ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240969

Enhancing Stance Detection on Social Media via Core Views Discovery

Yu Yan[a,b], Yinghan Shen[b,c], Teli Liu[a], Xuhui Jiang[b,c] and Dechun Yin[a,*]

[a]School of Information and Network Security, People Public Security University of China
[b]Institute of Computing Technology, Chinese Academy of Sciences
[c]Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences

Abstract. Stance detection aims to identify the expressed attitude towards a target from the text, which is significant for learning public cognition from social media. The short and implicit nature of social media users' expressions potentially results in the stance understanding bias of the model. To address this problem, introducing external background information is helpful to mitigate these biases and enhance explainability. The core view, reflecting the motivations and reasons behind an individual's stance toward the target, can be summarized and extracted from collective tweets, which can serve as a reference for stance detection. In this study, we propose the Stance Detection via Core View Discovery (SD-CVM), where the core views are used for background information modeling. Specifically, we construct a joint classifier combining the semantic understanding of tweets and their relevant core views from the public. We utilize the Large Language Model (LLM) to extract core views with stances from tweets and use these core views as background references for tweets. To further optimize the tweet understanding, we develop the contrastive and rebalancing mechanism by incorporating stance supervision signals for training. Experiments on two representative datasets demonstrate the excellent performance of our method.

1 Introduction

Stance detection aims to identify the stances of individuals (Favor, Against, or None) expressed in the texts towards a designated target [37]. In the digital age, stance detection on social media platforms enables wide-ranging studies of collective cognition for different applications, including trends in public opinion [13], reactions to political events [20], and business analysis [8].

Stance detection requires understanding various aspects and evaluations in texts toward a specific target[16], which poses a challenge beyond simple text classification. Previous studies mainly focused on enhancing text understanding to improve stance detection performance. Early methods[28, 29] employed feature engineering to manually extract linguistic and structural features, but suffer from limited generalizability. To address this limitation, recent advancements proposed using deep neural networks[4], facilitating more profound text understanding and learning features between the text and the target. Furthermore, introducing pre-trained language models (PLMs) [6, 22] as text encoders enabled stance detection models to leverage more detailed and diverse information. However, despite these advancements, model performance became difficult to further improve by solely enhancing text understanding, due to the absence of background information. Current studies aim to further enhance model generalization through external knowledge injection. The advantage of using external knowledge lies in its ability to provide contextual backgrounds that enrich the understanding of the topic and the nuances of stance within the text. WS-BERT[10] introduces Wikipedia documents as the knowledge source to contextualize social media posts towards the target, further augmenting the information of texts. Moreover, leveraging their world knowledge and few-shot reasoning capabilities, large language models (LLMs) [16, 9, 40] significantly improved the accuracy of stance detection.

Stance expressions are primarily driven by individual viewpoints and motivations. While existing methods emphasize the introduction of external factual knowledge about the target, they rely heavily on the model's surface matching between knowledge and stance and neglect the modeling of emotional and subjective information in the knowledge preparation stage. To address this gap, in the knowledge preparation stage, the Core View (CV) could be introduced as the contextualized information, which is a summary of the motivations and reasons behind an individual's stance toward the target. These core views provide a deep insight for model's ability to explain and predict stances accurately.

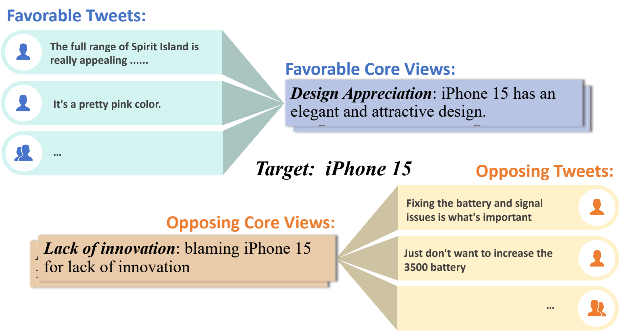


Figure 1. An example of summarizing core views from tweets related to the target *iPhone 15*. Different stances towards the iPhone 15 could be attributed to different reasons and motivations, which we refer to as the Core Views (CV).

Figure 1 shows the case of core views from social media tweets on the target '*iPhone 15*'. By analyzing tweets that show different

* Corresponding Authors. Email: yindechun@ppsuc.edu.cn. The research was financially supported by the Fundamental Research Funds for the Central Universities (Grant No. 2022JKF02039).

stances towards the product, we can conclude and summarize many different and clear core views. For instance, favorable sentiments are mainly attributed to the design and color appeal of the *iPhone 15*, while opposing sentiments were mainly caused by the perceived lack of innovation. These summarized views provide a condensed representation of the reasons behind a user's stance, which is used to augment the background modeling information for the stance detection model. Manually extracting core views from large datasets can be extremely time-consuming. To address this, we leverage the capabilities of Large Language Models (LLMs), which acquire rich semantic knowledge and contextual reasoning skills. By LLMs, we can reliably extract core views that capture the central reasons and motivations behind users' stances, providing valuable background information to enhance stance detection models.

To this end, we propose the Stance Detection by Core View Modeling (SD-CVM) approach, a novel stance detection framework that incorporates core views expressed in social media towards the given target. Specifically, we employ the LLM to extract the core views from a collection of tweets in the training data. Then, we construct a joint classifier that integrates the text's semantic understanding with core views as the background knowledge, where a classifying network is used to learn the relevance between text and target from semantic understanding and the kNN retrieves the core views for the text's background information. For further refining the model, we introduce Stance-Aware Contrastive Learning (SACL) and Rebalanced Focal Loss (RFL) to jointly optimize the loss calculation. We validate our SD-CVM approach by experimenting on the benchmark P-stance dataset, which demonstrates the state-of-the-art performance of our method.

The major contributions of our study are as follows:

- We propose Stance Detection via the Core View Modeling (SD-CVM) approach, where we automatically summarize the main motivations and reasons for users' different stances by using LLM, which we refer to as core views. Via core views discovering, we hope to achieve tweet background information augmentation.
- To jointly consider the text understanding and the background information, we adopt the joint prediction mechanism and further optimize the model's sample representations by introducing the hybrid loss strategy.
- Our experiments demonstrate a state-of-the-art performance over the benchmark dataset, underscoring the effectiveness of our core view modeling for stance detection.

2 Related Work

Stance Detection. Stance detection (SD) involves identifying a text author's stance towards a target, categorizing it as Favor, Against, or None [23]. This field is closely related to sentiment analysis, sarcasm detection, irony detection, and argument mining. Initially, SD relied on feature engineering with methods like the Bag of Words model and TF-IDF for semantic representation. Based on the statistic of the word, algorithms such as Support Vector Machines (SVMs) [15, 25], Random Forest (RF) [1], and Logistic Regression (LR) [38] are used for feature extraction and stance detection. Then, the advent of deep learning significantly transformed stance detection. Models such as CNN-based models [32], RNN-based models [7, 30, 26, 3], transformer-based models [6, 22, 24] have become prevalent, for their effectiveness in supervised association of semantic information with stance labels in both in-domain and cross-domain datasets.

Current research in SD increasingly focuses on the background information of tweets, highlighting the importance of background knowledge beyond the tweets for interpreting and accuracy stance inference. Obtaining knowledge from Wikipedia [10, 43] is a useful approach for background information modeling. Concept graph [11, 31] and Knowledge graph [19] are two typical tools for background information organization and injection for stance detection.

For stance detection, LLMs have been proven particularly effective. COLA [17] enhances stance detection by providing detailed context that helps to distinguish between stances on targets. DEEM [35] generates the specific experts for joint prediction. One of the key factors that LLM-based models perform well in stance detection is due to their extensive general knowledge. Therefore, how to specifically acquire and use knowledge of LLMs can be considered as a crucial approach to enhancing stance detection model performance.

LLM-Based Information Extraction. Information Extraction (IE) is a task that automatically extracts structured and meaningful information from unstructured textual data. Benefiting from the Large Language Model (LLM), the IE approach becomes flexible and open-ended [33, 36]. For instance, TutorQA [39] uses LLM for Concept Graph Recovery. NERRE [5] explores how to use LLM for structural information extraction.

For existing knowledge augmentation approaches in stance detection, due to the concept drift [2], directly incorporating external information can lead to biases in understanding. Therefore, using the information extracted from original data for information augmentation and obtaining the contextualized explanation from LLM is a promising approach.

3 Preliminaries

Definition 1. ***(Stance Detection)*** *Given the labeled dataset* $X = \{(x_i, t_i, y_i)\}_{i=1}^{n}$, *where* x_i *is the text with a sequence of words* $\{w_i^1, w_i^2, \ldots\}$, y_i *is the stance label and* t_i *is the corresponding target. Formally, stance detection is the task of classifying a given input text* x_i *towards a specified target* t_i *into the stance label* y_i *(Favor, Against, None). In our work, we aim to explicitly extract the core view set* $C = \{c^1, c^2, \ldots, c^m\}$ *from X, and discover tweet* x_i*'s corresponding core views* $C_i = \{c_i^1, c_i^2, \ldots\}$ *towards the target* t_i*. These core views are used for background information augmentation to enhance the model's ability to analyze tweet stances.*

Definition 2. ***(Core View)*** *In the stance detection task, we define that Core Views (CVs)* $C = \{c^1, c^2, \ldots, c^m\}$ *represent the various reasons and motivations behind the stances toward the target* t*, which consist of two main attributes,* ***Representativeness*** *and* ***Subjectivity****. Below is a detailed exploration of these attributes:*

- ***Representativeness****: A core view* c *achieves representativeness if its inclusion notably impacts the accurate classification of the stance* y_i *towards* t_i *for certain quantity samples* x'*s in the dataset. Specifically, For the core view* c*, its representativeness criterion is quantified by the following condition:* $\frac{|\{(x_i,t_i)\in X: P(y_i|c,t_i,x_i)-P(y_i|t_i,x_i)\geq\theta\}|}{|X|} \geq \zeta$, θ *and* ζ *are thresholds.*
- ***Subjectivity****: A core view* c *expresses the opinion, sentiment, or bias toward the target* t *i.e., all the core views could be classified into stance label* $y \in \{Favor, Against\}$.

4 Methodology

As illustrated in Figure 2, we propose the Stance Detection by Core View Modeling (SD-CVM) approach. Specifically, we employ the large language model to summarize the core views from the raw data.

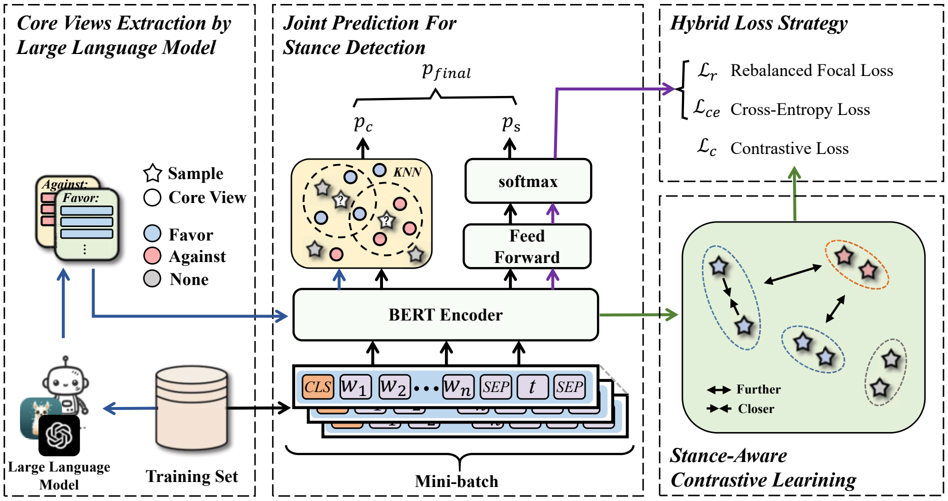


Figure 2. The architecture of the proposed SD-CVM approach: The Core View Extraction part(left), The Joint Prediction part(middle), The Stance-Aware Contrastive Learning and the Hybrid Loss Strategy part(right). SD-CVM inter the tweet's stance by combining the semantic understanding of the text and the relevant core views from the public.

Then, we construct the joint classifier with the neural network and k-nearest neighbor (kNN) method. This classifier is further optimized by Stance-Aware Contrastive Learning (SACL) and Rebalanced Focal Loss (RFL). Core views are used for voting on the potential stance of tweets according to the semantic similarity in kNN. Finally, SD-CVM jointly outputs the prediction of the neural network and kNN for stance detection.

4.1 Core View Extraction

Capitalizing on the advanced few-shot learning abilities of Large Language Models (LLMs), e.g., GPT-4, we conduct the context information modeling by employing LLM to extract the core views from the labeled tweet dataset.

Formally, given a labeled dataset $X = \{(x_i, t_i, y_i)\}_{i=1}^{n}$, we focus on subsets $X_{(t,y)} \subset X$ where all instances share the same target t and stance label y. Our objective is to discover the core views $C_{(t,y)} = \{(c_i, t, y)\}_{i=1}^{m}$ from the collected subset $X_{(t,y)}$. $C_{(t,y)}$ captures the shared views about the specific target t and stance y. The core view extraction consists of two stages: extracting and filtering stage.

In the extracting stage, denoted as f, which applies the LLM $\mathcal{G}$ with the prompt template P_{E} for the core view extraction. The prompt template P_{E} takes the batch instances from $X_{(t,y)}$ as input and guide $\mathcal{G}$ to generate the set of core views $C'_{(t,y)}$:

$$C'_{(t,y)} = f\left(\mathcal{G}, P_{\mathrm{E}}\left(X_{(t,y)}\right)\right). \tag{1}$$

To ensure the efficiency of the LLM's inference with core views, we adopt a multi-round sampling strategy from the training samples for core view extraction.

In the filtering stage, denoted as g, we conduct core view reranking and filtering to ensure $m \ll n$ after the extraction process:

$$C_{(t,y)} = g\left(\mathcal{G}, P_{\mathrm{F}}(C'_{(t,y)})\right)_{[:m]}, \tag{2}$$

where P_{F} is prompt template in the filtering stage. By extracting the core views $C_{(t,y)} = \{(c_i, t, y)\}_{i=1}^{m}$, which are annotated with the specific target t and stance label y, we could conduct the background augmentation with those core views.

For P_{E}, it is designed to batch input the samples for core view extraction as follows:

> **User:** Suppose you're a public opinion analyzer. The following tweets are from **[supporters/opponents]** of the topic [target], please summarize their core views. For each core view, give a brief explanation and supporting reasons. Make sure that the core views you provide are **representative** of the **[supporters/opponents]** toward the topic [target]:
> Tweet 1
> Tweet 2
> ...

For P_{F}, it is designed to as follows to rerank and filter the core views:

> **User:** Suppose you are a content auditor, and the following are the core views about the **[supporters/opponents]** under the topic [target] may exist duplicated or overly repetitive explanations, please carefully review and correct, and return your reviewed results according to the **representative**:
> Core view 1
> Core view 2
> ...

4.2 Encoder Module

We use the BERT $\mathcal{M}$ as our text encoder for semantic representation of sample and knowledge. Specifically, given a text $x = \{w_1, w_2, \ldots\}$, a target t, and the core views $c = \{r_1, r_2, \ldots\}$ where w and r are words in the text. We construct the input patterns "[CLS]x[SEP]t[SEP]" and "[CLS]c[SEP]". These patterns are fed into the text encoder module to obtain the d-dimensional hidden vectors(h_x, h_c) for each input instance:

$$h_x = \mathcal{M}([\mathrm{CLS}]x[\mathrm{SEP}]t[\mathrm{SEP}])_{[\mathrm{CLS}]}, \tag{3}$$

$$h_c = \mathcal{M}([\text{CLS}]c[\text{SEP}])_{[\text{CLS}]}, \tag{4}$$

where h_x and h_c the hidden vectors of [CLS] derived from the final layer of $\mathcal{M}$ for the input instance's semantic representation.

4.3 Core View Retrieval for Stance Detection

Stance Detection via Semantic Understanding. To infer stances from textual understanding, we develop a classifier with a fully connected layer and a BertPooler layer based on the text encoder, and use softmax as the activate function:

$$p_s = \text{softmax}(W \cdot \text{BertPooler}(h) + b), \tag{5}$$

where h is the hidden representation of the text instance, W and b are the weights and bias of the fully connected layer, respectively, and p_s is the predicted stance distribution via the semantic understanding.

Enhanced Stance Detection with Core View Information. We extract core views $C_t = \{(c_i, t, y_i)\}_{i=1}^{M_t}$ for each target t from our training data to enhance the inference capabilities of our model. For context information modeling for the specific text, we employ a k-nearest neighbor (kNN) mechanism for core view retrieval.

Specifically, during the inference phase, for a given test sample (x,t) with target t, the encoding model $\mathcal{M}$ is utilized to derive the vector representation h_x. Subsequently, the set of k-nearest core view neighbors $\mathcal{N}_x = \{(h_{c_i}, y_{c_i}) \mid y_{c_i} \neq \text{None}, i = 1, 2, \ldots, k\}$ is retrieved from C_t based on the cosine similarity distance. Based on core views, the prediction distribution p_c is computed as follows:

$$p_c = \sum_{i=1}^{k} \beta_i \cdot p_i^k, \quad \beta_i = \frac{\exp(\text{sim}(h_x, h_{c_i})/\tau_k)}{\sum_{j=1}^{k} \exp(\text{sim}(h_x, h_{c_j})/\tau_k)}, \tag{6}$$

where τ_k is the temperature hyperparameter for kNN, and β_i is the weighting factor for the i-th neighbor in the kNN. The similarity function $sim(u,v)$ is defined as the cosine similarity between vectors u and v: $\text{sim}(u,v) = \frac{u^\intercal v}{\|u\|\|v\|}$. Due to the tweets labeled with 'None' having no clear core views, we will not summarize these tweets. Alternatively, we introduce neutral samples directly into the kNN to represent the neutral data.

The final joint prediction integrates the information from both the semantic understanding and core view reference:

$$p_{\text{final}} = \lambda p_s + (1 - \lambda) p_c, \tag{7}$$

where the hyperparameter λ determines the relative contribution of two component.

4.4 Hybrid Loss Strategy

Stance-Aware Contrastive Learning. To guide the model in learning representations that are contextually rich and stance-informative, the Stance-Aware Contrastive Learning module (SACL) is designed to minimize the distance between embeddings of sentences with the same stance and maximize the distance between those with opposing stances. SACL leverages the stance labels of sentences as supervisory signals within mini-batches consisting of h, whose contrastive learning objective is defined as follows:

$$\ell_c(h_{x_i}) = \log \frac{\sum_{(x_i, x_i^+) \in \mathcal{P}_i} \exp(\text{sim}(h_{x_i}, h_{x_i}^+)/\tau_s)}{\sum_{x_j \in \mathcal{B} \setminus x_i} \exp(\text{sim}(h_{x_i}, h_{x_j})/\tau_s)}, \tag{8}$$

$$\mathcal{L}_C = -\frac{1}{N_{\mathcal{B}}} \sum_{h_i \in \mathcal{B}} \ell_c(h_{x_i}), \tag{9}$$

where τ_s is a temperature hyperparameter that scales the distribution of distances. The set $\mathcal{B}$ refers to the mini-batch training data, and $\mathcal{B} \setminus x_i$ denotes the set of all other samples in the mini-batch training data excluding sample i. The set $\mathcal{P}_i$ is the set of positive pairs for i-th sample, which consists of the sample with the same label in the mini-batch.

Rebalanced Focal Loss. To adaptively adjust our model's sensitivity to minority stances, we use a joint loss mechanism to integrate the Rebalanced Focal Loss (RFL) [12] and Cross-Entropy Loss. RFL loss prioritizes minority classes by adjusting loss weights, enhancing learning from minority stance examples. CE loss is employed to handle the learning of common stances. The RFL loss and CE loss are as follows:

$$\mathcal{L}_{CE} = -\frac{1}{N_{\mathcal{B}}} \sum_{i=1}^{N_{\mathcal{B}}} \sum_{c=1}^{C} y_i^c \log(p_i^c), \tag{10}$$

$$\mathcal{L}_{RFL} = -\frac{1}{N_{\mathcal{B}}} \sum_{i=1}^{N_{\mathcal{B}}} \sum_{c=1}^{C} r \cdot y_i^c (1 - p_i^c)^{\alpha} \log(p_i^c), \tag{11}$$

where $N_{\mathcal{B}}$ is the batch size, C is the number of classes, y_i^c is a binary indicator of whether class label c is the correct classification for i-th instance, and y_i^c is the predicted probability that i-th instance is of class c. In the RFL, $r = \frac{1-\beta_r}{1-\beta_r^{n_i}}$ is the rebalanced weight, where n_i is the total number of samples in the i-th class, $\beta_r \in [0,1)$ is used to adjust the influence of weight. $\alpha \in \mathbb{R}$ is the focusing parameter.

Combining Loss. We combine the stance-aware contrastive loss and classification loss as the final loss function $\mathcal{L}_{\text{final}}$:

$$\mathcal{L}_{\text{final}} = \gamma_r \mathcal{L}_{RFL} + \gamma_{ce} \mathcal{L}_{CE} + \gamma_c \mathcal{L}_C, \tag{12}$$

where γ_r, γ_{ce} and γ_c are the loss weight hyperparameters.

5 Experiments

In the experiment sections, we aim to answer the following key research questions:

- RQ-1: How effective is our SD-CVM for stance detection?
- RQ-2: How effective is the semantic modeling and hybrid loss strategy added to SD-CVM?
- RQ-3: Without the explicit training for core view matching, can the Core Views be understood and utilized by the model?
- RQ-4: How can the Core View represent the stance of users?
- RQ-5: How can the Core View explain the stance of users?

5.1 Experiment Setting

Datasets. To validate the effectiveness of our model, we conducted experiments on two stance detection datasets: **SemEval-2016** and **P-Stance**. The statistic details of datasets are presented in Table 1:

Table 1. The statistics for the SemEval-2016 and P-stance datasets.

Dataset	Target	Favor	Against	None
SemEval-16	AT	163	595	217
	CC	426	32	312
	FM	358	652	234
	HC	213	733	330
	LA	219	697	329
P-stance	Trump	3668	4286	-
	Biden	3219	4077	-
	Sanders	3558	2768	-

Table 2. Experimental results (%) of various models on SemEval-2016 datasets. The results with ‡ is from [41], and ♣ is from [16]. The best result is highlighted in **bold**. The second best result is highlighted in underline.

Method	Target					Average
	AT	CC	FM	HC	LA	
Fine-tuned-based Model						
BERT [6]	71.40	45.11	58.90	66.96	55.14	59.50
BERTweet [24]	70.81	53.11	65.05	68.67	65.24	64.58
RelNet [42]	70.55	57.20	61.25	62.33	63.65	63.00
S-MDMT [34]	69.50	52.49	63.78	67.20	67.19	64.03
TAPD [14]	73.87	**59.32**	63.93	70.01	67.23	<u>66.87</u>
KNN-TACL [31]	<u>74.33</u>	45.38	65.40	71.02	67.41	64.71
SD-CVM (ours)	**76.09**	<u>57.61</u>	<u>66.46</u>	<u>74.46</u>	**70.29**	**68.98**
GPT-3.5-turbo-based Model						
DQA♣	8.10	24.70	59.10	74.00	52.00	43.58
StSQA‡	-	-	**68.70**	**78.90**	61.50	-

- **SemEval-2016 dataset** [23] is a pioneering resource in English stance detection, comprised of 4,163 tweets across 5 targets: *Atheism* (AT), *Climate Change is a Real Concern* (CC), *Feminist Movement* (FM), *Hillary Clinton* (HC), and *Legalization of Abortion* (LA). Each tweet is associated with a target and has a manually annotated stance label: Favor, Against, or None. For comparison, we adopted the same data dividing method as [14, 31].
- **P-Stance dataset** [20] is a broadly annotated, large-scale collection consisting of 21,574 English tweets from the political domain. It is the dataset of tweets that captures public attitudes towards three politicians: *Donald Trump* (Trump), *Joe Biden* (Biden), and *Bernie Sanders* (Bernie). Each tweet is meticulously annotated to indicate whether it supports, opposes, or remains neutral towards one of these three target individuals. Similar to the existing studies [18], we eliminate samples labeled as "None" in our experiment.

Experimental Metrics and Setup. For evaluation metrics, we adopt F_{AVG}, which is employed in stance detection task[37, 23, 27] to evaluate our model performance. F_{AVG} is computed by averaging the average value of F1-scores for 'Favor' and 'Against' classes.

For experimental setups, we choose the Bertweet [24] as text encoder $\mathcal{M}$. The learning rate is set at 2e-5 with a warmup proportion of 0.1. The early stopping mechanism is adopted in the training process to prevent overfitting. The batch size for training is set to 16, and sequences are truncated or padded to the length of 128 tokens. For the hyperparameters, τ_s is set to 0.05, and τ_k is set to 0.1. λ,α, and β_r are set to 0.5, 0.1, and 0.2 [31, 12], respectively. For the hyperparameters in the hybrid loss strategy γ_r, γ_{ce} and γ_c are set to 1, 1, and 0.1 [21], respectively. For the cluster number of kNN, we choose the top 10 or 20 nearest core views as the reference, i.e., k is 10 or 20. To ensure performance stability, we utilize the labeled tweets and core views for retrieval. We use the GPT-3.5 for core view extraction. The core views number m under each target are 50 for P-stance and 30 for Semeval-2016. For CC's 'Against' stance, given the insufficient number of samples (30 in total), we extract 10 for CC. We use 50 samples per round to generate core views via GPT-3.5 for SemEval-2016, and 200 samples for P-stance.

Comparative Methods. We provide an overview of the baseline methods for comparison: 1) **BERT** [6], a widely-recognized pre-trained language model, is adapted for stance detection by incorporating a dropout layer followed by a linear classifier to the original architecture. 2) **BERTweet** [24], a BERT variant fine-tuned on English Twitter data, demonstrates an enhanced understanding of Twitter-specific language nuances. Our configuration of BERTweet mirrors the setup of the original BERT model. 3) **TAPD** [14] utilizes multi-prompt information extraction and representing labels as vectors rather than fixed words. 4) **KNN-TACL** [31] introduces the pretext task in contrastive learning to compute dynamic topic weights and use the kNN approach for stance classification 5) **WS-BERT** [10] integrates Wikipedia knowledge into document and target representations, and introduces two variants to accommodate varying textual styles within documents. 6) **RelNet** [42] augments the BERT framework with external knowledge integration, boosting its ability to associate opinion terms within a text to their respective targets effectively. 7) **S-MDMT** [34] employs built on BERT and target adversarial learning and a shared-private scheme to distinguish between common stance features and those unique to particular targets.

We also investigate the performance of the LLM-based stance detection, whose main difference is their constructed prompt: 1) **DQA** [16], directly conduct the question-answering, is implemented in strict accordance with [40] for target-specific zero-shot learning in the GPT-3.5. 2) **StSQA** [41] is a step-by-step question-answering method, which uses GPT-3.5 as the backbone.

5.2 Main results

For answering **RQ-1**, we conduct the main experiment. The main experiment results are shown in Table 2 and 3, which demonstrate the effectiveness of SD-CVM. We further list the results of GPT-3.5-turbo-based approaches (DQA, StSQA) as reference. The comparison of existing methods on two benchmarks shows the effectiveness of our SD-CVM.

Table 3. The Comparison of experimental results (%) on P-stance. For the methods using the LLM, results with † from [40], ‡ from [41]. The best result is highlighted in **bold**. The second best result is highlighted in underline.

Method	Target			Average
	Biden	Trump	Bernie	
Fine-tuned-based Model				
BERT [6]	78.00	77.19	69.77	74.99
BERTweet [24]	82.48	81.02	78.09	80.53
WS-BERT [10]	<u>83.50</u>	<u>85.80</u>	79.00	82.77
SD-CVM (ours)	**84.31**	**85.86**	<u>80.75</u>	**83.64**
GPT-3.5-turbo-based Model				
DQA†	82.30	82.80	79.40	81.50
StSQA‡	82.80	85.70	**80.80**	<u>83.10</u>

On the SemEval-2016 dataset, SD-CVM achieves an average F1 score of 68.98%, outperforming the previous best TAPD (66.87%). Our method improves by 2.22% on AT and 4.45% on HC, likely benefiting from better capturing nuanced semantics and background knowledge. However, for the target CC, our method performs slightly poorly with a 57.61% F1 score due to the limited 'Against' samples (only 11), making it difficult to effectively learn opposing views.

On the P-stance dataset, SD-CVM obtains an average F1 score of 83.64%, surpassing WS-BERT (82.77%). On the target Trump, our method yields an 85.86% F1 score, outperforming WS-BERT by 2.06%, likely due to leveraging large language models for core view extraction. However, on Bernie, our method performs relatively poorly with an 80.75% F1 score, lower than WS-BERT (79%), potentially due to the complex topics causing our extracted core views to struggle with comprehensive coverage.

5.3 Ablation Study

For answering **RQ-2**, we conduct the ablation study, which concludes that the semantic modeling and hybrid loss strategy added to SD-CVM do have significant benefits for its overall performance.

To examine the impacts of various modules, we conduct the ablation study on the P-stance dataset: 1) **w/o** *SACL*: removing the Stance-Aware Contrastive Learning, i.e., $\mathcal{L}_{\text{final}} = \gamma_r \mathcal{L}_{RFL} + \gamma_{ce}\mathcal{L}_{CE}$. 2) **w/o** *RFL*: removing the Rebalanced Focal Loss in the loss function, i.e., $\mathcal{L}_{\text{final}} = \gamma_{ce}\mathcal{L}_{CE} + \gamma_c \mathcal{L}_C$. 3) **w/o** p_c: removing the core view augmentation, the final output is only from the text-specific understanding, i.e., $p_{\text{final}} = p_s$. 4) **w/o** p_s: only using the core views as the inferences for stance detection, i.e., $p_{\text{final}} = p_c$. 5) **w/o** *SACL,RFL*, p_c: only using the BERT-based classifier for stance detection.

Table 4. The results (%) of ablation study on P-stance.

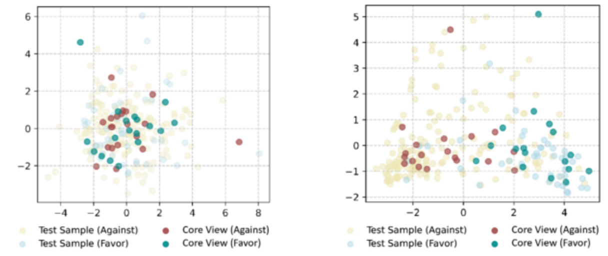
Method	Target			Average
	Biden	Trump	Bernie	
SD-CVM	84.31	85.86	80.75	83.64
w/o p_c	-1.17	-2.72	-1.80	-1.90
w/o p_s	-3.16	-3.76	-3.40	-3.44
w/o *SACL*	-0.87	-3.25	-3.75	-2.62
w/o *RFL*	-2.92	-3.09	-0.93	-2.31
w/o *SACL,RFL*, p_c	-3.37	-4.39	-4.85	-4.20

From the ablation results in Table 4, we can observe that: 1) As the results of **w/o** p_c and **w/o** p_s indicate, core views are helpful for information augmentation, but learning text-specific knowledge is also necessary. Removing text-specific knowledge will lead to a generalization degradation, while in cases of the target *Biden*, SD-CVM still makes relatively accurate inferences based on core views as guidance. 2) The hybrid loss strategy is useful for the model's stability performance in different datasets. Experiment **w/o** *SACL* and **w/o** *RFL* show different pronounced drops in three targets, indicating that stance-aware contrastive learning and rebalanced focal loss have their advantages for different datasets. Jointly considering them can be helpful for model learning in different data distributions.

5.4 Core View Availability Analysis

In the knowledge preparation stage, we generate core views for information augmentation. A key question is whether the stance information in the core views can be learned and utilized by the model. For answering **RQ-3**, we conduct a semantic distribution visualization where core views are analyzed in comparison with test samples before and after training. We conclude that the model can use the stance information of the core view for accurate stance detection.

We use PCA to reduce sentence embeddings to 2-D as Figure 3, which shows the obvious semantic distribution alignment of core views with tweets for the HC and FM datasets after training, in contrast to their overlapping before training. Test samples are consistent with this distribution. These observations suggest that the model is indeed leveraging the latent stance information in the core views during the training process. Without explicit supervision to match the surface forms of core views and text, the model has learned to align the semantic representations of text samples with the stance-oriented core views, enabling accurate stance detection. The visual



(a) HC (before training) (b) HC (after training)

(c) FM (before training) (d) FM (after training)

Figure 3. The comparison of text representation distributions influenced by the core views' latent stance information. Subfigures (a) and (b) present the experimental results on the target FM, while Subfigures (c) and (d) correspond to the target HC.

result highlights that the core views are representative enough for those samples with the same stance, as long as the model can identify the difference between 'Favor' and 'Against'.

5.5 Core View Effectiveness Analysis

For answering **RQ-4**, we conduct the parameter analysis, which concludes that using core views as external information effectively improves the model's effectiveness in stance detection.

Representativeness Analysis. Our core view modeling enhances stance detection accuracy and provides result explanations. To validate the effectiveness of incorporating core view data, we conduct experiments with different retrieval sample numbers for kNN, as shown in Figure 4. To further show the role of the core view, we only use the p_c, i.e., $p_{\text{final}} = p_c$. We under various setups, varying the number of core views (C) and labeled tweets (T) in kNN retrieval: 1) 100% T+C, 2) 50% T, 3) 10% T, and 4) Only C. More analysis is as follows:

1) For HC (Figure 4(a)), combining core view data with training data (100%T+C) substantially improves F1 Favor and F1 Against scores compared to using training data alone (100%T), emphasizing the significance of core views in enhancing stance detection. 2) For FM (Figure 4(b)), the model's performance using only core view data (C) is remarkably high, particularly in detecting Against stances, suggesting that core views effectively capture nuanced expressions of opposing views. 3) For LA (Figure 4(c)), including core view data (100%T+C) enhances the F1 Favor score compared to using training data alone (100%T). However, when training data is reduced (50%T and 10%C), a slight performance decrease is observed, indicating a synergistic relationship between core views and training data.

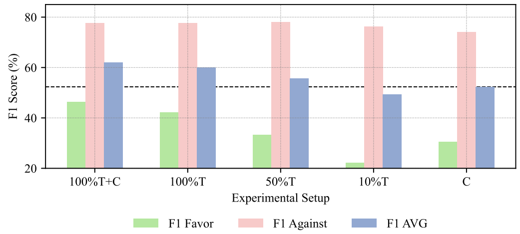
These experimental results demonstrate the effectiveness of core views in improving stance detection performance.

Performance Contribution Analysis. To validate the contribution of core view in the external knowledge augmentation, we adjust the k values in the kNN classifier to analysis, as shown in Figure 5. The used model is the original SD-CVM. Detailed analysis is as follows:

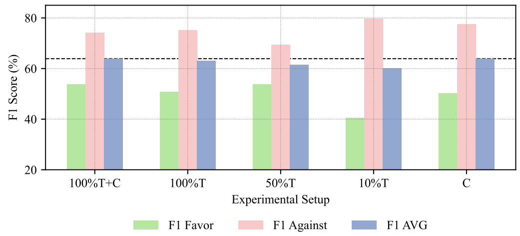
1) When k is 0, SD-CVM outputs the result only based on the tweet

Table 5. The case study on the target *Donald Trump* with different stances. We list two representative tweets(denoted as '**C**') and corresponding core views(denoted as 'C'). In this case, we can observe that the extracted core views are supportive of this positional categorization result of the tweet.

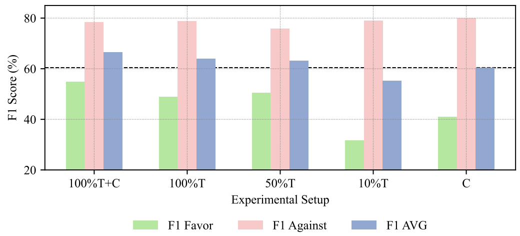
Target	Tweet Instance & Related Core Views	Stance
Donald Trump	**T1: Remind me again about how Russian bots are not real. The man does a better job #midterms #Trump** C1: Election Interference: The tweet refers to concerns about Russian bots influencing U.S. elections. C2: Conspiracy Theories: Tweets mention claims like Iran downing a Ukrainian flight to distract from Trump's impeachment. C3: Social Media Misinformation: It underscores the challenge of identifying the truth due to prevalent online misinformation.	**Favor**
	T2: 'Trump Foundation' has been forced to cease and desist, Trump perhaps has found another way to build the slush fund. C1: Accusations of Corruption and Incompetence: Trump is repeatedly labeled as corrupt. C2: Lack of Transparency: Critics accuse Trump of concealing critical information from the public C3: Personal attacks on character and integrity: Trump is described as cowardly, dishonest, and morally bankrupt.	**Against**



(a) HC



(b) FM



(c) LA

Figure 4. Comparative Analysis of F1 Scores across Different Experimental Setups. Subfigure (a) shows the comparison for HC, (b) for FM, and (c) for LA. The bar charts illustrate the variance in F1 Favor, F1 Against, and F1 AVG scores, highlighting the performance of each experimental setup.

semantic information, and there is no external knowledge for reference. The model does not perform the best. 2) The best performance is observed when k is between 10 and 20 across three target datasets. This range achieves a balance between core views and neighboring samples to avoid the overfitting of tweet semantic information and external noise. 3) When the k exceeds 20, the model's performance starts to decline across three target datasets. This trend indicates that a large k dilutes the contribution of core views to the predictive results, thereby introducing noise or irrelevant information and reducing the model's ability to discern accurately.

5.6 *Case Study*

For answering **RQ-5**, we conduct the case study, which concludes that the core views are explanatory background knowledge, which can provide clarification for the stance held by users.

The SD-CVM method provides explainability for stance detection

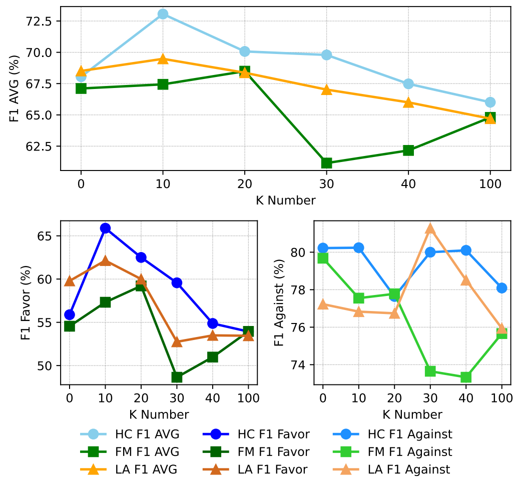


Figure 5. Comparison of F1 scores for HC, FM, and LA datasets under different numbers of nearest neighbors settings in KNN classifier.

by summarizing the core views from massive samples and using them as background information modeling for the text.To demonstrate its explainable results, we conduct a brief case study on the target *Donald Trump* with different stances ('Favor' and 'Against') in Table 5.

In this case study, for tweet T1 labeled as 'Favor', core views like 'Election Interference', 'Conspiracy Theories' and 'Social Media Misinformation' are highlighted, which capture the essence of the supportive stance. Conversely, for tweet T2 labeled as 'Against', core views with negative sentiment such as 'Accusations of Corruption and Incompetence', 'Lack of Transparency' and 'Personal Attacks on Character and Integrity' are identified. Our case study exemplifies the reasonable of introducing the core views from the public as background knowledge.

6 Conclusion

In this study, we propose the Stance Detection by Core View Modeling (SD-CVM) approach, which utilizes the public core views as background knowledge augmentation. To obtain the public's core views towards the given target, we summarize the information from the training samples by using the Large Language Model (LLM). For joint combing the semantic information and background knowledge of the text, we employ a neural network and k-nearest neighbor (kNN) to jointly predict. To further optimize the model, we adopt the hybrid loss strategy, introducing Stance-Aware Contrastive Learning Loss (SCAL) and Rebalanced Focal Loss (RFL) to further optimize the sample representation. The experiments in the benchmark dataset P-stance demonstrate the effectiveness of our SD-CVM.

In future work, we intend to explore from the perspective of explainability in multi-modal information for knowledge augmentation in the stance detection task.

References

[1] A. Aker, L. Derczynski, and K. Bontcheva. Simple open stance classification for rumour analysis. *arXiv preprint arXiv:1708.05286*, 2017.

[2] A. Bechini, A. Bondielli, P. Ducange, F. Marcelloni, and A. Renda. Addressing event-driven concept drift in twitter stream: a stance detection application. *IEEE Access*, 9:77758–77770, 2021.

[3] L. Borges, B. Martins, and P. Calado. Combining similarity features and deep representation learning for stance detection in the context of checking fake news. *Journal of Data and Information Quality (JDIQ)*, 11(3):1–26, 2019.

[4] W.-F. Chen and L.-W. Ku. Utcnn: a deep learning model of stance classification on social media text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1635–1645, 2016.

[5] J. Dagdelen, A. Dunn, S. Lee, N. Walker, A. S. Rosen, G. Ceder, K. A. Persson, and A. Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418, 2024.

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

[7] J. Du, R. Xu, Y. He, and L. Gui. Stance classification with target-specific neural attention networks. International Joint Conferences on Artificial Intelligence, 2017.

[8] R. Feldman. Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89, 2013.

[9] J. Gatto, O. Sharif, and S. M. Preum. Chain-of-thought embeddings for stance detection on social media. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[10] Z. He, N. Mokhberian, and K. Lerman. Infusing knowledge from wikipedia to enhance stance detection. *arXiv preprint arXiv:2204.03839*, 2022.

[11] H. Huang, B. Zhang, Y. Li, B. Zhang, Y. Sun, C. Luo, and C. Peng. Knowledge-enhanced prompt-tuning for stance detection. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–20, 2023.

[12] Y. Huang, B. Giledereli, A. Köksal, A. Özgür, and E. Ozkirimli. Balancing methods for multi-label text classification with long-tailed class distribution. *arXiv preprint arXiv:2109.04712*, 2021.

[13] M. Jang and J. Allan. Explaining controversy on social media via stance summarization. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1221–1224, 2018.

[14] Y. Jiang, J. Gao, H. Shen, and X. Cheng. Few-shot stance detection via target-aware prompt distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 837–847, 2022.

[15] M. Lai, V. Patti, G. Ruffo, and P. Rosso. # brexit: Leave or remain? the role of user's community and diachronic evolution on stance detection. *Journal of Intelligent & Fuzzy Systems*, 39(2):2341–2352, 2020.

[16] X. Lan, C. Gao, D. Jin, and Y. Li. Stance detection with collaborative role-infused llm-based agents. *arXiv preprint arXiv:2310.10467*, 2023.

[17] X. Lan, C. Gao, D. Jin, and Y. Li. Stance detection with collaborative role-infused llm-based agents, 2024.

[18] A. Li, B. Liang, J. Zhao, B. Zhang, M. Yang, and R. Xu. Stance detection on social media with background knowledge. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15703–15717, 2023.

[19] J. Li, X. Li, L. Hu, Y. Zhang, and J. Wang. Knowledge graph enhanced language models for sentiment analysis. In *International Semantic Web Conference*, pages 447–464. Springer, 2023.

[20] Y. Li, T. Sosea, A. Sawant, A. J. Nair, D. Inkpen, and C. Caragea. P-stance: A large dataset for stance detection in political domain. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2355–2365, 2021.

[21] B. Liang, Q. Zhu, X. Li, M. Yang, L. Gui, Y. He, and R. Xu. Jointcl: A joint contrastive learning framework for zero-shot stance detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 81–91. Association for Computational Linguistics, 2022.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[23] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 31–41, 2016.

[24] D. Q. Nguyen, T. Vu, and A. T. Nguyen. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, 2020.

[25] E. W. Pamungkas, V. Basile, and V. Patti. Stance classification for rumour analysis in twitter: Exploiting affective information and conversation structure. *arXiv preprint arXiv:1901.01911*, 2019.

[26] L. Poddar, W. Hsu, M. L. Lee, and S. Subramaniyam. Predicting stances in twitter conversations for detecting veracity of rumors: A neural approach. In *2018 IEEE 30th international conference on tools with artificial intelligence (ICTAI)*, pages 65–72. IEEE, 2018.

[27] P. Sobhani, D. Inkpen, and X. Zhu. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557, 2017.

[28] S. Somasundaran and J. Wiebe. Recognizing stances in ideological online debates. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 116–124, 2010.

[29] D. Sridhar, J. Foulds, B. Huang, L. Getoor, and M. Walker. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 116–125, 2015.

[30] L. Sun, X. Li, B. Zhang, Y. Ye, and B. Xu. Learning stance classification with recurrent neural capsule network. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part I 8*, pages 277–289. Springer, 2019.

[31] Y. Sun, J. Lu, L. Wang, S. Li, and N. Huang. Topic-aware contrastive learning and k-nearest neighbor mechanism for stance detection. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2362–2371, 2023.

[32] P. Vijayaraghavan, I. Sysoev, S. Vosoughi, and D. Roy. Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns. *arXiv preprint arXiv:1606.05694*, 2016.

[33] Z. Wan, F. Cheng, Z. Mao, Q. Liu, H. Song, J. Li, and S. Kurohashi. Gpt-re: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3534–3547, 2023.

[34] L. Wang and D. Wang. Solving stance detection on tweets as multi-domain and multi-task text classification. *IEEE Access*, 9:157780–157789, 2021.

[35] X. Wang, Y. Wang, S. Cheng, P. Li, and Y. Liu. Deem: Dynamic experienced expert modeling for stance detection. *arXiv preprint arXiv:2402.15264*, 2024.

[36] T. Xie, Q. Li, Y. Zhang, Z. Liu, and H. Wang. Self-improving for zero-shot named entity recognition with large language models. *arXiv preprint arXiv:2311.08921*, 2023.

[37] R. Xu, Y. Zhou, D. Wu, L. Gui, J. Du, and Y. Xue. Overview of nlpcc shared task 4: Stance detection in chinese microblogs. In *Natural language understanding and intelligent applications*, pages 907–916. Springer, 2016.

[38] K. Xuan and R. Xia. Rumor stance classification via machine learning with text, user and propagation features. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 560–566. IEEE, 2019.

[39] R. Yang, B. Yang, S. Ouyang, T. She, A. Feng, Y. Jiang, F. Lecue, J. Lu, and I. Li. Leveraging large language models for concept graph recovery and question answering in nlp education. *arXiv preprint arXiv:2402.14293*, 2024.

[40] B. Zhang, D. Ding, and L. Jing. How would stance detection techniques evolve after the launch of chatgpt? *arXiv preprint arXiv:2212.14548*, 2022.

[41] B. Zhang, X. Fu, D. Ding, H. Huang, Y. Li, and L. Jing. Investigating chain-of-thought with chatgpt for stance detection on social media. *arXiv preprint arXiv:2304.03087*, 2023.

[42] X. Zhang, J. Yuan, Y. Zhao, and B. Qin. Knowledge enhanced target-aware stance detection on tweets. In *Knowledge Graph and Semantic Computing: Knowledge Graph Empowers New Infrastructure Construction: 6th China Conference, CCKS 2021, Guangzhou, China, November 4-7, 2021, Proceedings 6*, pages 171–184. Springer, 2021.

[43] Q. Zhu, B. Liang, J. Sun, J. Du, L. Zhou, and R. Xu. Enhancing zero-shot stance detection via targeted background knowledge. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2070–2075, 2022.