A Topic-Aware Reinforced Model for Weakly Supervised Stance Detection

Penghui Wei, Wenji Mao, Guandan Chen
[†]SKL-MCCS, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[‡]University of Chinese Academy of Sciences, Beijing, China
{weipenghui2016, wenji.mao, chenguandan2014}@ia.ac.cn

Abstract

Analyzing public attitudes plays an important role in opinion mining systems. Stance detection aims to determine from a text whether its author is in favor of, against, or neutral towards a given target. One challenge of this task is that a text may not explicitly express an attitude towards the target, but existing approaches utilize target content alone to build models. Moreover, although weakly supervised approaches have been proposed to ease the burden of manually annotating large-scale training data, such approaches are confronted with noisy labeling problem. To address the above two issues, in this paper, we propose a Topic-Aware Reinforced Model (TARM) for weakly supervised stance detection. Our model consists of two complementary components: (1) a detection network that incorporates target-related topic information into representation learning for identifying stance effectively; (2) a policy network that learns to eliminate noisy instances from auto-labeled data based on off-policy reinforcement learning. Two networks are alternately optimized to improve each other's performances. Experimental results demonstrate that our proposed model TARM outperforms the state-of-the-art approaches.

Introduction

Analyzing and mining public attitudes has attracted increasing attention from opinion mining research community. Stance detection is the task of automatically inferring from an opinionated text whether its author is *in favor of*, *against*, or *neither of them* towards a given target. The target of interest may be a person, a government policy, a product, a claim and so on (Mohammad et al. 2016). Unlike sentiment analysis which centers on identifying the sentiments towards entities (Liu 2012), stance analysis aims to mine the essential viewpoints people stand, which reflects what people think and believe intrinsically. Thus, mining the stances in user-generated contents has broad applications such as social media monitoring and government decision making.

Recent stance detection studies have focused on analyzing online contents from social media platforms, and typically Twitter (Rajadesingan and Liu 2014; Sasaki et al. 2018). Figure 1 shows an example of Twitter stance detection task defined by Mohammad et al. (2016). According to the tweet content, we can infer from it that the tweet author is opposed to "*Atheism*". As we can see, a tweet may *not* explicitly express an attitude towards the target of interest, which is a major difference between stance detection and aspect-level sentiment classification. Because the semantic information carried by target content itself is limited, it may not be able to provide enough information for classifying stance accurately. Existing approaches for Twitter stance detection usually utilize target content alone to build models (Augenstein et al. 2016; Du et al. 2017; Zhou, Cristea, and Shi 2017; Wei, Mao, and Zeng 2018; Dey, Shrivastava, and Kaushik 2018). Therefore, capturing *target-related implicit expression* in text has become a challenging research issue in targeted stance detection task.

Target: *Atheism*
Tweet: Be still. Be patient. Watch and let God work.
Stance: *against*   **Overall sentiment:** *positive*

Figure 1: Twitter stance detection: An example.

Meanwhile, as manually annotating large-scale training data is time-consuming and labor-intensive, *weakly supervised* approaches for stance detection are in great need. In real-world scenarios, it is often required to tackle new targets which do not have ready-made labeled data. For instance, in a general election, the candidates are probably new targets and there is no labeled data with respect to them. Through building detection models on auto-labeled large-scale datasets, weakly supervised stance detection approaches are advantageous in such scenarios.

Related studies that focus on weakly supervised stance detection (Wei et al. 2016; Augenstein et al. 2016; Ebrahimi, Dou, and Lowd 2016a; 2016b) adopt Distant Supervision (DS) (Go, Bhayani, and Huang 2009) to construct auto-labeled training data based on manually-selected stance-indicative patterns. For instance, if a tweet content contains the pattern "#blessed", its author is probably opposed to "*Atheism*" and DS will annotate the tweet with an *against* label. Obviously, DS avoids the drawback of manual annotation, however, it also brings noisy instances annotated with incorrect stance labels. Thus, reducing the negative impact of *noisy stance labeling problem* is another challenging research issue in weakly supervised stance detection.

To address the above two challenging issues, in this paper, we propose a novel **T**opic-**A**ware **R**einforced **M**odel (**TARM**) for weakly supervised stance detection task, which consists of two complementary components: a topic-aware detection network and a stance revision policy network.

Intuitively, although a tweet may not explicitly express an attitude towards the given target, it usually talks about one or more *target-related topics*, and expresses attitudes towards these topics to implicitly stand its viewpoint towards the target. If we take such topic information into account, it will contribute to understanding target-related implicit expressions. Motivated by this, we design a topic-aware detection network (TDNet) that incorporates topic information into the tweet representation learning process. Specifically, we first extract target-related topics from large-scale corpus. Then, TDNet learns to capture text spans talking about these topics, and obtains tweet representations with respect to different topics. Compared to the models that only consider target information, TDNet represents a tweet via various views (i.e., multiple topics), and through which we can effectively identify targeted stance based on captured implicit expressions.

Further, unlike previous approaches that directly build detection models on noisy labeled data, we resort to learn a denoising policy that can *eliminate noisy tweets* for improving the capability of TDNet. Because no supervised signal can inform whether a tweet's label annotated by DS is correct, we need a trial-and-error process to explore a reliable policy. Moreover, tweets containing the same pattern are partially related, thus it is natural to make sequential decisions instead of individual ones on them. Motivated by the *exploration* ability of reinforcement learning (RL), we introduce a stance revision policy network (SRNet) that learns an RL-based policy to revise the auto-labeled data. Specifically, we formulate this as a sequential decision process: for each tweet, SRNet makes a decision to indicate whether it should be eliminated, where the current decision is affected by previous decisions. After obtaining a revised dataset, TDNet measures its quality by producing a reward signal, and the goal of SRNet is to achieve higher cumulative reward.

To make the exploration (i.e., trial-and-error) process of SRNet achieve timely reward signals from TDNet, we *alternately* optimize TDNet and SRNet after pre-training them on auto-labeled data: TDNet provides tweet representations and rewards to SRNet, and guides it to learn a good revision policy; SRNet produces revised data for training a better TDNet. Such procedure is alternate, aiming to make TDNet and SRNet improve each other's performances. Moreover, to accelerate the training process of SRNet, we leverage *off-policy* algorithm to optimize SRNet, which makes the training process significantly faster than traditional on-policy algorithms.

The contributions of this work are as follows:

- To address the target-related implicit expression issue in stance detection, we integrate topic information into tweet representation for effectively identifying targeted stance.
- To reduce the negative impact of noisy stance labeling in weakly supervised setting, we introduce a policy network that learns to eliminate noisy instances from auto-labeled data, and employ off-policy strategy to accelerate training.
- Experimental results show that our proposed model TARM is superior to the state-of-the-art approaches.

Related Work

Stance detection aims to identify the position expressed in a text towards a target, which is a research field related to argument mining (Lippi and Torroni 2016) and aspect-level sentiment analysis (Ma, Peng, and Cambria 2018), and also plays a crucial role in fact checking (Mohtarami et al. 2018).

Previous studies mainly focus on debates (Thomas, Pang, and Lee 2006; Somasundaran and Wiebe 2009; Hasan and Ng 2013). They utilize lexical and structural features extracted in text to build stance classifiers. Recent studies pay more attention to online contents from social media (Rajadesingan and Liu 2014; Chen and Ku 2016; Sasaki et al. 2018; Xu et al. 2018; Wei, Lin, and Mao 2018). In this paper we center on detecting stance towards a pre-defined proposition or entity, e.g., a government policy or a specific person.

Existing studies on supervised stance detection in Twitter mainly adopt neural networks. Zarrella and Marsh (2016) pre-train LSTM on unlabeled corpus by predicting hashtags. Du et al. (2017) and Zhou, Cristea, and Shi (2017) employ target-specific attention models to attend important words in tweets, and Dey, Shrivastava, and Kaushik (2018) extend their work to a two-phase solution. Wei, Mao, and Zeng (2018) adopt memory networks to iteratively polish the representations of stance clues and tweets. Sun et al. (2018) combine linguistic factors (sentiment, argument and dependency features) by an elaborate hierarchical attention model. Benton and Dredze (2018) present a user embedding method capturing user behaviors and social features for pre-training GRU.

Different approaches are proposed to tackle the weakly supervised setting, and they utilize distant supervision (Go, Bhayani, and Huang 2009; Xia, Jiang, and He 2017) to automatically annotate data for training models. Ebrahimi, Dou, and Lowd (2016a) model the interactions among stance, target and sentiment by an undirected graphical model. Further, they (2016b) collectively classify tweet stances and user stances by Markov random fields with the constraint of user relationships. To leverage the semantic information of target, Augenstein et al. (2016) propose the model BiCond that encodes tweets by a bidirectional LSTM in which initial cell states are conditional on the target. BiCond achieves the best performance of this task among related studies.

Existing neural network models for stance detection usually consider target alone to build models, which suffers from target-related implicit expression issue. In contrast, we tackle this issue by incorporating topic information into representation learning. Moreover, weakly supervised approaches are also confronted with the noisy stance labeling problem. Recently, some work applies reinforcement learning (RL) to distantly supervised tasks such as topic segmentation with topic labeling (Takanobu et al. 2018) and relation extraction (Feng et al. 2018; Qin, Xu, and Wang 2018), which inspires us to leverage RL in our specific task. In addition, these studies use on-policy training to optimize RL components, which is less efficient in time. To overcome this problem, we employ off-policy optimization which has the potential to significantly improve the training efficiency.

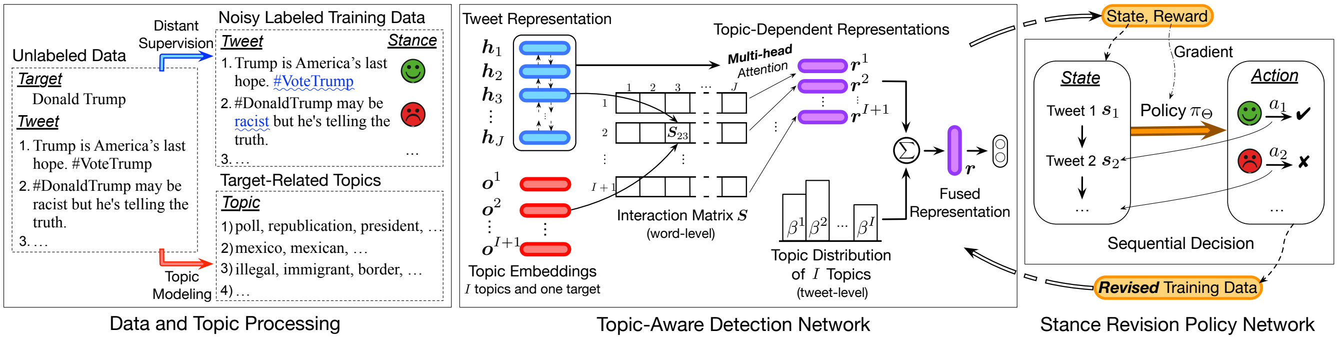

Unlabeled Data
Target
Donald Trump
Tweet
1. Trump is America's last hope. #VoteTrump
2. #DonaldTrump may be racist but he's telling the truth.
3. …
Distant Supervision
Topic Modeling
Noisy Labeled Training Data
Tweet
Stance
1. Trump is America's last hope. #VoteTrump
2. #DonaldTrump may be racist but he's telling the truth.
3. …
Target-Related Topics
Topic
1) poll, republication, president, …
2) mexico, mexican, …
3) illegal, immigrant, border, …
4) …
Data and Topic Processing
Tweet Representation
h_1
h_2
h_3
h_J
o^1
o^2
o^{I+1}
Topic Embeddings
I topics and one target
Interaction Matrix S
(word-level)
S_{23}
Multi-head Attention
Topic-Dependent Representations
r^1
r^2
r^{I+1}
β^1 β^2 … β^I
Topic Distribution of I Topics
(tweet-level)
r
Fused Representation
Topic-Aware Detection Network
State, Reward
Gradient
State
Tweet 1 s_1
Tweet 2 s_2
Policy π_Θ
Action
a_1 ✔
a_2 ✘
Sequential Decision
Revised Training Data
Stance Revision Policy Network

Figure 2: Overview of our topic-aware reinforced model (TARM) for weakly supervised stance detection.

Task Definition

Stance detection task is defined as: given a tweet and a target, identify whether the tweet author is *in favor of*, *against*, or *neither of them* towards the target. In weakly supervised setting, a *domain corpus* containing large-scale target-related tweets without stance labels is available for building models.

Proposed Approach

Overview

Figure 2 gives an overview of our **T**opic-**A**ware **R**einforced **M**odel (**TARM**). To obtain training data, we first automatically annotate domain corpus and extract topics from it. Then, a topic-aware detection network (TDNet) learns tweet representation by integrating topic information for identifying stance effectively, and a stance revision policy network (SRNet) learns to eliminate noisy instances from the auto-labeled data via maximizing the total reward computed by TDNet. Subsequently, TDNet continues to be fine-tuned with the revised data provided by SRNet, and SRNet can be further optimized using the better TDNet. The above optimization procedure is alternate, and the two networks can improve each other's performances.

Data and Topic Processing

Noisy Stance Labeling To obtain large-scale labeled training data without manual annotation, we adopt Distant Supervision (DS) (Go, Bhayani, and Huang 2009) to automatically annotate the domain corpus by stance-indicative patterns.

We manually select a set of favor-target patterns and a set of against-target patterns from the domain corpus. Tweets that contain at least one favor-/against- target pattern and do not simultaneously contain two types of patterns will be annotated with *favor/against* stance. This constructs a noisy labeled training set $\mathcal{D}$. $\mathcal{D}$ is a two-class dataset, i.e., no instance in $\mathcal{D}$ is labeled with *neither* stance[1]. The strategy of classifying *neither* stance will be detailed in next section.

Enriching Target Content via Topic Modeling To enrich the target content with target-related topic information, we extract topics[2] from the domain corpus via Biterm Topic Model (BTM) (Cheng et al. 2014), a classical topic modeling method over short texts. BTM models the generation of word co-occurrence to avoid the data sparsity in short texts.

We extract I topics using BTM, and each topic $\mathcal{T}^i$ is formally denoted as a word sequence $\mathcal{T}^i = (w_1^i, \ldots, w_N^i)$, where N means that we select top-N words for each topic.

As a result, we use these topics to enrich target content for building a topic-aware stance detection model. For the convenience of description, we denote the target itself as the $(I+1)$-th topic. Hence, we have a total of $I+1$ topics now.

Topic-Aware Detection Network (TDNet)

We design a topic-aware detection network (TDNet) for learning to represent the input tweet and topics. It consists of four layers and effectively incorporates topic information into the representation learning process.

Input Layer All words are mapped into their vector representations (i.e., word embeddings) using a look-up matrix. TDNet then represents tweet and topics in the following way.

Tweet encoder. We adopt a bi-directional GRU (biGRU), an effective sequence modeling method, to encode the input tweet. It captures contextual information for each word and outputs a sequence of hidden states (suppose that the length of the input tweet is J and the output dimension is $2d$):

$$\boldsymbol{H} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_J) \in \mathbb{R}^{2d \times J} \tag{1}$$

where each hidden state $\boldsymbol{h}_t$ is the concatenation of the t-th hidden states in two directions, i.e., $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}_t}; \overleftarrow{\boldsymbol{h}_t}] \in \mathbb{R}^{2d}$.

Topic embeddings. We adopt the same biGRU used for tweet to encode each topic. For the topic $\mathcal{T}^i = (w_1^i, \ldots, w_N^i)$ $(i \in [1, I+1])$, we represent it using the concatenation of the last step hidden states in two directions, named topic embedding $\boldsymbol{o}^i \in \mathbb{R}^{2d}$:

$$\boldsymbol{o}^i = [\overrightarrow{\boldsymbol{h}_N^i}; \overleftarrow{\boldsymbol{h}_1^i}] \tag{2}$$

Let $\boldsymbol{O} = \{\boldsymbol{o}^1, \ldots, \boldsymbol{o}^I, \boldsymbol{o}^{I+1}\} \in \mathbb{R}^{2d \times (I+1)}$ denote all $I+1$ topic embeddings. Next, we detail how to utilize such topic information to learn a topic-aware tweet representation.

Interaction Layer The interaction layer in TDNet learns to align tweet words and topics, inspired by recent models for reading comprehension task (Seo et al. 2017).

[1] It is hard to select patterns for *neither* tweets because they vary from objective contents to totally irrelevant contents.

[2] Here, a topic is represented as a set of correlated words.

We compute an interaction matrix $\boldsymbol{S} \in \mathbb{R}^{(I+1)\times J}$ to score how well a tweet word and a topic match, with bi-linear function (parametrized by $\mathbf{W}$) as the score function:

$$\boldsymbol{S} = \boldsymbol{O}^\top \mathbf{W} \boldsymbol{H} \tag{3}$$

where $\boldsymbol{S}_{ij} = (\boldsymbol{o}^i)^\top \mathbf{W} \boldsymbol{h}_j$ measures the semantic relatedness between the topic $\mathcal{T}^i$ and the j-th word $\boldsymbol{h}_j$ of the input tweet.

Representation Layer Based on the above alignment computing, this layer further incorporates topic information into representation learning for effective stance detection.

Topic-dependent tweet representations. According to various topics, a tweet can be represented with various views. Specifically, for each topic $\mathcal{T}^i$, TDNet learns a topic-dependent tweet representation $\boldsymbol{r}^i \in \mathbb{R}^{2d}$ ($i \in [1, I+1]$) via attention (Bahdanau, Cho, and Bengio 2015), a commonly-used technique to capture informative parts of input.

The i-th row of $\boldsymbol{S}$ (denoted as $\boldsymbol{S}_{i:} \in \mathbb{R}^J$) stores semantic relatedness information between topic $\mathcal{T}^i$ and the input tweet. For the topic $\mathcal{T}^i$, we obtain the attentive vector $\boldsymbol{\alpha}^i \in \mathbb{R}^J$ by normalizing $\boldsymbol{S}_{i:}$, and then compute the weighted sum of $(\boldsymbol{h}_1, \ldots, \boldsymbol{h}_J)$ to get the representation $\boldsymbol{r}^i$ dependent on this topic, aiming to assign higher weights to topic-related spans:

$$\boldsymbol{\alpha}^i = \mathrm{softmax}(\boldsymbol{S}_{i:}) \tag{4}$$

$$\boldsymbol{r}^i = \boldsymbol{H}\boldsymbol{\alpha}^i \tag{5}$$

Finally we get $I+1$ representations $\{\boldsymbol{r}^1, \ldots, \boldsymbol{r}^I, \boldsymbol{r}^{I+1}\}$ with respect to these topics. The above mechanism can be regarded as a variant of *multi-head* attention (Vaswani et al. 2017): each topic embedding $\boldsymbol{o}^i$ plays the role of the query vector in one attention head (totally $I+1$ heads). In TDNet, different attention heads share the same parameter matrix $\mathbf{W}$, while each attention head has its own query vector.

Topic distribution over a tweet. A tweet content usually centers on a tiny number of topics that its author intends to discuss. Note that topics $\mathcal{T}^1, \ldots, \mathcal{T}^I$ are extracted by training a BTM on the domain corpus. Hence, given the input tweet, we use the trained BTM to infer its topic distribution of such I topics, denoted as $(\beta^1, \ldots, \beta^I)$, where $\sum_{i=1}^{I} \beta^i = 1$.

Fused representation for a tweet. Further, we fuse the topic-dependent tweet representations $\{\boldsymbol{r}^1, \ldots, \boldsymbol{r}^I\}$ by utilizing the topic distribution as weights, and then concatenate $\boldsymbol{r}^{I+1}$. As a result, we obtain the fused tweet representation $\boldsymbol{r}$:

$$\boldsymbol{r} = \left[\boldsymbol{r}^{I+1}; \sum_{i=1}^{I} \beta^i \boldsymbol{r}^i\right] \tag{6}$$

Detection Layer Because the noisy labeled training set $\mathcal{D}$ is a two-class dataset, a softmax layer is used to output the predicted stance distribution $(y_{favor}, y_{against})$, with parameters $\mathbf{W}_D, \mathbf{b}_D$ to be learned:

$$(y_{favor}, y_{against})^\top = \mathrm{softmax}(\mathbf{W}_D \boldsymbol{r} + \mathbf{b}_D) \tag{7}$$

At test time, if a tweet's predicted stance distribution meets the condition of $|y_{favor} - y_{against}| \leqslant 0.1$, we classify it to *neither* class. This is a usual strategy in previous work (Wei et al. 2016; Ebrahimi, Dou, and Lowd 2016a).

TDNet is pre-trained on the auto-labeled training set $\mathcal{D}$ using cross entropy criterion, and then it will be fine-tuned during the joint training procedure of TDNet and SRNet.

Stance Revision Policy Network (SRNet)

Motivated by the *exploration* ability of reinforcement learning (RL), we design a stance revision policy network (SRNet), aiming at learning a policy to eliminate noisy tweets from the auto-labeled dataset $\mathcal{D}$ for building a cleansed dataset $\mathcal{D}'$.

Revision of Noisy Labeled Data We formulate the revision of the noisy labeled data $\mathcal{D}$ as a sequential decision process: for each tweet in $\mathcal{D}$, given its *state*, SRNet samples an *action* to decide whether it should be eliminated from $\mathcal{D}$; the current action is affected by previous actions and will affect following actions. After the revision process, TDNet gives a *reward* signal to indicate how well the actions do, and SRNet is optimized via maximizing the cumulative reward.

To improve the frequency of receiving reward, we group all tweets in $\mathcal{D}$ by their stance-indicative patterns, creating a set of subsets: $\mathcal{D} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots\}$. After revising one subset, TDNet computes a reward, and we sum the cumulative rewards of all subsets to obtain the objective function for the whole revision process. Based on a subset $\mathcal{G}$ that consists of T tweets, we detail the revision process that has T time steps.

State. For the t-th tweet in $\mathcal{G}$, its state vector $\boldsymbol{s}_t$ represents the current information after previous actions from time step 1 to $t-1$. Formally, $\boldsymbol{s}_t$ is the concatenation of three parts:

$$\boldsymbol{s}_t = [\boldsymbol{r}_t; \bar{\boldsymbol{r}}_{1\to t-1}; \boldsymbol{p}_t], \quad t \in [1, T] \tag{8}$$

where $\boldsymbol{r}_t$ is the tweet representation by Eq. (6), $\bar{\boldsymbol{r}}_{1\to t-1}$ is the average representation of tweets which are not eliminated by the policy from step 1 to $t-1$, and $\boldsymbol{p}_t$ is the sum of word embeddings in this tweet's pattern.

Policy and Action. The goal of our revision policy is to eliminate the tweets annotated with incorrect stance labels from $\mathcal{G}$, and keep the tweets having correct labels. Hence, we use a binary action space $\mathcal{A} = \{\text{`}elimination\text{'}, \text{`}keep\text{'}\}$: tweets assigned with the action '$elimination$' by the policy will be discarded from $\mathcal{G}$, and other tweets assigned with the action '$keep$' compose the cleansed subset $\mathcal{G}'$.

At each time step t, given the state $\boldsymbol{s}_t$, a corresponding action $a_t \in \mathcal{A}$ is sampled from a stochastic policy π_Θ which outputs a probability distribution over two actions in $\mathcal{A}$:

$$a_t \sim \pi_\Theta(a_t|\boldsymbol{s}_t) \tag{9}$$

where $\pi_\Theta(a_t|\boldsymbol{s}_t)$ is the probability of choosing the action a_t. We utilize a single-layer network $\pi_\Theta(a_t = \text{`}keep\text{'} \mid \boldsymbol{s}_t) = \sigma(\mathbf{W}_\pi \boldsymbol{s}_t + \mathbf{b}_\pi)$ to parametrize the RL policy with $\Theta = \{\mathbf{W}_\pi, \mathbf{b}_\pi\}$ to be learned, where $\sigma(\cdot)$ is logistic function.

Reward. After obtaining the revised subset $\mathcal{G}'$ by the current policy π_Θ, TDNet gives a reward r_T to measure its quality. Intuitively, if we feed a set which consists of high-quality $\langle tweet, label\rangle$ pairs into TDNet, we will obtain a lower average negative log-likelihood (NLL). Hence, we use the average of log-likelihood values in the subset $\mathcal{G}'$ as the reward function to indicate the quality of instances in $\mathcal{G}'$:

$$r_T = \frac{1}{|\mathcal{G}'|} \sum\nolimits_{\langle tweet, label\rangle \in \mathcal{G}'} \log(y_{label}) \tag{10}$$

where the predicted stance probability y_{label} of an instance $\langle tweet, label\rangle$ is computed by Eq. (7). This reward makes the objective of SRNet consistent with TDNet because the objective function of TDNet is to minimize the average NLL.

Algorithm 1 Joint Training Procedure of TARM

```
Require: 1) the noisy labeled dataset $\mathcal{D}$, which is represented by a set of subsets $\{\mathcal{G}_1, \mathcal{G}_2, \ldots\}$. Each subset $\mathcal{G}$ contains $T$ tweets.
    2) the TDNet $\Psi$; the SRNet $\Theta$; the "old" SRNet $\Theta'$.
 1: for all episode $e \leftarrow 1$ to $E$ do
 2:     for all subset $\mathcal{G} \in \mathcal{D}$ do
 3:         for all time step $t \leftarrow 1$ to $T$ do
 4:             Compute the state vector $\boldsymbol{s}_t$ by the TDNet $\Psi$;
 5:             Sample an action $a_t \sim \pi_{\Theta'}(a_t|\boldsymbol{s}_t) \in \mathcal{A}$ from the old policy $\pi_{\Theta'}$;
 6:         end for
 7:         Compute the reward $r_T$ of the revised $\mathcal{G}'$ by the TDNet;
 8:     end for
 9:     Sum the objectives of all subsets, obtain $J^{\Theta'}(\Theta)$;
10:     Update the SRNet $\Theta$ by optimizing $J^{\Theta'}(\Theta)$ ($K$ epochs);
11:     $\Theta' \leftarrow \Theta$;
12:     Update the TDNet $\Psi$ using the revised data $\mathcal{D}'$;
13: end for
```

On-Policy Optimization For a subset $\mathcal{G}$, let τ denote a trajectory $\tau = (\boldsymbol{s}_1, a_1, \ldots, \boldsymbol{s}_T, a_T)$ determined by the revision policy π_Θ. The cumulative reward for τ is $R(\tau) = \sum_{t=1}^{T} r_t = r_T$. The objective of SRNet is to maximize the expected cumulative reward $J(\Theta) = \mathbb{E}_{\tau\sim\pi_\Theta}[R(\tau)]$, which can be optimized by stochastic gradient ascent (SGA):

$$\nabla_\Theta J(\Theta) = \mathbb{E}_{\tau\sim\pi_\Theta}[\sum_{t=1}^{T}\big(\gamma^{T-t} r_T\big)\nabla_\Theta \log \pi_\Theta(a_t|\boldsymbol{s}_t)] \tag{11}$$

where γ is discount factor ($\gamma < 1$) (Sutton et al. 2000).

Traditional policy gradient method is "on-policy" learning: in one episode, after sampling a trajectory τ from the policy π_Θ, Θ will be updated *one time* by SGA. The updated π_Θ continues to sample a new trajectory, and then will be updated again. In other words, each sampling trajectory can only be used one time during training, which leads to low utilization.

Off-Policy Optimization If the noisy labeled dataset $\mathcal{D}$ is very large, on-policy revision is inefficient in time. For reusing each sampling trajectory to update the policy *multiple times* in *one* episode, we employ proximal policy optimization (PPO) (Schulman et al. 2017) and modify the original revision policy to "off-policy" learning. More specifically, we use an "old policy" $\pi_{\Theta'}$ to sample trajectories, and the objective can be rewritten via importance sampling trick:

$$J^{\Theta'}(\Theta) = \mathop{\mathbb{E}}_{\tau\sim\pi_{\Theta'}}[\frac{\pi_\Theta(\tau)}{\pi_{\Theta'}(\tau)}R(\tau)] = \mathop{\mathbb{E}}_{\tau\sim\pi_{\Theta'}}[\frac{\prod_{t=1}^{T}\pi_\Theta(a_t|\boldsymbol{s}_t)}{\prod_{t=1}^{T}\pi_{\Theta'}(a_t|\boldsymbol{s}_t)}R(\tau)] \tag{12}$$

In one episode, after sampling a trajectory τ from the old policy $\pi_{\Theta'}$, Θ will be updated *multiple times*, which takes full advantage of each trajectory. Then, the policy $\pi_{\Theta'}$ for sampling is set to the updated π_Θ, and next episode begins.

Stance	Patterns
favor	#gotrump, i will vote trump, #leftists, #trumpforpresident, #trumpisright, #makeamericagreatagain, #benghazi, #boycottnbc, #illegalimmigration
against	#dontvotetrump, #boycotttrump, idiot, #dumptrump, racist, fired, #narcissist, #proudmexican

Table 1: Examples of stance-indicative patterns.

Dataset	# all	**Stance**		
		# *favor*	# *against*	# *neither*
Training	9,624	3,633	5,991	–
Test	707	148	299	260

Table 2: Statistics of the datasets.

Training Procedure of TARM

To provide a warm-start for optimization, we pre-train TDNet and SRNet on the auto-labeled data. Then, we alternately train the two networks (detailed in Algorithm 1), which can provide timely rewards to SRNet for exploring a reliable revision policy. In each episode, we first optimize SRNet guided by TDNet (lines 2 to 11); we then fine-tune TDNet using the revised data given by SRNet (line 12). Such procedure is alternate and the two networks can be improved mutually.

Experiments

Experimental Setup

Dataset We evaluate our TARM on SemEval-2016 task 6.B dataset (Mohammad et al. 2016), the benchmark of weakly supervised stance detection task. The target of interest is "*Donald Trump*". The official organizers provided 78,156 tweet-ids for constructing domain corpus. Table 1 lists the representative stance-indicative patterns we selected. After automatically annotating the domain corpus, we obtain a noisy labeled training set containing 9,624 tweets. Table 2 gives the full statistics of these datasets. Note that both the auto-labeled training set and the test set are class-imbalanced: the number of instances with *against* labels is almost twice as much as that of instances with *favor* labels.

Baseline Approaches We compare TARM with the related approaches of weakly supervised stance detection task:

- **pkudblab** (Wei et al. 2016) The winner system of SemEval 2016 task 6.B is pkudblab, which adopts a convolutional neural network to classify stance in tweets.
- **Gen-STS** (Ebrahimi, Dou, and Lowd 2016a) It models the interactions among stance, target of stance and overall sentiment by an undirected graphical model.
- **SVM-RB-N** (Ebrahimi, Dou, and Lowd 2016b) It employs Markov random fields to integrate user relationships, collectively classifying tweet stances and user stances.
- **BiCond** (Augenstein et al. 2016) The state-of-the-art approach of this task. To learn tweet representation conditioned on the given target, BiCond first uses a biLSTM

Id	Topic Words
$\mathcal{T}^1$	vote, president, need, america, people
$\mathcal{T}^2$	poll, republican, new, campaign, presidential
$\mathcal{T}^3$	mexican, mexico, people, drug, elchapo
$\mathcal{T}^4$	nbc, univision, fire, miss, comment
$\mathcal{T}^5$	illegal, immigration, immigrant, border, american

Table 3: Top-5 words in five topics extracted by BTM.

to encode the target, and then utilizes its cell states as the initial cell states of another biLSTM to encode tweets.

Further, to make full use of target information, we extend BiCond by an attention mechanism used in text entailment:

- **A-BiCond (Attentive BiCond)** We add word-by-word attention (Rocktäschel et al. 2016) into BiCond, modeling the connection between tweet and each target word.

For ablation test, we also compare TARM with its two variants, i.e., TAM and TAM−:

- **TAM** TAM removes the SRNet in TARM, that is, directly training TDNet on the auto-labeled data.
- **TAM−** Compared to TAM, it does not enrich target content and utilizes target content alone to build model.

Model Configuration For BTM, we set the number of topics to $I = 5$ via Elbow method, and we remove both high- and low-frequency words in the domain corpus. Table 3 lists top-5 words in five topics extracted by BTM. For TDNet, hyper-parameters are tuned by 5-fold cross-validation. We first pre-train 200 dimensional word embeddings using Skip-Gram (Mikolov et al. 2013) on the domain corpus. GRU hidden states are also 200 dimensional, and N is set to 2. The optimizer is Adam with 64 mini-batch size and 5e-4 learning rate. We add an ℓ_2 penalty term with 1e-5 coefficient and use dropout with 0.5 ratio after the input layer and the representation layer to relieve overfitting. For SRNet, we set the max number of tweets in one subset to $T = 128$. The update times of PPO in one episode is $K = 10$ (see line 9 in Algorithm 1). The learning rate is 2e-5, and the discount factor is $\gamma = 0.9$.

Evaluation Metrics The average of the F_1-score for *favor* class and the F_1-score for *against* class is the official metric of SemEval-2016 Task 6. We denoted it as F_{avg}.

Experimental Results

Table 4 gives the performance comparison of different approaches for weakly supervised stance detection. A-BiCond achieves marginal improvement over BiCond, indicating that the semantic information carried by the target itself is limited and may not give enough clues to identify stance accurately because more sophisticated attention model also does not achieve great improvement.

Compared with TAM− and BiCond which consider target content alone, TAM incorporates topic information for learning topic-aware tweet representation and significantly outperforms them, especially for *favor* class (over 3% improvement) which is the minority class in the training set. Moreover, the number of trainable parameters in TAM is comparable with that in TAM−. Such two observations demonstrate that taking target-related topic information into account during learning tweet representation is helpful for stance detection and can boost performance without increasing the parameter size of neural network models.

Model	# Para.	Metrics		
		F_{avg}	F_{favor}	$F_{against}$
pkudblab	241K	0.5635	0.5404	0.5866
Gen-STS	–	0.5673	0.5708	0.5638
SVM-RB-N	–	0.5752	0.5427	0.6077
BiCond	1.29M	0.5794	0.5455	0.6133
A-BiCond	1.45M	0.5819	0.5508	0.6130
TAM−	643K	0.5772	0.5418	0.6125
TAM	644K	$0.5958^{\ddagger}$	$0.5778^{\dagger}$	0.6138
TARM	649K	$\mathbf{0.6078}^{\ddagger,*}$	$\mathbf{0.5978}^{\ddagger,**}$	**0.6178**

Table 4: Performance comparison. The column "# Para." lists the number of trainable parameters (excluding the word embedding matrix) in neural models. "†" ("$*$") and "‡" ("$**$") mean that the results outperform BiCond (TAM) by paired t-test at the significance levels of 0.01 and 0.001, respectively.

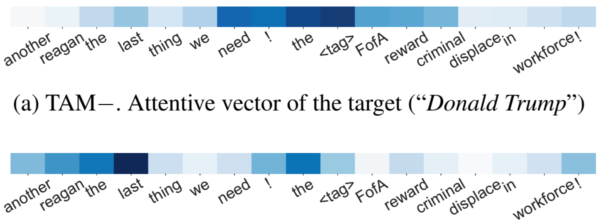
Further, TARM introduces an RL-based revision policy to eliminate noisy instances and achieves the best performance of all three metrics. This result illustrates that TARM is superior to the models which are directly built on noisy stance labeled data. Next, we will give more concrete analysis of different components in TARM.

Effectiveness of Incorporating Topic Information

Although the above results show that TAM performs better than TAM− and BiCond, we need to take a further step for understanding the effect of topic information. Table 5 shows two tweets whose stances are wrongly predicted by TAM− and BiCond but accurately predicted by TAM. The overall sentiment of the first tweet is *negative* but the targeted stance is *favor*. As we can see, although this tweet mentions the target "*Donald Trump*", it stands the viewpoint through expressing a negative attitude towards "*NBC*" (topic $\mathcal{T}^4$). Both TAM− and BiCond confuse the targeted stance with the overall sentiment of the tweet.

Tweet	Stance
Kudos to Donald Trump for telling off NBC. The media HATES the GOP'S & will do ANYTHING to destroy them. #NBC #mediaatfault	True: *favor* TAM: *favor* TAM−: *against* BiCond: *against*
Another Reagan the last thing we need! The #FatherofAmnesty rewarded 3m criminals, displaced 1.9m in workforce! @username	True: *against* TAM: *against* TAM−: *favor* BiCond: *favor*

Table 5: Two examples: Only TAM predicts the stances accurately, while both TAM− and BiCond fail to predict.



(a) TAM−. Attentive vector of the target ("*Donald Trump*")

(b) TAM. Attentive vector of the topic $\mathcal{T}^1$ ("*vote*, *president*, ...").

Figure 3: Visualization results of attentive vectors produced by TAM− and TAM, respectively.

The second tweet does not explicitly express an attitude towards the target, but it exists a span "*we need*". In Figure 3, we visualize the attentive vectors produced by TAM− and TAM to investigate how attention mechanism captures tweet spans with the influences of the target itself and the topic "*vote*, *president*, *need*, *america*, *people*" ($\mathcal{T}^1$), respectively. We can clearly observe that the topic $\mathcal{T}^1$ accurately captures the span "*the last*". However, the target highlights "*need ! the #FofA*" and fail to identify targeted stance. Therefore, integrating topic information is effective to overcome the target-related implicit expression issue in stance detection.

Effectiveness of RL-based Revision Policy

We discuss the effectiveness of the proposed revision policy for noisy labeled data from two perspectives. First, we compare off-policy optimization with the on-policy one on performance and training time cost. Second, we measure the utility of the revised dataset provided by TARM.

On-Policy vs. Off-Policy Revision Table 6 shows the comparisons between on-policy and off-policy training strategies. The advantage of off-policy optimization is evident: (1) off-policy training is eight times faster than on-policy training in the pre-training phase, and three times faster than that in the joint training phase; (2) off-policy training also achieves higher stance detection performance. In practice, on-policy TARM needs around 100 episodes for pre-training and 15 episodes for joint training, while off-policy TARM only needs 10 episodes and 5 episodes respectively. Consequently, off-policy revision shortens the training time cost significantly and meets the need of real-world scenarios.

Optimization Method	Metric	Revision Time	
	F_{avg}	Pre-train	Joint
TARM (On-policy)	0.6003	∼60min	∼12min
TARM (Off-policy)	0.6078	∼7min	∼4min

Table 6: Performance and training time comparisons between on-policy and off-policy optimization for TARM. Both of them are trained on a single GPU.

Utility of the Revised Dataset We train TAM, BiCond and A-BiCond on the revised dataset, and compare their performance with the original models trained on the noisy labeled dataset. As shown in Table 7, replacing the noisy labeled data by the revised data to train models can provide a performance boost, thus our RL-based revision policy can improve the quality of auto-labeled data. We also note that the performance of TAM trained on the revised dataset does not outperform TARM, which indicates that first pre-training on large-scale auto-labeled data and then jointly training is more adequate for the weakly supervised setting.

Model	Dataset for Training	
	Auto-labeled dataset	Revised dataset
BiCond	0.5794	0.5869
A-BiCond	0.5819	0.5897
TAM	0.5958	0.5989

Table 7: Performance comparison between training on the auto-labeled dataset and the revised dataset. Metric: F_{avg}.

Noisy Labeled Tweet
I don't believe #Donald Trump is really a racist. Probably going to profit off this BorderWall he's proposing regardless of who wins. (Pattern: racist)
@realDonaldTrump Mr. Donald Trump only you can say, "You are fired," to illegals. Thank you for standing with Americans. #Trump2016 (Pattern: fired)

Table 8: Two noisy labeled tweets eliminated by SRNet.

Illustration Table 8 presents two tweets eliminated from the auto-labeled dataset by SRNet during training. Using the first tweet as an example, although this tweet matches the against-target pattern "racist", it does not express an *against* stance towards the target. This shows that TARM can effectively eliminate incorrectly labeled instances.

Conclusion

This paper focuses on two challenging issues of weakly supervised stance detection: target-related implicit expression issue and noisy stance labeling issue. To address them, we propose a novel topic-aware reinforced model TARM, which overcomes the first issue by integrating topic information into representation learning, and introduces an RL-based revision policy to reduce the negative impact caused by the second issue. Experimental results show that TARM outperforms existing approaches. In further work, we shall explore how to construct high-quality stance-indicative patterns automatically, and consider the confidence of different patterns.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology of China under Grant #2016QY02D0305, NSFC Grants #71621002, #71702181 and #71472175, and CAS Key Program under Grant #ZDRW-XH-2017-3. We are grateful to the anonymous reviewers for their valuable comments. We also thank Junjie Lin for his kind assistance.

References

Augenstein, I.; Rocktäschel, T.; Vlachos, A.; and Bontcheva, K. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of EMNLP*, 876–885.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Benton, A., and Dredze, M. 2018. Using author embeddings to improve tweet stance classification. In *Proceedings of W-NUT@EMNLP*, 184–194.

Chen, W.-F., and Ku, L.-W. 2016. UTCNN: A deep learning model of stance classification on social media text. In *Proceedings of COLING*, 1635–1645.

Cheng, X.; Yan, X.; Lan, Y.; and Guo, J. 2014. BTM: Topic modeling over short texts. *IEEE Transactions on Knowledge & Data Engineering (TKDE)* 26(12):2928–2941.

Dey, K.; Shrivastava, R.; and Kaushik, S. 2018. Topical stance detection for Twitter: A two-phase LSTM model using attention. In *Proceedings of ECIR*, 529–536.

Du, J.; Xu, R.; He, Y.; and Gui, L. 2017. Stance classification with target-specific neural attention networks. In *Proceedings of IJCAI*, 3988–3994.

Ebrahimi, J.; Dou, D.; and Lowd, D. 2016a. A joint sentiment-target-stance model for stance classification in tweets. In *Proceedings of COLING*, 2656–2665.

Ebrahimi, J.; Dou, D.; and Lowd, D. 2016b. Weakly supervised tweet stance classification by relational bootstrapping. In *Proceedings of EMNLP*, 1012–1017.

Feng, J.; Huang, M.; Zhao, L.; Yang, Y.; and Zhu, X. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*, 5779–5786.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*.

Hasan, K. S., and Ng, V. 2013. Extra-linguistic constraints on stance recognition in ideological debates. In *Proceedings of ACL*, 816–821.

Lippi, M., and Torroni, P. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)* 16(2):10.

Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5(1):1–167.

Ma, Y.; Peng, H.; and Cambria, E. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *Proceedings of AAAI*, 5876–5883.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 3111–3119.

Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; and Cherry, C. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of SemEval*, 31–41.

Mohtarami, M.; Baly, R.; Glass, J.; Nakov, P.; Màrquez, L.; and Moschitti, A. 2018. Automatic stance detection using end-to-end memory networks. In *Proceedings of NAACL*, 767–776.

Qin, P.; Xu, W.; and Wang, W. Y. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of ACL*, 2137–2147.

Rajadesingan, A., and Liu, H. 2014. Identifying users with opposing opinions in Twitter debates. In *Proceedings of SBP*, 153–160.

Rocktäschel, T.; Grefenstette, E.; Hermann, K. M.; Kočiskỳ, T.; and Blunsom, P. 2016. Reasoning about entailment with neural attention. In *Proceedings of ICLR*.

Sasaki, A.; Hanawa, K.; Okazaki, N.; and Inui, K. 2018. Predicting stances from social media posts using factorization machines. In *Proceedings of COLING*, 3381–3390.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of ICLR*.

Somasundaran, S., and Wiebe, J. 2009. Recognizing stances in online debates. In *Proceedings of ACL-IJCNLP*, 226–234.

Sun, Q.; Wang, Z.; Zhu, Q.; and Zhou, G. 2018. Stance detection with hierarchical attention network. In *Proceedings of COLING*, 2399–2409.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NIPS*, 1057–1063.

Takanobu, R.; Huang, M.; Zhao, Z.; Li, F.-L.; Chen, H.; Zhu, X.; and Nie, L. 2018. A weakly supervised method for topic segmentation and labeling in goal-oriented dialogues via reinforcement learning. In *Proceedings of IJCAI*, 4403–4410.

Thomas, M.; Pang, B.; and Lee, L. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of EMNLP*, 327–335.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of NIPS*, 5998–6008.

Wei, W.; Zhang, X.; Liu, X.; Chen, W.; and Wang, T. 2016. pkudblab at SemEval-2016 task 6: A specific convolutional neural network system for effective stance detection. In *Proceedings of SemEval*, 384–388.

Wei, P.; Lin, J.; and Mao, W. 2018. Multi-target stance detection via a dynamic memory-augmented network. In *Proceedings of SIGIR*, 1229–1232.

Wei, P.; Mao, W.; and Zeng, D. 2018. A target-guided neural memory model for stance detection in Twitter. In *Proceedings of IJCNN*, 2068–2075.

Xia, R.; Jiang, J.; and He, H. 2017. Distantly supervised lifelong learning for large-scale social media sentiment analysis. *IEEE Transactions on Affective Computing (TAC)* 8(4):480–491.

Xu, C.; Paris, C.; Nepal, S.; and Sparks, R. 2018. Cross-target stance classification with self-attention networks. In *Proceedings of ACL*, 778–783.

Zarrella, G., and Marsh, A. 2016. MITRE at SemEval-2016 task 6: Transfer learning for stance detection. In *Proceedings of SemEval*, 458–463.

Zhou, Y.; Cristea, A. I.; and Shi, L. 2017. Connecting targets to tweets: Semantic attention-based model for target-specific stance detection. In *Proceedings of WISE*, 18–32.