A Target-Guided Neural Memory Model for Stance Detection in Twitter

Penghui Wei*†, Wenji Mao*†, and Daniel Zeng*‡
*State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
†University of Chinese Academy of Sciences, Beijing 100049, China
‡Department of Management Information Systems, The University of Arizona, Tucson, AZ 85721, USA
Email: {weipenghui2016, wenji.mao, dajun.zeng}@ia.ac.cn

Abstract—**Exploring user stances and attitudes is beneficial to a number of Web related research and applications, especially in social media platforms such as Twitter. Stance detection in Twitter aims at identifying the stance expressed in a tweet towards a given target (e.g., a government policy). A key challenge of this task is that a tweet may not explicitly express opinion about the target. To effectively detect user stances implied in tweets, target content information plays an important role. In previous studies, conventional feature-based methods often ignore target content. Although more recent neural network-based methods attempt to integrate target information using attention mechanism, the performance improvement is rather limited due to the underuse of this information. To address this issue, we propose an end-to-end neural model, *TGMN-CR*, which makes better use of target content information. Specifically, our model first learns conditional tweet representation with respect to specific target. It then employs a target-guided iterative process to extract crucial stance-indicative clues via multiple interactions between target and tweet words. Experimental results on SemEval-2016 Task 6.A Twitter Stance Detection dataset show that our proposed method outperforms the state-of-the-art alternative methods, and substantially outperforms the comparative methods when a tweet does not explicitly express opinion about the given target.**

I. Introduction

Exploring user stances and attitudes has attracted increasing research attention in recent years [1], [2]. Stance detection is the task of determining from a text whether its author is in favor of, against, or neutral towards a target entity [1]. The target of interest may be a movement, a government policy, a person, a product, etc. Automatically identifying user stances towards a specific target by analyzing user-generated contents online can help us understand what people think and believe. It is not only a fundamental research topic, but also has widespread applications in decision-making processes, business intelligence, social event monitoring, public security and many others.

Over the past few years, stance detection has been widely studied from offline scenarios such as congressional debates [3] and student essays [4] to online ones such as debate forums [5]–[9]. More recently, stance detection centers on analyzing textual data in popular social media platforms, e.g., Twitter [10], Sina Weibo [11] and Facebook [9], and Twitter has become the representative platform for stance detection research. SemEval-2016 introduced a shared task aiming at detecting stance in tweets [12]. Consider the following tweet-target pair:

Tweet: *It's not always the guys job. #equality*
Target: *Feminist Movement*

From the content of this tweet, one can figure out that its author is in favor of feminist movement, and find some characteristics of this task. First, stance detection is a task different from sentiment classification. The tweet in this example has negative sentiment but its stance towards the target is supportive. Second, a tweet may not explicitly express opinion about the target, thus stance detection is also different from aspect-based sentiment analysis (ABSA) [13] in which aspects are always assumed to be discussed in texts. Due to these characteristics, detecting stance in tweets is a challenging task in social media analytics.

Previous studies have proposed different methods for stance detection in Twitter, mainly including feature-based methods and neural network based models. Feature-based methods like [1] and [14] usually extract hand-crafted text features from tweets and then train stance classifiers. Bag-of-words (BoW) and its n-gram extensions are commonly used. As discrete features fail to capture the semantic relatedness between words or phases, they often perform poorly when a tweet implicitly expresses opinion about the target.

In contrast, neural networks possess the capacity of semantic composition through distributed vector representation [15]. Moreover, attention mechanism [16] is promising in many tasks, attributed to its capability of capturing salient parts of input information. Typical work in [17] and [18] utilizes neural networks to encode targets and tweets, and then learns to focus on important words of a tweet given the specific target using attention model. Both work gets better performance than other neural models which do not consider target information.

However, previous neural models for Twitter stance detection usually did not take full advantage of target information due to two main reasons. First, they usually did not take target content into account during tweet encoding. Intuitively, learning target-specific tweet representation is beneficial to detecting stance towards the specific target. Second, the ways of using attention, e.g., in [17] and [18], did not adequately model the interaction process between target and tweet words, and thus may not correctly extract the most vital information for

Accepted by IJCNN 2018.

stance detection. Consequently, although feature-based methods for stance detection often ignore target information, they can achieve performance comparable with those in [17] and [18] due to the underuse of target content information in current neural stance detection models.

To address the above issues, in this paper, we propose an end-to-end neural model for Twitter stance detection, i.e., *TGMN-CR* (Target-Guided Memory Network with Conditional Representation), which leverages the power of target content information. *TGMN-CR* first uses a bidirectional gated recurrent unit network (BiGRU) to learn the vector representation of target, called *target embedding*. To impart target information into tweet representation, another BiGRU whose hidden state is initialized with target embedding is used to obtain tweet representation. It then employs a target-guided iterative process which maintains an updatable *memory vector* to store crucial clues for stance detection. During each iteration, a soft attention mechanism guided by target first learns to highlight the salient parts of input tweet and summarizes its contextual information, and then a memory update mechanism incorporates previous clues, current contextual information, and target information to obtain new clues. By means of this iterative process, multiple interactions of target and tweet words are modeled, and clues for detecting stance are accumulated during each iteration. At the end of the last iteration, the newest clues and target embedding are used to predict stance distribution.

Our work has made several contributions. (1) We propose a neural memory model *TGMN-CR* for detecting stance in tweets, which makes better use of target content information. (2) *TGMN-CR* learns target-specific tweet representation, and extracts crucial stance-indicative clues via multiple interactions between target and tweet words. (3) We conduct experiments on SemEval-2016 Task 6.A benchmark dataset, and the performance of *TGMN-CR* outperforms the state-of-the-art alternative methods. Moreover, it performs substantially better than comparative methods when a tweet does not explicitly express opinion about the target.

II. Related Work

We first briefly review early stance analysis studies in debate texts, and then focus on recent studies of detecting stance in tweets, including feature- and neural network- based methods.

Stance detection has been widely studied on congressional debates [3] and online debate forums [5]–[9]. Both textual features (e.g., n-grams) and extra-linguistic features are utilized in previous work. To explore arguing opinion, Somasundaran and Wiebe [5] constructed an arguing lexicon, and used arguing features such as modal verbs and sentiment features to build a classifier. To take into account the dialogic context of posts, Anand *et al.* [6] constructed both parent features and post features. Because textual features are sometimes limited, Hasan and Ng [7] used extra-linguistic features such as user-interaction constraints and ideology constraints to improve performance. Recently, neural networks are also used for stance detection on forums. Chen and Ku [9] proposed a neural model that utilizes user behavior, topic distribution and comment information.

With the fast development of social media platforms, identifying user attitudes from social media contents like Twitter has gained increasing research attention in recent years. The earlier work [10] studied stance analysis in Twitter, which proposed a retweet-based label propagation method coupled with a classifier to gauge users' stance. More recently, some studies proposed different kinds of methods for detecting stance in tweets. Feature-based methods extracted text features, e.g., n-grams [1] and repeated vowels [14], to train stance classifiers. Because discrete features can not capture the semantic relatedness between words, their performance are relatively low when a tweet implicitly talks about the target. Although these methods are commonly used, another drawback of feature-based method is that they rarely use target content information. Several other studies employed graphical models to integrate extra information, e.g., sentiment polarity labels [19], user networks [20], as supplement for Twitter stance detection.

Compared with feature-based methods, many recent studies apply neural networks, e.g., convolutional neural networks (CNNs) [21] and recurrent neural networks (RNNs) [17], [18], [22], [23], to learn tweet representation automatically. In these methods, they proposed various ways to integrate target information in neural models. To tackle the sparseness problem of target-related data, Zarrella and Marsh [22] used transfer learning to pre-train a long short-term memory network (LSTM) on a large unlabelled corpus for target-related hashtag prediction task. To address the weakly supervised task that no labeled training data is available for the targets existing in testing data, Augenstein *et al.* [23] utilized conditional encoding [24] to model target and tweet jointly, and demonstrated that it performs better than independent modeling.

The neural models mentioned above do not consider that not all words make equal contribution to the semantic meaning of a tweet. Attention mechanism [16] provides a means for finding important parts of input information in neural models. To find more useful words in a tweet, Du *et al.* [18] proposed a target-specific attention model, which performs better than neural models that do not integrate target information. For the similar consideration, Zhou *et al.* [17] extended the token-level attention by a gated structure. It outperforms previous neural models, but does not significantly perform better than feature-based methods. This is partially due to the "single-use" attention (i.e., using attention one time) utilized in [17] and [18] may not highlight the information for detecting stance correctly. To overcome the limitations in previous work, we consider that crucial information for stance detection towards a target should be updated and accumulated with the process of multiple interactions between the target and tweet words.

Based on the ideas of attention and external memory, Sukhbaatar *et al.* [25] proposed end-to-end memory networks that adaptively select the relevant memory slices from an external memory multiple times to tackle question answering task. Variants of memory networks [26], [27] have also been proposed and applied on other tasks. Motivated by these work, we propose a neural memory model which obtains crucial clues via multiple interactions between target and tweet words.

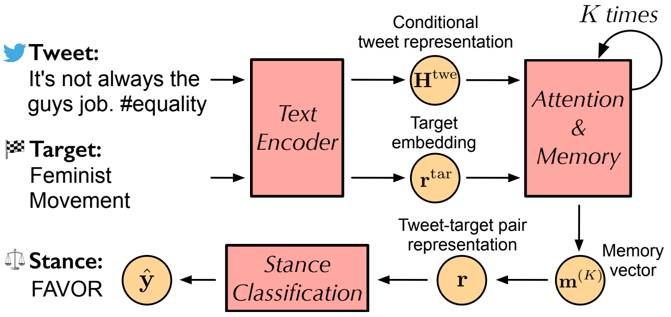


Fig. 1. Overall architecture of our model.

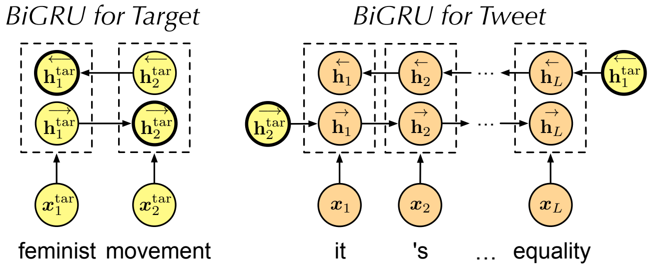

BiGRU for Target
BiGRU for Tweet
$\overleftarrow{\mathbf{h}}_1^{\text{tar}}$
$\overleftarrow{\mathbf{h}}_2^{\text{tar}}$
$\overrightarrow{\mathbf{h}}_1^{\text{tar}}$
$\overrightarrow{\mathbf{h}}_2^{\text{tar}}$
$\boldsymbol{x}_1^{\text{tar}}$
$\boldsymbol{x}_2^{\text{tar}}$
feminist movement
$\overrightarrow{\mathbf{h}}_2^{\text{tar}}$
$\overleftarrow{\mathbf{h}}_1$
$\overleftarrow{\mathbf{h}}_2$
$\overleftarrow{\mathbf{h}}_L$
$\overleftarrow{\mathbf{h}}_1^{\text{tar}}$
$\overrightarrow{\mathbf{h}}_1$
$\overrightarrow{\mathbf{h}}_2$
$\overrightarrow{\mathbf{h}}_L$
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$\boldsymbol{x}_L$
it 's ... equality

Fig. 2. Target embedding and conditional tweet representation.

III. Problem Formulation

Twitter stance detection task can be defined as follows. Given a tweet and a pre-defined target, a model classifies the tweeter's stance towards the target into one of the following three classes: FAVOR, AGAINST, and NONE. Note that a tweet may not explicitly express opinion about the target. In this paper, we focus on supervised task that the targets existing in testing set have labeled samples in training set.

Formally, we denote the input tweet as $(w_1, w_2, \ldots, w_L)$, where w_i is the i-th token in the tweet and L is the length of the tweet. Similarly, we denote the input target as $(w_1^{\text{tar}}, w_2^{\text{tar}}, \ldots, w_l^{\text{tar}})$. We use a word embedding matrix $\boldsymbol{X} \in \mathbb{R}^{e \times |\mathbb{V}|}$ to map each word w into a continuous real-valued vector $\boldsymbol{x} \in \mathbb{R}^e$, where $|\mathbb{V}|$ is the vocabulary size. The tweet and the target then can be denoted as $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L)$ and $(\boldsymbol{x}_1^{\text{tar}}, \boldsymbol{x}_2^{\text{tar}}, \ldots, \boldsymbol{x}_l^{\text{tar}})$, respectively.

IV. Proposed Method

Fig. 1 shows the overall architecture of *TGMN-CR*. There are three main modules in our model. *Text Encoder* module first learns the fix-length vector representation of the given target, namely target embedding, and then learns the target-specific tweet representation which is conditional on the target. Afterwards, *Attention and Memory* module, the core component of our model, runs a target-guided iterative process on the conditional tweet representation to iteratively update the memory vector which stores crucial clues for detecting the stance towards the given target. The target embedding and the lastest memory vector are concatenated to get the vector representation of the tweet-target pair, and it is fed into *Stance Classification* module to predict the stance. The proposed model is fully differentiable and trained end-to-end.

A. Text Encoder Module

*1) **Target embedding**:* Gated recurrent unit network (GRU) is an effective model for encoding sequence data [28]. It introduces a gating mechanism to control the way of information updating without using extra memory cells. A GRU is used over the input sequence to generate a hidden state vector sequence.

For learning the representation by using both past and future input features efficiently, we use a bidirectional GRU (BiGRU) which contains a forward GRU and a backward GRU to obtain the vector representation of a target, depicted in the left part of Fig. 2. The forward GRU reads the input sequence in the original direction and the backward GRU reads it in the opposite direction. Hence, BiGRU can learn both the past and the future features of a position in a sequence.

Given the target $(\boldsymbol{x}_1^{\text{tar}}, \boldsymbol{x}_2^{\text{tar}}, \ldots, \boldsymbol{x}_l^{\text{tar}})$, at each time step $t \in [1, l]$, the forward GRU for target uses the input vector $\boldsymbol{x}_t^{\text{tar}}$ and the previous hidden state vector $\overrightarrow{\mathbf{h}}_{t-1}^{\text{tar}}$ to compute the current hidden state vector $\overrightarrow{\mathbf{h}}_t^{\text{tar}} \in \mathbb{R}^d$:

$$\overrightarrow{\mathbf{h}}_t^{\text{tar}} = \overrightarrow{\text{GRU}}^{\text{tar}}(\boldsymbol{x}_t^{\text{tar}}, \overrightarrow{\mathbf{h}}_{t-1}^{\text{tar}}) \,, \tag{1}$$

where t ranges from 1 to l. While the backward GRU for target runs this process in the opposite direction:

$$\overleftarrow{\mathbf{h}}_t^{\text{tar}} = \overleftarrow{\text{GRU}}^{\text{tar}}(\boldsymbol{x}_t^{\text{tar}}, \overleftarrow{\mathbf{h}}_{t+1}^{\text{tar}}) \,, \tag{2}$$

where t ranges from l to 1.

The initial hidden states of the BiGRU for target are zero vectors, i.e., $\overrightarrow{\mathbf{h}}_0^{\text{tar}} = \overleftarrow{\mathbf{h}}_{l+1}^{\text{tar}} = \mathbf{0}$. We concatenate the last hidden states to obtain the target embedding $\mathbf{r}^{\text{tar}}$:

$$\mathbf{r}^{\text{tar}} = [\overrightarrow{\mathbf{h}}_l^{\text{tar}}; \overleftarrow{\mathbf{h}}_1^{\text{tar}}] \in \mathbb{R}^{2d} \,, \tag{3}$$

where $[\cdot;\cdot]$ is the concatenation operator of two vectors.

*2) **Conditional tweet representation**:* We then use another BiGRU for encoding tweets, depicted in the right part of Fig. 2. As we mentioned in Section I, intuitively, incorporating the target information into tweet representation is beneficial to stance detection. To integrate target semantic information to the process of modeling tweet, in contract to the BiGRU for target in which initial hidden states are zero vectors, the initial states of the BiGRU for tweet depend on target embedding. Specifically, the states are initialized as follows:

$$\overrightarrow{\mathbf{h}}_0 = \overrightarrow{\mathbf{h}}_l^{\text{tar}} \,, \quad \overleftarrow{\mathbf{h}}_{L+1} = \overleftarrow{\mathbf{h}}_1^{\text{tar}} \,. \tag{4}$$

The BiGRU for tweet encodes the tweet in the similar way:

$$\overrightarrow{\mathbf{h}}_t = \overrightarrow{\text{GRU}}(\boldsymbol{x}_t, \overrightarrow{\mathbf{h}}_{t-1}) \,, \quad \overleftarrow{\mathbf{h}}_t = \overleftarrow{\text{GRU}}(\boldsymbol{x}_t, \overleftarrow{\mathbf{h}}_{t+1}) \,, \tag{5}$$

where $t \in [1, L]$. We obtain the hidden state of each time step by concatenating the forward state and the backward state:

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t] \in \mathbb{R}^{2d} \,, \tag{6}$$

$$\mathbf{H}^{\text{twe}} = \text{concat\_col}(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L) \in \mathbb{R}^{2d \times L} \,. \tag{7}$$

The tweet representation $\mathbf{H}^{\text{twe}}$ is conditional on the given target. Hence, target-specific information is incorporated into the process of tweet modeling.

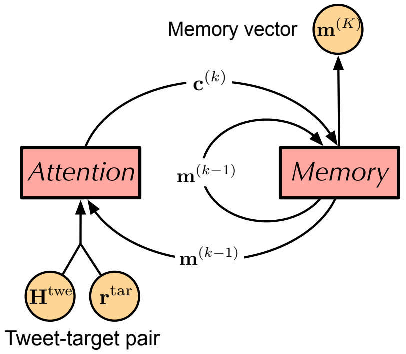


Fig. 3. Target-guided iterative process for extracting clues.

B. Attention and Memory Module

This module aims at extracting crucial information for detecting stance exactly. It is evidently that not all words in a tweet make the same contribution to the attitude expression of this tweet. Target content can guide the model to know such words that are more useful than other words in a tweet. Because a tweet may implicitly express stance towards the given target, we suggest that multiple interactions between target and tweet words can be conducive to extract crucial clues for stance detection.

To achieve this objective, inspired by memory networks [25]–[27] which utilize multiple computational steps to obtain useful information, we employ a target-guided iterative process, depicted in Fig. 3. Specifically, this module maintains an updatable vector called *memory vector* to store crucial clues for stance detection. In each iteration, there is a two-stage process. First, a soft attention mechanism guided by target learns to highlight the words which make obvious contributions to stance expression, and summarizes the semantic information of tweet as a *contextual vector*. Then a memory update mechanism updates the memory vector by incorporating previous memory, current contextual information and target information.

With the running of the process of multiple interactions, the fusion of low-level representation can obtain the high-level representation of clues.

*1) **Target-guided attention mechanism**:* We denote the total number of iterations in this module as K. In the k-th iteration ($k \in [1, K]$), a one-layer network is used as the score function to compute the importance scores $(e_1^{(k)}, e_2^{(k)}, \ldots, e_L^{(k)})$ for each token in the tweet $(w_1, w_2, \ldots, w_L)$, guided by the target. Softmax function is used for normalization to produce the attention weights $(\alpha_1^{(k)}, \alpha_2^{(k)}, \ldots, \alpha_L^{(k)})$:

$$e_i^{(k)} = \tanh(\boldsymbol{w}_{\text{att}}^{\top}[\mathbf{r}^{\text{tar}}; \mathbf{m}^{(k-1)}; \mathbf{h}_i] + b_{\text{att}}), \quad i \in [1, L], \tag{8}$$

$$\alpha_i^{(k)} = \text{Softmax}(e_i^{(k)}) = \frac{\exp(e_i^{(k)})}{\sum_{j=1}^{L} \exp(e_j^{(k)})}, \tag{9}$$

where $\mathbf{m}^{(k-1)} \in \mathbb{R}^{2d}$ is the memory produced by the previous iteration, $\boldsymbol{w}_{\text{att}} \in \mathbb{R}^{6d}$ and $b_{\text{att}} \in \mathbb{R}$ are trainable parameters. We then compute the weighted sum of the tweet representation $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L)$ to generate the contextual vector $\mathbf{c}^{(k)} \in \mathbb{R}^{2d}$:

$$\begin{aligned} \mathbf{c}^{(k)} &= \sum_{i=1}^{L} \alpha_i^{(k)} \mathbf{h}_i \\ &= \mathbf{H}^{\text{twe}} \boldsymbol{\alpha}^{(k)}, \end{aligned} \tag{10}$$

where $\boldsymbol{\alpha}^{(k)} = (\alpha_1^{(k)}, \alpha_2^{(k)}, \ldots \alpha_L^{(k)})^{\top}$ and $\sum_{i=1}^{L} \alpha_i^{(k)} = 1$. In this way, our model learns to attend more useful words in tweets and give them higher weights when building contextual vector.

*2) **Memory update mechanism**:* In the k-th iteration, after obtaining the contextual information $\mathbf{c}^{(k)}$, we need to update the memory vector, i.e., the representation of clues. To update the memory from $\mathbf{m}^{(k-1)}$ to $\mathbf{m}^{(k)}$ by taking into account both the current contextual information and the target information, we define a memory update function $f_{\text{m}}(\cdot)$:

$$\mathbf{m}^{(k)} = \begin{cases} \mathbf{r}^{\text{tar}}, & k = 0, \\ f_{\text{m}}(\mathbf{m}^{(k-1)}, \mathbf{c}^{(k)}, \mathbf{r}^{\text{tar}}), & k \in [1, K]. \end{cases} \tag{11}$$

The update function $f_{\text{m}}(\cdot)$ can be set to an RNN unit like GRU or LSTM, a fully-connected layer, etc. Here, we use the following ReLU layer to obtain new memory:

$$\mathbf{m}^{(k)} = \text{ReLU}(\boldsymbol{W}_{\text{m}}^{(k)}[\mathbf{m}^{(k-1)}; \mathbf{c}^{(k)}; \mathbf{r}^{\text{tar}}] + \boldsymbol{b}_{\text{m}}^{(k)}), \tag{12}$$

where $\boldsymbol{W}_{\text{m}}^{(k)} \in \mathbb{R}^{2d \times 6d}, \boldsymbol{b}_{\text{m}}^{(k)} \in \mathbb{R}^{2d}$ are trainable parameters, and $\text{ReLU}(\cdot) = \max(0, \cdot)$.

After K iterations, the final memory vector $\mathbf{m}^{(K)}$ stores the newest clues for stance detection.

C. Stance Classification Module

We concatenate the target embedding $\mathbf{r}^{\text{tar}}$ and the final memory vector $\mathbf{m}^{(K)}$ to produce the tweet-target pair vector presentation $\mathbf{r}$, and we then feed it into a fully-connected layer with Softmax function for stance classification:

$$\mathbf{r} = [\mathbf{m}^{(K)}; \mathbf{r}^{\text{tar}}], \tag{13}$$

$$\hat{\mathbf{y}} = \text{Softmax}(\boldsymbol{W}_{\text{sc}}\mathbf{r} + \boldsymbol{b}_{\text{sc}}), \tag{14}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^3$ is the predicted stance distribution. $\boldsymbol{W}_{\text{sc}} \in \mathbb{R}^{3 \times 4d}$ and $\boldsymbol{b}_{\text{sc}} \in \mathbb{R}^3$ are trainable parameters.

D. End-to-End Training

The goal of the training process is to minimize the objective function $\mathcal{J}$, which uses cross-entropy as the loss function:

$$\mathcal{J} = -\frac{1}{N} \sum_{i=1}^{N} {\mathbf{y}_i}^{\top} \log \hat{\mathbf{y}}_i + \lambda \left\|\Theta\right\|^2, \tag{15}$$

where N is the number of training samples. $\hat{\mathbf{y}}_i \in \mathbb{R}^3$ denotes the predicted stance distribution of the i-th training sample, and $\mathbf{y}_i \in \{0, 1\}^3$ denotes the ground truth which is a one-hot vector. Θ denotes all trainable weights, and ℓ_2-regularization term with coefficient λ is used to alleviate overfitting.

Our model is trained end-to-end through stochastic gradient descent (SGD) with back-propagation to compute gradients.

TABLE I
STATISTICS OF INSTANCES IN THE STANCE DETECTION DATASET

Dataset	#total	Opinion towards (%)			% of instances in Train				% of instances in Test			
		Target	Other	No one	#train	FAVOR	AGAINST	NONE	#test	FAVOR	AGAINST	NONE
A.	733	49.3	46.4	4.4	513	17.9	59.3	22.8	220	14.5	72.7	12.7
C.C.C.	564	60.8	30.5	8.7	395	53.7	3.8	42.5	169	72.8	6.5	20.7
F.M.	949	68.3	27.4	4.3	664	31.6	49.4	19.0	285	20.4	64.2	15.4
H.C.	984	60.3	35.1	4.6	689	17.1	57.0	25.8	295	15.3	58.3	26.4
L.A.	933	63.7	31.0	5.4	653	18.5	54.4	27.1	280	16.4	67.5	16.1
Total	4163	61.0	33.8	5.2	2914	25.8	47.9	26.3	1249	23.1	51.8	25.1

V. Experiments

We conduct experiments on a benchmark dataset and analyze the effectiveness of different modules in our *TGMN-CR* model.

A. Dataset Description

We adopt SemEval-2016 Task 6.A dataset [12], the benchmark for Twitter stance detection, as the experimental data. It contains five targets related to several different topics such as ideology, movement and specific person: “Atheism” (A.), “Climate Change is a Real Concern” (C.C.C.), “Feminist Movement” (F.M.), “Hillary Clinton” (H.C.) and “Legalization of Abortion” (L.A.). Each tweet-target pair is labeled with one of the stance labels: “FAVOR”, “AGAINST” and “NONE”.

Furthermore, according to whether the tweet of a tweet-target pair explicitly expresses opinion about the target, each pair is annotated with “Target”, “Other”, or “No one”. A pair labeled with “Target” means that the tweet expresses opinion about the target directly, while labeled with “Other” means that the tweet does not express opinion towards the target but it has opinion about something other than this target. If the tweet of a pair does not express any opinion, the pair will be labeled with “No one”. Table I lists the statistics of instances.

B. Baseline Methods

We do not utilize extra domain corpus, or extra supervised information beyond stance labels. Therefore, we choose the following methods as baselines for comparison. According to whether the model considers target information, we classify all methods used for comparison into two types.

*1) **Type I: without target information**:* Methods which only consider tweet information belong to this type. They include:

- *SVM* [12]. It extracts common text features and then trains a support vector machine (SVM) classifier.
- *NBoW*. Neural bag-of-words model (NBoW) sums word embeddings in a tweet to obtain tweet representation which is fed into a Softmax layer for classification.
- *CNN* [21]. A CNN is used as feature extractor, which contains a convolutional layer and a max-pooling layer [29].
- *biGRU* [17]. It uses a BiGRU to model tweet and feeds the last hidden state into a Softmax layer.

*2) **Type II: with target information**:* Methods considering both tweet and target content belong to this type. They include:

- *AT-biGRU* [17]. Two BiGRUs are used to represent target and tweet respectively. A token-level attention mechanism is applied to find the important words in a tweet.
- *TAN* [18]. It is similar to the *ATAE-LSTM* proposed by [30]. A target-specific attention function scores the importance of each word in a tweet.
- *AS-biGRU-CNN* [17]. It extends the attention used in *AT-biGRU* by a gated structure, and stacks a CNN on top.
- *MemN2N*. We implemented end-to-end memory networks (MemN2N) [25] for stance detection, which encode tweet into an external memory and utilize target to make multiple computational steps (hops) on the memory. The number of hops was set to 5.

C. Implementation Details

The whole dataset can be divided into five subsets through five different targets. For each target, we built model using the subset corresponding to this target. Separate models for different targets share the same configuration.[1]

Word embeddings were initialized by GloVe 100-dimensional pre-trained embeddings on Twitter corpus [31], and out-of-vocabulary words were initialized by sampling from the uniform distribution $U(-0.25, 0.25)$. We fine-tuned word embeddings during training. The dimension of hidden states in one direction was set to 60. The total number of iterations in *Attention and Memory* module was set to 5. We trained all models for max 20 epochs with a mini-batch size of 16, ℓ_2-regularization weight of 0.001 and initial learning rate of 0.0005 by Adam optimizer [32]. To alleviate overfitting, we used dropout strategy with a ratio of 0.3 before both the BiGRU for tweet and the last fully-connected layer.

D. Evaluation Metric

We use the average of the F_1-score for “FAVOR” class and the F_1-score for “AGAINST” class as the evaluation metric, which is the official metric of SemEval-2016 Task 6:

$$F_{\text{AVG}} = \frac{1}{2}(F_{\text{FAVOR}} + F_{\text{AGAINST}})\,. \tag{16}$$

This metric does not disregard the “NONE” class because it is a negative class during computing F_{FAVOR} and F_{AGAINST}.

[1] We randomly sampled 30% of training data as validation data. We first trained the model on the retained 70% training data and used validation data for model selection. Afterwards, we retrained the model on the complete training data.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT METHODS. BEST SCORES ARE IN BOLD

Type	Model	Overall	Opinion towards		Dataset				
		(%)	Target	Other	A.	C.C.C.	F.M.	H.C.	L.A.
I	*SVM* [12]	68.98	74.54	43.20	65.19	42.35	57.46	58.63	**66.42**
	NBoW	63.22	68.71	41.90	51.86	39.84	48.95	59.71	60.95
	CNN [21]	67.33	71.07	46.66	63.34	52.69	51.33	64.41	61.09
	biGRU [17]	67.65	N/A[a]	N/A	65.26	43.08	56.53	55.60	61.39
II	*AT-biGRU* [17]	67.97	N/A	N/A	62.32	43.89	54.15	57.94	64.05
	TAN [18]	68.79	N/A	N/A	59.33	**53.59**	55.77	65.38	63.72
	AS-biGRU-CNN [17]	69.42	N/A	N/A	**66.76**	43.40	58.83	57.12	65.45
	MemN2N	68.67	74.06	48.31	46.97	42.41	57.82	55.77	62.59
	TGMN-CR	**71.04**	**75.53**	**49.57**	64.60	43.02	**59.35**	**66.21**	66.21

[a]"N/A" means that the corresponding paper doesn't provide the result.

E. Results and Discussions

Table II shows the performance comparison of *TGMN-CR* with baselines. To examine the performance of methods from different perspectives, we also report the F_{AVG} on two subsets of testing set, one of which contains tweet-target pairs labelled with "Target" (called *target subset*), and the other contains tweet-target pairs labelled with "Other" (called *other subset*). Moreover, we also show the performance on different targets respectively for reference.

In general, methods belonging to type II perform better than type I, which verifies that integrating target information can boost the performance. Moreover, results on *other subset* are significantly lower than results on *target subset*, demonstrating that the major difficulty of this task is that identifying the stance when a tweet does not talk about the given target directly.

The performance of *SVM* on full testing set (68.98%) and *target subset* (74.54%) outperform a significant number of neural network-based methods. However, its performance on *other subset* (43.20%) is usually worse than neural methods. These results give us two enlightenments. First, *SVM* is a strong baseline because n-gram features are effective to represent important words and phrases related to target, but discrete feature representation can not capture the semantic relatedness between words. Second, although unmodified neural methods such as *CNN* can not achieve very high performance directly, they show the potential of processing the major difficulty of this task due to the capacity of semantic composition by using distributed representation. The performance of *NBoW* is poor, so the importance of each word in a tweet should not be equal.

Attention-based methods including *AT-biGRU*, *TAN* and *AS-biGRU-CNN* incorporate target information into neural models by "single-use" attention mechanism, and outperform ordinary neural models. However, compared with *SVM*, both *AT-biGRU* and *TAN* perform worse than it, demonstrating that adding attention mechanism into neural architectures for stance detection in the similar way to ABSA models like [30] is not particularly efficient. Although *AS-biGRU-CNN* extends the attention mechanism, it does not perform significantly better than *SVM* (only 0.44%). These results verify that target information is underused in them.

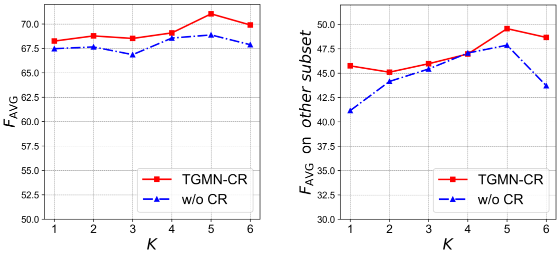
TABLE III
PERFORMANCE COMPARISON OF OUR ORIGINAL AND VARIANT MODELS

Metric	Model	Overall	Opinion towards	
			Target	Other
F_{FAVOR}	*TGMN-CR*	**65.52**	**70.59**	**32.88**
	w/o CR	62.77	67.72	32.00
F_{AGAINST}	*TGMN-CR*	**76.55**	**80.47**	**66.27**
	w/o CR	74.98	79.48	63.74

Our *TGMN-CR* model strengthens the use of target information through learning conditional tweet representation and employing memory vector to represent clues for detecting stance. *TGMN-CR* achieves the best F_{AVG} on full testing set (71.04%), *target subset* (75.53%) and *other subset* (49.57%) among all methods. Especially, it provides superior performance on *other subset* to other methods by a significant margin, demonstrating the effectiveness of our proposed method. Compared with *SVM*, *TGMN-CR* outperforms it by 2.06% on full testing set and 6.37% on *other subset*, which indicates that our method can automatically learn powerful feature representation for stance detection task. *TGMN-CR* performs better than *MemN2N*, for two reasons. First, *MemN2N* only stacks word embeddings as the external memory, while *TGMN-CR* can produce higher quality text representation. Furthermore, the memory update mechanism in our model is more effective than the multiple computational steps of *MemN2N* in this task.

F. Effects of Different Modules in Our Model

1) ***Effect of conditional representation:*** To demonstrate the effect of conditional tweet representation, we further conduct ablation experiments. Table III shows that how conditional representation influences the performance on both "FAVOR" class and "AGAINST" class, where "w/o CR" means that the initial hidden states of BiGRU for tweet are zero vectors in these models. As we can see, for both classes, using conditional representation can improve the performance on both *target subset* and *other subset*. Consequently, learning tweet representation that is conditional on target embedding is an effective way to make full use of target information.



(a) F_{AVG} (b) F_{AVG} on *other subset*

Fig. 4. Performance of different models with various iterations.

*2) **Effect of memory update mechanism**:* The total number of iterations in *Attention and Memory* module, i.e., the value of K, is a critical parameter of our model. To analyze how it influences the performance of *TGMN-CR* and *TGMN-CR* w/o CR, we varies the value of K from one to six and Fig. 4 illustrates the results of different models with various K. We can observe that more iterations can generally lead to better performance, demonstrating the necessity of making multiple interactions to update memory vector iteratively. However, we can also see that the performance is decreased when K exceeds five. We suggest that this is mainly caused by overfitting, so the generalization of model becomes poor. Moreover, gradient vanishing is also a possible reason.

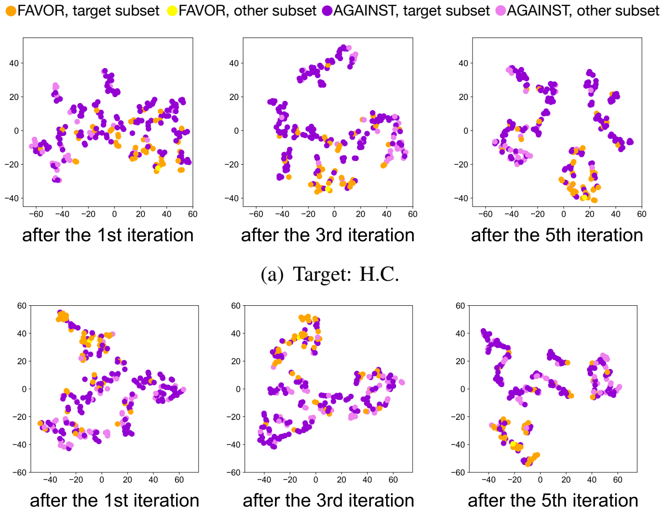
G. Qualitative Analysis

*1) **Visualization of memory vectors updating**:* We apply t-SNE [33] to memory vectors on test samples of two targets “H.C.” and “L.A.”, and Fig. 5 illustrates the 2D visualization. With the running of the iterative process in *Attention and Memory* module (the total number of iterations K is five here), samples with same stance label (“FAVOR” or “AGAINST”) can be gradually clustered, which indicates that the discrimination of memory vectors is gradually increased through extracting more distinguishing features of two classes. Some samples belonging to different classes are grouped together, which shows the limitation of our method and the difficulty of this task. The class-imbalance of dataset also causes negative impact.

*2) **Case study and error analysis**:* To understand more about the behaviour of our method, we choose three test samples and visualize the final attention weights for them in Table IV. “+1” means “FAVOR” and “-1” means “AGAINST”.

The first tweet does not explicitly express opinion about “Atheism” , but *TGMN-CR* accurately captured the useful words including `praise`, `thank` and `god`, and identified that the stance towards the target is opposition although the sentiment polarity is positive. This example shows the difference of sentiment classification and stance detection. For the latter, automatic system also need to handle the situation that the given target and the opinion target of text are inconsistent.

In the second tweet, the words `abortion` and `genocide` make obvious contributions to the author’s attitude towards “Legalization of Abortion”, and *TGMN-CR* can give higher attention scores to them. Note that the word `genocide` appears twice in this tweet, and the position of the second one is far from the target word `abortion` which is located at the beginning of the tweet. Our method can notice both the first and the second `genocide`, demonstrating that *TGMN-CR* has the capability of extracting useful expression in text for detecting stance correctly.



(a) Target: H.C.

(b) Target: L.A.

Fig. 5. Visualization of memory vectors on testing set using t-SNE.

We also show an example that *TGMN-CR* gave wrong prediction result. In the third tweet, the author expresses opposition to “Feminist Movement” by using the rhetorical device of irony. Our method can highlight `ladies`, `men` and `equality`, but failed to understand the implication of this tweet. Sometimes automatic system need decipher more nuanced forms of language to identify the attitude correctly, which is a challenging task.

TABLE IV
VISUALIZATION OF ATTENTION WEIGHTS FOR THREE TEST SAMPLES (DEEPER COLOR MEANS HIGHER WEIGHT)

Tweet	**Target**	**Stance**	
		Truth	Predict
Praise and thank God for everything in your life today. #grateful	Atheism	-1	-1
Abortion is genocide - I don’t think you know what genocide means. #ygk	Legalization of Abortion	-1	-1
Ladies in front of men is equality apparently.	Feminist Movement	-1	+1

VI. Conclusion

In this paper, we propose an end-to-end neural model *TGMN-CR* for target-specific stance detection in Twitter, which leverages the power of target content information. It first learns tweet representation that is conditional on specific target of interest, and then utilizes a target-guided iteration process to extract crucial clues for detecting stance. Experimental results on the benchmark dataset demonstrate that our proposed method outperforms the state-of-the-art alternative methods, especially when a tweet does not explicitly express opinion about the given target. Furthermore, ablation test and visualization analysis show the effectiveness of different modules in our model. In future work, we shall integrate extra information, e.g., user profile and relationship network, for improving the performance of stance detection.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant #2016QY02D0305, NSFC Grant #71621002, #71702181 and #61671450, and CAS Key Grant #ZDRW-XH-2017-3.

References

[1] P. Sobhani, S. Mohammad, and S. Kiritchenko, "Detecting stance in tweets and analyzing its interaction with sentiment," in *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*SEM)*, 2016, pp. 159–169.

[2] C. Li, X. Guo, and Q. Mei, "Deep memory networks for attitude identification," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM)*, 2017, pp. 671–680.

[3] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from congressional floor-debate transcripts," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006, pp. 327–335.

[4] A. Faulkner, "Automated classification of stance in student essays: An approach using stance target information and the wikipedia link-based measure," in *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2014, pp. 174–179.

[5] S. Somasundaran and J. Wiebe, "Recognizing stances in ideological on-line debates," in *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 2010, pp. 116–124.

[6] P. Anand, M. Walker, R. Abbott, J. E. F. Tree, R. Bowmani, and M. Minor, "Cats rule and dogs drool!: Classifying stance in online debate," in *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA), ACL-HLT*, 2011, pp. 1–9.

[7] K. S. Hasan and V. Ng, "Extra-linguistic constraints on stance recognition in ideological debates," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013, pp. 816–821.

[8] A. Sasaki, J. Mizuno, N. Okazaki, and K. Inui, "Stance classification by recognizing related events about targets," in *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, 2016, pp. 582–587.

[9] W.-F. Chen and L.-W. Ku, "UTCNN: A deep learning model of stance classification on social media text," in *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, 2016, pp. 1635–1645.

[10] A. Rajadesingan and H. Liu, "Identifying users with opposing opinions in Twitter debates," in *Proceedings of International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP)*, 2014, pp. 153–160.

[11] R. Xu, Y. Zhou, D. Wu, L. Gui, J. Du, and Y. Xue, "Overview of NLPCC shared task 4: Stance detection in Chinese microblogs," in *Proceedings of the 5th Conference on Nature Language Processing and Chinese Computing and the 24th International Conference on Computer Processing of Oriental Languages (NLPCC-ICCPOL)*, 2016, pp. 907–916.

[12] S. Mohammad, S. Kiritchenko, P. Sobhani, X.-D. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluations (SemEval)*, 2016, pp. 31–41.

[13] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "Semeval-2014 task 4: Aspect based sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluations (SemEval)*, 2014, pp. 27–35.

[14] M. Tutek, I. Sekulic, P. Gombar, I. Paljak, F. Culinovic, F. Boltuzic, M. Karan, D. Alagić, and J. Šnajder, "Takelab at Semeval-2016 task 6: Stance classification in tweets using a genetic algorithm based ensemble," in *Proceedings of the 10th International Workshop on Semantic Evaluations (SemEval)*, 2016, pp. 464–468.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 3111–3119.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[17] Y. Zhou, A. I. Cristea, and L. Shi, "Connecting targets to tweets: Semantic attention-based model for target-specific stance detection," in *Proceedings of the 18th International Conference on Web Information Systems Engineering (WISE)*, 2017, pp. 18–32.

[18] J. Du, R. Xu, Y. He, and L. Gui, "Stance classification with target-specific neural attention networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 3988–3994.

[19] J. Ebrahimi, D. Dou, and D. Lowd, "A joint sentiment-target-stance model for stance classification in tweets," in *Proceedings the 26th International Conference on Computational Linguistics (COLING)*, 2016, pp. 2656–2665.

[20] ——, "Weakly supervised tweet stance classification by relational bootstrapping." in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 1012–1017.

[21] W. Wei, X. Zhang, X. Liu, W. Chen, and T. Wang, "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 Evaluations (SemEval)*, 2016, pp. 384–388.

[22] G. Zarrella and A. Marsh, "MITRE at Semeval-2016 task 6: Transfer learning for stance detection," in *Proceedings of the 10th International Workshop on Semantic Evaluations (SemEval)*, 2016, pp. 458–463.

[23] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 876–885.

[24] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiský, and P. Blunsom, "Reasoning about entailment with neural attention," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2016.

[25] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 2440–2448.

[26] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *Proceedings of International Conference on Machine Learning (ICML)*, 2016, pp. 1378–1387.

[27] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *Proceedings of International Conference on Machine Learning (ICML)*, 2016, pp. 2397–2406.

[28] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[29] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[30] Y. Wang, M. Huang, X. Zhu, and L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 606–615.

[31] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[33] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.