Knowledge-enhanced Prompt-tuning for Stance Detection

HU HUANG, School of Cyberspace Science and Technology, University of Science and Technology of China, China
BOWEN ZHANG, College of Big Data and Internet, Shenzhen Technology University, China
YANGYANG LI, Academy of Cyber, China
BAOQUAN ZHANG, YUXI SUN, and CHUYAO LUO, School of Computer Science and Technology, Harbin Institute of Technology, China
CHENG PENG, University of Electronic Science and Technology of China, Zhongshan Institute, China

Investigating public attitudes on social media is important in opinion mining systems. Stance detection aims to analyze the attitude of an opinionated text (e.g., favor, neutral, or against) toward a given target. Existing methods mainly address this problem from the perspective of fine-tuning. Recently, prompt-tuning has achieved success in natural language processing tasks. However, conducting prompt-tuning methods for stance detection in real-world remains a challenge for several reasons: (1) The text form of stance detection is usually short and informal, which makes it difficult to design label words for the verbalizer. (2) The tweet text may not explicitly give the attitude. Instead, users may use various hashtags or background knowledge to express stance-aware perspectives. In this article, we first propose a prompt-tuning-based framework that performs stance detection in a cloze question manner. Specifically, a knowledge-enhanced prompt-tuning framework (KEprompt) method is designed, which consists of an automatic verbalizer (AutoV) and background knowledge injection (BKI). Specifically, in AutoV, we introduce a semantic graph to build a better mapping from the predicted word of the pretrained language model and detection labels. In BKI, we first propose a topic model for learning hashtag representation and introduce ConceptGraph as the supplement of the target. At last, we present a challenging dataset for stance detection, where all stance categories are expressed in an implicit manner. Extensive experiments on a large real-world dataset demonstrate the superiority of KEprompt over state-of-the-art methods.

CCS Concepts: • **Information systems** → **Social networks;**

Additional Key Words and Phrases: Stance detection, deep learning, prompt-tuning framework

This research is supported by the Stable Support Project for Shenzhen Higher Education Institutions (SZWD2021011) and the Research Promotion Project of Key Construction Discipline in Guangdong Province (2022ZDJS112).

Authors' addresses: H. Huang, School of Cyberspace Science and Technology, University of Science and Technology of China, Hefei, 230000, China; B. Zhang (corresponding author), College of Big Data and Internet, Shenzhen Technology University, Shenzhen, 518000, China; email: zhang_bo_wen@foxmail.com; Y. Li, National Engineering Research Center for Risk Perception and Prevention, CAEIT, Academy of Cyber, Beijing, 100000, China; B. Zhang, Y. Sun, and C. Luo, School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, 518000, China; C. Peng, University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan, 528400, China.

2375-4699/2023/06-ART159 $15.00
https://doi.org/10.1145/3588767

ACM Reference format:
Hu Huang, Bowen Zhang, Yangyang Li, Baoquan Zhang, Yuxi Sun, Chuyao Luo, and Cheng Peng. 2023. Knowledge-enhanced Prompt-tuning for Stance Detection. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 22, 6, Article 159 (June 2023), 20 pages.
https://doi.org/10.1145/3588767

1 INTRODUCTION

Stance detection tasks in **natural language processing (NLP)** aim to carry out attitude classification toward a certain target given opinionated input texts [21]. Early research on stance detection was concentrated on online debates, where the sentence format is normalized and the user's attitude is usually clearly expressed [36, 39]. With the rapid development of the Internet, increasingly, research works have focused on mining from social media, such as Twitter [45, 49]. Generally, the sentence structure for social media is usually short and informal, which poses a challenge.

Conventional methods can be classified into non- and **pretrained language models (PLMs)**. Non-pretrained models conduct deep neural networks, for example, **long short-term memory (LSTM)**, **attention-based models (Att)**, and **graph convolutional network (GCN)**, for building stance classification models. For example, Du et al. [12] proposed an attention method by utilizing target-specific knowledge for stance classification. Dey et al. [11] utilized two RNNs to filter the non-neutral text and classify attitudes separately. Sun et al. [37] developed a hierarchical attention method to learn text representation via carefully designed linguistic factors. Liang et al. [26] presented the effective GCN-based method to differentiate target-invariance or target-specific features to learn informative stance features. Inspired by the recent success of PLMs, fine-tuning methods have led to improvements [28]. Fine-tuning models adapt PLM by building a stance classification head on top of the "<cls>" token and fine-tuning the whole model.

Recently, some works have shown that one of its critical challenges is the substantial gap of objective forms between pre-training and fine-tuning, which restricts PLMs from reaching their full potential [16, 44]. More recently, a new paradigm, prompt-based learning, has achieved great success on text classification tasks by reformulating classification tasks as cloze questions [33]. A typical way to employ prompts is to pack the input text into a natural language template and let the PLM carry out masked language modeling. For example, to assort the stance polarity of a sentence "*we should support it*" with the target "*Feminist Movement*" into the "*favor*" category, we combine the sentence with a template: "*we should support it, the attitude for Feminist Movement is* [MASK]." The prediction is made based on the probability that the word "support" is filled in the "[MASK]" token. The mapping from label words (e.g., "support") to the specific class is called the Verbalizer, which bridges a projection between the vocabulary and the label space and has a great influence on the performance of classification. To the best of our knowledge, no research work has conducted the prompt-tuning method for stance detection tasks, which motivates this study.

Despite the effectiveness of prior work, conducting a prompt-tuning method for stance detection remains a challenge for several reasons: (1) Different from the traditional text classification task, the stance detection task often deals with texts from social media that are short and informal. Thus, it makes it challenging to design adequate and suitable label words for verbalizers based on limited information in both manual-defined and automatically generated manner [16]. (2) Second, the tweet text may not explicitly give the attitude. Instead, users may use various hashtags[1] or background knowledge to express stance-aware perspectives in practice. However, such information is not fully leveraged in these prompt-tuning methods. Therefore, the performance improvement of directly employing these existing text classification methods in stance detection tasks is limited.

[1] A special symbol starting with #.

To address the above challenge, in this article, we propose a **knowledge-enhanced prompt-tuning framework (KEprompt)** method for stance detection. KEprompt is a novel framework that consists of two main components: **automatic verbalizer (AutoV)** and **background knowledge injection (BKI)**. AutoV can automatically select apposite label words of the verbalizer, and BKI explores background knowledge of hashtags and targets. Specifically, (1) AutoV contains two steps: construction and refinement. In the construction stage, we introduce a semantic knowledge graph as a supplement to construct label words. Second, to cope with the noise in the unsupervised expansion of label words, we propose refinement methods. (2) In BKI, we propose a neural topic model to learn the representation of the hashtag. Second, inspired by Reference [15], we introduce *ConceptGraph*[2] as a supplement to the target to integrate the background knowledge. Finally, to clearly verify the effectiveness of our method on implicit sentiment texts, we annotate a new **implicit stance detection dataset (ISD)**, where all stance categories are expressed in an implicit manner.

The main contributions of this article are summarized as follows:

- We propose the KEprompt framework for stance detection. In KEprompt, we propose an automatic verbalizer to automatically define the label words and a background knowledge injection method to integrate the external background knowledge.
- We annotate a new stance detection dataset ISD for evaluating the effectiveness of all stance detection methods on implicit sentiment texts.
- We conduct extensive experiments on widely used benchmarks to verify the effectiveness of our model for stance detection, which shows the effectiveness of our model. The code and ISD dataset will be released at https://share.weiyun.com/dHuomknT.

The remainder of this article is organized as follows: Section 2 describes the related work, including some traditional and recent methods of stance detection, prompt-tuning networks and datasets used in stance detection. In Section 3, we provide the details of our KEprompt. In Section 4, we provide the details of the ISD dataset. In Section 5, we give the experimental results. Finally, Section 6 presents the conclusions.

2 RELATED WORK

2.1 Stance Detection

Inferring a text's attitude toward a certain target is the goal of stance detection, which is related to argument mining, fact-checking, and aspect-level sentiment analysis [19, 31]. (1) For in-domain setup, conventional methods can be classified into two categories: non- and pretrained methods. The non-pretrained methods mainly conduct deep neural networks, such as Att and GCN, to train a stance classifier. The Att methods mainly utilize target-specific information as the attention query and deploy an attention mechanism for inferring the stance polarity [11, 12, 37, 41]. The GCN methods propose a graph convolutional network to model the relation between target and text [6, 8, 23]. (2) Several studies are also being conducted for **cross-target stance detection (CTSD)** tasks, which can be classified into two categories. The first class of methods is word-level transfer, which uses the common words shared by two targets to bridge the knowledge gap [3]. Second, some approaches handle this cross-target problem with concept-level knowledge shared by two targets [5, 42, 46]. (3) **Zero-shot stance detection (ZSSD)** is the special case of targets during the inference of unseen to a trained stance detection model, which is more challenging. Specifically, Allaway and McKeown [2] delivered a large-scale human-labeled stance detection dataset in the zero-shot scenario. Allaway et al. [2] utilized a target-specific stance detection dataset to ZSSD

[2] https://concept.research.microsoft.com/.

Table 1. Details of the Existing Dataset

	Authors	Targets	Type	Size
1	Mohammad et al. [29]	Atheism, Climate change is a real concern, Feminist movement, Hillary Clinton, Legalization of abortion, Donald Trump	Target-specific	4,870
2	Sobhani et al. [35]	Trump-Clinton, Trump-Cruz, Clinton-Sanders	Multi-target	4,455
3	Conforti et al. [7]	Merger of companies: Cigna-Express Scripts, Aetna-Humana, CVS-Aetna, Anthem-Cigna, Disney-Fox	Target-specific	51,284
4	Li et al. [25]	Donald Trump, Joe Biden, Bernie Sanders	Target-specific	21,574
5	**Ours (ISD)**	Donald Trump, Joe Biden	Target-specific	6,027

and employed adversarial learning to mine target-invariance information. Liu et al. [28] proposed a common sense knowledge-enhanced graph model based on BERT to utilize both the inter- and extra-semantic information. Liang et al. [26] presented an effective method to differentiate target-invariance or target-specific features to better learn transferable stance features.

2.2 Prompt-tuning

Prompt-tuning has been conducted for various natural language processing tasks such as text classification [16], natural language understanding [33], and sentiment analysis [22]. The verbalizer is a crucial part of prompt-tuning and has a significant impact on how well it works [13]. These methods can be classified into two types: (1) human-designed verbalizers, which are highly biased to personal expertise and do not have enough coverage; for example, Schick et al. [33] manually define label words for text classification. (2) Automatic verbalizer, which adopts automatic searching methods to obtain a better verbalizer. However, current methods require a large number of training and validation sets for optimization [34].

So far, several previous studies have been conducted for stance detection [14, 20]. For example, Jiang et al. [20] first proposed the prompt-tuning framework TAPD for stance detection, in which the verbalizer maps each label to a hidden vector for predicting the label. Hardalov et al. [14] proposed a prompt-based framework for cross-lingual stance detection. Additionally, our work is also closely related to KPT [16] and AutoPT [34], which were developed for aspect-level sentiment analysis (ABSA) and the task form is related to stance detection. KPT introduces external sentiment lexicons to enrich label words for verbalizer, while AutoPT generates the words from the training texts via PLM. The difference lies in several aspects: (1) For the ABSA task, sentiment-carry words are clearly expressed, so KPT can directly refer to the corresponding sentiment-lexicons according to such words. However, for the stance detection task, attitude words are often implicitly expressed and lack relate lexicons. (2) AutoPT generates the words from training data. However, the text form is short and informal for stance detection, which makes the generated label words difficult to distinguish between categories.

2.3 Dataset for Stance Detection on Social Media

Several datasets have been constructed and have become benchmark datasets for stance detection on social media. The comparison of our ISD dataset with some existing stance detection datasets is summarized in Table 1. **SemEval-2016 Task 6 (SEM16)** is the first stance detection dataset

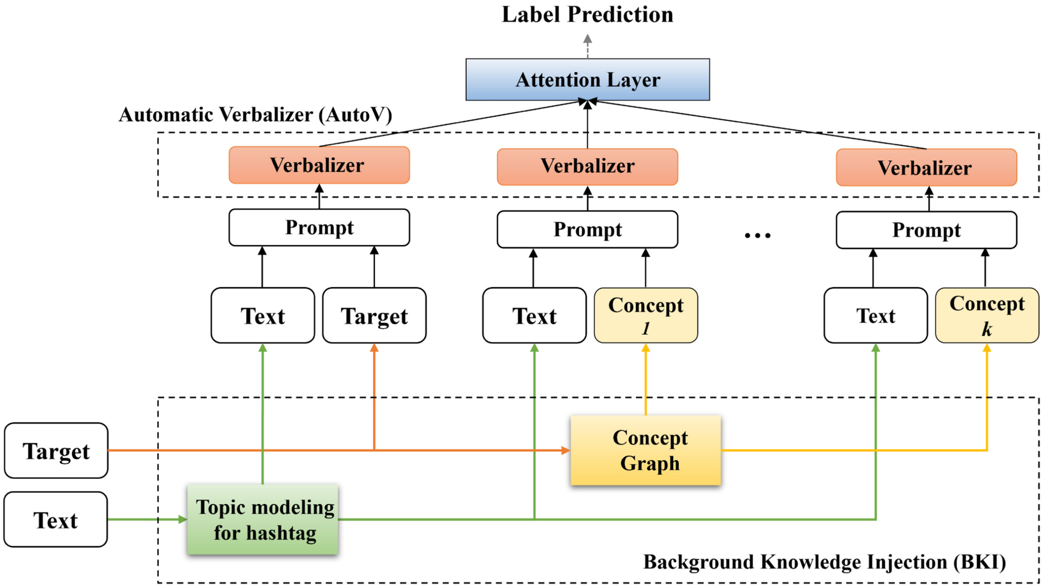


Fig. 1. Framework overview of KEprompt.

collected from Twitter and widely used as the benchmark, which contains 4,870 stance-bearing tweets toward different targets [29]. Subsequently, to utilize large-scale annotated datasets, Zhang et al. then extended SemEval-6 by adding the *Trade Policy* target. Conforti et al. [7] proposed a WT-WT dataset that contains a larger labeled corpus. Li et al. [25] proposed the P-Stance dataset, which is specific to the political domain and contains the longer average length for each tweet. We summarized the difference between a similar dataset and the proposed ISD in Table 4.

3 OUR METHODOLOGY

We use $X = \{x_i, q_i\}_{i=1}$ to represent the labeled dataset, where each x denotes the input text and q denotes the corresponding target. Each sentence-target pair $(x, q) \in X$ is labeled with a stance label y. Given an input sentence x and a corresponding target q, the goal of stance detection is to predict a stance label for the input sentence with a given target.

3.1 Model Overview

As illustrated in Figure 1, our KEprompt consists of an automatic verbalizer, AutoV, and BKI. Here, AutoV aims to automatically define the label words, which contain construction and refinement stages. Specifically, in verbalizer construction, we define the label by utilizing an external semantic graph, while the verbalizer refinement accounts for rectifying these collected label words to alleviate the effects of noises. In BKI, we introduce two types of background knowledge to represent hashtags and targets.

3.2 Preliminary: Prompt-tuning with PLM

Prompt-tuning formulates the stance detection task into a masked language modeling task. In particular, prompt-tuning packs the given text x, q with a template p, which is a designed text. For example, when we need to classify the sentence x =“*We should support this.*” into the stance label “favor” or “against.” The prompt-tuning method wraps the text x with the defined templete:

x_p = "*We should support this. The attitude to the* <Target q> *is* $\underline{[MASK]}$." Let M be PLM, it provides the probability that each word v in the vocabulary being filled in [MASK] given P_M([MASK]= $v|x_p$). Here, v is the defined label word in the verbalizer. To map the probabilities of such words to the probabilities of the labels, here, a verbalizer is a mapping f from the defined words in the vocabulary, which forms the label word set V, to the label space Y, i.e., $f : V \rightarrow Y$. Formally, the probability $P(y|x_p)$ of label y, is computed as:

$$P(y|x_p) = \mu(P_M([\text{MASK}] = v|x_p)|v \in y), \tag{1}$$

where μ is a function transforming the probability of label words into the probability of the label. In the above example, prompt-tuning may define V_1 = "{support, agree}," V_2 = "{opposition}" and μ as an identity function, then if the average probability of the words in V_1 is larger than the words in V_2, we classify the instance into *favor* class. For prompt tuning, the learning objective is to minimize:

$$l(y|x_p) = -logP_M([\text{MASK}] = v|x_p). \tag{2}$$

3.3 Automatic Verbalizer

Verbalizer Construction. Predicting masked words prompt-based context is not a single-choice procedure, because various words may fit this text. Most existing methods [17, 32] mainly focus on leveraging limited information to construct verbalizer (insufficient coverage). Therefore, we introduce SenticNet [4] as prior knowledge to expand the label words. SenticNet can obtain its related semantically related words according to the given word. For example, the semantic-related words of "mad" from SenticNet are "*resent*, *malice*, *rage*, *temper*." Specifically, to utilize semantic knowledge, we send the word W_g in candidate (i.e., support, against, etc.) into the SenticNet graph and acquire the λ-hop semantic-related words, which are denoted as W_s.

Verbalizer Refinement. Although we have constructed a verbalizer that contains comprehensive label words by the step of verbalizer construction, the collected label words are very noisy; thus, it is necessary to refine such a verbalizer to retain high-quality label words. To this end, we proposed to organize massive label words into a tree structure, which can provide an explicit quality evaluation (i.e., the lower-level leaf nodes represent lower importance) for filtering label words. Specifically, each node of the tree denotes words and is rooted on the class label (e.g., "favor"), and the ith layer nodes are the word extracted by i-hop from SenticNet. Based on the above tree structure, we conduct a refinement strategy to filter label words. The main idea is to calculate the quality score P_I of label words from lower- to high-level nodes and then delete these words whose quality scores are less than a threshold α. Formally, as a simple solution, we first randomly sample a small-size text $\hat{D}$ and then leverage it to calculate the average probability (denoted as $P_D(v)$) of the predicted probability P_{avg} for each word in the verbalizer:

$$P_D(v) = \frac{1}{|\hat{D}|} \sum_{x \in \hat{D}} P_{avg}([MASK] = v|x_p). \tag{3}$$

To further strengthen the refinement, along with the learning process, we compute the growth rate (denoted as P_G) of the predicted probability for each label word. We argue that a small growth rate indicates that the word is rare to the PLM. Therefore, the predicted words of PLM tend to be inaccurate. Based on this, we compute the final quality score of each word v_i by:

$$P_I(v_i) = P_D(v_i) + P_G(V_i). \tag{4}$$

At last, we remove the label words whose quality scores are less than a threshold from lower layers to higher layers of the tree.

3.4 Background Knowledge Injection

People have wide-ranging background knowledge that can be used it to understand the implicit stance in a text. In this article, we introduce background knowledge to help the model's more profound understanding of the text, thus enhancing its performance on the stance detection task. Here, we introduce two types of background knowledge: First, we acquire the background knowledge to enrich the knowledge of the target. Second, we propose a topic modeling method for hashtag representation.

Target-related background knowledge. To precisely capture the target-related background knowledge, we send the target word into *ConceptGraph* to acquire the related background knowledge (c_i). For example, background knowledge of the target "*Feminist Movement*" is "e*galitarian movement, publicized revolution*," and so on. Then, we construct the template for the prompt-tuning method. For example, given the input "Men and women should have equal rights," with the target "Feminist Movement," the template can be: "Men and women should have equal rights. The attitude of egalitarian movement is [MASK]" or "*Men and women should have equal rights. The attitude of publicized revolution is* [MASK]."

Topic modeling for hashtag. To represent the hashtag, we utilize the **neural topic model (NTM)** to learn hashtag representation. Specifically, we select unlabeled dataset for each hashtag and propose the topic modeling method to learn a representation for each hashtag.

We use K to denote the topic numbers in the topic modeling process, and we use t_k, which is a learnable parameter, as the topic embedding for each of the topics k $(k = 1, 2, ..., K)$. Specifically, we denote T assembled from such topic embeddings $T = \{t_1, t_2, ..., t_K\}$. Here, the word embedding matrix is denoted as ε. By calculating the semantic similarity between the topic and words, we may determine word distribution γ_k for each topic k:

$$\gamma_k = softmax(\varepsilon t_k), k = 1, 2, ..., K. \tag{5}$$

Finally, the topic words' distributions are: $\gamma = (\gamma_1, \gamma_2, ..., \gamma_K)$.

We deploy LDA-style topic modeling in our method by utilizing a **variational autoencoder (VAE)**.

(1) We first study the latent variable ζ from the prior distribution: $\zeta \sim N(0, I)$.
(2) Then, we can acquire the topic distribution $\eta = softmax(W\zeta)$, where W is a trainable parameter.
(3) For the word at n position of the sentence, $n = 1, 2, ..., N$, we acquire a word $w_n \sim \gamma\eta$. Finally, we calculate the probability of w_n by:

$$p(w_n|\eta, \gamma) = \sum_{k=1}^{K} p(k|\eta)p(w_n|\gamma_k) = [\eta\gamma]_{w_n}, \tag{6}$$

and thus we have $w_n \sim \gamma\eta$, where $\gamma\eta \in R^V$.

Because it is challenging to infer the posterior for ζ, VAE uses a variational distribution $q(\zeta|x)$ to approximate the true posterior. Then, $q(\zeta|x)$ is a diagonal Gaussian: $q(\zeta|x) = N(\zeta; \mu, \sigma^2 I)$, and μ, σ^2 are parameterized by **neural network layers (NN)**: $\mu = NN_\mu(x), log(\sigma^2) = NN_\sigma(x)$.

During the training process, the goal of topic model is to maximize the variational lower bound:

$$\gamma = \mathbb{E}_{\zeta \sim q(\zeta|x)}[xlog(\gamma\eta)] - D_{KL}[q(\zeta|x)||p(\zeta)]. \tag{7}$$

Finally, NVI's objective function is:

$$min_{\{T, W, NN_\mu, NN_\sigma\}} = \mathbb{E}_{\zeta \sim q(\zeta|x)}[xlog(\gamma\eta)] - D_{KL}[q(\zeta|x)||p(\zeta)]. \tag{8}$$

3.5 Attention Layer

To effectively integrate the background knowledge with prompt-tuning, we propose an attention layer. Specifically, after acquiring the background knowledge, we train a separate language model M_i with each c_i via Equation (2). After each M_i is trained separately, the predicted words are then sent to the attention layer for ensembling. Formally, assume E_i denotes the embedding of the predicted word of M_i. We use a learnable representation h as the attention query to compute the attention weight α_t for the tth word:

$$\alpha_t = softmax(h^{\mathrm{T}} E_i) \tag{9}$$

$$emb = \sum_{t=1}^{n} \alpha_t E_i \ . \tag{10}$$

Given words from the verbalizer, we produce the probability that the token v can be selected as the label words,

$$\delta = \frac{\exp{(v_i \cdot emb)}}{\sum_{v_j \in V} \exp{(v_j \cdot emb)}} \ , \tag{11}$$

where v is the embedding of the token in Verbalizer. Then, we sum the words' probabilities of each label, which denotes as $\hat{y}$. Finally, the loss function of ensemble network can be standard cross-entropy methods:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{j=1}^{C} y_{ij} \log \hat{y}_{ij}, \tag{12}$$

where N represents the number of samples for training, C means the number of stance classes, y_i denotes the one-hot represented ground-truth label for the ith sample. Finally, the attention layer is optimized by the standard gradient descent algorithm.

4 A NEW DATASET FOR IMPLICIT STANCE DETECTION (ISD)

One major challenge in stance detection is that the tweet text may not explicitly give the attitude words. Therefore, it requires a deep understanding of the background knowledge of the text and target. However, in existing stance detection datasets, most sentences explicitly contain attitude words, making stance detection degenerate to sentence-level text classification. In this article, we present a challenging ISD dataset for stance detection. The text does not contain explicit sentiment words; thus, it is necessary to understand the relationship between the text and the background knowledge of the target to predict the stance polarity effectively.

4.1 Data Collection

The tweets are collected by using the Twitter streaming API. Following the prior works [3, 25, 43] that target presidential candidates, we focus on two targets in the presidential race of 2020: "Donald Trump (DT)" and "Joe Biden (JB)." We filter out tweets that are very short and simple (e.g., "vote for Joe") and contain no or ineffective hashtags (e.g., the hashtag for crawling). In sum, we gathered approximately 762,255 tweets for the two targets combined.

4.2 Data Preprocessing

The preprocessing stage contained several steps: (i) We deleted the hashtag that was used for crawling the text. The reason is that such a hashtag may appear in most sentences, so its presence in a sentence is meaningless, equivalent to the inclusion of a symbol in all texts. (ii) Duplicates and retweets were removed. Twitter data are noisy owing to repeated tweets in addition to the

Table 2. Statistics of the Datasets

Dataset		Favor	None	Against	SUM	Avglen	h-tags
Trump	L	875	1,134	1,096	3,105	17.16	467
	U		\		391,163	19.06	
Biden	L	1,046	525	912	2,922	17.45	432
	U		\		365,065	19.02	

SUM and *Avglen* denote the total number of instances and average length for all texts in the dataset. *h-tags* represent the number of hashtags from two targets, *L* and *U* denote labeled and unlabeled text, respectively.

Table 3. Statistics of the Training and Test Datasets

	Training				Test			
Target	Favor	Against	None	SUM	Favor	Against	None	SUM
Trump	28.1%	35.4%	36.5%	2637	28.2%	35.3%	36.5%	468
Biden	31.9%	33.3%	34.8%	2483	31.9%	33.3%	34.8%	439

inventive spellings, lingo, and URLs. We must purge duplicates from the dataset to clean it up, since these duplicate data make it harder for us to create trustworthy models. (iii) We save only the English tweets. This research aims to create an English stance detection dataset; thus, we leave multilingual stance detection to future research.

4.3 Data Annotation and Quality Assurance

We invited three experienced natural language processing researchers to annotate the stance polarity with “Favor,” “Against,” and “None.” To guarantee the labeled quality, we raised two strict annotation requirements: (1) following Tang et al., [38], the annotators were asked to discard the “conflict sentences,” which contain too many hashtags that are unrelated to two targets. (2) They were asked to select tweets with certain hashtags that contain underlying information, such as implicit stance-aware topics. Thus, the data in ISD usually contain at least one hashtag with the underlying information.

To ensure the data quality, we applied the following steps after the data annotation: (i) We delete the text that only contains the hashtag used for crawling the text. The reason is that such a hashtag may appear in most sentences, so its presence in a sentence is meaningless, equivalent to the inclusion of a symbol in all texts. (ii) We manually select the text that contains the hashtag with an underlying meaning and delete the hashtag used for crawling the text inside one sentence.

4.4 Data Analysis

The statistics of the ISD dataset are given in Tables 2 and 3. ISD consists of 756,228 unlabeled tweets and 6,027 labeled tweets for *DT* and *JB*, respectively. All sentences in the ISD dataset contain at least one hashtag that contains attitude-bearing topics. The sentences contain 2.14 hashtags on average. The average length is 17.45 and 16.31 for *DT* and *JB*, respectively. We created the training and testing sets following an 80/20 split for both Trump and Biden targets.

4.5 Comparison with Existing Datasets

The main difference between ISD and existing datasets is that the user attitude in ISD is referred to in a more implicit way. First, it may not contain attitude-bearing words in the text. Moreover, the attitudes may be reflected in the hashtags. For example, given a tweet “Everyone has rights, it’s time for us to act #StopTrump” with target *DT*. The text is a neutral description, but the attitude

Table 4. Example of Hashtags

Target	Hashtag for crawling	Hashtag with implicit meaning
Trump	#DonaldTrump #Republican	#MAGA
Biden	#JoeBiden #Democrats	#SleepyJoe

is reflected in the label "#StopTrump." Second, the attitude tendencies of the text may be opposite toward the given target. Considering the example, "It's a huge waste #theWall #VoteJoe" with the target Biden, it is difficult to correctly infer the stance considering only the text, because the stance is reflected by the relation of #theWall and the target Biden. These characteristics contribute to making ISD a challenging dataset for stance detection.

However, for the conventional datasets, such as SEM16 [29] and P-Stance [25], the hashtags only use for crawling. Table 4 shows the difference between these hashtags. For example, the hashtags may #JoeBiden for the conventional dataset, which contains no stance-aware meaning. While for ISD, the hashtags may #SleepyJoe, which contains implicit stance polarity. Therefore, the conventional datasets treat the stance detection task as a sentence-level text classification task. It can be seen from prior works that the simple sentence-level text classification classifier can still earn competitive results with many recent stance detection methods in the existing datasets [27].

5 EXPERIMENTS

5.1 Experimental Data

In this article, we conduct experiments on strong benchmark datasets from **SemEval-2016 Task 6 (SEM16)**, P-stance [25], ISD, and VAST [1].

- **SEM16** consists of 4,870 tweets with different targets. Each tweet is labeled with "*favor*," "*against*" or "*neutral*." Following the setup from Reference [42], we select four targets: *Donald Trump* (D), *Hillary Clinton* (H), *Legalization of Abortion* (L), and *Feminist Movement* (F). These targets are widely utilized to evaluate the stance detection task. Sepcifically, for the cross-target setup [26, 42, 46], we constructed eight cross-target stance detection tasks (*D→H, H→D, F→L, L→F, T→H, H→T, T→D, D→T*). The source target is represented by the left side of the arrow in this instance, while the destination target is represented by the right side.
- **VAST** is the zero-shot stance detection dataset. Each sample contains a sentence, a target, and a stance polarity from *"Pro," "Con,"* or *"Neutral."* There are 4,003 samples for training and 383, 600 as the *dev* and *test* sets, respectively.
- **P-stance** contains 21,574 tweets, with the *"Donald Trump (DT)," "Joe Biden(JB),"* and *"Bernie Sanders"* targets.

5.2 Compared Baseline Methods

We assess and contrast our model against a number of reliable baselines, as follows:

Statistics-based methods:

- **BiLSTM** [3] uses Bi-LSTM to encode the sentence and the target separately. Then, **Bicond** uses the conditional encoding method for learning the target-dependent representation.
- **MemNet** [38] employs a memory network that uses the multi-hop attention mechanism to encode the text.
- **AOA** [18] models the target and context with two LSTMs, respectively. Then, the interactive attention is introduced to model the relation.

- **ASGCN** [47] employs the dependency tree to model dependencies and applies GCN for text representation.
- **TAN** [12] proposes target-specific attention with the LSTM model for stance detection.
- **TPDG** [27] proposes a target-adaptive graph convolutional network for stance detection, which utilizes the shared features from other similar targets.
- **AT-JSS-Lex** [24] proposes a target-adaptive graph convolutional network for stance detection, which utilizes the shared features from other similar targets.

Fine-tuning-based methods:

- **BERT-FT** & **RoBERTa-FT** [10] uses a pretrained BERT or RoBERTa model to perform stance detection. To adapt to the training and fine-tuning of the Bert model, we convert the given context and target to "[CLS] + text + [SEP] + target + [SEP]."
- **S-MDMT** [40] proposes a target adversarial learning method based on BERT, which can acquire stance-toward information.
- **RelNet** [48] develops a BERT-based knowledge-aware framework for stance detection.
- **STANCY** [30] proposes a BERT-based model for stance detection, which is pre-trained with additional corpora.
- **PT-HCL** [26] proposes a contrastive learning method for cross-target and zero-shot stance detection.

Prompt-tuning-based methods:

- **MPT** develops prompt-tuning-based PLM to perform stance detection, where humans define the verbalizer.
- **AutoPT** [16] proposes an auto-prompt method for stance detection, where the label word is generated from the data corpus.
- **KPT** [34] introduces external lexicons to define the verbalizer. Different from the lexicon utilized in Reference [34], we utilize SenticNet instead of sentiment lexicons.
- **PIN-POM** [9] develops soft prompt methods for short text classification.
- **TAPD** [20] develops the prompt-tuning method for stance detection.

5.3 Evaluation Metrics

Following References [26, 46], we use the micro average F1-score in evaluation. First, we compute the F1-score for *Favor* and *Against*:

$$\begin{aligned} F1_{favor} &= \frac{2P_{favor}R_{favor}}{P_{favor} + R_{favor}} \\ F1_{against} &= \frac{2P_{against}R_{against}}{P_{against} + R_{against}} \end{aligned}, \tag{13}$$

where P and R are precision and recall, respectively, and the final F1-score can be computed by:

$$F1_{avg} = \frac{F1_{favor} + F1_{against}}{2}. \tag{14}$$

Second, because the targets in the dataset are unbalanced, we compute the micro-averaged F1 and the macro-averaged F1 and regard their average as another evaluation metric: $F1_m = (F1_{micro} + F1_{macro})/2$.

$$F1_m = \frac{(F1_{micro} + F1_{macro})}{2} \tag{15}$$

Table 5. Prompt Templates

1. Input text x. It was [MASK].
2. Input text x. target is [MASK].
3. Input text x. The target made me fell [MASK].
4. Input text x. Its attitude to target is [MASK].
5. Input text x. I think target is [MASK].
6. Input text x. I felt the target is [MASK].

5.4 Implementation Details

In the experiments, we select pretrained language models with BERT-base, BERT-large, RoBERTa-base, and RoBERTa-large. The Adam optimizer is applied to train the model with a mini-batch size of 32 and a learning rate of 0.0001. We select the SecticNet lexicon for defining label words and ConceptGraph for enriching the background knowledge of the target. In this article, we manually define the templates for prompting PLM. Details of the templates are shown in Table 5.

5.5 Overall Performance

5.5.1 In-domain Setup. Table 6 shows the results of in-domain stance detection with several strong benchmarks. From the results, we draw the following conclusions. (1) Compared with statistic-based methods, pretrained models can significantly improve the performance of stance detection for most setups. For example, BERT-FT (BERT-base) achieves 5.3% improvements on average (9.9% with BERT-large) compared with the best competitor of statistic-based methods (TPDG) on ISD dataset. This verifies the effectiveness of the pretrained model in stance detection. (2) Prompt-based PLM methods achieve stable improvement in multiple tasks compared with fine-tuning PLM. For example, compared with TAPD, KEprompt improves 5.9% for $\mathrm{F1}_{avg}$ with BERT-base on average of SEM16 datasets. The result shows that the Prompt framework can better release the performance of PLM. (3) After introducing SenticNet as label words of Verbalizer, the performance of KPT has been significantly improved. As it can be seen from the results of $\mathrm{F1}_{avg}$, KPT achieves 3.6%, 3.8%, and 4.6% improvements compared with MPT on SEM16, ISD, and P-stance on average, respectively. (4) The proposed KEprompt method yields better performance than all the baselines in most of the tasks. For example, our method improves 13.2% over the best neural network-based model (TPDG), 7.79% over the best fine-tuned PLM model (RoBERTa-large), 4.98% over the best prompt-tuning method (MPT), on average of six tasks. The advantage of KEprompt comes from its two characteristics: (i) We develop an automatic verbalizer method to improve the coverage and reduce the bias of the manual Verbalizer. (ii) Background knowledge is introduced into the prompt-tuning framework.

5.5.2 Cross-target Setup. Acquiring large annotated data is a time-consuming and labor-intensive process. Hence, we propose to study how our method works in a cross-target setup. The cross-target setup aims to infer the attitude of the destination target by utilizing labeled data from the source target. The $\mathrm{F1}_{avg}$ and $\mathrm{F1}_{m}$ results are reported in Tables 7 and 8, respectively. From the result, we can observe that our method stably exceeds the other baseline by a significant margin. Among them, compared with previous promising statistical method (TPDG), our proposed model (KEprompt RoBerta-large) improves $\mathrm{F1}_{avg}$ by 8.4% and $\mathrm{F1}_{m}$ by 8.0% on average, which verifies that utilizing a prompt-tuning framework could lead to improvements in cross-target setup. Compared with fine-tuning-based methods (BERT-base, BERT-large, RoBERTa-base, and RoBERTa-large), KEprompt improves 14.6%, 11.0%, 13.1%, and 11.4% over $\mathrm{F1}_{m}$, on average. The result further highlights the crucial role of using the prompt-tuning framework in cross-target stance detection.

Table 6. Performance Comparison on F1$_{avg}$

Embedding	Methods	SEM16			ISD		P-stance	
		F	L	H	DT	JB	DT$_p$	JB$_p$
Statistic.	BiLSTM †	51.6	59.1	55.8	28.6	35.0	69.7	68.6
	BiCond †	52.9	61.2	56.1	55.2	50.5	70.6	68.4
	MemNet †	51.1	58.9	52.3	53.5	52.2	76.8	77.2
	AoA †	55.4	58.3	51.6	55.9	57.6	77.2	77.6
	TAN †	55.8	63.7	65.4	50.4	52.5	77.1	77.6
	ASGCN †	56.2	59.5	62.2	55.1	58.2	76.8	78.2
	AT-JSS-Lex‡	61.5	68.4	68.3	-	-	-	-
	TPDG	67.3	**74.7**	73.4	64.2	60.0	76.8	78.1
BERT-base	FT	62.3	62.4	67.0	69.1	65.6	79.4	79.4
	S-MDMT ‡	63.8	67.2	67.2	-	-	-	-
	STANCY ‡	61.7	63.4	64.7	-	-	-	-
	TAPD ‡	63.9	63.9	70.1	-	-	-	-
	MPT	63.1	62.9	70.4	69.0	65.9	79.3	79.9
	PIN-POM	62.1	62.9	69.2	67.6	65.2	79.2	79.4
	AutoP	62.4	62.4	70.0	67.2	64.8	79.0	79.6
	KPT	63.3	63.5	71.3	69.4	66.4	80.2	80.4
	KEprompt	**72.1**¶	69.1	**74.4**	**70.5**¶	**67.4**¶	**81.0**¶	**81.2**¶
BERT-large	FT	63.5	65.3	72.1	70.7	73.3	81.2	81.3
	MPT	64.7	63.2	71.5	73.4	67.7	81.9	81.4
	AutoP	64.5	60.1	67.2	67.6	64.8	81.6	81.9
	KPT	65.3	65.7	74.9	75.8	73.9	81.9	82.1
	KEprompt	76.8¶	69.4	76.2	77.3	74.8	82.2	82.6
RoBERTa-base	FT	61.8	64.4	76.6	64.2	71.8	74.2	83.4
	MPT	62.6	66.7	75.4	66.8	71.6	75.6	82.7
	AutoP	62.4	67.1	76.4	66.5	71.4	75.3	82.8
	KPT	64.0	67.9	76.9	69.9	72.4	77.9	84.1
	KEprompt	68.3	70.3	77.1	72.4	73.5	83.2	84.4
RoBERTa-large	FT	72.3	67.6	81.5	72.4	72.4	88.1	86.5
	MPT	73.3	71.4	81.3	72.8	72.1	86.0	87.0
	AutoP	72.8	72.6	81.4	71.2	70.8	86.3	86.4
	KPT	75.2	74.2	82.6	75.7	74.4	88.4	88.1
	KEprompt	**80.7**¶	**76.5**¶	**84.2**¶	**77.4**¶	**76.9**¶	**89.5**¶	**88.6**¶

The results with † are retrieved from Reference [46], ‡ are retrieved from Reference [20]. The ¶ mark refers to a p-value < 0.05. The best scores are in bold. Note that, to evaluate the stability of the model, following Reference [46], we run the method three times and report the average score for our proposed KEprompt.

Additionally, note that our proposed model achieves stability superior to KPT and MPT. For example, KEprompt (BERT-base) improves 1.1 % and 4.5% over KPT and MPT on average of all eight setups, respectively. This reveals that our proposed model, which automatically refines the knowledgeable verbalizer and utilizes background knowledge, could potentially enhance the inferring ability of the unseen target.

5.5.3 Zero-shot Stance Detection. In some extreme special cases, the target of the given text may be unseen in the training dataset. Therefore, we also compare our method with the

Table 7. Performance Comparison of CTSD (F1_{avg}) on Eight Tasks

Embedding	Methdos	F→L	L→F	H→D	D→H	H→T	T→H	D→T	T→D
Statistic.	BiLSTM †	44.8	41.2	29.8	35.8	29.1	39.5	31.1	34.1
	BiCond †	45.0	41.6	29.7	35.8	29.2	40.2	31.7	34.7
	CrossNet †	45.4	43.3	43.1	36.2	29.8	41.7	31.4	37.4
	SEKT †	53.6	51.3	47.7	42.0	33.5	46.0	44.4	39.5
	TPDG	58.3	54.1	50.4	52.9	**59.5**	49.8	51.2	48.9
BERT-base	BERT	47.9	33.9	43.6	36.5	26.1	23.1	24.1	45.6
	MPT	42.1	47.6	47.1	58.7	43.4	52.8	51.3	50.5
	KPT	43.1	44.1	46.1	61.4	44.9	54.3	51.6	51.8
	KEprompt	49.1	54.2	54.6	60.9¶	44.7	57.2¶	51.9¶	53.9¶
BERT-large	BERT	46.2	43.4	50.0	44.5	28.3	38.6	30.2	50.9
	MPT	42.5	49.0	47.2	59.9	47.6	56.3	50.9	53.2
	KPT	43.7	50.9	49.7	61.7	49.1	57.7	52.1	56.5
	KEprompt	49.3	54.9¶	50.7	63.3¶	41.6	56.3¶	**53.7**¶	55.5¶
RoBERTa-base	RoBERTa	44.8	42.6	64.9	60.0	26.8	38.5	27.1	51.1
	MPT	47.8	56.4	64.0	65.7	50.3	59.7	51.4	60.6
	KPT	48.3	56.9	64.4	66.0	52.6	61.6	51.7	61.8
	KEprompt	48.9	56.7¶	65.1¶	67.3¶	47.2	64.4¶	51.6	61.8¶
RoBERTa-large	RoBERTa	49.1	55.3	65.9	71.1	35.4	47.6	34.9	56.3
	MPT	63.4	68.9	68.2	74.0	46.2	70.3	51.2	62.2
	KPT	65.0	68.5	69.3	74.2	45.1	70.0	50.5	62.7
	KEprompt	**67.8**¶	**69.6**¶	**68.1**¶	**75.5**¶	49.1	**71.7**¶	53.3¶	**62.9**¶

The results with † are retrieved from Reference [46]. The ¶ mark refers to p-value < 0.05. The best scores are in bold.

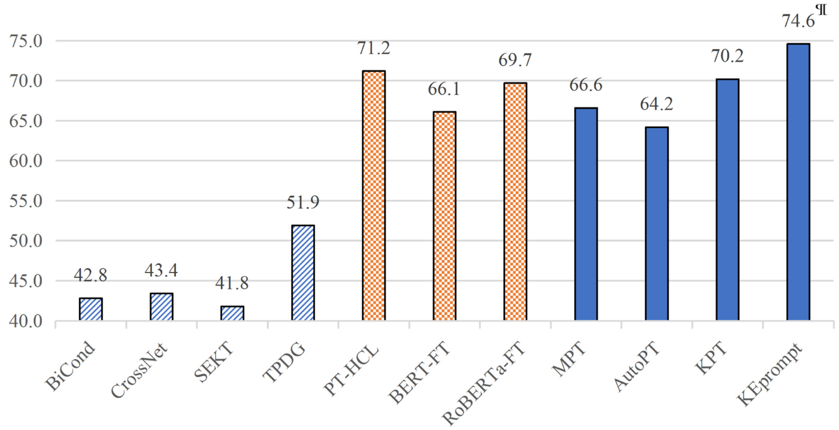


Fig. 2. $F1_m$ results of zero-shot stance detection for the VAST dataset. The ¶ mark refers to p-value < 0.05.

competitors in ZSSD. The results are shown in Figure 2. We can observe that the performance is inferior compared to in-target and cross-target setups because of the limitations and difficulties of ZSSD. Specifically, statistic-based methods perform poorly, since they do not consider the external background knowledge. The fine-tuning-based methods (such as PT-HCL, BERT-FT, and RoBERTa-FT) stably exceed the statistic-based methods by a significant margin, which verifies the effectiveness of the information learned from a large corpus. Despite the challenges and

Table 8. Performance Comparison of CTSD ($F1_m$) on Eight Tasks

Embed	Methods	F→L	L→F	H→D	D→H	H→T	T→H	D→T	T→D
Statistic.	BiLSTM †	40.1	37.9	43.3	40.1	23.6	41.8	20.7	38.9
	BiCond †	40.3	39.2	44.2	40.8	23.9	42.4	20.7	39.6
	CrossNet †	44.2	43.1	46.1	41.8	24.4	42.5	21.1	40.7
	SEKT †	52.3	51.0	46.3	43.2	30.0	48.9	39.1	43.5
	TPDG	62.4	55.9	51.0	51.0	**58.4**	54.5	51.0	50.4
BERT-base	BERT	49.9	39.5	41.2	39.9	35.3	29.5	39.1	47.8
	MPT	50.7	47.8	46.4	51.8	43.6	57.7	52.4	52.2
	KPT	53.8	51.1	49.1	64.1	46.9	58.3	53.1	54.3
	KEprompt	54.6	54.9	49.6	64.9¶	48.0	60.8¶	53.2¶	52.8
BERT-large	BERT	50.1	45.3	51.4	52.8	34.6	39.4	34.8	51.1
	MPT	53.1	53.7	48.5	61.4	48.8	59.4	51.7	55.6
	KPT	55.2	54.1	51.6	64.6	49.8	60.1	52.6	57.1
	KEprompt	56.0	56.6¶	52.0	66.7¶	45.5	60.2	54.4¶	56.0¶
RoBERTa-base	RoBERTa	48.8	42.6	64.5	61.9	33.3	39.4	33.8	51.5
	MPT	55.0	54.9	60.8	64.6	51.4	62.3	51.1	59.6
	KPT	55.2	55.3	62.1	66.5	53.7	64.7	52.9	61.2
	KEprompt	55.6	56.1	64.6¶	68.9¶	55.6	67.2¶	52.8¶	61.3¶
RoBERTa-large	RoBERTa	53.9	58.2	66.2	71.9	40.3	47.6	40.1	55.9
	MPT	65.3	68.5	66.8	73.8	44.5	71.7	51.1	58.5
	KPT	67.6	69.0	67.2	75.5	47.7	71.4	51.9	61.8
	KEprompt	**70.3**¶	**70.5**¶	**68.1**¶	**76.6**¶	50.8	**72.9**¶	**54.3**¶	**62.3**¶

The results with † are retrieved from Reference [46]. The ¶ mark refers to p-value < 0.05. The best scores are in bold.

difficulties of ZSSD, our KEprompt still shows promise and improves significantly when compared to all baselines on the VAST dataset. This implies that our KEprompt is effective in the more challenging ZSSD with the help of utilizing background knowledge and a prompt-tuning framework.

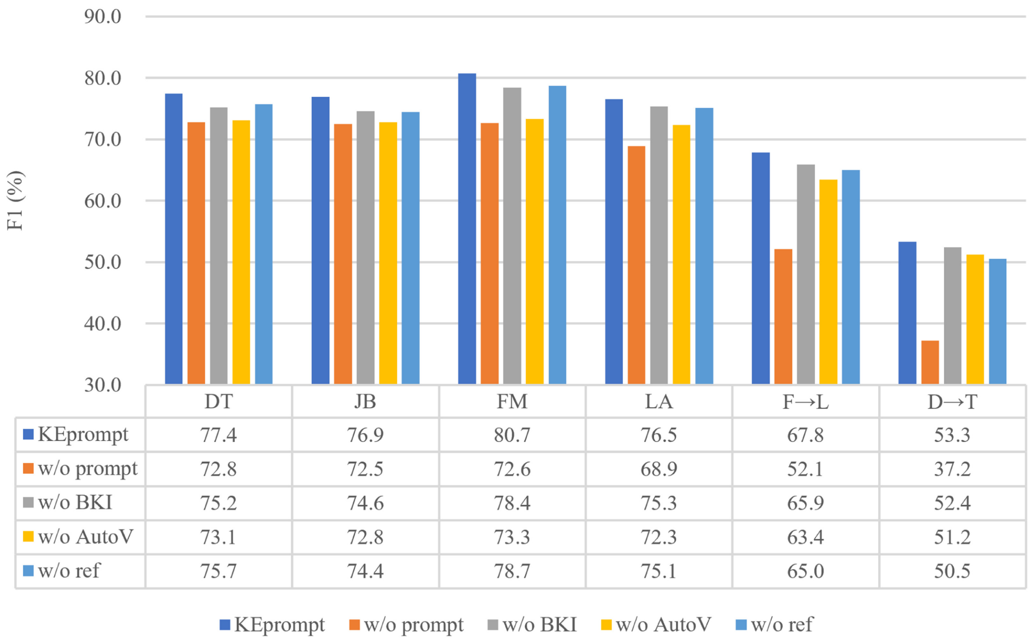
5.6 Ablation Study

To study the impact of each component of the proposed KEprompt method, we implement the ablation test to remove the proposed component denoted as w/o.

The variants of KEprompt:

- **w/o Prompt:** SILTN without the prompt-tuning framework; instead, we use standard fine-tuning method and use the Verbalizer selection method followed by Reference [34].
- **w/o BKI:** SILTN without the background knowledge injection method.
- **w/o AutoV:** SILTN without automatic verbalizer, and the label words are "favor, against, and neutral."
- **w/o ref:** SILTN without verbalizer refinement strategy, and we select the label words from SenticNet with the 1-hop connection.

The ablation results are summarized in Figure 3. From the results, we observe that the prompt-tuning framework makes great improvements to our KEprompt method. Specifically, we can observe that the removal of the automatic verbalizer (w/o AutoV) sharply degrades performance. This verifies the effectiveness and significance of introducing an external lexicon for label words and selecting appropriate label words automatically, which helps fully incorporate the background knowledge and semantic-related knowledge, reducing the bias of human expertise. In addition,



	DT	JB	FM	LA	F→L	D→T
■ KEprompt	77.4	76.9	80.7	76.5	67.8	53.3
■ w/o prompt	72.8	72.5	72.6	68.9	52.1	37.2
■ w/o BKI	75.2	74.6	78.4	75.3	65.9	52.4
■ w/o AutoV	73.1	72.8	73.3	72.3	63.4	51.2
■ w/o ref	75.7	74.4	78.7	75.1	65.0	50.5

Fig. 3. F1_{avg} of the ablation test. DT and JB are from ISD, FM and LA are from SEM16. F, L, D, and T selected from SEM16.

Table 9. The Experimental Results with Different Lexicons

Embed	Lexicons	F→L	H→D	T→D	DT	JB
BERT-base	SenticNet	54.6	49.6	52.8	70.9	68.9
	WordNet	51.7	47.4	53.6	67.8	67.1
RoBERTa-base	SenticNet	55.2	62.1	61.2	71.4	70.7
	WordNet	55.1	61.5	61.0	69.8	69.5

the performance declines considerably when the background information is not considered in the prompt-tuning framework, which reveals that the information of target and hashtag is important for stance detection. Furthermore, note that the removal of the verbalizer refinement strategy (- w/o ref) leads to an evident decline in performance. This implies that the verbalizer refinement strategy can help KEprompt reduce noisy label words and maintain high-quality and suitable label words. Not surprisingly, combining all factors achieves the best performance for all the experiments.

5.6.1 *Impact of Lexicon Selected in Verbalizer Construction.*

Based on the verbalizer refinement strategy, the lexicons we selected are one of the most important parts of the overall performance. To investigate the impact of the external knowledge selected on the performance of our proposed model, we select two widely used lexicons (WordNet and SenticNet) and investigate the performance of each lexicon. In particular, we evaluate the F1_m performance of KEprompt on F→L, H→D, and T→D. From Table 9, we can observe that the performance of utilizing SenticNet is better than that of selecting WordNet. This is because, for some seed words (label words), WordNet missed acquiring the words with multi-hop relation. In contrast, SenticNet has more semantically related words and thus better coverage of label words. Details of label words can be found in Table 10.

Table 10. Example of Label Words Selected with 2-hops (We Filtered out Phrases)

Lexicons	Seeds	2-hop words
SenticNet	against	bad, crap, dumb, sick, stupid, nuts, useless, grim, disease, worse, shocking, against, horrible, suck, ghostly
	neutral	detachment, stationary, withdrawal, inactive, motionless, neutral, indifferent, steady, settled, lazy, stabilize, stable
	favor	favorable, happily, favorably, favor, pleased, sanctioned, agree, affirmative
WordNet	against	/
	neutral	torpid, neutral, unbiased, electroneutral, so-so, sluggish, achromatic, unbiased, inert, immaterial, apathetic, impersonal, indifferent, soggy, deaf
	favor	party-favor, privilege, prerogative, perquisite, prefer, favor, exclusive-right, choose, party-favour, opt, favour

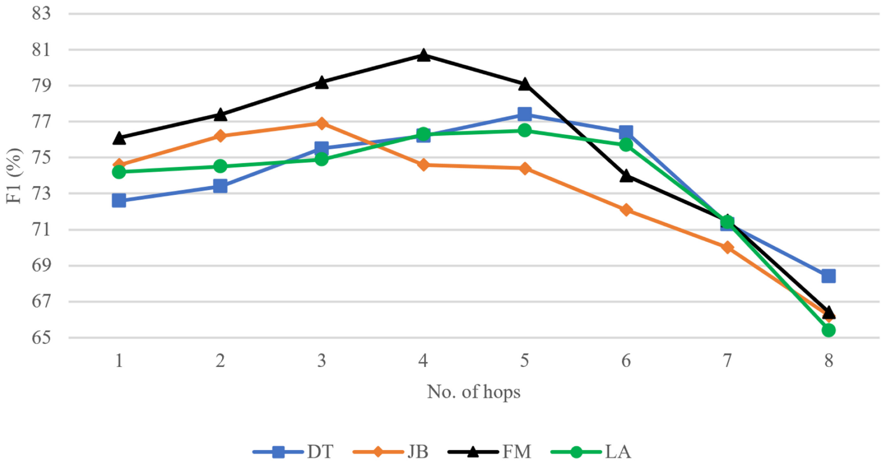


Fig. 4. The experimental results ($F1_{avg}$) with respect to the varying number of hops in verbalizer construction.

5.6.2 Impact of Number of hops in Verbalizer Construction. The number of hops is an important hyper-parameter of Verbalizer construction, since it helps to select the label words for the initial construction. In this article, we would like to investigate its impact on the proposed KE-prompt. Specifically, we report its performance on 4 targets by increasing the number of hops from 1 to 8. F1 results are the average value over 3 runs with random initialization. Figure 4 shows the results. We observe that KEprompt can obtain the best performance within 5 hops. After 6 iterations, the performance tends to decline steadily. We observe that as the number of hops increases, the overlap between label words of each category increases significantly. This makes it difficult for the model to learn the differences between categories, resulting in rapid performance degradation.

6 CONCLUSION

In this article, we propose a knowledge-enhanced automatic verbalizer (KEprompt) method for stance detection, which expands the verbalizer in prompt-tuning using external semantic knowledge and infusing background knowledge. In addition, we annotate a new dataset ISD for stance detection on implicit attitude sentiment expression. The experimental results demonstrated that the KEprompt model significantly outperformed the state-of-the-art methods for stance detection. In future work, we plan to mine the implicit attitude from creative spellings, jargon, and URLs. Furthermore, we may also devote our effort to exploiting the human reading cognitive process in stance detection, which helps us comprehend and understand the text in depth.

REFERENCES

[1] Emily Allaway and Kathleen Mckeown. 2020. Zero-shot stance detection: A dataset and model using generalized topic representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* 8913–8931.

[2] Emily Allaway and Kathleen R. McKeown. 2020. Zero-shot stance detection: A dataset and model using generalized topic representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 8913–8931.

[3] I. Augenstein, T. Rocktaeschel, A. Vlachos, and K. Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[4] Erik Cambria, Qian Liu, Sergio Decherchi, Frank Xing, and Kenneth Kwok. 2022. SenticNet 7: A commonsense-based neurosymbolic AI framework for explainable sentiment analysis. In *Proceedings of the International Conference on Language Resources and Evaluation*.

[5] Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. 2018. SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *Proceeedings of the 32nd AAAI Conference on Artificial Intelligence*.

[6] Alessandra Teresa Cignarella, Cristina Bosco, and Paolo Rosso. 2022. Do dependency relations help in the task of stance detection? In *Proceedings of the 3rd Workshop on Insights from Negative Results in NLP*. Association for Computational Linguistics, 10–17.

[7] Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2020. Will-they-won't-they: A very large dataset for stance detection on Twitter. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1715–1724.

[8] Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2021. Synthetic examples improve cross-target generalization: A study on stance detection on a Twitter corpus. In *Proceedings of the 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, 181–187.

[9] Yuhao Dan, Jie Zhou, Qin Chen, Qingchun Bai, and Liang He. 2022. Enhancing class understanding via prompt-tuning for zero-shot text classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 4303–4307.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 4171–4186.

[11] Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. 2018. Topical stance detection for Twitter: A two-phase LSTM model using attention. In *Proceedings of the European Conference on Information Retrieval*. Springer, 529–536.

[12] Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention networks. In *Proceedings of the International Joint Conferences on Artificial Intelligence*.

[13] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 3816–3830.

[14] Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2022. Few-shot cross-lingual stance detection with sentiment-based pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 10729–10737.

[15] Zihao He, Negar Mokhberian, and Kristina Lerman. 2022. Infusing Wikipedia knowledge to enhance stance detection. *arXiv preprint arXiv:2204.03839* (2022).

[16] Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035* (2021).

[17] Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2225–2240.

[18] Binxuan Huang, Yanglan Ou, and Kathleen M. Carley. 2018. Aspect level sentiment classification with attention-over-attention neural networks. In *Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 197–206.

[19] Rachna Jain, Deepak Kumar Jain, Dharana, and Nitika Sharma. 2022. Fake news classification: A quantitative research description. *ACM Trans. Asian Low Resour. Lang. Inf. Process.* 21, 1 (2022), 3:1–3:17.

[20] Yan Jiang, Jinhua Gao, Huawei Shen, and Xueqi Cheng. 2022. 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*. 837–847.

[21] Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Comput. Surv.* 53, 1 (2020), 1–37.

[22] Chengxi Li, Feiyu Gao, Jiajun Bu, Lu Xu, Xiang Chen, Yu Gu, Zirui Shao, Qi Zheng, Ningyu Zhang, Yongpan Wang et al. 2021. Sentiprompt: Sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. *arXiv preprint arXiv:2109.08306* (2021).

[23] Chen Li, Hao Peng, Jianxin Li, Lichao Sun, Lingjuan Lyu, Lihong Wang, Philip S. Yu, and Lifang He. 2022. Joint stance and rumor detection in hierarchical heterogeneous graph. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 6 (2022), 2530–2542.

[24] Yingjie Li and Cornelia Caragea. 2019. Multi-task stance detection with sentiment and stance lexicons. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 6299–6305.

[25] Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021. P-stance: A large dataset for stance detection in political domain. In *Findings of the Association for Computational Linguistics*.

[26] Bin Liang, Zixiao Chen, Lin Gui, Yulan He, Min Yang, and Ruifeng Xu. 2022. Zero-shot stance detection via contrastive learning. In *Proceedings of the ACM Web Conference*. 2738–2747.

[27] Bin Liang, Yonghao Fu, Lin Gui, Min Yang, Jiachen Du, Yulan He, and Ruifeng Xu. 2021. Target-adaptive graph for cross-target stance detection. In *Proceedings of the Web Conference*. 3453–3464.

[28] Rui Liu, Zheng Lin, Yutong Tan, and Weiping Wang. 2021. Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In *Findings of the Association for Computational Linguistics*. 3152–3157.

[29] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiao-Dan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation*. 31–41.

[30] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2019. STANCY: Stance classification based on consistency cues. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 6413–6418. DOI: https://doi.org/10.18653/v1/D19-1675

[31] Sujata Rani and Parteek Kumar. 2022. Aspect-based sentiment analysis using dependency parsing. *ACM Trans. Asian Low Resour. Lang. Inf. Process.* 21, 3 (2022), 56:1–56:19.

[32] Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*. 5569–5578.

[33] Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 255–269.

[34] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 4222–4235.

[35] Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2017. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. 551–557.

[36] Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, 226–234.

[37] Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018. Stance detection with hierarchical attention network. In *Proceedings of the 27th International Conference on Computational Linguistics*. 2399–2409.

[38] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[39] Marilyn A. Walker, Pranav Anand, Robert Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 592–596.

[40] Limin Wang and Dexin Wang. 2021. Solving stance detection on tweets as multi-domain and multi-task text classification. *IEEE Access* 9 (2021), 157780–157789.

[41] Penghui Wei, Junjie Lin, and Wenji Mao. 2018. Multi-target stance detection via a dynamic memory-augmented network. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 1229–1232.

[42] Penghui Wei and Wenji Mao. 2019. Modeling transferable topics for cross-target stance detection. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1173–1176.

[43] Chang Xu, Cecile Paris, Surya Nepal, and Ross Sparks. 2018. Cross-target stance classification with self-attention networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 778–783.

[44] Min Yang, Qingnan Jiang, Ying Shen, Qingyao Wu, Zhou Zhao, and Wei Zhou. 2019. Hierarchical human-like strategy for aspect-level sentiment classification with sentiment linguistic knowledge and reinforcement learning. *Neural Netw.* 117 (2019), 240–248.

[45] Min Yang, Wei Zhao, Lei Chen, Qiang Qu, Zhou Zhao, and Ying Shen. 2019. Investigating the transferring capability of capsule networks for text classification. *Neural Netw.* 118 (2019), 247–261.

[46] Bowen Zhang, Min Yang, Xutao Li, Yunming Ye, Xiaofei Xu, and Kuai Dai. 2020. Enhancing cross-target stance detection with transferable semantic-emotion knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3188–3197.

[47] Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. *arXiv preprint arXiv:1909.03477* (2019).

[48] Xin Zhang, Jianhua Yuan, Yanyan Zhao, and Bing Qin. 2021. Knowledge enhanced target-aware stance detection on tweets. In *Proceedings of the China Conference on Knowledge Graph and Semantic Computing*. Springer, 171–184.

[49] Yazhou Zhang, Prayag Tiwari, Dawei Song, Xiaoliu Mao, Panpan Wang, Xiang Li, and Hari Mohan Pandey. 2021. Learning interaction dynamics with an interactive LSTM for conversational sentiment analysis. *Neural Netw.* 133 (2021), 40–56.

Received 8 October 2022; revised 8 January 2023; accepted 13 March 2023