

# A multi-target stance detection based on Bi-LSTM network with position-weight<sup>①</sup>

Xu Yilong (徐翼龙)\*, Li Wenfa<sup>②\*\*</sup>, Wang Gongming<sup>\*\*\* \*\*\*\*</sup>, Huang Lingyun<sup>\*\*\*\*\*</sup>

(\* Smart City College, Beijing Union University, Beijing 100101, P. R. China)

(\*\* College of Robotics, Beijing Union University, Beijing 100101, P. R. China)

(\*\*\* Tianyuan Network Co., Ltd., Beijing 100193, P. R. China)

(\*\*\*\* Beijing Tianyuan Network Co., Ltd., Beijing 100193, P. R. China)

(\*\*\*\*\* Chinatelecom Information Development Co., Ltd., Beijing 100093, P. R. China)

## Abstract

In the task of multi-target stance detection, there are problems the mutual influence of content describing different targets, resulting in reduction in accuracy. To solve this problem, a multi-target stance detection algorithm based on a bidirectional long short-term memory (Bi-LSTM) network with position-weight is proposed. First, the corresponding position of the target in the input text is calculated with the ultimate position-weight vector. Next, the position information and output from the Bi-LSTM layer are fused by the position-weight fusion layer. Finally, the stances of different targets are predicted using the LSTM network and softmax classification. The multi-target stance detection corpus of the American election in 2016 is used to validate the proposed method. The results demonstrate that the Bi-LSTM network with position-weight achieves an advantage of 1.4% in macro average F1 value in the comparison of recent algorithms.

**Key words:** long short-term memory (LSTM), multi-target, natural language processing, stance detection

## 0 Introduction

In recent years, the continuous improvement and development of social media has led an increasing number of people to use social media to share and discuss their attitudes toward different people, events, and objects. Analysis of such texts containing stances in the social media may help us understand their preferences and opinions. Such information plays an important role in public opinion analysis. A large number of researchers, such as Wang et al.<sup>[1]</sup> and Li et al.<sup>[2]</sup>, have used stance detection technology to find such information.

Multi-target stance detection<sup>[3]</sup> is a sub-task of stance detection. It is used to mine the stance classifications of different targets in one text. Typical examples include mining the opinions of different politicians in elections and analyzing user recognition of different brands in similar products. In addition to determining the stance of the target, multi-target stance detection

identifies the corresponding positions of different targets in the same text.

In the fields of multi-target stance detection, most methods combine the 2 tasks of target location (determine the context that describes the different goals) and stance detection (determine stance label based on a goal) into one task during execution. So, these methods tend to enlarge the structure of the model to enhance its ability to mine features. The result, however, provides the comprehensive optimal solution of the 2 tasks rather than the optimal solution of stance detection. Thus, stance detection of a certain target is easily affected by other target descriptions, which may reduce the accuracy of the result. Therefore, target location and stance detection should be executed successively and independently.

To enable such execution, the proposal is as follows. First, the context range in the text corresponding to the different goals needs to be located. In the case of the above example presented here, the context range concerns ‘Hillary Clinton being a liar.’ Then, the

① Supported by the National Natural Science Foundation of China (No. 61972040), the Science and Technology Projects of Beijing Municipal Education Commission (No. KM201711417011), and the Premium Funding Project for Academic Human Resources Development in Beijing Union University (No. BPHR2020AZ03).

② To whom correspondence should be addressed. E-mail: 644085325@qq.com  
Received on Oct. 8, 2019

stance is determined by analyzing the target text in the context range. Based on the above statement, a bidirectional long short-term memory (Bi-LSTM) with position-weight is proposed to carry out the multi-target stance detection. Bi-LSTM can describe the dependency relationship between words from front to back and from back to front, and the position-weight vector can describe the impact of words on the different targets of stance detection. The multi-target stance detection database of the American election in 2016 is used to validate the proposed method.

## 1 Related work

In recent years, given the rapid growth in the number of social media users, researchers have begun to focus on stance detection from social media texts. In 2016, the international conferences SemEval<sup>[4]</sup> and the 5th CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC)<sup>[5]</sup> provided annotated data on stance detection from the social media. Thereafter, some researchers began studying different types of data for stance detection<sup>[6,7]</sup>. Given the similarity between stance detection and sentiment analysis, researchers made efforts to distinguish between them<sup>[8]</sup>, and attempted using sentiment analysis to obtain improved stance detection effects<sup>[9]</sup>.

With regard to the social media, deep learning methods, including the traditional CNN and recurrent neural network (RNN)<sup>[10-13]</sup> as well as fusion models<sup>[14-17]</sup>, are typically used to detect stance.

Subsequently, researchers began applying stance detection to texts containing different targets in the same category. This is called multi-target stance detection, and is regarded as a sub-task of stance detection<sup>[18-21]</sup>. For example, Liu et al.<sup>[19]</sup> proposed an approach to automatically obtain zones of each target and combined it with the LSTM method to obtain the corresponding stance. The results demonstrated the effectiveness of two-target stance detection. Lin et al.<sup>[21]</sup> designed a topic-based approach to detect multiple standpoints in Web text to generate a stance classifier according to the distribution of the standpoint-related topic-term. They produced the parameter values of the classifier with this adaptive method and proved its effectiveness through experiments.

The above methods do not use the positional relationship between words in the text and targets to help the algorithm enhance the ability to distinction between the content describing the different targets. Therefore, in order to obtain the best stance detection effect, it is necessary to extract the appropriate clause as the input

text according to the context range corresponding to the target so as to avoid the influence of unrelated text. Therefore, in this paper, an unsupervised method to extract the context ranges of different targets in the text is proposed. Then, the Bi-LSTM network with position-weight is generated by combining position-weight with the Bi-LSTM approach. Finally, the stance labels of different targets are predicted with LSTM and Softmax classification. The details of the approach are explained in the next section.

## 2 Proposed method

The architecture of our model is shown in Fig. 1. It consists of the following 5 modules: embedded layer, Bi-LSTM layer, position-weight fusion layer, LSTM layer, and Softmax classifier. The result of combining the word embeddings of all the target topics and the input text serves as the input of the model, and the output consists of the author's stance labels for all the possible target topics.

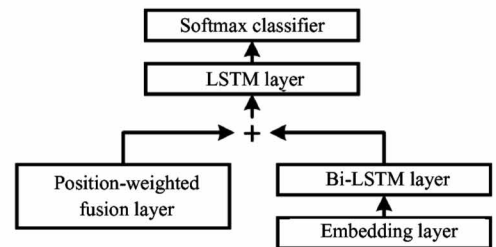


Fig. 1 Model architecture

### 2.1 Embedding layer

The embedding layer transforms every word in the input text into one vector, each of which expresses the relationship between the words applicable to the context. By representing each word in a text as a  $1 \times n$  vector, the text can be represented as an  $l \times n$  matrix ( $l$  is the number of words in the text, and  $n$  is the dimension of each lexical vector). Accordingly, the input text can be converted into a numerical matrix, which facilitates the feature extraction by the algorithm.

### 2.2 Bi-LSTM layer

To extract valid features from unstructured text, LSTM is used to encode the text. The input of the LSTM is a tensor formed by arranging the embedded vectors of the words to be processed in order from front to back. The corresponding output is a tensor composed of implicit states of the LSTM units in order from front to back. LSTM can describe long-distance lexical dependency in the text and is suitable for text data modeling<sup>[22]</sup>.

The Bi-LSTM network consists of a forward LSTM and a backward LSTM. The input of the forward LSTM network is composed of the embedded vectors of words arranged in order from front to back, and the input of the backward LSTM network is a series of the embedded vectors of words arranged in the opposite order. Thus, Bi-LSTM can describe the dependency relationship between words in the front to back and back to front directions. The output of Bi-LSTM is the result of the splicing of the output of the forward and backward LSTM units. Therefore, in Bi-LSTM network, each word will be first transmitted to a forward LSTM unit and then to a backward LSTM unit, and its output is the result of the splicing of the hidden states of the 2 LSTM units.

In the LSTM model<sup>[23]</sup>, a unit  $t$  is calculated as follows:

$$i_t = \sigma(x_t U^i + h_{t-1} W^i + b_i) \quad (1)$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f + b_f) \quad (2)$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o + b_o) \quad (3)$$

$$q_t = \tanh(x_t U^q + h_{t-1} W^q + b_q) \quad (4)$$

$$p_t = f_t \times p_{t-1} + i_t \times q_t \quad (5)$$

$$h_t = o_t \times \tanh(p_t) \quad (6)$$

where  $U \in R^{d \times n}$  and  $W \in R^{n \times n}$  are weight matrixes,  $b \in R^n$  is the offset vector,  $d$  is the dimension of the word embedding,  $\sigma$  and  $\tanh$  represent sigmoid and tanh activation functions and  $n$  is the output size of the LSTM network. The LSTM model consists of input gate  $i_t$ , forgetting gate  $f_t$ , and output gate  $o_t$ .

### 2.3 Position-weight fusion layer

When using Bi-LSTM alone to extract the features, it becomes difficult to analyze the differences between multiple targets of the text. This leads to lack of pertinence when the algorithm processes multiple tar-

gets in the same text. Therefore, in order to reflect the differences among the corresponding features of different targets in the text, a two-stage method is designed. The first step calculates the ultimate position-weight vector, and the second step concatenates the position-weight vector and output of the Bi-LSTM layer.

#### 2.3.1 Calculating the final position-weight vector

The positional features of different targets in the text are obtained with the unsupervised method. Generally speaking, the closer a word is to a target, the better it can express its stance on the target. This means that the weight of the word's influence on the target can be expressed as the distance between the word and the target in the sentence. The number of words between the target and word is regarded as the distance between them.  $w_k^m$  represents the weight of the influence of the  $k$ -th word in the text on the  $m$ -th target, which is expressed as

$$w_k^m = 1 - \frac{|k - \tau_m|}{k_{MAX}} \quad (7)$$

where  $\tau_m$  is the serial number of the word corresponding to the  $m$ -th target, and  $k_{MAX}$  is the number of words included in the entire text. There may be multiple targets in one sentence. Thus, the maximum of  $w_k^m$  among all targets is taken as the maximal weight of the  $k$ -th word, which is expressed as  $w_k^{max}$ .

$$w_k^{max} = \max\{w_k^1, w_k^2, \dots, w_k^m, \dots, w_k^{m_{MAX}}\} \quad (8)$$

Then, all the maximum weights  $w_k^{max}$  are spliced into the position-weight vector  $E$ , which is shown as

$$E = \{w_1^{max}, w_2^{max}, \dots, w_{k_{MAX}}^{max}\} \quad (9)$$

At this point, each component of vector  $E$  represents the influence of each word on the target, as shown in Fig. 2. Each element in  $E$  is a value between 0 and 1.

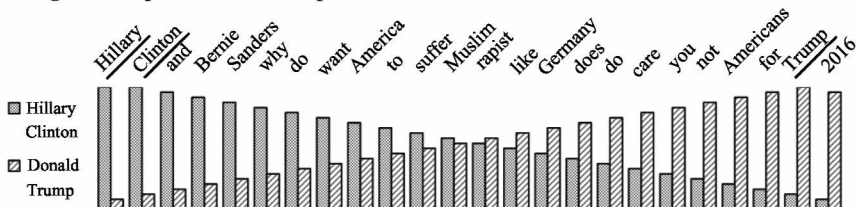


Fig. 2 Position-weight vector of 2 targets in the same text

In order to control the effect of vector  $E$  on the prediction result, the coefficient  $\mu$  is used to expand each component of vector  $E$  by a factor of  $\mu$ . The influence of positional weight on system can be changed by adjusting  $\mu$ , as follows:

$$E_\mu = E \times \mu \quad (10)$$

where  $E_\mu$  is the ultimate position-weight vector. Each element in  $E_\mu$  is a value between 0 and  $\mu$ .

#### 2.3.2 Concatenating position-weight vector and output of Bi-LSTM

In the Bi-LSTM network, the output of each word is composed of the spliced hidden states of the forward and backward LSTM units. In addition, each word corresponds to one position-weight in the  $E_\mu$ . Thus, a new vector is produced by concatenating the position-weight of each word and the Bi-LSTM output. This vector is

taken as the output of the position-weight fusion layer. This vector can not only describe the dependency between words in the different directions, but it can also describe the impact of a word on the different targets of stance detection.

## 2.4 LSTM layer

To determine the stance labels from the fusion of the position-weight and the output of Bi-LSTM, the LSTM is used for re-encoding<sup>[23]</sup>. This process will re-extract features from the fused tensor from the previous section in the order of the text. The input of this layer is the output vector of the position-weight fusion layer,

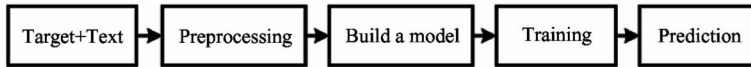


Fig. 3 Flow chart for multi-target position detection

The experiment used the stance detection corpus for the US 2016 general election constructed by Sobhani et al.<sup>[3]</sup>. This corpus contains 3 datasets, each of which is a collection of tweets and stance labels of 2 candidates. In the original corpus, 2 target words of each sentence were combined for analysis in Ref. [3]. Distribution of data are shown in Table 1. In addition, the model parameters are shown in Table 2.

Table 1 Details of the experimental datasets

Target	Train	Dev	Test	Total
Clinton - Sanders	957	137	272	1 366
Clinton - Trump	1 240	177	355	1 722
Cruz - Trump	922	132	263	1 317

Table 2 Main parameter setting in our experiment

Hyper-parameter	Values
Word embedding	fastText <sup>[25]</sup>
Dimension of word embedding	300
Output size of Bi-LSTM/LSTM	256
Iterations	15

## 3.2 Evaluation metric

As a category task, stance detection is more inclined to improve the classification accuracy of the “favor” and “against” stances. Therefore, the average  $F1$  scores of “favor” and “against” ( $F_{avg}$ ) were used as the evaluation indicators<sup>[4]</sup>.

## 3.3 Baselines

The selected baselines are as follows.

Sequence-to-sequence (Seq2Seq)<sup>[26]</sup>. Recently, the Seq2seq model has achieved good performance when dealing with timing problems. Therefore, Ref. [3]

and the output is the hidden state of the last LSTM unit.

## 2.5 Softmax classifier

The output of the LSTM layer is taken as the input of this layer, and the Softmax classifier<sup>[24]</sup> is used to predict the stance labels of the different targets.

## 3 Experiment

### 3.1 Experimental setting

The specific process of completing the multi-objective position detection task is shown in Fig. 3.

applied this model to the multi-target stance detection problem. In this method, text is used as the input of the model, and the stance labels representing different targets are output. The advantage of this algorithm is that it not only mines the stance related to the target from the text, but also refers to the relationship between multiple targets.

Target-related zone modeling (TRZM)<sup>[19]</sup>. This model is proposed for multi-target stance detection tasks. It uses a region segmentation method to divide a text containing 2 targets into 4 parts, and then a multi-input LSTM is used to process these parts to detect the stance results.

## 3.4 Results and discussion

In order to verify the effectiveness of the proposed method, the following 2 experiments are carried out; comparison between the proposed method and the related baselines, and comparisons of different parameters in the position-weight fusion layer.

### 3.4.1 Comparison between the proposed method and the related baselines

The experimental result of the algorithm is compared with those of the other algorithms, as shown in Table 3, where PW-Bi-LSTM is the bidirectional LSTM network with position-weight proposed in this paper. There is the result of PW-Bi-LSTM > TRZM > Seq2Seq, when comparing the  $F1$  value of different methods. The conclusions drawn from these results are as follows.

1) Although this method has the ability to refer to different labels to detect the stance, the method does not take into account the effect of the positional relationship between the text and the target. This may be

the reason why its effect is lower than TRMZ and the model in this article.

2) The effect of TRMZ is not good, but it can meet the actual requirement, because the combination of feature extraction and deep learning is a good way to improve multi-target stance detection. But this method splits the integrity of the text, which may be the reason for its poor performance.

3) The proposed method outperformed the other methods in 3 datasets and macro  $F_{avg}$  are at 1.4%

higher than the corresponding values in the other algorithms. Compared with the other methods, the proposed method can automatically extract the position features of different targets in the text and expand the tolerance for input difference. For input text with different targets, other methods may be impacted by other targets when detecting the given target stance, and their accuracies decrease subsequently. However, the proposed method can avoid the influence of irrelevant targets, and the accuracy does not change much.

Table 3 Performances of our approach and the compared methods

Methods	Clinton-Sanders			Clinton-Trump			Cruz-Trump			Macro
	Clinton	Sanders	Avg	Sanders	Trump	Avg	Cruz	Trump	Avg	
Seq2 Seq <sup>[26]</sup>	55.59	53.86	54.73	54.46	58.74	56.60	47.02	59.21	53.12	54.81
TRMZ <sup>[19]</sup>	52.26	58.38	55.32	60.63	61.98	61.31	52.88	50.30	51.59	56.07
PW-Bi-LSTM	57.10	54.64	55.87	58.70	65.33	62.02	52.67	56.63	54.65	57.51

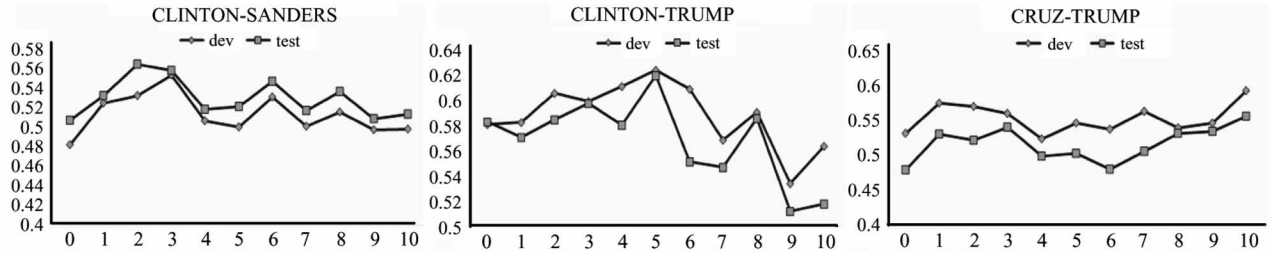


Fig.4  $F_{avg}$  for different coefficients ( $\mu$ ) in the proposed method. The  $x$ -axis denotes the coefficient size, and the  $y$ -axis refers to  $F_{avg}$

### 3.4.2 Comparisons of different parameters in the position-weight fusion layer

One of the key parameters affecting the performance of the proposed method is the coefficient  $\mu$ , mentioned in Section 2.3. In order to determine the influence of the ultimate position-weight vector on the algorithm and to find the optimal coefficient  $\mu$  in the position-weight fusion layer,  $F_{avg}$  for different values of  $\mu$  in the development and test sets in the 3 datasets are compared, as shown in Fig.4. The figure shows that when the ultimate position-weight vector is added to our algorithm (i. e.,  $\mu \neq 0$ ),  $F_{avg}$  are significantly improved, which indicates that this addition can improve the result of the multi-target stance detection. In addition, the effect of the proposed algorithm is related to the coefficient  $\mu$ . In the 3 datasets, the best results in development sets are achieved when  $\mu$  equals 3, 5 and 10, respectively. Thus, the effect of the proposed algorithm can be improved by adjusting the coefficient  $\mu$ .

## 4 Conclusions

In this study, Bi-LSTM network with position-

weight for multi-target stance detection is proposed. The positional relationship between word and target is represented as a vector. And then this vector is embedded into the Bi-LSTM model to refine the stance detection. The experimental results demonstrate the validity of the proposed method, which states that adding the multi-target information can expand the tolerance for the input difference and diversity. In the future, additional position feature extraction methods and actual data covering a wider range of topics will be adopted for continuous improvement and optimization of the algorithm. In addition, it leads to a large volatility of the experiment that the number of data sets used in this paper is small. Therefore, in the follow-up work, the study of the volatility of the results will be considered.

## References

- [1] Wang R, Zhou D, Jiang M, et al. A survey on opinion mining: from stance to product aspect[J]. *IEEE Access*, 2019, 7: 41101-41124
- [2] Li C, Guo X, Mei Q. Deep memory networks for attitude identification[C]//Proceedings of the 10th ACM International Conference on Web Search and Data Mining, Cambridge, UK, 2017: 671-680

- [ 3 ] Sobhani P, Inkpen D, Zhu X. A dataset for multi-target stance detection[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 2017: 551-557
- [ 4 ] Mohammad S, Kiritchenko S, Sobhani P, et al. Semeval-2016 task 6: detecting stance in tweets[C]//Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, USA, 2016: 31-41
- [ 5 ] Xu R, Zhou Y, Wu D, et al. Overview of nlpcc shared task 4: stance detection in chinese microblogs[J]. *Lecture Notes in Computer Science*, 2016, 10102: 907-916
- [ 6 ] Jannati R, Mahendra R, Wardhana C W, et al. Stance classification towards political figures on blog writing[C]//2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia, 2018: 96-101
- [ 7 ] Xu J, Zheng S, Shi J, et al. Ensemble of feature sets and classification methods for stance detection[J]. *Lecture Notes in Computer Science*, 2016, 10102: 679-688
- [ 8 ] Ebrahimi J, Dou D, Low D. A joint sentiment-target-stance model for stance classification in tweets[C]//Proceedings of the 26th International Conference on Computational Linguistics, Osaka, Japan, 2016: 2656-2665
- [ 9 ] Sun Q, Wang Z, Li S, et al. Stance detection via sentiment information and neural network model[J]. *Frontiers of Computer Science*, 2019, 13(1): 127-138
- [ 10 ] Zarrella G, Marsh A. Mitre at semeval-2016 task 6: transfer learning for stance detection[J]. *arXiv* :1606.03784, 2016
- [ 11 ] Yan Y, Chen J, Shyu M L. Efficient large-scale stance detection in tweets[J]. *International Journal of Multimedia Data Engineering and Management*, 2018, 9(3): 1-16
- [ 12 ] Igarashi Y, Komatsu H, Kobayashi S, et al. Tohoku at SemEval-2016 task 6: feature-based model versus convolutional neural network for stance detection[C]//Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, USA, 2016: 401-407
- [ 13 ] Augenstein I, Rocktäschel T, Vlachos A, et al. Stance detection with bidirectional conditional encoding[J]. *arXiv* :1606.05464, 2016
- [ 14 ] Zhou Y, Cristea A I, Shi L. Connecting targets to tweets: semantic attention-based model for target-specific stance detection[C]//International Conference on Web Information Systems Engineering, Puschino, Russia, 2017: 18-32
- [ 15 ] Yu N, Pan D, Zhang M, et al. Stance Detection in Chinese Microblogs with neural networks[J]. *Lecture Notes in Computer Science*, 2016, 10102: 893-900
- [ 16 ] Siddiqua U A, Chy A N, Aono M. Tweet stance detection using an attention based neural ensemble model[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, USA, 2019: 1868-1873
- [ 17 ] Li W, Xu Y, Wang G. Stance detection of microblog text based on two-channel CNN-GRU fusion network[J]. *IEEE Access*, 2019, 7: 145944-145952
- [ 18 ] Rajendran G, Chitturi B, Poornachandran P. Stance-in-depth deep neural approach to stance classification[J]. *Procedia Computer Science*, 2018, 132: 1646-1653
- [ 19 ] Liu H, Li S, Zhou G. Two-target stance detection with target-related zone modeling[C]//China Conference on Information Retrieval, Guilin, China, 2018: 170-182
- [ 20 ] Wei P, Lin J, Mao W. Multi-target stance detection via a dynamic memory-augmented network[C]//The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, USA, 2018: 1229-1232
- [ 21 ] Lin J, Kong Q, Mao W, et al. A topic enhanced approach to detecting multiple standpoints in web texts[J]. *Information Sciences*, 2019
- [ 22 ] Dey K, Shrivastava R, Kaushik S. Topical stance detection for Twitter: a two-phase LSTM model using attention[C]//European Conference on Information Retrieval, Grenoble, France, 2018: 529-536
- [ 23 ] Yin W, Kann K, Yu M, et al. Comparative study of CNN and RNN for natural language processing[J]. *arXiv* :1702.01923, 2017
- [ 24 ] Jiang Y. Using Machine Learning for Stance Detection[D]. The Texas:University of Texas, 2019:14-20
- [ 25 ] Bojanowski P, Grave E, Joulin A, et al. Enriching word vectors with subword information[J]. *Transactions of the Association for Computational Linguistics*, 2017, 5: 135-146
- [ 26 ] Sobhani P, Inkpen D, Zhu X. Exploring deep neural networks for multitarget stance detection[J]. *Computational Intelligence*, 2019, 35(1): 82-97

**Xu Yilong**, born in 1993. He is currently a master's student at Beijing Union University. His research interests include natural language processing and machine learning.