Title
A Unified Framework for Unseen Target Stance Detection based on Feature Enhancement via Graph Contrastive Learning

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 45(45)

Authors
Zhao, Xuechen
Zou, Jiaying
tian, lei
et al.

Publication Date
2023

A Unified Framework for Unseen Target Stance Detection based on Feature Enhancement via Graph Contrastive Learning

Xuechen Zhao[1,3]**, Jiaying Zou**[1]**, Lei Tian**[1]**, Feng Xie**[1]**, Haiyang Wang**[1]**,**
HongZhou Wu[1]**, Bin Zhou**[1,2,✉]**, Jie Tian**[3]

[1] School of Computer, National University of Defense Technology, Changsha, China
[2] Key Lab. of Software Engineering for Complex Systems, National University of Defence Technology, Changsha, China
[3] School of Data and Computer Science, Shandong Womens University, Jinan, China
{zhaoxuechen, zoujiaying20, leitian129, xiefeng, wanghaiyang19, whz, binzhou}@nudt.edu.cn, 29052@sdwu.edu.cn

Abstract

Stance detection for unseen targets is designed to automatically identify the user's stance or attitude towards various new targets that are constantly appearing with no labels. Inspired by work in cognitive science, we distinguish functions between systems for syntactic and semantic to enhance stance detection. First, we construct a dual-view graph and utilize unsupervised graph contrastive learning to capture target-invariant features influencing stance expression from a syntactic structure perspective. Second, we use an attention mechanism to learn the relationship between syntactic pattern features and a given target, and fuse the two parts to enhance the model's ability to predict unseen targets. Meanwhile, we employ the interactive GCN to maintain the global semantics of the dual-view graph fusion and ensure the stability and validity of the learned syntactic representations. Comprehensive experiments on stance detection of unseen targets verify the effectiveness and superiority of our proposed method.

Keywords: Stance Detection; Dual-View Graph; Contrastive Learning

Introduction

Stance detection aims to automatically identify attitudes or stances (i.e., pro, con, or neutral, etc.) in texts for a specific target. Traditional stance detection training and inference are all on the same target, which requires a lot of labeled data (Wei, Mao, & Chen, 2019). With the development of social media, the data on the internet is exploding, and people are discussing a wide range of targets, many of which have yet to be seen before and for which there is no labeled data (Mohammad, Kiritchenko, Sobhani, Zhu, & Cherry, 2016). This problem corresponds to two tasks: zero-shot stance detection and cross-target stance detection, which we collectively refer to as unseen target stance detection (UTSD).

The key to solving UTSD problems is learning transferable, target-invariant knowledge from the labeled seen target data. Some existing approaches try to improve the model's predictive ability for unseen targets by employing attention mechanisms (Xu, Paris, Nepal, & Sparks, 2018a) or fusing external knowledge (Liu, Lin, Tan, & Wang, 2021a). However, due to the coupling of target-specific features, the prediction effect of transferring knowledge directly from a specific target to unseen targets is usually limited. (Allaway, Srikanth, & Mckeown, 2021) uses adversarial learning to guide the model to learn the feature distribution of unseen targets, and this model is less effective in the case of unbalanced label distribution. (Liang et al., 2021) refers to shallow pragmatics features such as word frequency statistics in unseen target data to achieve feature sharing among different targets; (Liang et al., 2022) performs stance detection through hierarchical supervised contrastive learning. They earn better classification results, while more natural general target-invariant features, such as syntactic patterns, are underutilized.

Both linguistics and psychology consider that language can be divided into "syntactic representation" and "semantic representation". Text semantics is the unity of syntactic and semantic representations, which interact with each other to complete the process of sentence comprehension and expression. Even sentences oriented to different targets may have the same or similar syntactic patterns, which is a natural target-invariant textual feature that plays an essential role in stance expression. Inspired by this, we propose a Unified Framework for Unseen Target Stance Detection based on Feature Enhancement via Graph Contrastive Learning (FEGCL). Our method integrates unsupervised contrastive learning and graph neural network techniques to model the syntactic patterns of texts and employs such syntactic pattern features to enhance model prediction and thus improve the performance of UTSD task.

The main contributions of this paper are summarized as follows:

- We unify the stance detection task for unseen targets and propose a novel end-to-end unification framework. The model constructs a dual-view graph focusing on "syntactic representation" and "semantic representation" by masking keywords and builds a bridge for knowledge transfer between seen and unseen targets (i.e., syntactic pattern features) through unsupervised graph contrastive learning.
- Based on GCN for interactive convolution of the dual-view graph, we fuse syntactic presentation and semantic presentation and keep the fused features consistent with the global semantics, thus ensuring the effectiveness of the acquired features. Further, our model employs target-invariant syntactic pattern features to enhance global semantic features and facilitate the generalization of stance features from seen to unseen targets.
- Extensive experiments on typical benchmark datasets show that our model performs well on zero-shot stance detection. We even apply the model to the cross-target stance

In M. Goldwater, F. K. Anggoro, B. K. Hayes, & D. C. Ong (Eds.), *Proceedings of the 45th Annual Conference of the Cognitive Science Society*.

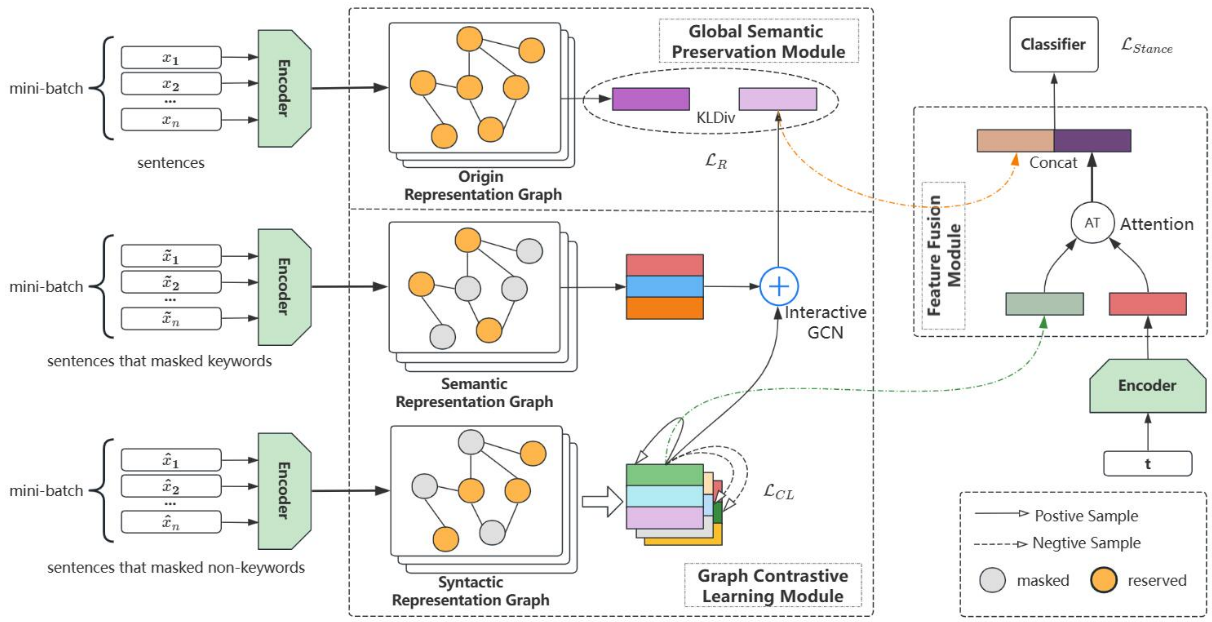


Figure 1: The framework of the proposed FEGCL model.

detection tasks, demonstrating our model's superiority and generalizability.

Task Description

The labeled "seen target dataset" $D_s = \{u_i^s = (t_i^s, x_i^s, y_i)\}|_{i=1}^{N_s}$ is the training set, where u_i^s is the i-th sample in the training set, which consists of the utterance x_i^s towards the target t_i^s and stance label $y_i \in \{pro, con, neutral\}$. The "unseen targets dataset" $D_d = \{u_i^d = (t_i^d, x_i^d)\}|_{i=1}^{N_d}$ is the testing set, where the target has not been seen in the training set, and u_i^d is the i-th sample in the testing set, meaning the utterance x_i^d towards the target t_i^d. Our goal is to train on seen targets dataset and generalize the model's inference ability to unseen targets. In particular, the cross-target stance detection is trained on a training set with seen targets and generalizes the inference capability to a testing set with unseen but relevant targets(Liang et al., 2021). Zero-shot stance detection aims to learn from the training set of many targets with labels and automatically identify previously unseen targets that have little relevance to the training targets(Allaway & Mckeown, 2020).

Methodology

In this section, We introduce the proposed unified Framework for Unseen Target Stance Detection(UTSD) based on Feature Enhancement via Graph Contrastive Learning. As demonstrated in Figure 1, FEGCL consists of four main components: 1)graph contrastive learning module, which is based on the dual-view graph to separate and learn target-invariant features as shared features for migration from seen to unseen targets; 2)global semantic preservation module, which is based on interactive GCN to avoid distortion of the learned syntactic pattern features; 3)Feature fusion module, which fuses syntactic pattern features with specific targets based on an attention mechanism; and 4)stance classifier.

Encoder

Given a sentence x towards the specific target t, where $t = \{w_i\}|_{i=1}^{m}$ consists of m words and $x = \{w_i\}|_{i=1}^{n}$ consists of n words[1]. To take full advantage of contextual information, we adopt BERT as the encoder $f_\theta(\cdot)$.

$$z, \mathbf{X} = f_\theta(input) = \mathrm{BERT}(input) \tag{1}$$

where the input can be a single x, t or (x, t), a combination of the two. $\mathbf{X} = [v_1, v_2, \ldots, v_{|input|}] \in \mathbb{R}^{|input| \times d_m}$ is the feature matrix from the last hidden layer of the encoder outputs, and $|input|$ is the length of the input. v_i is the last hidden layer vector corresponding to the i-th input token. d_m is the dimension of the hidden layer feature representation. $z \in \mathbb{R}^{d_m}$ stands for the semantic aggregate representation of the input, that is, the representation of the special token [CLS] in BERT.

Contrastive Learning based on Dual-view Graph

Dual-View Graphs Construction The expression of text semantics results from the coupling of multiple meta-features (Cope, Beaver, & Fintel, 2013), and it is crucial for UTSD task to obtain the target-invariant and shared features. Here, we focus on the syntactic patterns that affect the semantic expression of sentences, which is naturally target-invariant and manifests as a graph structure. Therefore, we decompose the sentence into two views, the syntactic representation graph

[1] In this paper, we will use SpaCy toolkit to generate the dependency tree of a input sentence. In order to ensure the alignment of SpaCy and Bert, we will take the word segmentation result of SpaCy as the input of Bert.

and the semantic representation graph, to represent the text features from the perspective of target-invariant and target-specific features, respectively.

First, we construct graphs for each sentence based on syntactic dependency trees[2] to capture the dependencies between words in the sentences. The $\mathbf{A} = \mathbb{R}^{n\times n}$ for the adjacency matrix of each sentence can be expressed as:

$$\mathbf{A}_{i,j} = \begin{cases} 1, & x(w_i, w_j) \\ 0, & otherwise \end{cases} \tag{2}$$

where $\mathbf{A}_{i,j}$ represents adjacency matrix between i-th row and j-th column , and $x(w_i, w_j)$ indicates that the word w_i in the syntactic dependency tree of the sentence x is connected to the word w_j with edge. Here, for simplicity, we define the adjacency matrix as a symmetric matrix (i.e., $\mathbf{A}_{i,j} = \mathbf{A}_{j,i}$). Following (Welling & Kipf, 2016), we set that each word node has a self-looping edge (i.e., $\mathbf{A}_{i,i} = 1$).

Secondly, since text syntax does not depend on keywords or topic words related to any target, to focus on the syntactic patterns, we use KeyBert (Grootendorst, 2020) to extract the keywords in the sentence x and mask the keywords with the particular token [MASK] to obtain the auxiliary sentence $\tilde{x}$. At the same time, we reverse the mask to keep the keywords of the sentence, while the non-keywords are masked with a special token [MASK] to obtain the other auxiliary sentence $\hat{x}$, aiming to focus more on the expression of the target-specific content. Based on the above data preprocessing operations, our training set and testing set are adjusted as $D_s = \{u_i^s = (t_i^s, x_i^s, \tilde{x}_i^s, \hat{x}_i^s, y_i)\}|_{i=1}^{N_s}$ and $D_d = \{u_i^d = (t_i^d, x_i^d, \tilde{x}_i^d, \hat{x}_i^d)\}|_{i=1}^{N_d}$.

Finally, the sentences masking keywords $\tilde{x}$ and the adjacency $\mathbf{A}$ form the "syntactic representation graph". The sentences masking non-keywords $\hat{x}$ and the adjacency matrix $\mathbf{A}$ form the "semantic representation graph". Each word of a sentence is a node in the graph whose features are the word-level feature of the last layer by encoding.

$$\begin{aligned} \tilde{\mathbf{X}}^{(\mathbf{0})} &= [\tilde{v}_0, \tilde{v}_1, \dots, \tilde{v}_n] = f_\theta(\tilde{x})[1] \\ \hat{\mathbf{X}}^{(\mathbf{0})} &= [\hat{v}_0, \hat{v}_1, \dots, \hat{v}_n] = f_\theta(\hat{x})[1] \end{aligned} \tag{3}$$

where $\tilde{\mathbf{X}}^{(\mathbf{0})} \in \mathbb{R}^{n\times d_m}, \hat{\mathbf{X}}^{(\mathbf{0})} \in \mathbb{R}^{n\times d_m}$ are the initial feature matrices of the "syntactic representation graph" and "semantic representation graph", respectively, and $\tilde{v}_i, \hat{v}_i$ correspond to the feature vectors of the nodes in the two views, respectively.

Syntactic Representation Based on Graph Contrastive Learning The "syntactic representation graph" helps us to learn target-invariant syntactic pattern features that are naturally target-invariant and transferable. Acquiring syntactic pattern features with good discriminative ability is the key to affecting stance classification. Since contrastive learning has the advantage of distinguishing ability in feature space and has been effective in many fields, we formulate the process of learning based on "syntactic representation graph" as a self-supervised graph contrastive learning problem. Further, we construct positive and negative examples of contrastive learning from node attribute-level enhancement and network topology-level enhancement.

Data augmentation 1) node-attribute-level augmentation. (Gao, Yao, & Chen, 2021) considers that any deletion or alteration of a word will harm the data augmentation performance and takes dropout as the data augmentation of the minimal form of text representation. So, we feed a sentence into the encoder twice, and the meaning of the two representations obtained is identical, which can be considered as a pair of positive samples. In contrast, the other samples in the same batch are taken as negative samples. Specifically, given the encoder $f_\theta(\cdot, m)$ with dropout mask m, we encode the sentence x twice with different dropout masks m, m' and convey positive sample pairs $\{\tilde{\mathbf{X}}, \tilde{\mathbf{X}}'\}$. 2) network-topology-level augmentation. To adapt to the diversity and irregular language expressions of social media, for each sample, we randomly remove 1% of the edges in the syntactic dependency tree to enhance the graph and obtain the positive sample pairs of the adjacency matrix $\{\mathbf{A}, \mathbf{A}'\}$.

Given an mini-batch input $\mathcal{B} = \{u_i\}_{i=1}^{N_b}$, where N_b is the size of the mini-batch, after data augmentation, the mini-batch for contrastive learning is doubled to $\mathcal{B}'$, which size is $2N_b$. For each sample expressed in the feature matrix and adjacency matrix, we refer to $\{(\tilde{\mathbf{X}}, \mathbf{A}), (\tilde{\mathbf{X}}', \mathbf{A}')\}$ as a positive sample pair. In contrast, the other $2N_b - 2$ samples in $\mathcal{B}'$ are negative samples about the positive sample pair.

Syntactic pattern representation We feed the feature matrix $\tilde{\mathbf{X}}$ and the normalized adjacency matrix $\hat{\mathbf{A}}$ of the "syntactic representation graph" into the GCN (Graph Convolutional Network) (Welling & Kipf, 2016) to learn the target-invariant syntactic feature matrix in the context.

$$\tilde{\mathbf{X}}^{(l+1)} = \mathbf{GCN}(\tilde{\mathbf{X}}^{(l)}; \mathbf{A}) = \sigma(\hat{\mathbf{A}}\tilde{\mathbf{X}}^{(l)}\mathbf{W}^{(l)}) \tag{4}$$

where $\tilde{\mathbf{X}}^{(l)}$ is the syntactic feature matrix of the l layer convolution. $\hat{\mathbf{A}} = \mathbf{D}^{(-\frac{1}{2})}\mathbf{A}\mathbf{D}^{(-\frac{1}{2})}$, $\mathbf{D}$ is the degree matrix, $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. $\sigma(\cdot)$ is ReLU nonlinear activation function. $\mathbf{W}^{(l)} \in \mathbb{R}^{d_m\times d_m}$ are trainable parameters.

Further, the target t is encoded as $\tilde{z} = f_\theta(t)[0]$, and we use a retrieval-based attention mechanism(C. Zhang, Li, & Song, 2019) to learn the relationship between syntactic feature and the specific target.

$$\begin{aligned} \alpha_j &= \frac{\exp(\beta_j)}{\sum_{k=1}^{n}\exp(\beta_t)} \\ \beta_j &= (\mathbf{W}_q\tilde{z})^T(\mathbf{W}_k\tilde{X}_j) \end{aligned} \tag{5}$$

where α_j is the attention of j-th node feature in syntactic feature matrix to the $\tilde{z}$, T represents the transpose operation of the vector, and $\mathrm{W}_q, \mathrm{W}_k \in \mathbb{R}^{d_s\times d_m}$ are the learnable parameters. Further, we compute the fusion features that represent the syntactic expression patterns.

$$\tilde{f} = \mathbf{ATN}(t, \tilde{\mathbf{X}}) = \sum_{k=1}^{n}\alpha_k(\mathbf{W}_v\tilde{X}_k) \tag{6}$$

[2]In this work, we use spaCy toolkit for generating dependency tree of the input sentence: https://spacy.io/.

where $W_v \in \mathbb{R}^{d_s \times d_m}$ is the learnable parameter and $\tilde{f} \in \mathbb{R}^{d_s}$.

Contrastive Learning We define a neural network projection head $h = g_\psi(f) = W^{(2)}\sigma(W^{(1)}f)$, which maps the feature vectors to the space for calculating the loss of contrastive learning, where $W^{(1)}$ and $W^{(2)}$ are trainable parameters. For each sample, the contrastive loss is expressed as:

$$\ell_i = -log\frac{e^{sim(\tilde{h}_i,\tilde{h}'_i)/\tau}}{\sum_{j=1}^{2N_b}\mathbb{1}_{j\neq i}\cdot e^{sim(\tilde{h}_i,\tilde{h}'_j)/\tau}} \tag{7}$$

where $\mathbb{1}_{j\neq i}$ is an indicator function, $sim(u,v) = u^T v/\|u\|\|v\|$ indicates the cosine similarity of vectors u and v normalized by L_2, τ is a temperature parameter that controls the penalty intensity for hard samples in contrastive learning. Thus, the contrastive learning loss for each mini-batch is:

$$\mathcal{L}_{CL} = \frac{-1}{2N_b}\sum_{i\in\mathcal{B}'}\ell_i \tag{8}$$

Global Semantic Preservation based on Interactive GCN

Global semantic features are essentially the result of some fusion of target-invariant features and target-specific features, which correspond to the "syntactic representation graph" and "semantic representation graph" in FEGCL, respectively. To avoid the prediction instability caused by feature distortion, inspired by (Liang et al., 2021), we utilize GCN-based dual-view interactive fusion to reconstruct the global semantic features and ensure that the reconstructed feature distribution is consistent with the original feature distribution. Here, we take $z = f_\theta(x)[0]$ as the original global semantic feature, and the interaction fusion features are defined as:

$$\begin{aligned}
\tilde{\mathbf{X}}^{(l+1)} &= \mathbf{GCN}_s(\tilde{\mathbf{X}}^{(l)}, \mathbf{A}) \\
\tilde{\mathbf{Z}}^{(l+1)} &= \mathbf{GCN}_t(\tilde{\mathbf{X}}^{(l+1)}, \mathbf{A}) \\
\hat{\mathbf{X}}^{(l+1)} &= \mathbf{GCN}_t(\hat{\mathbf{X}}^{(l)}, \mathbf{A}) \\
\hat{\mathbf{Z}}^{(l+1)} &= \mathbf{GCN}_s(\hat{\mathbf{X}}^{(l+1)}, \mathbf{A})
\end{aligned} \tag{9}$$

where $\mathbf{GCN}_s, \mathbf{GCN}_t$ are the graph convolution modules for "syntactic representation graph" and "semantic representation graph", respectively. $\tilde{\mathbf{Z}}^{(l+1)}$ is the result of merging syntactic pattern representation into semantic representation, and $\hat{\mathbf{Z}}^{(l+1)}$ is the result of merging semantic representation into syntactic pattern representation. Additionally, we read out the reconstructed feature by average pooling, defined as:

$$z' = \mathbf{Mean}_{pool}(\frac{\tilde{\mathbf{Z}}^{(l+1)} + \hat{\mathbf{Z}}^{(l+1)}}{2}) \tag{10}$$

We maintain the distribution consistency between the original global semantic features and the reconstructed semantic features based on the KL divergence.

$$\ell_i^R = \mathbf{KL}[p(z_i)||p(z'_i)] \tag{11}$$

Thus, the learning objective for global semantic preservation is:

$$\mathcal{L}_R = \frac{1}{N_b}\sum_{i=1}^{N_b}\ell_i^R \tag{12}$$

Stance Classifier

To give full play to the role of target-invariant and target-specific features in stance classification, we enhance the original sentence feature z with the syntactic pattern feature $\tilde{f}$, which is easy to transfer from different targets. Furthermore, we use a fully-connected layer with softmax normalization to generate the probability distribution of stance detection:

$$\hat{y}_i = \text{softmax}(\mathbf{W}_o(\tilde{f}_i \oplus z_i) + b_o) \tag{13}$$

where $\hat{y}_i \in \mathbb{R}^{d_p}$ is predicted stance probability distribution for the input sample u_i, d_p is the dimension of stance labels, $\mathbf{W}_o \in \mathbb{R}^{d_p\times(d_m+d_s)}$ and $b_o \in \mathbb{R}^{b_p}$ are learnable parameters, $\oplus$ represents the concatenation.

Finally, we train the classifier by cross-entropy loss between the predicted distribution $\hat{y}$ and the ground-truth distribution y of each sample in mini-batch:

$$\mathcal{L}_{cls} = -\sum_{i=1}^{N_b}\sum_{j=1}^{d_p} y_i^j log\hat{y}_i^j \tag{14}$$

Table 1: The data statistics of VAST dataset.

VAST	Train	Dev	Test
#Samples	13477	2062	3006
#Unique Comments	1845	682	786
# Distinct Topics	4641	497	759
#Zero-shot Topics	4003	383	600

Table 2: The data statistics of WT-WT dataset.

DataSet	Target	Favor	Against	Neutral
WT-WT	CA	2469	518	5520
	CE	773	253	947
	AC	970	1969	3098
	AH	1038	1106	2804

Learning Objective

The learning objective is to train the model by jointly optimizing the supervised stance classification loss $\mathcal{L}_{cls}$, the self-supervised graph contrastive learning loss $\mathcal{L}_{CL}$ and the global semantic perservation loss $\mathcal{L}_R$. The overall target $\mathcal{L}$ can be formulated as the sum of three losses:

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{CL} + \beta\mathcal{L}_R + \lambda\|\Theta\|^2 \tag{15}$$

where α, β are tunable hyper-parameters, Θ denotes all trainable parameters in the model, and λ denotes the L2-regularization coefficient.

Table 3: Performance comparison of unseen targets stance detection. **Bold face** indicates the best result of each column and underlined the second-best. The results with $*$ represent our implementation.

Models	WT-WT(Cross-target)				VAST(Zero-shot)			
	AC	AH	CA	CE	Pro	Con	Neu	All
BiCond	64.9	63.0	56.5	52.5	44.6	47.4	34.9	42.8
CrossNet	65.1	62.3	59.1	54.4	46.2	43.4	40.4	43.4
SEKT	-	-	-	-	50.4	44.2	30.8	41.8
TOAD	59.2*	62.0*	58.1*	57.8*	42.6	36.7	43.8	41.0
BERT	67.1	67.3	56.0	60.5	54.6	58.4	85.3	66.1
BERT-GCN	-	-	-	-	58.3	60.6	86.9	68.6
BERT-DAN	72.2*	74.1*	<u>73.5</u>*	<u>70.4</u>*	<u>60.6</u>*	58.4*	89.3*	69.4*
TPDG	74.2	73.1	66.8	65.6	-	-	-	-
PT-HCL	**76.7**	<u>76.3</u>	73.1	69.2	**61.7**	<u>63.5</u>	**89.6**	<u>71.6</u>
FEGCL(ours)	<u>75.4</u>	**77.5**	**74.3**	**73.9**	59.8	**65.8**	<u>89.4</u>	**71.7**

Experiment

Datasets and Evaluation Metrics

To verify the effectiveness of our model on unseen targets, we used VAST (Allaway & Mckeown, 2020) dataset for the zero-sample stance detection and WT-WT (Conforti et al., 2020) dataset for cross-target stance detection, respectively.

VAST is a public dataset for zero-shot stance detection, which is composed of a large number of targets. The statistics of the dataset are shown in Table 1. We compute the Macro F1-Score for each label to measure the model's performance.

WT-WT is a financial dataset that is used to detect the attitudes of M&A operations among companies. There are four targets in WT-WT, namely CVS_AET (CA), CI_ESRX (CE), ANTM_CI (AC), and AET_HUM (AH). The statistics of the dataset are shown in Table 2. We use the average of the Favor and Against Macro F1-Scores to evaluate the performance of our model. Following (Conforti et al., 2020), we adopt an evaluation setup of leave-one-target-out and randomly divide the seen target data into the training set and validation set according to the ratio of 85:15.

Baselines

We select several baselines with good performance to compare with FEGCL. These include BiLSTM-based models such as BiCond (Augenstein, Rocktäschel, Vlachos, & Bontcheva, 2016), CrossNet (Xu, Paris, Nepal, & Sparks, 2018b), and SEKT (B. Zhang et al., 2020); a model based on adversarial learning: TOAD (Allaway et al., 2021); a model based on graph neural network: TPDG (Liang et al., 2021); BERT-based models: BERT (Devlin, Chang, Lee, & Toutanova, 2019), BERT-GCN (Liu, Lin, Tan, & Wang, 2021b) and BERT-DAN (the BERT version of DAN(Xu et al., 2020)); contrastive learning based model: PT-HCL (Liang et al., 2022).

Implementation and Reproducibility

All programs are implemented using python 3.9.13 and pytorch 1.12.1 with CUDA 11.8 in an Ubuntu 20.04.5 with an nvidia 3090 GPU.

For data augmentation, we adopt keybert (Grootendorst, 2020) to extract the keywords of texts and select 30% of all keywords to mask for every sentence. We use a pre-trained uncased bert-base (Devlin et al., 2019) as an encoder with 768-dimensional embedding. We train our model for 15 epochs, using the Adam (Kingma & Ba, 2015) optimizer with a learning rate of 2e-5. The mini-batch size is set to 32. The hyper-parameters for contrastive learning is $\tau = 0.14$ and dropout probability $p = 0.3$. The hyper-parameters combination for the total loss is $\alpha = 0.2$ and $\beta = 0.5$.

Experimental Results

Main Experimental Results

Table 3 shows the experimental results of our model and baselines for cross-target and zero-shot tasks. FEGCL results in all scenarios are significantly better than the baselines.

For the cross-target stance detection on WT-WT dataset, we can see that TPDG has achieved good results due to the use of graph structure to mine some pragmatic information. Our model improves by 4.7% in F1-Score compared to the PT-HCL model based on contrastive learning, showing that our model based on the graph contrastive learning can effectively extract high-quality target-invariant features and thus improve the ability of cross-target stance detection. Finally, Our model achieves a better performance in most cases.

We also validate the zero-shot stance detection in the VAST dataset. We can see that the BERT-based models has better results compared with the traditional LSTM-based model, which indicates that the BERT is able to capture more valuable semantic information from datasets. Further, models such as PT-HCL, which introduce contrastive learning, further improve the performance of stance detection. And FEGCL model achieves the best results in most of the results, which shows that our dual-view graph-based contrastive learning can effectively model target-invariant syntactic features, thus promoting feature transfer between seen and unseen targets.

Table 4: Experimental results of ablation study. A check indicates that the model uses the appropriate module. When MKeys is unchecked, the words in the sentence are masked by 10% at random.

MKeys	CL	Recon	Cross-target				Zero-shot
			AC	AH	CA	CE	ALL
	✓	✓	68.5	70.0	62.8	61.2	68.2
✓		✓	73.9	75.8	71.1	70.6	69.2
✓	✓		74.8	76.2	73.8	73.8	70.2
✓	✓	✓	**75.4**	**77.5**	**74.3**	**73.9**	**71.7**

Ablation Study

We designed several variants of the EFGCL model to analyze the effect of each component of our method and different combinations through ablation experiments. **MKeys** denotes the module for masking keywords; when this module is not selected, it represents masking words randomly. **CL** indicates the graph contrastive learning module and **Recon** denotes the global semantic preservation module.

The results of ablation study are shown in table 4. We can see that both the removal of graph contrastive loss and the global semantic reconstruction loss significantly degrade the performance of the model. Therefore, both learning objectives are critical to our model. We find that models without masking keywords in a targeted way have a significant degradation in prediction performance, indicating that targeted masking target-specific keywords are crucial. In contrast, targeted mask keywords can shield target-specific information without destroying the syntactic structure, which is beneficial to learning target-invariant syntactic pattern features. In addition, removal of the graph contrastive learning module **CL** also leads to a decrease in prediction performance, which shows that the features obtained from contrastive learning can effectively enhance the model learning quality. Similarly, the global semantic reconstruction module **Recon** also affects the quality of feature learning, and removal still leads to a decrease in prediction performance.

Visualization Analysis

Analysis of Syntactic Representation To analyze how the contrastive learning module in our model plays a role in improving the quality of the learned representations. Taking AH in WTWT as an example, we randomly selected 500 instances from the training set and test set, respectively, and visualized the syntactic pattern representations by T-SNE. As shown in the figure 2, the distribution of "syntactic pattern representation " in the WT-WT dataset without AH (i.e., the training set) overlaps mostly with the feature representation of AH (i.e., the testing set), indicating that the syntactic pattern features learned in the training data can effectively cover the testing data and thus obtain a good feature migration effect.

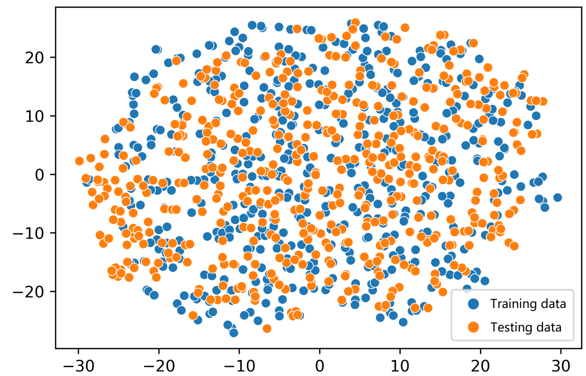
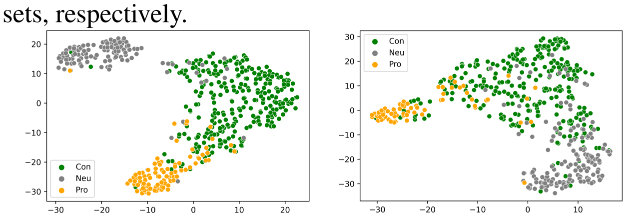


Figure 2: Visualization of syntactic representation learned by FEGCL. The blue and orange colors indicate the distribution of the syntactic pattern features of the training and testing sets, respectively.



(a) Training data. (b) Testing data.

Figure 3: Visualization of FEGCL classification ability in the training data (a) and testing data (b).

Analysis of Classification Again using the WTWT dataset as an example, we further demonstrate the classification capability of the proposed model. We randomly selected 500 samples from the training and test sets, respectively, and presented the t-SNE visualization of the classification presentation. As shown in Figure 3, for samples with different stances in the training set, our FEGCL can distinguish them well. For samples with different stances in the training set, our FEGCL can distinguish them well. FEGCL also achieves satisfactory classification results on the testing set, indicating that our model has good generalization ability.

Conclusion

In this paper, we propose a novel unified framework model for the unseen target stance detection, which models the syntactic and semantic representations affecting the textual pose representation by constructing a dual-view graph through a keyword masking approach, respectively. We further employ unsupervised graph contrastive learning to learn target-invariant features (i.e., syntactic pattern features) from syntactic representation graphs and fuse them with specific targets as augmented features based on an attention mechanism as the primary source of transferable knowledge. Meanwhile, global semantic feature maintenance based on interactive GCN ensures the validity of the learned features and promotes prediction stability. Experimental results of unseen targets show that our model significantly improves performance and achieves the best results, demonstrating the excellent generality of our model.

Acknowledgements

This work is supported by the Key R&D Program of Guangdong Province No.2019B010136003, the National Natural Science Foundation of China No.62172428, 62006143, 61732004, 17320222 and the Natural Science Foundation of Shandong Province of China No.ZR2020MF152.

References

Allaway, E., & Mckeown, K. (2020). Zero-shot stance detection: A dataset and model using generalized topic representations. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 8913–8931).

Allaway, E., Srikanth, M., & Mckeown, K. (2021). Adversarial learning for zero-shot stance detection on social media. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 4756–4767).

Augenstein, I., Rocktäschel, T., Vlachos, A., & Bontcheva, K. (2016). Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 876–885).

Conforti, C., Berndt, J., Pilehvar, M. T., Giannitsarou, C., Toxvaerd, F., & Collier, N. (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* (pp. 1715–1724).

Cope, J., Beaver, D., & Fintel, K. (2013, 01). Semantics and pragmatics. In (p. 333-351).

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019* (pp. 4171–4186).

Gao, T., Yao, X., & Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 6894–6910).

Grootendorst, M. (2020). *Keybert: Minimal keyword extraction with bert.* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.4461265` doi: 10.5281/zenodo.4461265

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd international conference on learning representations, ICLR 2015.*

Liang, B., Chen, Z., Gui, L., He, Y., Yang, M., & Xu, R. (2022). Zero-shot stance detection via contrastive learning. In *Proceedings of the acm web conference 2022* (pp. 2738–2747).

Liang, B., Fu, Y., Gui, L., Yang, M., Du, J., He, Y., & Xu, R. (2021). Target-adaptive graph for cross-target stance detection. In *Proceedings of the web conference 2021* (pp. 3453–3464).

Liu, R., Lin, Z., Tan, Y., & Wang, W. (2021a). Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 3152–3157).

Liu, R., Lin, Z., Tan, Y., & Wang, W. (2021b). Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 3152–3157).

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

Wei, P., Mao, W., & Chen, G. (2019). A topic-aware reinforced model for weakly supervised stance detection. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 7249–7256).

Welling, M., & Kipf, T. N. (2016). Semi-supervised classification with graph convolutional networks. In *J. international conference on learning representations (iclr 2017).*

Xu, C., Paris, C., Nepal, S., & Sparks, R. (2018a). Cross-target stance classification with self-attention networks. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 778–783).

Xu, C., Paris, C., Nepal, S., & Sparks, R. (2018b). Cross-target stance classification with self-attention networks. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 778–783).

Xu, C., Paris, C., Nepal, S., Sparks, R., Long, C., & Wang, Y. (2020). DAN: dual-view representation learning for adapting stance classifiers to new domains. In *ECAI 2020 - 24th european conference on artificial intelligence* (Vol. 325, pp. 2260–2267).

Zhang, B., Yang, M., Li, X., Ye, Y., Xu, X., & Dai, K. (2020). Enhancing cross-target stance detection with transferable semantic-emotion knowledge. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 3188–3197).

Zhang, C., Li, Q., & Song, D. (2019). Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 4568–4578).