Stance Classification with Target-Specific Neural Attention Networks

Jiachen Du1,2, Ruifeng Xu1,3, Yulan He4, Lin Gui1
1 Laboratory of Network Oriented Intelligent Computation, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
2 Department of Computing, the Hong Kong Polytechnic University, Hong Kong
3 Guangdong Provincial Engineering Technology Research Center for Data Science, Guangzhou, China
4 School of Engineering and Applied Science, Aston University, United Kingdom
dujiachen@stmail.hitsz.edu.cn, xuruifeng@hit.edu.cn, y.he9@aston.ac.uk, guilin.nlp@gmail.com

Abstract

Stance classification, which aims at detecting the stance expressed in text towards a specific target, is an emerging problem in sentiment analysis. A major difference between stance classification and traditional aspect-level sentiment classification is that the identification of stance is dependent on target which might not be explicitly mentioned in text. This indicates that apart from text content, the target information is important to stance detection. To this end, we propose a neural network-based model, which incorporates target-specific information into stance classification by following a novel attention mechanism. In specific, the attention mechanism is expected to locate the critical parts of text which are related to target. Our evaluations on both the English and Chinese Stance Detection datasets show that the proposed model achieves the state-of-the-art performance.

1 Introduction

With the rapidly development of Internet and Web 2.0, more and more people post their opinions online. To detect sentiment or retrieve opinions from online text, sentiment analysis and opinion mining [Gui et al., 2016; Wang et al., 2016] has become a hot research topic in natural language processing. Various techniques have been proposed to identify the polarity of a given text. However, in many practical applications, we are interested to learn the position of an author to a specific target or topic rather than the polarity of the whole text. For example, during the US general election, we would like to find out from someone’s online posts whether she supports Trump or not. This is referred to as target-specific stance detection.

Previous work on stance detection mostly focused on online debates [Walker et al., 2012] or news articles [Ferreira and Vlachos, 2016]. Spurred by the growth in the use of microblogging platforms such as Twitter and Weibo, companies and media organizations are increasingly seeking ways to mine microblogs for information about what people think of and feel about their products and services. Studying how stance is expressed on microblogging platforms can be beneficial in many application areas.

Stance detection is formalized as the task of assigning a stance label to a piece of text with respect to a specific target, i.e. whether a text is in favor of or against the given target, or neither of them. A major difference between stance detection and traditional aspect-level sentiment classification [Liu, 2012] is that stance detection is dependent on both subjective expressions found in text and the associated target which might not be explicitly mentioned. It may cause the model make wrong decision when predicting the stance. For example, the text below implies stance against “abortion”, but the target “abortion” does not appear anywhere in text and needs to be inferred:

“We remind ourselves that love means to be willing to give until it hurts.”

In the stance detection research, various models were proposed. Some of them used feature engineering to extract features manually [Mohammad et al., 2016], and some used classical neural network-based models such as Recurrent Neural Networks (RNNs) [Augenstein et al., 2016] and Convolutional Neural Networks (CNNs) [Vijayaraghavan et al., 2016]. Nevertheless, most of these methods perform stance detection based on features extracted from text and ignore the target information. As such, they sometimes produce spurious results especially when text expresses stance towards other target instead of the given one. To alleviate this problem, we propose a neural network based model named Target-specific Attentional Network (TAN) to make full use of the target information in stance detection. Our model utilizes a novel target-specific attention extractor to focus on the important parts in text which are highly related to the target topic. Firstly, we concatenate the embedding vectors of text and target to learn the target-specific embedding for modelling both text and target. Then we use a fully-connected network to learn attention signal for driving the classifier to focus on the salient parts in text and finally determine the stance. Experimental results on both English and Chinese datasets show that the proposed model outperforms a number of competitive baselines and gives the state-of-the-art performance to the best of our knowledge.

The main contributions of our work can be summarized as follows:

* We propose a neural attention model to extract target-
related information for stance detection. This model is able to extract core parts of given text when different targets are concerned.

- We propose a supervised model, TAN, which combines RNN with long-short memory (LSTM) and target-specific attention extractor.
- Experimental results on the datasets of SemEval-2016 (English) and NLPCC-2016 (Chinese) show that our model outperforms several strong baselines including the top performed systems in both stance detection shared tasks. Furthermore, the visualization of selected instances illustrates why the proposed model works well.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work and Section 3 presents our proposed Target-specific Attentional Network model. Section 4 discusses evaluation results. Finally, Section 5 concludes the paper.

2 Related Work

In this section, we will review related work on stance detection and neural attentional models briefly.

**Stance Detection:** Previous work in stance detection mostly focused on debates [Hasan and Ng, 2013; Walker et al., 2012] or student essays [Faulkner, 2014]. There is a growing interest in performing stance classification on microblogs. SemEval-2016 Task 6 [Mohammad et al., 2016] involved two stance detection subtasks in tweets in supervised and weakly supervised settings. The majority of existing approaches attempt to detect the stance label of an entire sentence, regardless of the target information. NLPCC-2016 Chinese Stance Detection shared task released the first Chinese dataset for stance detection [Xu et al., 2016]. [Augenstein et al., 2016] used two bidirectional RNN to model both target and text for stance detection. However this model requires a very large unlabeled Twitter corpus in order to predict the task-relevant hashtags as an auxiliary task to initialize the word embeddings.

**Neural Attentional Model:** In the general domain of sentiment analysis, many deep learning approaches have been proposed. [Tang et al., 2015] used gated RNN to model documents for sentiment classification. [Tai et al., 2015] explored the structure of a sentence and used a tree-structured RNN with long-short term memory (LSTM) for sentiment classification. The advantage of RNN is its ability to better capture the contextual information, especially the semantics of long texts. However RNNs cannot pay attention to the salient parts of text. This limitation influences the performance of RNN when applied to text classification. To address this problem, a new direction of incorporating attentions into neural networks has emerged. Neural networks with attention mechanism show promising results on sequence-to-sequence (seq2seq) tasks in NLP, including machine translation [Bahdanau et al., 2014], caption generation [Xu et al., 2015] and text summarization [Rush et al., 2015]. For text classification, [Yang et al., 2016] applies the attention model used in seq2seq tasks to document-level classification. However, there is no neural attention model for stance detection task up to now.

3 Model

As has been previously discussed, the performance of stance detection could be potentially improved by considering both text content features and target related features. Motivated by this, we propose an RNN-based model which can concentrate on salient parts in text corresponding to a given target. We name our model as Target-specific Attention Neural Network (TAN). The overall architecture of our model is shown in Figure 1. It consists of two main components: a recurrent neural network (RNN) as the feature extractor for text and a fully-connected network as the target-specific attention selector. These two components are combined by an element-wise multiplication operation in the classification layer. We describe the details of these two components in the following subsections.

3.1 Recurrent Neural Network with Long-short Time Memory

An Recurrent Neural Network (RNN) [Elman, 1990] is a kind of neural network that processes sequences of arbitrary length by recursively applying a function to its hidden state vector $h_t \in \mathbb{R}^d$ of each element in the input sequences. In neural network-based models, a text sequence of length $T$ (padded where necessary) is normally represented as $[x_1, x_2, \ldots, x_T]$, where $x_t \in \mathbb{R}^d (t = \{0, 1, \ldots, T-1\})$ corresponds to the $d$-dimensional vector representation of the $t$-th word in the text sequence. The hidden state vector at the time step $t$ depends on the input symbol $x_t$ and the hidden state vector at the last time step $h_{t-1}$ and is computed by the recurrent function $g$:

$$ h_t = \begin{cases} 0 & t = 0 \\ g(h_{t-1}, x_t) & \text{otherwise} \end{cases} \quad (1) $$

A fundamental problem in traditional RNN is that gradients propagated over many steps tend to either vanish or explode. It affects RNN to learn long-dependency correlations in a sequence. Long short-term memory network (LSTM) was proposed by [Hochreiter and Schmidhuber, 1997] to alleviate this problem. LSTM has three gates: an input gate $i_t$, a forget gate $f_t$, and an output gate $o_t$. Each gate is typically a sigmoid applied to a linear combination of the current input and the previous hidden state. The cell state $c_t$ is a weighted sum of the previous cell state and the current input, and is output through a tanh activation. The cell state carries information from previous time steps, while the hidden state is a transformation of the cell state that is used to output the current prediction.
a forget gate $f_t$, an output gate $o_t$ and a memory cell $c_t$. They are all vectors in $\mathbb{R}^d$. The LSTM transition equations are:

$$
\begin{align*}
    i_t &= \sigma (W_i x_t + U_i h_{t-1} + V_i c_{t-1}), \\
    f_t &= \sigma (W_f x_t + U_f h_{t-1} + V_f c_{t-1}), \\
    o_t &= \sigma (W_o x_t + U_o h_{t-1} + V_o c_{t-1}), \\
    c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(c_t), \\
    h_t &= o_t \odot \tanh(c_t), \\
\end{align*}
$$

(2)

where $x_t$ is the input at the current time step, $\sigma$ is the sigmoid function and $\odot$ is the elementwise multiplication operation. $W_{i,f,o,c}, U_{i,f,o,c}, V_{i,f,o}$ are all sets of learned weight parameters. In our model, we use the hidden state vector of each time step as the representation of the corresponding word in a sentence.

In this study, we employ bi-directional LSTM model to better capture the information in text. The bi-directional LSTM has a forward and a backward LSTM. The annotation for each word are obtained by concatenating the forward hidden state and the backward one.

### 3.2 Target-augmented Embedding

The target information is vital for determining the stance of a given text. To combine the information of target and text, we propose to learn a target-augmented embedding for each target. A target sequence of length $N$ is represented as $[z_1, z_2, \ldots, z_N]$ where $z_n \in \mathbb{R}^d$ is the $d$-dimensional vector of the $n$-th word in the target sequence. Since the common word embedding representations exhibit linear structure that make it possible to meaningfully combine words by an element-wise addition of their vector representations, we use the average vector $\bar{z}$ to obtain a more compact target representation:

$$
\bar{z} = \frac{1}{N} \sum_{n=1}^{N} z_n
$$

(3)

In order to better take advantage of target information, we append the target representation to the embedding of each word in original text. The target-augmented embedding of a word $t$ for a specific target $z$ is $e_t^z = x_t \oplus \bar{z}$ where $\oplus$ is the vector concatenation operation. Notice that the dimension of $e_t^z$ is $(d + d')$.

### 3.3 Target-specific Attention Extraction

Traditional RNN model cannot capture the important parts in sentences. In order to address this problem, we design an attention mechanism which drives the model to concentrate on salient parts in text with respect to a specific target. To make full use of target information, this model uses a bypass network which takes the target-augmented embeddings discussed in Section 3.2 as input to extract target-specific attention signal.

Here, we use a linear transformation to map the $(d + d')$-dimensional target-augmented embedding of each word to a scalar value:

$$
a'_t = W_a e_t^z + b_a
$$

(4)

where $W_a$ and $b_a$ are learned set of weights and bias terms for attention extraction.

To obtain more stable attention signal, we then feed the attention vector $[a'_1, a'_2, \ldots, a'_T]$ into a softmax transformation to get the final attention signal for each word:

$$
a_t = \text{softmax}(a_t) = \frac{e^{a'_t}}{\sum_{t'=1}^{T} e^{a'_{t'}}}
$$

(5)

### 3.4 Stance Classification

We use the product of attention signal $a_t$ and the corresponding hidden state vector of RNN $h_t$ to represent the word $t$ in a sequence with attention signal. The representation of the whole sequence can be obtained by averaging the word representations:

$$
s = \frac{1}{T} \sum_{t=0}^{T-1} a_t h_t
$$

(6)

where $s \in \mathbb{R}^d$ is the vector representation of the text sequence and it can be used as features for text classification:

$$
p = \text{softmax}(W_{at}s + b_{at})
$$

(7)

where $p \in \mathbb{R}^C$ is the vector of predicted probability for stance. Here $C$ is the number of classes of stance labels, and $W_{at}$ and $b_{at}$ are parameters of the classification layer.

### 3.5 Model Training

We use cross-entropy loss to train our model end-to-end given a set of training data $\{x^i, z^i, y^i\}$, where $x^i$ is the $i$-th text to be predicted, $z^i$ is the corresponding target and $y^i$ is one-hot representation of the ground-truth stance for target $z^i$ and text $x^i$. We represent this model as a black-box function $f(x, z)$ whose output is a vector representing the probability of stance. The goal of training is to minimize the loss function:

$$
\mathcal{L} = - \sum_{i} \sum_{j} y^i_j \log f_j(x_i, z_i) + \lambda \|\theta\|^2
$$

(8)

where $i$ is the index of data and $j$ is the index of class. $\lambda \|\theta\|^2$ is the $L_2$-regularization term and $\theta$ is the parameter set.

Apart from the parameter sets of standard LSTM $\{W_{i,f,o,c}, U_{i,f,o,c}, V_{i,f,o}\}$ and softmax classification $\{W_a, b_a\}$, our model only has additional parameters $\{W_a, b_a\}$ for attention extractor.

### 4 Performance Evaluation

In this section, we compare the performance of TAN with several strong baselines on stance detection. We firstly describe the experimental setting, then present the comparative results, and finally show some visualization results where the learned attention signals can be visualized to illustrate the validity of the proposed attention extractor.

#### 4.1 Experimental Setting

In this section, we first describe the datasets used in our experiments, then introduce the evaluation metrics and baseline methods, and finally present the details of the training process of our proposed model.
Datasets

To validate the effectiveness of the proposed model, we conduct experiments on datasets of stance detection task in English and Chinese.

English Dataset. Semeval-2016 Task 6 [Mohammad et al., 2016] released the first dataset for stance detection from English tweets. In this dataset, more than 4,000 tweets are annotated for whether one can deduce favorable or unfavorable stance towards one of five targets “Atheism”, “Climate Change is a Real Concern”, “Feminist Movement”, “Hillary Clinton”, and “Legalization of Abortion”. Task 6 has two subtasks including subtask-A supervised learning and subtask-B unsupervised learning. In this evaluation, we only use the dataset of subtask-A in which the targets provided in the test set can all be found in the training set. Table 1 shows the statistics of this dataset.

Chinese Dataset. To show the stability and language independence of our model, we also conduct experiment on a Chinese dataset for stance detection. We use the dataset of NLPCC-2016 Chinese Stance Detection Shared Task. The construction of this dataset followed the same procedure as in the Semeval-2016 Task 6. There are 3,000 Chinese tweets of 5 targets annotated for 3 stance labels. For each target, there are 600 training and 200 test samples. Table 2 shows the statistics of this dataset.

Metrics

The micro average of $F_1$-score across targets which is utilized in Semeval evaluation is adopted as the metrics. Firstly, the $F_1$-score for Favor and Against categories for all instances in the dataset is calculated as:

$$F_{\text{Favor}} = \frac{2P_{\text{Favor}}R_{\text{Favor}}}{P_{\text{Favor}} + R_{\text{Favor}}}$$

$$F_{\text{Against}} = \frac{2P_{\text{Against}}R_{\text{Against}}}{P_{\text{Against}} + R_{\text{Against}}}$$

where $P$ and $R$ are precision and recall. Then the average of $F_{\text{Favor}}$ and $F_{\text{Against}}$ is calculated as the final metrics:

$$F_{\text{average}} = \frac{F_{\text{Favor}} + F_{\text{Against}}}{2}$$

(10)

Note that the final metrics does not disregard the None class. By taking the average F-score for only the Favor and Against classes, we treat None as a class that is not of interest.

Baselines

We compare the following baseline methods:

- Neural Bag-Of-Words (NBOW): The NBOW sums the word vectors within the sentence and applies a softmax classifier.
- LSTM: LSTM without target-specific embedding and target-specific attention.
- LSTM$_t$: LSTM with target-specific embedding.
- TOP: The best performing model in the shared tasks.

Training details

We use ad-hoc strategy to train one model for each target. The final result is obtained by concatenating all the predicted results of these models. Although different models are used...
for different targets, they all share the same sets of hyper-
parameters. All hyper-parameters are tuned to obtain the best
performance by 5-fold cross validation on the training set.
In our experiments, all word vectors are initialized by
word2vec [Mikolov et al., 2013]. The word embedding vec-
tors are pre-trained on unlabelled corpora which is crawled
from Twitter and Sina Microblogging. The other parameters
are initialized using a uniform distribution $U(-0.01, 0.01)$.
The dimension of word and target embeddings are 300 and
the size of units in LSTM is 100. Adam is used for our opti-
mization method, and its learning rate is $5e-4$, $\beta_1$ is 0.9, $\beta_2$
is 0.999, $\epsilon$ is $1e-8$. All models are trained by mini-batch of
30 instances.

4.2 Results

The performance on the English dataset of all baselines and
our proposed model are listed in Table 3. Firstly, it is ob-
erved that NBOW and ordinary LSTM perform unsatisfacto-
"rily, since they only use features extracted from the text but
ignore the information expressed by the targets. LSTM utilizes
target-specific embeddings and thus improves upon or-
dinary LSTM by 3.03%. TAN performs better than LSTM
with target-specific embeddings. It shows that the attention
mechanism of TAN can further capture the target information
to improve the performance of stance detection. TAN also
outperforms MITRE which is the top performing model on
this shared task. In particular, we observe that TAN improves
upon MITRE’s by 3.54% on the target “Hillary Clinton”. For
this target, most tweets tend to compare other candidates of
presidency election with Hillary Clinton. This obviously af-
facts the performance of the models which cannot find the
important words corresponding to the given target. TAN ap-
plies the novel attention mechanism to extract key words cor-
responding to targets, and uses the information obtained from
stance using back-propagation to link the attention signals of
targets with stance. Overall, our method TAN outperforms
all baselines significantly. The empirically results show that
target-specific attention could benefit stance detection.

The performance on the Chinese dataset are shown in Table
4. Firstly, we notice that the results on the Chinese dataset are
generally better than those on the English dataset. One pos-
sible reason is that the annotated data in Chinese is more bal-
anced than those in English. We also observe that our model
performs the best among all methods. In specific, TAN out-
performs the first-place system of NLPCC-2016’s by 1.8%.
The top performing system of NLPCC is a relatively strong
baseline that used carefully chosen hand-crafted features and
optimal parameters tuned by grid search. The results demon-
strates that TAN is a language-independent model that per-
form consistently well across different languages.

4.3 Qualitative Analysis

Learning curve

To show the effectiveness of TAN, we plot the learning curves
of selected targets for each dataset in Figure 2, which com-
pares the training and test costs of standard LSTM and TAN.
It is obvious that TAN achieves lower training costs compared
to standard LSTM after a fixed number of iterations and it has
faster convergence rate. In Figure 2(b), TAN converges after
only 3 iterations. But the standard LSTM needs more than
5 iterations to converge. This shows that TAN has a more
powerful fitting capacity. We also notice that TAN needs less
iterations to achieve the best test performance. In both exam-
pies, TAN reaches the best performance on the test set after 3 iterations compared to the 5 iterations for standard LSTM.
The above results show that TAN is superior than the standard

\begin{table}
\centering
\caption{Performance comparison of stance detection on the English Dataset.}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textbf{Target} & \textbf{NBOW} & \textbf{LSTM} & \textbf{LSTM$_E$} & \textbf{TOP} & \textbf{TAN} \\
\hline
E.1 & 55.12 & 58.18 & 59.77 & \textbf{61.47} & 59.33 \\
E.2 & 39.93 & 40.05 & 48.98 & 41.63 & \textbf{53.59} \\
E.3 & 50.21 & 49.06 & 52.04 & \textbf{62.09} & 55.77 \\
E.4 & 55.98 & 61.84 & 56.89 & 57.67 & \textbf{65.38} \\
E.5 & 55.07 & 51.03 & 60.34 & 57.28 & \textbf{63.72} \\
\hline
Overall & 60.19 & 63.21 & 66.24 & 67.82 & \textbf{68.79} \\
\hline
\end{tabular}
\end{table}

\begin{table}
\centering
\caption{Performance comparison of stance detection on the Chinese Dataset.}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textbf{Target} & \textbf{NBOW} & \textbf{LSTM} & \textbf{LSTM$_E$} & \textbf{TOP} & \textbf{TAN} \\
\hline
C.1 & 55.07 & 51.03 & 73.78 & 77.30 & \textbf{77.50} \\
C.2 & 55.12 & 58.18 & 55.23 & 57.80 & \textbf{59.33} \\
C.3 & 39.93 & 40.05 & 55.23 & 58.14 & \textbf{59.19} \\
C.4 & 50.21 & 80.36 & 63.59 & 62.09 & \textbf{65.00} \\
C.5 & 55.98 & 61.84 & 71.23 & \textbf{76.52} & 72.38 \\
\hline
Overall & 62.53 & 65.27 & 68.12 & 71.06 & \textbf{72.88} \\
\hline
\end{tabular}
\end{table}

Figure 2: Learning curves of the selected targets.
LSTM in both accuracy and time complexity.

Visualization of attention

In order to validate that our model is able to select target-specific parts in a text sequence, we visualize the attention layers for several sentences in both English and Chinese whose labels are correctly predicted by our model in Figure 3. We choose three examples, two in English and one in Chinese. Since the Semeval Dataset also provided sentiment annotations, we show at the top of each example the actual stance label and sentiment label. It can be observed that the stance label and sentiment label do not agree with each other in the two examples shown here. This shows that stance detection is fundamentally different from traditional sentiment analysis. The stance detection results generated by TAN and standard LSTM are displayed in the right half of Figure 3. We can see that the standard LSTM generated wrong results on these three examples, but TAN identified the stance labels correctly. In particular, TAN can select words that have strong relation with a given target. For example, in the first sentence, TAN highlights “Democrats”, “Republicans” and “power” which are non-trivial words related to “Hillary Clinton”. In the second sentence, “Religion” is selected by our model as strongly related to “Atheism”. For the example in Chinese, our model not only identified the important word “Syria” but also highlighted other related words “Putin” and “ISIS”.

4.4 Error Analysis

We also analyze the sentences where our model failed to predict the correct stance labels. We show an attention visualization in Figure 4, consisting of one English example and one Chinese example. For the English example, the true stance label is “Against”, but our model predicted its label as “Favor”. In this example, the original sentence was a quotation from the Bible. Hence, some background knowledge would be required in order to predict the stance label correctly. For the Chinese example, the author is against the “Two-Child Policy” because of the difficulties that will be arising from education, future employment and healthcare costs. Interestingly, although TAN has correctly identified the important words such as “education”, “healthcare” and “second child” that are strongly related to the target, it gives a neural result that neither supports or against the “Two-Child Policy”.

5 Conclusion

In this paper, we proposed an attention based neural network for stance detection. The main contribution of this model is to learn target-augmented embeddings for text and use attention mechanism to extract target-specific parts in text to improve classification performance. Experimental results show that our model outperforms several strong baselines. Meanwhile, the visualization of some attentions extracted by our model shows the impressive capability of our model to extract the important parts which are helpful to improve stance detection.

In future work, we will focus on combining the proposed attention mechanism with other state-of-the-art models in stance detection and explore a feasible way in incorporating external knowledge to improve the stance detection performance.

Figure 3: Visualization of learned attention in datasets. Gray patches highlight the words strongly related to a given target.

Figure 4: Error Analysis: Visualization of the learned attention in both English and Chinese examples.
Acknowledgements

This work was supported by the National Natural Science Foundation of China 61370165, U1636103, 61632011, 61528302, Shenzhen Foundational Research Funding JCYJ20150625142543470, Guangdong Pro vincial Engineering Technology Research Center for Data Science 2016KF09 and grant from the Hong Kong Polytechnic University (G-YBJP).

References


