilized to capture the weight of each feature in each type of feature set and the corresponding sentence representation for each feature type.

Firstly, calculate the scores S_i^O, S_j^S, and S_k^T for the features T_i^O, T_j^S, and T_k^T in the basic feature T^O, target feature T^T, and sentiment feature T^S of the text according to the output of multi-feature representation layer.

$$S_i^O = ka\_s\left(T_i^O, Q^O, skv_i\right) \tag{11}$$

$$S_j^S = s\left(T_j^S, Q^S\right) \tag{12}$$

$$S_k^T = s\left(T_k^T, Q^T\right) \tag{13}$$

Then, calculate the weights a_i^O, a_j^S, and a_k^T for the features T_i^O, T_j^S, and T_k^T according to the scores S_i^O, S_j^S, and S_k^T. Meanwhile, the feature representations $T_i^{O'}$, $T_j^{S'}$, and $T_k^{T'}$ are updated. The calculation process for a_i^O and $T_i^{O'}$ is as follows:

$$a_i^O = \frac{\exp\left(S_i^O\right)}{\sum_{l=0}^{n} \exp\left(S_l^O\right)} \tag{14}$$

$$T_i^{O'} = a_i^O T_i^O \tag{15}$$

The calculation process for a_j^S, a_k^T,$T_j^{S'}$ and $T_k^{T'}$ is the same as a_i^O and $T_i^{O'}$.

3.3.4 Multi-feature joint attention

Multi-feature joint attention is employed to capture the comprehensive weight of multiple features corresponding to words and the comprehensive representation of texts. As shown in Eq. (16), the various features $T_i^{O'}$, $T_i^{S'}$, and $T_i^{T'}$ corresponding to the i-th word are concatenated to obtain a comprehensive feature T_i' for the word w_i in text $Text$.

$$T_i' = T_i^{O'} \left| T_i^{S'} \right| T_i^{T'} \tag{16}$$

Under the guidance of stance knowledge, calculate the attention weight of each word's comprehensive features via Eq. (17) and Eq. (18).

$$S_i' = ka\_s\left(T_i', Q', skv_i\right) \tag{17}$$

$$a_i' = \frac{\exp\left(S_i'\right)}{\sum_{j=0}^{n} \exp\left(S_j'\right)} \tag{18}$$

Finally, the comprehensive representation H of the text $Text$ is calculated according to the comprehensive feature T_i' and the weights a_i'.

$$H = \sum_{i=0}^{n} a_i' T_i' \tag{19}$$

where T_i' represents the comprehensive feature of word w_i in text $Text$, a_i' represents the weight of the comprehensive feature T_i'.

3.4 Stance classification

In order to determine the stance polarity of the text, the essential representation T_0' and multi-feature comprehensive representation H are concatenated as the text representation. Meanwhile, $softmax$ is employed to realize the stance classification of the text and obtain the stance classification result R_{Text}. The calculation process is as follows:

$$\bar{H} = \tanh\left(W\left[H \mid T_0'\right] + b\right) \tag{20}$$

$$R_{Text} = \mathrm{softmax}(\bar{W}\bar{H} + \bar{b}) \tag{21}$$

where $\mid$ denotes the concatenation operation, the weights W, $\bar{W}$ and the bias b, $\bar{b}$ are learnable variables.

3.5 Training

When training the model, we utilize the cross-entropy loss function to adjust the model parameters. The loss function is shown in Eq. (22).

$$L(\theta) = -\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log p_{ij} + \frac{\lambda}{2}\|\theta\|^2 \tag{22}$$

where n represents the number of training samples, m represents the number of stance polarity categories, y_{ij} denotes the actual probability of the i-th sample belonging to the j-th category, and p_{ij} indicates the predicted probability of the i-th sample belonging to the j-th category. To prevent overfitting of the model, L2 regularization is utilized to limit the parameters. Variable θ denotes all model parameters that need to be optimized, and λ indicates the regularization coefficient. In addition, Adam is employed to optimize the parameter search process during model training.

4 Experiments and dataset

In this section, we firstly introduce the datasets, evaluation indicators, and the experimental parameter settings. Then, we introduce several competitive baselines on stance classification and compare our model's performance with these competitive baseline models according to the experimental results. Finally, ablation experiments and case studies are given to illustrate our model's effectiveness.

4.1 Dataset

To confirm the performance of our model, we conduct experiments on the Semeval-2016 Task6.A dataset[24], similar to the baseline methods. In addition, we employ pseudo-label technology to generate a target text classification dataset based on the stance classification dataset, which is employed to train the target feature extractor T-BERT. A sentiment dataset is utilized to train the sentiment features extractor S-BERT. Subsequently, the S-BERT is transferred to the stance classification dataset to extract the text sentiment features.

(1) Semeval-2016 Task6.A: Twitter stance classification dataset

In this dataset, the stance ("Against", "Favor", "None") conveyed by more than 4000 tweets is labeled for five different targets: "Atheism" ("Atheism"), "Climate Change is a Real Concern"("Climate"), "Feminist Movement"("Feminist"), "Hillary Clinton"("Hillary"), and "Legalization of Abortion"("Abortion"). The dataset "Five_All" combines the data from all five targets into one dataset. The details of the dataset are shown in Table 2.

(2)Target text classification dataset

To extract the target feature, which determines whether the text is related to the target, we utilize pseudo-labeling technology to create a text classification dataset based on the stance dataset.This dataset is then used to train the target feature extractor T-BERT. The specific generation method of the dataset has been thoroughly described in Section 3.2. The distribution of the dataset is shown in Table 3.

(3)Sentiment dataset

Since the stance dataset does not include sentiment labels, it is challenging to obtain a sentiment dataset that aligns perfectly with the stance dataset. Therefore, we utilize the publicly available sentiment dataset SST-2 [8] to train the sentiment feature extractor S-BERT.

4.2 Evaluation metrics

The micro average F1-score which is employed in the evaluation of Semeval-2016 Task6.A, is adopted as the metrics. Assuming that P and R represent precision and recall, the F1-score for "Favor" and "Against" categories is calculated as follows:

$$F_{Favor} = \frac{2 \times P_{Favor} \times R_{Favor}}{P_{Favor} + R_{Favor}} \tag{23}$$

$$F_{Against} = \frac{2 \times P_{Against} \times R_{Against}}{P_{Against} + R_{Against}} \tag{24}$$

Then, the average of F_{Favor} and $F_{Against}$ is calculated as the final metrics:

$$F_{Average} = \frac{F_{Favor} + F_{Against}}{2} \tag{25}$$

Similar to the evaluation criteria in Semeval-2016 Task6.A, we use $F_{Average}$ to evaluate the prediction performance of the model on each target dataset and "Five_All" dataset.

Table 2 The details of the Semeval-2016 Task6.A dataset.

Target	Train				Test				Total
	Total	Against	None	Favor	Total	Against	None	Favor	
Atheism	513	304	117	92	220	160	28	32	733
Climate	395	15	168	212	169	11	35	123	564
Feminist	664	328	126	210	285	183	44	58	949
Hillary	689	393	178	118	295	172	78	45	984
Abortion	653	355	177	121	280	189	45	46	933
Five_All	2914	1395	766	753	1249	715	230	304	4163

Table 3 The details of the target text classification dataset.

Target	Train			Test			Total
	Total	No	Yes	Total	No	Yes	
Atheism	513	116	397	220	28	192	733
Climate	395	166	229	169	35	134	564
Feminist	664	124	540	285	44	241	949
Hillary	689	27	662	295	76	219	984
Abortion	653	175	478	280	43	237	933
Five_All	2914	608	2306	1249	226	1023	4163

4.3 Parameter setting

In our implementation, we utilize BERT as the primary model for text representation. The word vector's dimension is 768. Besides, five-fold cross-validation is utilized to validate our model, and the prediction result is acquired by voting. The batch sizes and dropout rates are 16 and 0.1, respectively. The optimizer is Adam Optimizer, and the learning rate is 5e-5. The number of hidden units in multi-feature joint attention is set to 256. In the stance knowledge extraction module, the parameters *min_df* and *max_df* of TF_IDF are set to 1 and 1.0, respectively. The threshold N indicating the number of samples containing a potential stance word is set to 8. The maximum probability P for the stance polarity of a word is set to 0.9. In addition, five random seeds were randomly selected for all experiments in this paper, and the mean value of experimental results was taken as the final experimental result.

4.4 Baselines

In order to comprehensively evaluate the performance of SC-KMAN, we compare our model with the following state-of-the-art models.

SVM [24] employs character-level and word-level n-gram as features and SVM as a classifier for text stance classification. This model achieves the best performance in Semeval-2016 Task6.A.

CNN [34] is utilized to represent text semantics. The final stance classification results are obtained by voting the prediction results of the models trained by different sub-datasets.

LSTM [40] is employed to capture text semantics. This model trained by hashtag prediction task on the additional large dataset is transferred to the stance detection dataset.

TAN [10] is the original work introducing the target information into the neural network-based stance classification model. It utilizes the attention mechanism to effectively capture the words related to the target in tweets.

AS-biGRU-CNN [43] introduces the target information by embedding target word. On this foundation, the final representation of the text is extracted by CNN for stance classification.

HAN [29] makes full use of the sentiment, dependence and argument features to represent the text. In addition, hierarchical attention is utilized to capture the internal and mutual importance of various feature sets.

TGMN-CR [33] is a stance detection model with external memory, which learns the stance-related information according to the target-tweet vector representation.

CCNN-ASA [42] utilizes self-attention and CNN to realize stance detection. Additionally, an attention-based compression module is introduced to make the stance-indicative words closer.

PNEM [27](Siddiqua et al., 2019) utilizes a multi-kernel convolution network to represent text and employs two variants of LSTM with attention to obtain higher-level representations of sentences, achieving better text stance classification.

BERT [8] is a widely-used pre-training model in natural language processing, exhibiting significant performance across various tasks. Stance classification can be completed through fine-tuning.

BERT-TAN is a variant of TAN [10], replacing LSTM in TAN with BERT. We employ this model to verify the performance of stance classification model with target word embedding.

BERT-TFAN is another variant of TAN [10], replacing the target word embedding in BERT-TAN with a T-BERT that indicates whether the text is related to the target. We design this model to verify the performance of stance classification model with target feature embedding.

PE-HCN [41] proposes a multi-dynamic routing capsule network architecture simulating hierarchical clustering. It aggregates the features of each word in the tweet and transfers the aggregated feature vectors to the category capsules as the text representation in various categories to achieve stance classification.

SCN [36] utilizes BERT to learn text representation with target word embedding and employs three convolution networks to learn various stance representations of text.

CBERT-ASDA [22] employs ASDA and CBERT for data augmentation and completes the training of the stance detection model according to the enhanced data.

4.5 Results and analysis

The performance of all baselines and our proposed model is listed in Table 4. Note that the results for BERT, BERT-TAN, BERT-TFAN, and SC-KMAN are obtained through our experiments, while the results of other models are directly taken from the original paper. Firstly, we can observe that the basic neural network models, CNN and LSTM, have a general performance on stance classification. TAN and HAN respectively verified that target word embedding and various beneficial features could improve the performance of stance detection. On this basis, various models were designed to achieve considerable performance in stance detection, including TGMN, CCNN-ASA, PNEM, PE-HCN, SCN, and CBERT-ASDA. To compare the impact of target word embedding and target feature embedding on stance detection, we designed two variants of TAN, namely BERT-TAN and BERT-TFAN. BERT-TAN is a stance detection model based on target word embedding, while BERT-TFAN is a stance detection model based on target feature embedding. The performance comparison between BERT-TAN and BERT-TFAN demonstrates that target feature embedding can represent more target information than target word embedding in multiple datasets. By introducing target feature embedding and proposing knowledge-aware attention networks, our model achieved the best performance on the "Five_All" and "Abortion" datasets, the second-best performance on the "Hillary" dataset, and the third-best performance on the "Atheism" and "Feminist" datasets. In the "Climate" dataset, the proportion of "Against" data in the training and test datasets is 3.8% and 6.51%, respectively. From this, it can be seen that there is a problem with data distribution significant imbalance in the "Climate" dataset. Such significant imbalanced data distribution is relatively rare. Therefore, our model does not deal with a data distribution significant imbalance, resulting in average performance on the "Climate" dataset. Excluding the "Climate" dataset, our model has achieved relatively considerable performance on various target datasets. It means that compared with other models, our model not only achieves the best performance on the "Five_all" dataset but also shows insensitivity to the performance ranking across various target datasets.

4.6 Ablation experiments

To investigate the impact of each component of our SC-KMAN model, we compare the full SC-KMAN model with its ablations. The results are shown in Table 5.

Table 4 Performance comparison with baseline models on Semeval-2016 Task6.A dataset.

Model	Atheism	Climate	Feminist	Hillary	Abortion	Five_All
SVM	65.19%	42.35%	57.46%	58.63%	66.42%	68.98%
CNN	63.34%	52.69%	51.33%	64.41%	61.09%	67.33%
LSTM	61.47%	41.63%	62.09%	57.67%	57.28%	67.82%
TAN	59.33%	**53.59%**	55.77%	65.38%	63.72%	68.79%
AS-biGRU-CNN	66.76%	43.40%	58.58%	57.12%	65.45%	69.42%
HAN	70.53%	49.56%	57.50%	61.23%	66.16%	69.79%
TGMN-CR	64.60%	43.02%	59.35%	66.21%	66.21%	71.04%
CCNN-ASA	67.25%	50.05%	61.37%	67.94%	65.61%	71.82%
PNEM	67.73%	44.27%	66.76%	60.28%	64.23%	72.11%
BERT	71.67%	44.86%	59.02%	65.17%	62.47%	72.23%
BERT-TAN	70.73%	44.68%	58.91%	68.25%	65.41%	72.31%
BERT-TFAN	71.35%	44.81%	59.40%	68.34%	65.93%	72.73%
PE-HCN	69.24%	45.31%	**67.52%**	64.93%	61.01%	72.74%
SCN	73.55%	48.41%	61.36%	**71.30%**	65.34%	73.73%
CBERT-ASDA	**74.93%**	-	56.43%	67.01%	61.66%	-
SC-KMAN (ours)	72.53%	45.26%	63.68%	69.01%	**67.11%**	**74.53%**

Table 5 Performance comparison with various experiment setting.

Model	Atheism	Climate	Feminist	Hillary	Abortion	Five_All
BERT	71.67%	44.86%	59.02%	65.17%	62.47%	72.23%
M-BERT	73.28%	44.51%	61.49%	67.19%	65.31%	73.35%
SC-MAN	71.99%	44.55%	60.05%	67.89%	64.85%	73.83%
SC-KMAN-OS	72.20%	44.64%	63.34%	67.05%	66.67%	73.69%
SC-KMAN-OT	**76.93%**	44.75%	63.55%	68.68%	62.92%	73.85%
SC-KMAN	72.53%	**45.26%**	**63.68%**	**69.01%**	**67.11%**	**74.53%**

BERT is a commonly utilized natural language processing model that has achieved remarkable performance in various tasks. M-BERT represents that the knowledge-aware multi-feature attention networks are removed from the SC-KMAN model. Comparing the results of M-BERT and SC-KMAN, we observe that the performance of M-BERT is incomparable with SC-KMAN. It indicates that knowledge-aware multi-feature attention networks are incredibly significant for the SC-KMAN model.

The SC-MAN model is obtained by replacing the knowledge-aware attention in the SC-KMAN model with general attention. The experimental results show that the performance of the SC-MAN model is reduced by 0.7% compared with the SC-KMAN model. It means that the knowledge-aware attention can help the model better recognize the stance conveyed in the text.

To investigate the impact of various features on the performance of SC-KMAN model, S-BERT and T-BERT are removed from SC-KMAN to obtain SC-KMAN-OT and SC-KMAN-OS, respectively. Experimental results show that sentiment features and target features have an impact on the performance of SC-KMAN. Especially, the impact of target features is particularly significant.

4.7 Case study

In this part, we take a review tweets as examples to visualize the knowledge-aware attention network of our SC-KMAN model, as shown in Fig. 2 and Fig. 3. The abbreviations TA, SA, and OA represent internal attention to the original stance feature, sentiment feature, and target feature. JA represents the multi-feature joint attention.

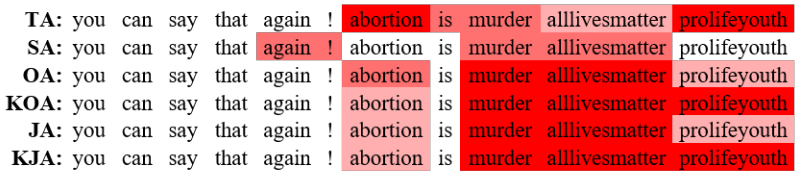

Fig. 2 Visualization of attention weights over a tweet with target word in the SC-KMAN model.

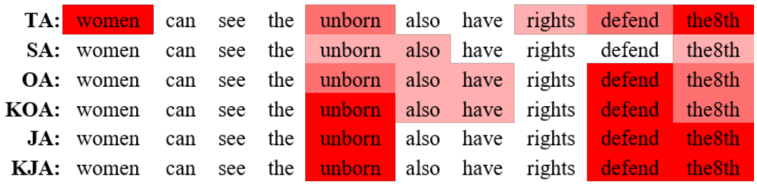

Fig. 3 Visualization of attention weights over a tweet without target word in the SC-KMAN model.

KOA and KJA represent the original stance feature's internal attention with knowledge-aware score and the multi-feature joint attention with knowledge-aware score, respectively. From Fig. 2, we can observe that T-BERT can recognize target word "abortion" from tweets. As shown in Fig. 3, T-BERT can also identify target-related words, such as "unborn" and "the8th", even when the target word is not explicitly mentioned in the tweets. In addition, both figures show that stance knowledge-aware attention helps the model accurately learn stance words, such as "prolifeyouth" and "unborn". The performance of multi-feature joint attention with knowledge-aware score is particularly outstanding compared to internal attention.

In conclusion, our SC-KMAN model can effectively identify the target-related information in tweets and comprehensively analyze the multiple features of each word to obtain more accurate stance classification results.

5 Conclusion

This paper proposes a novel Stance Classification model with Knowledge-aware Multi-feature Attention Network (SC-KMAN). The main idea of SC-KMAN is to obtain various beneficial stance classification clues and comprehensively analyze these clues to detect the stance of the text. Firstly, we design a target feature extractor T-BERT for extracting richer target features. Comparative experiments reveal that T-BERT performs better in stance detection on multiple datasets compared to the target word embedding in previous work. Additionally, transfer learning is utilized to design a sentiment feature extractor S-BERT. Through ablation experiments, we conclude that T-BERT has a more significant impact on the performance of our model than S-BERT. Furthermore, we propose a knowledge-aware multi-feature attention network to capture the comprehensive weight of multiple features corresponding to words. In ablation experiments and case study, we verify that the knowledge-aware multi-feature attention network is helpful for our model to identify more effective stance classification clues.

However, our model only detects the user's stance from the social text. Many other clues can still be utilized in practical applications, including user's attribute information, social relations, social behavior, etc. In future work, we will simultaneously utilize users' attribute information, social text, social behavior, and social relations to realize user-level stance detection.

Acknowledgments This work was supported by The National Key Research and Development Program of China [Grant No. 2016QY03D0501].

Author contributions Chao Meng: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft. Binxing Fang: Conceptualization, Writing – review & editing. Hongli Zhang: Supervision, Writing – review & editing. Yuchen Yang: Software, Validation. Gongzhu Yin: Software, Validation. Kun Lu: Software, Validation.

Data availability The data is available in http://www.saifmohammad.com/WebPages/StanceDataset.htm

Declarations

Conflict of interest The authors declare that they have no known conflicts of interests.

References

1. Al-Ghadir AI, Azmi AM, Hussain A (2021) A novel approach to stance detection in social media tweets by fusing ranked lists and sentiments. Inf Fusion 67:29–40

2. Aldayel A, Magdy W (2022) Characterizing the role of bots' in polarized stance on social media. Soc Netw Anal Min 12(1):30

3. Alturayeif NS, Luqman H, Ahmed MA (2023) A systematic review of machine learning techniques for stance detection and its applications. Neural Comput Appl 35(7):5113–5144

4. Bai N, Meng F, Rui X, et al (2023) A multitask attention tree neural net for stance classification and rumor veracity detection. Appl Intell 53(9):10715–10725

5. Chen P, Ye K, Cui X (2021) Integrating n-gram features into pre-trained model: a novel ensemble model for multi-target stance detection. In: Proceedings of International conference on artificial neural networks, Springer, pp 269–279

6. Chen Z, Chen CC (2016) SCIFNET: stance community identification of topic persons using friendship network analysis. Knowl Based Syst 110:30–48

7. Conforti C, Berndt J, Pilehvar MT, et al (2022) Incorporating stock market signals for twitter stance detection. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 4074–4091

8. Devlin J, Chang M, Lee K, et al (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 4171–4186

9. Ding S, Yue Z, Yang S, et al (2020) A novel trust model based overlapping community detection algorithm for social networks. IEEE Trans Knowl Data Eng 32(11):2101–2114

10. Du J, Xu R, He Y, et al (2017) Stance classification with target-specific neural attention networks. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17, p 3988–3994

11. Dutta S, Li B, Nagin DS, et al (2022) A murder and protests, the capitol riot, and the chauvin trial: Estimating disparate news media stance. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, pp 5059–5065

12. Elfardy H, Diab MT (2016) CU-GWU perspective at semeval-2016 task 6: Ideological stance detection in informal text. In: Proceedings of the 10th International Workshop on Semantic Evaluation, pp 434–439

13. Fu Y, Li X, Li Y, et al (2022) Incorporate opinion-towards for stance detection. Knowl Based Syst 246:108657

14. Glandt K, Khanal S, Li Y, et al (2021) Stance detection in COVID-19 tweets. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing,, pp 1596–1611

15. Gómez-Suta M, Correa JDE, Mejía JAS (2023) Stance detection in tweets: A topic modeling approach supporting explainability. Expert Syst Appl 214:119046

16. Guo M, Hwa R, Lin Y, et al (2020) Inflating topic relevance with ideology: A case study of political ideology bias in social topic detection models. In: Proceedings of the 28th International Conference on Computational Linguistics, pp 4873–4885

17. Guo Y (2023) A mutual attention based multimodal fusion for fake news detection on social network. Appl Intell 53(12):15311–15320

18. Hardalov M, Arora A, Nakov P, et al (2021) Cross-domain label-adaptive stance detection. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp 9011–9028

19. Hasan KS, Ng V (2013) Extra-linguistic constraints on stance recognition in ideological debates. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp 816–821

20. Kawintiranon K, 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, pp 4725–4735

21. Li C, Peng H, Li J, et al (2022) Joint stance and rumor detection in hierarchical heterogeneous graph. IEEE Trans Neural Networks Learn Syst 33(6):2530–2542

22. Li Y, Caragea C (2021) Target-aware data augmentation for stance detection. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 1850–1860

23. Liang B, Chen Z, Gui L, et al (2022) Zero-shot stance detection via contrastive learning. In: Proceedings of WWW '22: The ACM Web Conference 2022, pp 2738–2747

24. Mohammad SM, Kiritchenko S, Sobhani P, et al (2016) Semeval-2016 task 6: Detecting stance in tweets. In: Proceedings of the 10th International Workshop on Semantic Evaluation, pp 31–41

25. Mohammad SM, Sobhani P, Kiritchenko S (2017) Stance and sentiment in tweets. ACM Trans Internet Techn 17(3):26:1–26:23

26. Ng LHX, Carley KM (2022) Is my stance the same as your stance? A cross validation study of stance detection datasets. Inf Process Manag 59(6):103070

27. Siddiqua UA, Chy AN, Aono M (2019) Tweet stance detection using an attention based neural ensemble model. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 1868–1873

28. Sirrianni J, Liu X, Adams D (2020) Agreement prediction of arguments in cyber argumentation for detecting stance polarity and intensity. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp 5746–5758

29. Sun Q, Wang Z, Zhu Q, et al (2018) Stance detection with hierarchical attention network. In: Proceedings of the 27th International Conference on Computational Linguistics, pp 2399–2409

30. Thomas M, Pang B, Lee L (2006) Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp 327–335

31. Wang Z, Sun Q, Li S, et al (2020) Neural stance detection with hierarchical linguistic representations. IEEE ACM Trans Audio Speech Lang Process 28:635–645

32. Wei P, Lin J, Mao W (2018) Multi-target stance detection via a dynamic memory-augmented network. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp 1229–1232

33. Wei P, Mao W, Zeng D (2018) A target-guided neural memory model for stance detection in twitter. In: 2018 International Joint Conference on Neural Networks, pp 1–8

34. Wei W, Zhang X, Liu X, et al (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, pp 384–388

35. Xiao Z, Song W, Xu H, et al (2020) TIMME: twitter ideology-detection via multitask multi-relational embedding. In: Proceedings of KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp 2258–2268

36. Yang D, Wu Q, Chen W, et al (2020) Stance detection with stance-wise convolution network. In: Proceedings of Natural Language Processing and Chinese Computing - 9th CCF International Conference, pp 555–567

37. Yang R, Ma J, Lin H, et al (2022) A weakly supervised propagation model for rumor verification and stance detection with multiple instance learning. In: Proceedings of SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 1761–1772

38. Yildirim G (2023) A novel hybrid multi-thread metaheuristic approach for fake news detection in social media. Appl Intell 53(9):11182–11202

39. Yuan J, Zhao Y, Lu Y, et al (2022) SSR: utilizing simplified stance reasoning process for robust stance detection. In: Proceedings of the 29th International Conference on Computational Linguistics, pp 6846–6858

40. Zarrella G, Marsh A (2016) MITRE at semeval-2016 task 6: Transfer learning for stance detection. In: Proceedings of the 10th International Workshop on Semantic Evaluation, pp 458–463

41. Zhao G, Yang P (2020) Pretrained embeddings for stance detection with hierarchical capsule network on social media. ACM Trans Inf Syst 39(1):1:1–1:32

42. Zhou S, Lin J, Tan L, et al (2019) Condensed convolution neural network by attention over self-attention for stance detection in twitter. In: Proceedings of International Joint Conference on Neural Networks, pp 1–8

43. Zhou Y, Cristea AI, Shi L (2017) Connecting targets to tweets: Semantic attention-based model for target-specific stance detection. In: Proceedings of Web Information Systems Engineering - WISE 2017 - 18th International Conference, pp 18–32

44. Zhu L, He Y, Zhou D (2020) Neural opinion dynamics model for the prediction of user-level stance dynamics. Inf Process Manag 57(2):102031

45. Zhu Q, Liang B, Sun J, et al (2022) Enhancing zero-shot stance detection via targeted background knowledge. In: Proceedings of SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 2070–2075