Improving Multi-task Stance Detection with Multi-task Interaction Network

Heyan Chai[1], **Siyu Tang**[1], **Jinhao Cui**[1], **Ye Ding**[2], **Binxing Fang**[1,3] and **Qing Liao**[1,3]*
[1] Harbin Institute of Technology, Shenzhen, China
[2] Dongguan University of Technology, China
[3] Peng Cheng Laboratory, Shenzhen, China
{chaiheyan,tangsiyu999}@stu.hit.edu.cn,gcjh777@163.com
dingye@dgut.edu.cn,fangbx@cae.cn,liaoqing@hit.edu.cn

Abstract

Stance detection aims to identify people's standpoints expressed in the text towards a target, which can provide powerful information for various downstream tasks. Recent studies have proposed multi-task learning models that introduce sentiment information to boost stance detection. However, they neglect to explore capturing the fine-grained task-specific interaction between stance detection and sentiment tasks, thus degrading performance. To address this issue, this paper proposes a novel multi-task interaction network (MTIN) for improving the performance of stance detection and sentiment analysis tasks simultaneously. Specifically, we construct heterogeneous task-related graphs to automatically identify and adapt the roles that a word plays with respect to a specific task. Also, a multi-task interaction module is designed to capture the word-level interaction between tasks, so as to obtain richer task representations. Extensive experiments on two real-world datasets show that our proposed approach outperforms state-of-the-art methods in both stance detection and sentiment analysis tasks.

1 Introduction

Stance detection is a task that identifies the standpoint or attitude (*e.g. favor, against, or none, etc.*) of user towards various targets (*e.g. entity, event, opinion, people, etc.*) (Küçük and Can, 2020; Liang et al., 2021). Stance detection is crucial for machines to understand natural language and it has many practical application areas such as sarcasm detection (Agrawal et al., 2020; Tang et al., 2022), fake news detection (Liao et al., 2021a; Ma et al., 2022), relation extraction (Li et al., 2021b; Huang et al., 2021), etc. Some early studies attempt to extract the richer stance expressions by leveraging manually engineered features (Darwish et al., 2020; Kochkina et al., 2017; Zarrella and Marsh, 2016; Mohammad et al., 2016; Sen et al., 2018) or employing deep learning approaches (Wei et al., 2016; Augenstein et al., 2016; Zhou et al., 2019; Kawintiranon and Singh, 2021). However, limited annotated training tweets on social media platforms can not provide richer stance representations, leading to the degraded performance of models mentioned above.

	Tweet	Target	Stance	Sentiment
Example 1	Pregnant people have feelings, and the ability to make decisions about their health	Legalization of abortion	Favor	Positive
	They have not the ability and shouldn't make decisions that involve their health	Legalization of abortion	Favor	Negative
Example 2	I have an immune system that works fine, masks harm our immune system	Wearing a face mask	Against	Positive
	I have next to no immune system right now so thanks to all wearing masks	Wearing a face mask	Favor	Positive

Figure 1: Examples of MTL stance detection.

Motivated by recent advances in multi-task learning (MTL) (Ruder, 2017; Xu et al., 2020; Liao et al., 2021a), some studies jointly train sentiment analysis and stance detection task to alleviate the limited training data problem (Mohammad et al., 2017; Li and Caragea, 2019; Hosseinia et al., 2020). For example, Li and Caragea (2019) constructs a sentiment analysis task as an auxiliary task to help stance detection by introducing the external sentiment lexicon. Different from constructing auxiliary task, Sun et al. (2019) proposes a multi-task learning model that learns the representations of sentiment and stance simultaneously. Above MTL models only focus on extracting contextual task representations and sentence-based features as the shared information across different tasks to identify stances and classify sentiments of sentences, ignoring the word-level task-specific information. We argue that words may play different roles for different tasks. As shown in Example 1 in Figure 1, the identical words with colors in these two tweets express the same stances but opposite sentiment polarities. Likewise, in Example 2, the identical words express opposite stances but the same senti-

* Corresponding Author

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 2990–3000
December 7-11, 2022 ©2022 Association for Computational Linguistics

ment polarities. As such, it is desirable to leverage the task-related pragmatics relationship at the word level to capture the difference between tasks, with target of improving the performance of all tasks.

In this work, inspired by some graph neural network models proposed in other tasks (Ding et al., 2019; Zhang et al., 2020; Wu et al., 2021), we explore novel task-related pragmatics graphs to model the relationship between tasks at the word level. Based on it, we propose a novel Multi-Task Interaction Network (MTIN) framework with a word-level task interaction to automatically identify and adapt the roles that a word plays in different tasks, so as to obtain the richer task representation and thus improve the overall performance of all learning tasks. In summary, the main contributions of our work are three-fold:

- We construct novel stance-related and sentiment-related graphs (dubbed st-graph and se-graph respectively) based on task-related pragmatics weights of words.
- A novel multi-task interaction network is explored to capture the significant task-related information for each task from the word-level interaction between task-related graphs.
- Experimental results on a number of benchmark datasets verify the advantages of the proposed MTIN in both stance detection and sentiment analysis tasks.

2 Methodology

In this section, we describe our proposed MTIN in detail. As demonstrated in Figure. 2 , the architecture of the proposed MTIN contains three main components: 1) *input representation*, which derives the sentence representation of the input sentence-target pair with pre-trained language model, 2) *multi-task interaction module*, which constructs task-related graphs by using word-level pragmatics weights and captures the interaction between tasks, 3) *task-related attention*, which captures the richer stance and sentiment representations and outputs the final predicts.

2.1 Task Definition

Given a set of annotated sentences towards the corresponding targets $\mathcal{D}_s = \{(s_i, a_i, y_i^{st}, y_i^{se})\}_{i=1}^{N_s}$ for training set and $\mathcal{D}_q = \{(s_i, a_i)\}_{i=1}^{N_q}$ for testing set, where s_i and a_i are the sentence and corresponding target text respectively, y_i^{st} and y_i^{se} are

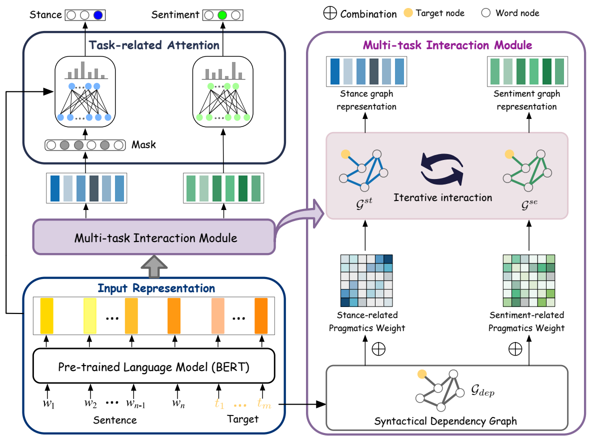


Figure 2: The architecture of the proposed MTIN. The left side shows the overall architecture of multi-task stance detection; and the right side shows the details of how to perform task interaction with multi-task interaction module.

the stance label and sentiment label of a i-th annotated instance, N_s and N_q are the numbers of training and testing samples, respectively. The aim of multi-task stance detection is to find a model $\mathcal{F}$ that predicts the stance label $\hat{y}_i^{st}$ and sentiment label $\hat{y}_i^{se}$ in $\mathcal{D}_q$ simultaneously, such that $\mathcal{F}(s_i, a_i) = \{\hat{y}_i^{se}, \hat{y}_i^{st}\} \approx \{y_i^{se}, y_i^{st}\}$. Note that, we use $t \in \{st, se\}$ to denote the stance task and sentiment task for simplicity.

2.2 Input Representation

For a given sentence consists of n words $s = \{w_i\}_{i=1}^{n}$ and the corresponding target a consisted of m words, $a = \{w_i\}_{i=1}^{m}$, n and m are the length of the text s and target a, respectively. In this paper, we use pre-trained language model BERT (Devlin et al., 2019) as the encoder to encode both the sentence s and the target a. Specifically, we feed "$[CLS]s[SEP]a[SEP]$" as input into the BERT to acquire a d_m-dimensional hidden representation $\boldsymbol{h} \in \mathbb{R}^{(n+m) \times d_m}$ of each input pair:

$$\boldsymbol{h} = BERT([CLS]s[SEP]a[SEP]) \quad (1)$$

where $\boldsymbol{h} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_{(n+m)}\}$ denotes the representation of the input pair and $\boldsymbol{h}_i \in \mathbb{R}^{d_m}$ denotes the vector representation of i-th word.

2.3 Task-related Graphs Construction

For an input sentence-target pair (s, a), we obtain the corresponding dependency tree $\mathcal{T}^s$ over s by syntactical dependency parser[1]. Noting that we cannot construct a dependency tree for target text a, since it is not a full sentence. Therefore, we con-

[1]We use spaCy toolkit: https://spacy.io/.

struct the dependent relations by adding the head-dependent arcs between words in a and root words of $\mathcal{T}^s$, and finally obtain dependency tree $\mathcal{T}$ for input sentence-target pair (s, a). Accordingly, a syntactical dependency graph $\mathcal{G}_{dep}$ is built by taking each word as node and treating the head-dependent relation in $\mathcal{T}$ as edges. Note that, the $\mathcal{G}_{dep}$ is a undirected graph with self loops in words. Formally, the adjacent matrix $\mathcal{A}_{dep} \in \mathbb{R}^{(n+m)\times(n+m)}$ is computed by,

$$\mathcal{A}_{dep}[i,j] = \begin{cases} 1 & \text{if } \mathcal{T}(w_i, w_j) \text{ or } \mathcal{T}(w_j, w_i), \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\mathcal{T}(w_i, w_j)$ denotes that there exists a dependent arc from word w_i to w_j.

Task-related Pragmatics Weights Computing To capture the importance of words and interaction between words, we compute the pragmatics weights of words focused on specific task by integrating the word frequency and task-related pragmatics weight. Firstly, we compute word frequency $p(w_i)$ by calculating the times of word w_i appearing in the corpus, which can be defined as,

$$p(w_i) = \frac{N(w_i)}{N},\ \varphi(w_i) = \frac{p(w_i) - \mu(p(\cdot))}{\sigma(p(\cdot))} \tag{3}$$

where $N(w_i)$ is the number of w_i appeared in the corpus, N is the total number of words in the corpus. $\mu(\cdot)$ and $\sigma(\cdot)$ are the mean value and standard deviation function of p, respectively. For alleviating the influence of outliers and extreme values (e.g. symbols, meaningless words), we use the normalization form $\varphi(w_i)$ of the distribution of $p(\cdot)$ later.

Secondly, inspired by (Liang et al., 2021), to reduce the interference of noise, we only utilize *Favor* ($label_+$) and *Against* ($label_-$) category to compute the stance-related pragmatics weight $\phi^t(w_i)|_{t=st}$, and utilize *Positive* ($label_+$) and *Negative* ($label_-$) category to compute the sentiment-related pragmatics weight $\phi^t(w_i)|_{t=se}$. For sentence s, we compute the $\phi^t(w_i)$ by,

$$\rho^t(w_i) = \left| \frac{N^t(w_i, label_+)}{N^t(label_+)} - \frac{N^t(w_i, label_-)}{N^t(label_-)} \right| \tag{4}$$

$$\phi^t(w_i) = 1 + \frac{\rho^t(w_i) - \mu(\rho^t)}{\sigma(\rho^t)}, \quad w_i \in s \tag{5}$$

where $t \in \{st, se\}$ denotes stance task st or sentiment task se. $N^t(w_i, label_\pm)$ and $N^t(label_\pm)$ are the number of occurrences of w_i and the total number of words in different stance or sentiment category respectively. For word w_j that only appears in the target text a, $w_j \in a$, we directly define the pragmatics weight of w_j by $\phi^t(w_j) = 1$.

To harmonize the syntactic structural information and pragmatics information of words, the task-related graphs $\mathcal{G}^t$ are built by integrating the pragmatics information $\phi^t(w_i)$ learned from the word-level perspective into the syntactic graph $\mathcal{G}_{dep}$. Here, the nodes of $\mathcal{G}^t$ are every word of the input pair, and adjacent matrices $\mathcal{A}^t \in \mathbb{R}^{(n+m)\times(n+m)}$ between nodes are given by,

$$\mathcal{A}^t[i,j] = \begin{cases} \sum\limits_{k\in\{i,j\}} \phi^t(w_k)\varphi(w_k) & \text{if } \mathcal{A}_{dep}[i,j] = 1, \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Note that, $\mathcal{A}^{st}$ and $\mathcal{A}^{se}$ are the adjacent matrices of st-graph $\mathcal{G}^{st}$ and se-graph $\mathcal{G}^{se}$, respectively.

2.4 Multi-task Interaction Module

For augmenting the importance of context words close to the target (Li et al., 2018), we conduct a positional-aware transformation e_i on the word representation $\boldsymbol{h}_i$ to mask out the target words and assign position weights. Formally, given an input sentence-pair $w_1 \cdots w_n w_{n+1} \cdots w_{n+m}$, the transformed $\boldsymbol{h}_i$ is $\hat{\boldsymbol{h}}_i = e_i \boldsymbol{h}_i$, where e_i is given by,

$$e_i = \begin{cases} 1 - \frac{n+1-i}{n+m} & 1 \le i < n+1 \\ 0 & n+1 \le i \le n+m \end{cases} \tag{7}$$

We denote the transformed word representations as $\hat{\boldsymbol{h}} = \{\hat{\boldsymbol{h}}_1, \hat{\boldsymbol{h}}_2, \cdots, \hat{\boldsymbol{h}}_{n+m}\}$, which is used for task-related graphs iterative update later.

Task-related Graphs Iterative Update We utilize an iterative task-related graphs interaction way to capture the fine-grained task interaction at the word level. We assemble st-graph layer and se-graph layer to form an interactive GCN block, which can interactively learn the task-related graph representations. The feature of each node in the l-th GCN block is iteratively and interactively updated by graph convolution networks (GCNs) (Kipf and Welling, 2017), which is given by,

$$\boldsymbol{g}^{st,l} = \mathrm{ReLU}\left(\boldsymbol{E}_{se}^{\frac{1}{2}} \tilde{\mathcal{A}}^{se} \boldsymbol{E}_{se}^{-\frac{1}{2}} \boldsymbol{g}^{se,l-1} \boldsymbol{W}_{st}^{l} \right) \tag{8}$$

$$\boldsymbol{g}^{se,l} = \mathrm{ReLU}\left(\boldsymbol{E}_{st}^{\frac{1}{2}} \tilde{\mathcal{A}}^{st} \boldsymbol{E}_{st}^{-\frac{1}{2}} \boldsymbol{g}^{st,l} \boldsymbol{W}_{se}^{l} \right) \tag{9}$$

where $\boldsymbol{g}^{se,l-1}$ is the node representations derived from the preceding GCN block. Adjacent matrix $\tilde{\mathcal{A}}^t = \mathcal{A}^t + \boldsymbol{I}$, where $\boldsymbol{E}_t = \sum_j (\mathcal{A}_j^t + \boldsymbol{I})$ and $\boldsymbol{I}$ is the identity matrix. Here, the initial input of first GCN block is $\boldsymbol{g}^{se,0} = \hat{\boldsymbol{h}} = \{\hat{\boldsymbol{h}}_1, \cdots, \hat{\boldsymbol{h}}_{n+m}\}$.

2.5 Task Representations

Target-specific Stance Representation To obtain the target-specific representation, we firstly mask out non-target words and keep the target word unchanged as follows,

$$mask_i = \begin{cases} 0 & 1 \leq i < n+1 \\ 1 & n+1 \leq i \leq n+m \end{cases} \tag{10}$$

Therefore, we obtain the masking graph representation $\hat{\boldsymbol{g}}^{st,L} = mask \times \boldsymbol{g}^{st,L}$, where $\boldsymbol{g}^{st,L}$ is the final output of st-graph. Then, we use a retrieval-based attention mechanism to retrieval significant target-specific stance clue α as below,

$$\beta_k = \sum_{i=1}^{m+n} \boldsymbol{h}_k^\top \hat{\boldsymbol{g}}_i^{st,L}, \ \alpha_k = \frac{\exp{(\beta_k)}}{\sum_{i=1}^{n} \exp{(\beta_i)}} \tag{11}$$

where $\boldsymbol{h}$ is the output of encoder in Section 2.2. The final target-specific stance presentation is formulated as : $\boldsymbol{r}^{st} = \sum_{k=1}^{m+n} \alpha_k \boldsymbol{h}_k$.

Sentence-based Sentiment Representation Different from stance task, we only retrieve significant contextual sentiment clues to predict the sentence-level sentiment label,

$$\beta_k^{'} = \sum_{i=1}^{m+n} \boldsymbol{h}_k^\top \boldsymbol{g}_i^{se,L}, \ \alpha_k^{'} = \frac{\exp(\beta_k^{'})}{\sum_{i=1}^{n} \exp(\beta_i^{'})} \tag{12}$$

where $\boldsymbol{g}^{se,L}$ denotes the final output of se-graph. Then we obtain the final richer sentiment representation by: $\boldsymbol{r}^{se} = \sum_{k=1}^{m+n} \alpha_k^{'} \boldsymbol{h}_k$.

2.6 Multi-task Learning Objective

The final objective function of stance detection and sentiment analysis task is defined by cross-entropy loss and L_2-regularization,

$$\mathcal{L} = -\sum_{i=1}^{N_s} (\lambda_1 y_i^{st} \log \hat{y}_i^{st} + \lambda_2 y_i^{se} \log \hat{y}_i^{se}) + \lambda_3 \|\Theta\|_2 \tag{13}$$

where λ_1, λ_2 and λ_3 are the coefficients of stance detection, sentiment analysis task and L_2-regularization term. $\hat{\boldsymbol{y}}^t$ is the probability distribution of task representation by a fully-connected layer, $\hat{\boldsymbol{y}}^t = softmax(\boldsymbol{W}^t \boldsymbol{r}^t + \boldsymbol{b}^t)$ and y_i^t is the ground-truth label. Θ is the total parameters.

3 Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of MTIN, with the goal of answering the following questions: **RQ1:** How are the performances of the proposed MTIN model compared to existing works? **RQ2:** What are the contributions of components of MTIN?

Dataset	Category	Stance Task			Sentiment Task		
		Favor	Against	Neither	Positive	Neutral	Negative
SemEval16	Train	753	1395	766	962	189	1763
	Test	304	715	230	368	90	791
COVID19	Train	1252	750	748	451	360	1939
	Test	306	205	176	108	82	497

Table 1: Statistics for the two datasets.

3.1 Datasets

We conduct experiments on two benchmark datasets for stance detection and sentiment analysis tasks, SemEval16 (Mohammad et al., 2016) and COVID19 (Glandt et al., 2021). These two datasets are twitter posts annotated with the stance and sentiment labels. The detailed statistics are listed in Table 1.

3.2 Implementation Details

For all the experiments, we use pre-trained BERT-base[2] model to initialize the word embedding. The dimensionality of the word embedding is set to 768. To alleviate overfitting, we apply dropout at a rate of 0.3 to input word embedding. The dropout rate of st-graph and se-graph are set to 0.2, and the number of GCN block is set to 2. We use a uniform distribution to initialize all parameters of model. We implement MTIN in PyTorch (v1.7.0 with Python 3.8.3) and use Adam with learning rate (0.00001), batch size (16), and L2 regularization weight (0.001). The parameters $(\lambda_1, \lambda_2, \lambda_3)$ are set as $(0.6, 0.4, 0.00001)$. Following by (Mohammad et al., 2016; Li and Caragea, 2019), we use official evaluation metrics of SemEval16 to evaluate the performance of stance detection task of MTIN, which is the F_{avg} for the *Favor* and *Against* categories under all of the testing set.

$$F_{avg} = \frac{F_{\text{favor}} + F_{\text{against}}}{2} \tag{14}$$

where $F_{\text{favor}} = \frac{2P_{\text{favor}} R_{\text{favor}}}{P_{\text{favor}} + R_{\text{favor}}}$ and $F_{\text{against}} = \frac{2P_{\text{against}} R_{\text{against}}}{P_{\text{against}} + R_{\text{against}}}$, where P and R are precision and recall respectively. In addition, we average the F_{avg} on each target to get $MacF_{avg}$ to evaluate the overall performance of MTIN toward targets. We average Accuracy (ACC) and F1 score (*F1-score*) on each target to get ACC_{avg} and *F1-score*$_{avg}$ to evaluate sentiment analysis performance of MTIN.

3.3 Baseline Methods

We have chosen a few mainstream and latest methods in stance detection and sentiment analysis for

[2]https://github.com/google-research/bert

Category	Model	SemEval16		COVID19	
		$MacF_{avg}$	F_{avg}	$MacF_{avg}$	F_{avg}
STL	SVM-ngram (Mohammad et al., 2016)	0.580	0.689	0.584	0.654
	MITRE (Mohammad et al., 2016)	0.560†	0.678†	-	-
	pkudblab (Mohammad et al., 2016)	0.586†	0.673†	-	-
	BERT (Devlin et al., 2019)	0.536	0.646	0.619	0.647
	SCN (Yang et al., 2020a)	0.545	0.613	0.512	0.533
	KEMLM (Kawintiranon and Singh, 2021)	0.556	0.641	0.619	0.644
	TextGCN (Yao et al., 2019)	0.599	0.653	0.557	0.588
	TextING (Zhang et al., 2020)	0.597	0.658	0.589	0.609
	FCS (Sun et al., 2019)	0.602	0.692	-	-
	BERTtoCNN (Li et al., 2021a)	-	0.678	-	-
MTL	AT-JSS (Li and Caragea, 2019)	0.513	0.600	0.566	0.595
	Tchebycheff (Mao et al., 2020)	0.504	0.573	0.581	0.605
	BanditMTL (Mao et al., 2021)	0.540	0.608	0.601	0.619
Ours	MTIN *w/o* SE	0.578	0.645	0.622	0.638
	MTIN-BiLSTM	<u>0.643</u>	<u>0.689</u>	<u>0.647</u>	<u>0.666</u>
	MTIN (Ours)	**0.649**	**0.703**	**0.653**	**0.679**

Table 2: Experimental results of stance detection task on two datasets. Average $MacF_{avg}$ and F_{avg} over 3 runs with random initialization. The best and second-best results are in bold and underlined, respectively. The results with † are retrieved from semantic evaluation (SemEval-2016).

comparison, including:

- Single-task learning (STL) methods: For ST task, we use KEMLM (Kawintiranon and Singh, 2021), BERT (Devlin et al., 2019), SCN (Yang et al., 2020a), TextGCN (Yao et al., 2019), and TextING (Zhang et al., 2020) for comparison methods. Also, we compared three best methods from the semantic evaluation challenge SemEval-2016 (Mohammad et al., 2016), SVM-ngram, MITRE, and pkudblab. For SE task, we use BERT (Devlin et al., 2019), ABCDM (Basiri et al., 2021), ISA (Barnes et al., 2021), TextGCN (Yao et al., 2019), and TextING (Zhang et al., 2020) for comparison.

- Multi-task learning (MTL) methods: There are a few multi-task learning methods for stance detection here, AT-JSS (Li and Caragea, 2019), Tchebycheff (Mao et al., 2020), and BanditMTL (Mao et al., 2021).

- Our methods: MTIN is our multi-task interaction network. MTIN-BiLSTM is our MTIN with the BERT replaced by Glove+BiLSTM. MTIN *w/o* ST and MTIN *w/o* SE are the model that removes the stance detection (ST) task and sentiment analysis (SE) task from MTIN, respectively.

3.4 RQ1: Performance Evaluation

3.4.1 Performance of Stance Detection

Table 2 shows experimental results on two datasets, which illustrate that MTIN achieves the best performance of stance detection over all metrics compared with other methods. The STL methods such as BERT, SCN, and KEMLM overall perform poorly since they ignore the importance of sentiment information. For example, MTIN improves upon the TextING model by 4.5% in F_{avg} on the SemEval16 dataset and has 3.2% improvement upon the BERT model in F_{avg} on the COVID19 dataset. MTL methods such as AT-JSS and Tchebycheff consider sentiment information, but they still obtain a poor performance. For example, MTIN has 9.5% improvement in F_{avg} upon the existing MTL models on the SemEval16 dataset. The key reason is that existing MTL methods ignore capturing the task interaction features at different levels. In addition, the MTIN *w/o* SE degrades the performance substantially, which indicates that multi-task interaction module can indeed help to obtain richer task representation for improved performance of

Category	Model	SemEval16		COVID19	
		ACC_{avg}	*F1-score*$_{avg}$	ACC_{avg}	*F1-score*$_{avg}$
STL	BERT (Devlin et al., 2019)	0.776	0.791	0.687	0.734
	ABCDM (Basiri et al., 2021)	0.685	0.651	0.633	0.562
	ISA (Barnes et al., 2021)	0.662	0.632	0.664	0.606
	TextGCN (Yao et al., 2019)	0.775	0.790	0.728	0.695
	TextING (Zhang et al., 2020)	0.774	0.753	0.736	0.564
MTL	AT-JSS (Li and Caragea, 2019)	0.709	0.644	0.669	0.562
	Tchebycheff (Mao et al., 2020)	0.699	0.565	0.604	0.501
	BanditMTL (Mao et al., 2021)	0.654	0.594	0.658	0.580
Ours	MTIN *w/o* ST	0.773	0.798	0.741	0.718
	MTIN-BiLSTM	<u>0.781</u>	**0.802**	<u>0.761</u>	<u>0.746</u>
	MTIN (Ours)	**0.807**	<u>0.796</u>	**0.785**	**0.762**

Table 3: Experimental results of sentiment analysis task on two datasets. Average *F1-score*$_{avg}$ and ACC_{avg} over 3 runs with random initialization. The best and second-best results are in bold and underlined, respectively.

Model	ST task		SE task	
	$MacF_{avg}$	F_{avg}	ACC_{avg}	*F1-score*$_{avg}$
MTIN with *cos sim*	0.573	0.640	0.737	0.725
MTIN with *TF-IDF*	0.526	0.594	0.730	0.727
MTIN with *Fixed*	0.582	0.657	0.771	0.771
MTIN (Ours)	**0.649**	**0.703**	**0.807**	**0.796**

Table 4: Ablation study results of ST and SE task on SemEval16 dataset.

all tasks. In summary, experimental results show the effectiveness of MTIN on stance detection task.

3.4.2 Performance of Sentiment Analysis

As shown in Table 3, we can observe that MTIN achieves tremendously better performance than all the comparison methods across all metrics. Among them, compared with STL methods, such as BERT, ABCDM, and ISA, our proposed MTIN improves 3.1% on ACC_{avg} on SemEval dataset and 9.8% on ACC_{avg} on COVID-19 dataset, which verifies the effectiveness of MTIN. Compared with MTL models, our MTIN has significantly improvement (15.2% and 18.2% on *F1-score*$_{avg}$) on the SemEval16 and COVID19 dataset, which further illustrates the effectiveness of task-related pragmatics graph that leverages the interaction between words and interaction between tasks. In addition, the MTIN *w/o* ST degrades the performance substantially, which indicates that MTIN can indeed help to obtain richer task representation for improved performance of all tasks.

3.5 RQ2: Ablation Study

To explore the impact of embedding model, we construct a variant of MTIN, named MTIN-BiLSTM, by replacing BERT of MTIN with Glove and Bi-LSTM model. As shown in Table 2 and Table 3, it is clear that MTIN-BiLSTM outperforms existing methods but worse than MTIN, which verifies the effectiveness of our proposed MTIN with more powerful BERT embedding model.

In addition, to verify the effectiveness of task-related pragmatics weights of words, we construct three variants of MTIN: (1) MTIN with *cos sim* that replaces task-related pragmatics weight with *cosine similarity*, (2) MTIN with *TF-IDF* that replaces task-related pragmatics weight with *TF-IDF*, (3) MTIN with *Fixed* that replaces task-related pragmatics weight with *Fixed* value ("1"). As shown in Table 4, it is clear that all of the variants of MTIN lead to the performance of all tasks drops evidently. This implies that leveraging task-related pragmatics weights of words with task interaction module properly improves the performance of all tasks. That is, our proposed task-related pragmatics dependency information is effective and it can indeed learn the task-related representations, so as to improve the performance of our proposed MTIN.

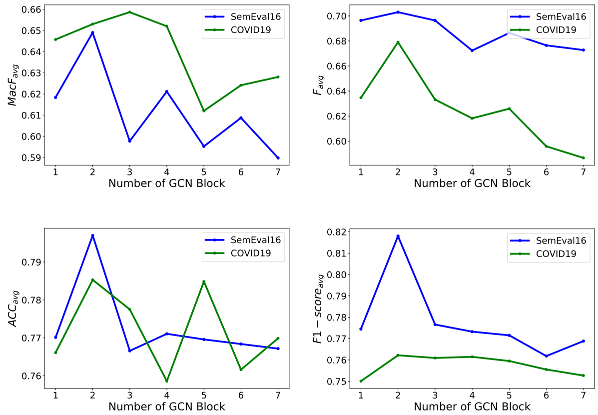
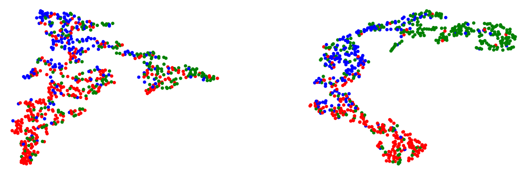


Figure 3: Performance of setting different numbers of interactions between task-related graphs.



(a) BERT model (b) MTIN (**Ours**)

Figure 4: Visualization of intermediate vector representations. Red=*Against*, blue=*Neither*, green=*Favor*.

4 Analysis and Case Study

4.1 Impact of the Number of Interaction between Task Graphs

To explore the impact of the number of interaction between task graphs over the performance of MTIN, we vary the GCN block number from 1 to 7 and show the results in Figure 3. Note that, the number of GCN block represents the number of interactions between task-related graphs. It is clear that model with 2 GCN blocks performs overall better than other numbers across all tasks and datasets, and thus we set the number of GCN block to 2 in our model. Comparatively, the model with 1 GCN block achieves unsatisfactory performance under all tasks and datasets, which potentially illustrates that too few interactions between task graphs is insufficient to leverage task-related pragmatic information to improve the performance of all tasks simultaneously. Moreover, in the cases of the number of GCN blocks greater than 2, the performance fluctuates with the increasing number of GCN blocks and essentially tends to decline when the number of GCN blocks is greater than 4. This implies that increasing the number of GCN blocks can easily degrade the learning ability of the model due to a sharp increase in model parameters.

4.2 Visualization

To qualitatively illustrate the effectiveness of our MTIN model over facilitating the learning of features, we visualize the intermediate vectors learned by **BERT** and our proposed **MTIN** by using t-SNE (Van der Maaten and Hinton, 2008) tool on the testing set of COVID19 dataset and show the visualization of stance detection task in Figure 4. We can clearly observe that there are three distinct clusters here, which represent the three different polarities of stance detection task. Note that, among the three stance polarities, the separations of vector representations learned by **MTIN** are substantially sharper than those learned by the **BERT** model. Also, the feature of each cluster learned by **MTIN** is more compact and distinguishable compared with features learned by **BERT** model. This verifies that MTIN model can capture more definite correlation between different stance classes and leverage the learned task-related features to capture the clearer difference of task representations, so as to achieve the better performance of all learning tasks.

4.3 Case Study

To better understand how MTIN works, we select two testing samples of stance detection task , whose targets are "*Feminist Movement*" and "*Wearing a Face Mask*" respectively. As shown in Table 5, we present a case study and visualize attention weights learned by three different models. The color represents the importance of a word towards the stance representation, the darker more important.

The first example "*I believe that every woman should have their own rights.*" has a subjective word "*believe*" that can explicitly express the stance of user toward the target, and a subjunctive word "*should*" that can bring extra difficulty in detecting implicit semantics. TextGCN fails to recognize the importance of words "*believe*" and "*should*", resulting in wrong predictions. BERT and MTIN accurately capture the relationship between those words and given target, so as to predict the correct stance.

The second example "*Not wearing a mask because I am in my own greenhouse.*" contains the negation "*Not*" in the sentence, which can easily mislead models into making wrong stance.

Model	Target	Attention Visualization	Prediction	Label
TextGCN	FM	I believe that every woman should have their own rights .	Against✗	Favor
	WFM	Not wearing a mask because I am in my own greenhouse .	Favor✗	Against
BERT	FM	I believe that every woman should have their own rights .	Favor✓	Favor
	WFM	Not wearing a mask because I am in my own greenhouse .	Favor✗	Against
MTIN	FM	I believe that every woman should have their own rights .	Favor✓	Favor
	WFM	Not wearing a mask because I am in my own greenhouse .	Against✓	Against

Table 5: Case study. Visualization of attention scores from BERT, TextGCN, and MTIN on testing examples of SemEval16 and COVID19 datasets, along with their predictions and corresponding ground truth labels. The target FM (Feminist Movement) and WFM (Wearing a Face Mask) are from SemEval16 and COVID19 datasets respectively. The marker ✓ and ✗ indicate the correct and incorrect predictions, respectively.

TextGCN and BERT are capable of capturing the negation, but fail to build the connection between negation word and given target, and finally make the wrong predictions. Our proposed MTIN correctly captures the relationship between words and given target by leveraging the task-related pragmatics dependency information, which implies that our MTIN effectively harmonizes the pragmatics dependency information and semantic information. This shows that our MTIN can make full use of pragmatics information of words to improve the performance of model.

5 Related Work

Stance detection Early efforts on stance detection focused on employing support vector machines (SVM) with manually engineered features to detect stance of users towards the given target (Mohammad et al., 2016; Sen et al., 2018; Kochkina et al., 2017; Darwish et al., 2020). Deep learning methods such as recurrent neural networks (RNNs) (Zarrella and Marsh, 2016), convolutional neural networks (CNNs) (Sun et al., 2018), and attention mechanism (Du et al., 2017; Siddiqua et al., 2019; Kawintiranon and Singh, 2021; Wei et al., 2016; Augenstein et al., 2016; Yang et al., 2020b) are proposed to capture the richer stance representation for improving the performance of stance detection. More recently, some pre-trained language models, for instance, BERT (Devlin et al., 2019) were applied to stance detection task for improved performance. Another line of work utilized GNNs model (Yao et al., 2019; Zhang et al., 2020) to achieve promising results via leveraging syntactical dependency parse to construct the relationship between words of sentences. For example, Yao et al. (2019) leverages the explicit dependency parse trees to model the syntactical connections of contextual words to obtain the better stance representations. Zhang et al. (2020) integrate syntactical structure information and semantic dependency information for stance detection.

Multi-task learning Multi-task learning (MTL) methods (Caruana, 1997; Thung and Wee, 2018; Liao et al., 2021b; Yang et al., 2021) aims at solving multiple tasks simultaneously via modeling the relationship between some learning tasks to improve the generalization of MTL models, which have gained much popularity in a variety of Natural Language Processing (NLP) downstream tasks (Liu et al., 2015; Luong et al., 2016). Li and Caragea (2019) proposed a MTL framework that takes sentiment analysis task as auxiliary task and incorporates target-specific attention mechanism to achieve promising performance of stance detection. Mao et al. (2020) proposed a novel Tchebycheff procedure to improve the performance of multi-task text classification. Mao et al. (2021) solved poor generalization performance of MTL, which caused uncontrolled task variance by jointly minimizing the empirical losses and regularizing the task variance. Different from the methods mentioned above, we study the relationship between joint learning tasks via task-related pragmatics weight of words and capture the word-level task interaction information through a proposed multi-task interaction module.

6 Conclusion

In this paper, we propose a novel multi-task learning model to improve the performance on both the stance detection and sentiment analysis task. We first construct heterogeneous task-related pragmatics graphs to capture more fine-grained task representation. Besides, a multi-task interaction module is designed to capture the word-level interaction between tasks, so as to improve the overall performance of model. Experimental results show the effectiveness of our proposed model.

Limitations

In this paper, we construct a stance graph and a sentiment graph for each given sentence and employ GCN module to achieve the interactions between stance detection and sentiment analysis. For a long text, the constructed graph is very large and sparse, which makes our model impossible to train, so as to fail to detect the stance and classify the sentiment. Therefore, the major limitation of our work is that it cannot be applied to long texts.

Acknowledgements

This work was supported by the National Natural Science Foundation of China(No. 62076079), Guangdong Major Project of Basic and Applied Basic Research (No.2019B030302002), and Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005).

References

Ameeta Agrawal, Aijun An, and Manos Papagelis. 2020. Leveraging transitions of emotions for sarcasm detection. In *SIGIR*, pages 1505–1508. ACM.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *EMNLP*, pages 876–885. The Association for Computational Linguistics.

Jeremy Barnes, Erik Velldal, and Lilja Øvrelid. 2021. Improving sentiment analysis with multi-task learning of negation. *Nat. Lang. Eng.*, 27(2):249–269.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U. Rajendra Acharya. 2021. ABCDM: an attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Gener. Comput. Syst.*, 115:279–294.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Kareem Darwish, Peter Stefanov, Michaël J. Aupetit, and Preslav Nakov. 2020. Unsupervised user stance detection on twitter. In *ICWSM*, pages 141–152, Atlanta, Georgia, USA. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A neural multi-digraph model for chinese NER with gazetteers. In *ACL*, pages 1462–1467. Association for Computational Linguistics.

Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention. In *Proc. 26th Int. Joint Conf. on Artificial Intel.*, pages 3988–3994, Melbourne, Australia. IJCAI.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in COVID-19 tweets. In *ACL/IJCNLP*, pages 1596–1611. Association for Computational Linguistics.

Marjan Hosseinia, Eduard C. Dragut, and Arjun Mukherjee. 2020. Stance prediction for contemporary issues: Data and experiments. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media, SocialNLP@ACL*, pages 32–40. Association for Computational Linguistics.

Yuan Huang, Zhixing Li, Wei Deng, Guoyin Wang, and Zhimin Lin. 2021. D-bert: Incorporating dependency-based attention into bert for relation extraction. *CAAI Trans. Intell. Technol.*, 6(4):417–425.

Kornraphop Kawintiranon and Lisa Singh. 2021. Knowledge enhanced masked language model for stance detection. In *NAACL-HLT*, pages 4725–4735.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR*.

Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm. In *Proceedings of the 10th International Workshop on Semantic Evaluation,*, pages 475–480. ACL.

Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Comput. Surv.*, 53(1):12:1–12:37.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *ACL*, pages 946–956. Association for Computational Linguistics.

Yang Li, Yuqing Sun, and Nana Zhu. 2021a. Berttocnn: Similarity-preserving enhanced knowledge distillation for stance detection. *Plos one*, 16(9):e0257130.

Yingjie Li and Cornelia Caragea. 2019. Multi-task stance detection with sentiment and stance lexicons. In *EMNLP-IJCNLP*, pages 6298–6304. Association for Computational Linguistics.

Ziyang Li, Feng Hu, Chilong Wang, Weibin Deng, and Qinghua Zhang. 2021b. Selective kernel networks for weakly supervised relation extraction. *CAAI Trans. Intell. Technol.*, 6(2):224–234.

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*, pages 3453–3464.

Qing Liao, Heyan Chai, Hao Han, Xiang Zhang, Xuan Wang, Wen Xia, and Ye Ding. 2021a. An integrated multi-task model for fake news detection. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Qing Liao, Heyan Chai, Hao Han, Xiang Zhang, Xuan Wang, Wen Xia, and Ye Ding. 2021b. An integrated multi-task model for fake news detection. *IEEE Trans. Knowl. Data Eng.*, pages 1–1.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *HLT-NAACL*, pages 912–921. The Association for Computational Linguistics.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR (Poster)*.

Xiaobo Ma, Yongbin Liu, and Chunping Ouyang. 2022. Capturing semantic features to improve chinese event detection. *CAAI Trans. Intell. Technol.*, 7(2):219–227.

Yuren Mao, Zekai Wang, Weiwei Liu, Xuemin Lin, and Wenbin Hu. 2021. Banditmtl: Bandit-based multi-task learning for text classification. In *ACL/IJCNLP*, pages 5506–5516. Association for Computational Linguistics.

Yuren Mao, Shuang Yun, Weiwei Liu, and Bo Du. 2020. Tchebycheff procedure for multi-task text classification. In *ACL*, pages 4217–4226. Association for Computational Linguistics.

Saif Mohammad, Svetlana Kiritchenko, and et al. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *SemEval@NAACL-HLT 2016*, pages 31–41. The Association for Computer Linguistics.

Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and sentiment in tweets. *ACM Trans. Internet Techn.*, 17(3):26:1–26:23.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Anirban Sen, Manjira Sinha, Sandya Mannarswamy, and Shourya Roy. 2018. Stance classification of multi-perspective consumer health information. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pages 273–281. ACM.

Umme Aymun Siddiqua, Abu Nowshed Chy, and Masaki Aono. 2019. Tweet stance detection using an attention based neural ensemble model. In *Proc. NAACL-HLT (1)*, pages 1868–1873.

Qingying Sun, Zhongqing Wang, Shoushan Li, and et al. 2019. Stance detection via sentiment information and neural network model. *Frontiers of Computer Science*, 13(1):127–138.

Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018. Stance detection with hierarchical attention network. In *Proc. 27th International Conference on Computational Linguistics*, pages 2399–2409, Santa Fe, New Mexico, USA. ACL.

Jingyao Tang, Yun Xue, Ziwen Wang, Shaoyang Hu, Tao Gong, Yinong Chen, Haoliang Zhao, and Luwei Xiao. 2022. Bayesian estimation-based sentiment word embedding model for sentiment analysis. *CAAI Trans. Intell. Technol.*, 7(2):144–155.

Kim-Han Thung and Chong-Yaw Wee. 2018. A brief review on multi-task learning. *Multim. Tools Appl.*, 77(22):29705–29725.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Eva Maria Vecchi, Neele Falk, Iman Jundi, and Gabriella Lapesa. 2021. Towards argument mining for social good: A survey. In *ACL/IJCNLP*, pages 1338–1352. Association for Computational Linguistics.

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

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24.

Chang Xu, Cécile Paris, Surya Nepal, Ross Sparks, Chong Long, and Yafang Wang. 2020. DAN: dual-view representation learning for adapting stance classifiers to new domains. In *ECAI/(PAIS*, volume 325, pages 2260–2267. IOS Press.

Dechuan Yang, Qiyu Wu, Wei Chen, Tengjiao Wang, Zhen Qiu, Di Liu, and Yingbao Cui. 2020a. Stance detection with stance-wise convolution network. In *NLPCC*, volume 12430, pages 555–567. Springer.

Dechuan Yang, Qiyu Wu, Wei Chen, Tengjiao Wang, Zhen Qiu, Di Liu, and Yingbao Cui. 2020b. Stance detection with stance-wise convolution network. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 555–567. Springer.

Min Yang, Weiyi Huang, Wenting Tu, Qiang Qu, Ying Shen, and Kai Lei. 2021. Multitask learning and reinforcement learning for personalized dialog generation: An empirical study. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):49–62.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *AAAI*, 01, pages 7370–7377.

Guido Zarrella and Amy Marsh. 2016. MITRE at semeval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation,*, pages 458–463, San Diego, CA, USA. The Association for Computer Linguistics.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every document owns its structure: Inductive text classification via graph neural networks. In *ACL*, pages 334–339. Association for Computational Linguistics.

Shengping Zhou, Junjie Lin, Lianzhi Tan, and Xin Liu. 2019. Condensed convolution neural network by attention over self-attention for stance detection in twitter. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.