Computers, Materials & Continua CMC, vol.65, no.1, pp.777-788, 2020

Multi-Level Feature-Based Ensemble Model for Target-Related Stance Detection

Shi Li[1], Xinyan Cao[1,*] and Yiting Nan[2]

Abstract: Stance detection is the task of attitude identification toward a standpoint. Previous work of stance detection has focused on feature extraction but ignored the fact that irrelevant features exist as noise during higher-level abstracting. Moreover, because the target is not always mentioned in the text, most methods have ignored target information. In order to solve these problems, we propose a neural network ensemble method that combines the timing dependence bases on long short-term memory (LSTM) and the excellent extracting performance of convolutional neural networks (CNNs). The method can obtain multi-level features that consider both local and global features. We also introduce attention mechanisms to magnify target information-related features. Furthermore, we employ sparse coding to remove noise to obtain characteristic features. Performance was improved by using sparse coding on the basis of attention employment and feature extraction. We evaluate our approach on the SemEval-2016Task 6-A public dataset, achieving a performance that exceeds the benchmark and those of participating teams.

Keywords: Attention, sparse coding, multi-level features, ensemble model.

1 Introduction

Stance detection is the task of automatic judgment of the attitude expressed in a text toward a specific target. Stance detection is similar to traditional emotional analysis, but there are also noticeable differences. Both of them use rules, machine learning, deep learning, and other methods to analyze and classify subjective text. The difference is that emotional analysis determines whether the author expresses a positive, negative, or neutral emotion, while stance detection aims to judge whether the author is in favor or against the given target. The same stance may contain different emotional states.

The classification of stance is a target-dependent sentiment classification problem. Although the text contains the emotional state of the author, some studies have proved that single emotional tendencies cannot express the author's exact stance [Mohammad, Sobhani and Kiritchenko (2017)]. Stance detection is based on sentiment analysis to further analyze the stance tendency of text related to the given target. Therefore, stance mining is used to distinguish the real stance of each party involved and has important

[1] School of Information Engineering, Northeast Forestry University, Harbin, China.

[2] Petabase LLC, Washington DC, 20001, USA.

[*] Corresponding Author: Xinyan Cao. Email: cxy@nefu.edu.cn.

Received: 02 April 2020; Accepted: 09 June 2020.

theoretical and practical significance in many fields such as elections, public social opinion, and product surveys. Recently, stance detection has been applied to fields such as rumor detection and fake news detection [Bourgonje, Schneider and Rehm (2017); Mohtarami, Baly, Glass et al. (2018); Li, Hu, Lu et al. (2020)].

Stance detection is a challenging task because the texts that express the subject points are diverse. The challenges of stance detection in social media are more complicated; this is especially seen in tweets, which are short and informal user-generated text that usually do not follow syntax rules and contain a large number of abbreviations. Moreover, sometimes the tweet stance may be toward a target that is not mentioned in the tweet.

In recent years, most of the related works of short text stance detection have explored machine learning and deep learning models in their methods. We propose a neural network ensemble method to solve the problems of non-standardization, text implication, and satirical recognition of short texts in social media such as tweets. This method can consider local features and global features. First, we employ different kernel sizes of convolutional neural networks (CNNs) to extract local features of different levels. Then, we put the local features into bidirectional long short-term memory (Bi-LSTM) to get the global features that are time-dependent. Finally, we obtain the multi-level features with the concatenation of global features. We also employ an attention mechanism to magnify target-related information and use sparse coding to remove noise. We experiment on a public dataset to verify the validity of the method. The performance is improved compared with the baseline methods, which shows the effectiveness of the target attention matrix. The method also effectively avoids the problem of incomplete consideration of global and local features. Furthermore, the use of sparse coding results in an improvement of 1.8% over the ensemble model based on the attention matrix.

2 Related works

Initially, stance detection was applied to debates and parliamentary elections [Sobhani, Mohammad and Kiritchenko (2016); Anand, Walker, Abbott et al. (2011); Rajadesingan and Liu (2014)]. With the popularity of social networks, stance detection has been applied to short texts on social media regarding parliamentary debates and elections, with the goal of finding the stances users take on hot topics. Therefore, stance detection receives widespread attention and extensive applications, such in as public opinion surveys and hot topic analysis.

Stance detection was derived from the task of sentiment analysis, which also belongs to the study of sentence classification. As a text classification task, stance detection was first formally proposed in SemeEval-2016 [Mohammad, Kiritchenko, Sobhani et al. (2016)]. NLPCC2016 [Xu, Zhou, Wu et al. (2016)] proposed a similar task on a Chinese dataset. The best baseline method trains five support vector machine (SVM) classifiers for the corresponding target with the characteristics of n-grams at the word-level and character-level, and then uses five-fold cross-validation and adjusts the parameters. Using the improved n-gram feature can significantly improve the average F-score. Training a separate classifier for each target is better than training a single classifier for all targets.

Early stance detection was based on feature engineering. A direct method to solve the stance detection problem involves manually designing a target-related feature set and

integrating these features into feature-based classifiers such as the SVM and K-nearest Neighbors [Xu, Zheng, Shi et al. (2016)]. However, these methods require a lot of feature engineering to extract language features manually, and the sparse, discrete features are few. With the increasing amount of text to be processed, the cost of manual extraction of features became too high, and therefore researchers started to apply the deep learning method. Augenstein et al. [Augenstein, Rocktäschel, Vlachos et al. (2016)] experimented with conditional LSTM encoding, which is dependent on the target. They improved the performance with bidirectional encoding. Zarrella et al. [Zarrella and Marsh (2016)] used two recurrent neural network (RNN) classifiers: the first one was employed to predict task-related targets on a substantial unmarked Twitter corpus and initialize the second RNN classifier. However, this method requires a lot of external data for training parameters, which are very dependent on external data. Vijayaraghavan et al. [Vijayaraghavan, Sysoev, Vosoughi et al. (2016)] studied a character-level CNN and word-level CNN to analyze the stance in tweets. However, this method requires separate verification of the sub-models for different targets of the dataset, and there are disadvantages in migration. Later, ensemble models began to be employed since they can combine the merits of the different neural networks [Ren, Zhang and Suganthan (2016)]. Vychegzhanin et al. [Vychegzhanin and Kotelnikov (2019)] proposed an ensemble model based on a cross-validation procedure to evaluate the effectiveness of each combination and select the optimal combination. Meanwhile, Siddiqua et al. [Siddiqua, Chy and Aono (2019)] combined multi-CNNs and two LSTM variants. These two methods outperformed the best approach of SemEval-2016.

In recent years, methods based on the combination of attention and deep learning have achieved excellent results in text classification [Liu, Yang, Lv et al. (2019); Xie, Hou, Wang et al. (2020)]. The combination makes the text feature extraction pertinent. Because each word in a sentence has different importance, the attention mechanism can focus on different relevant information in the text. Yang et al. and Chen et al. [Yang, Yang, Dyer et al. (2016); Chen, Sun, Tu et al. (2016)] used the attention mechanism to extract words that express essential information for the sentence and weigh the words to obtain the sentence vector. The experimental results indicate the effectiveness of applying the attention mechanism to text classification. Dey et al. [Dey, Shrivastava and Kaushik (2018)] proposed a two-phase LSTM-based model with attention embedding: the first phase classifies the tweets into neutral and non-neutral, while the second phase classifies the non-neutral tweets into favor and against stances. Sobhani et al. [Sobhani, Inkpen and Zhu (2019)] proposed an attention-based encoder-decoder framework, and employed different RNN models to solve the potential dependency among targets. Lastly, Wang et al. [Wang, Sun, Li et al. (2020)] proposed a hierarchical network with an attention mechanism to learn the mutual attention between document and linguistic information. The employment of attention measures the importance of different linguistic features.

Recently, researchers have studied user-level stance detection [Darwish, Stefanov, Michaël et al. (2019); Zhu, He and Zhou (2020)]. Darwish et al. propose that if multiple users forward the same controversial topic, then these users are likely to have the same stances. This method is employed to improve the stance detection performance at the

user level. However, in many situations, it is not straightforward to obtain related information at the user level, such as in forwarding. In this case, text-based stance detection is more effective.

Different from previous studies, we propose an ensemble model based on attention and sparse coding. We not only focus on the target-related information but also integrate local features and global features in feature extraction. Furthermore, we employ sparse coding to remove irrelevant features and noise because the features are sparse in the stance text of social media posts and the high-dimensional abstract features contain noise.

3 Methodology

In this section, we propose a method for stance detection in the social network by building a new neural ensemble model based on sparse coding and an attention mechanism. The method consists of three components. First, we employ LSTM to generate an attention matrix. Then, we use different convolution filters to extract features of different levels and separately input these features into Bi-LSTM to obtain the multi-level features, which contain word meanings and the sentence's meaning. Furthermore, we employ sparse coding to remove the noise and put characteristic features into *softmax* for classification.

3.1 Generating an attention matrix based on LSTM

Since stance detection is a target-related task, considering more information about the target can improve detection accuracy. In this work, we utilize the time-dependent ability of LSTM [Hochreiter and Schmidhuber (1997)] to learn information about the target. Furthermore, we generate an attention matrix as a weight matrix through target information learning. The weight matrix is a coefficient of vector stance text and makes each word focus on target-related information.

First, we obtain word embedding L_w ($L_w \in R^k \times |V|$) of each target. Here, k is the dimension of the word vector, and $|V|$ is the size of the word vector. Then, we feed target vectors into LSTM and select the last state representation as the word embedding expression of the target information after calculating the implicit representation of each position.

In the transmission, the sequence $\{w_1, w_2..., w_T\}$ is stance description information. The initial state is 0, h is the implicit output, and g is the implicit unit state. Moving from the previous state to the next state is expressed as follows:

$$[h_0, g_0] = 0 \tag{1}$$

$$[h_1, g_1] = LSTM(w_1, h_0, c_0) \tag{2}$$

...

$$[h_T, c_T] = LSTM(w_T, h_{T-1}, c_{T-1}) \tag{3}$$

Finally, we generate an attention matrix α with a weight matrix of stance text, α is shown in Eq. (4).

$$\alpha = [\alpha_1, \alpha_2 ... \alpha_n] (\alpha \in R^d) \tag{4}$$

where d is the dimension of the vector, n is the maximum target length, and $\alpha_{i(i\in 1,2...n)}$ is the weight of each target word.

3.2 Combining different sizes of CNNs with Bi-LSTM

In this section, we describe an ensemble model to obtain the multi-level features through combining different sizes of CNNs and Bi-LSTM. We employed convolution kernels of different sizes to extract features. Because the CNN has excellent feature extraction ability, and different filters have different sensitivities, the CNN can capture different features [Kim (2014)]. In this work, we feed different features into Bi-LSTM to obtain the global features, which include the local features. Then, we can obtain a higher-level global feature that concatenates different levels of global features. In the following, we describe the specific details of the different sizes of CNNs and Bi-LSTM in our model.

The purpose of convolution is to extract abstract expressions from the text. First, the word vector is fed into the convolution layer. The word vector matrix is the target-related vector matrix, which is the dot product of the pre-trained text and the attention matrix H. H is shown in Eq. (5).

$$H = \alpha \cdot X \tag{5}$$

where α is an attention matrix; X is the words matrix with maximum length n. X is shown in Eq. (6).

$$X = [x_1, x_2 ... x_n](x_{i(i\in 1,2...n)} \in R^k) \tag{6}$$

where $x_{i(i\in 1,2...n)}$ is the k-dimensional vector of the word in the sentence.

Then, we extract abstract features in H by utilizing different filter sizes. Here, we control the feature abstraction through the filter size and recalculate when the matrix H is invariant. Generally, in the single convolution filter, $\odot$ is used to represent the convolution operation between the matrix H and the filter F, that is, the abstract feature Y is defined as in Eq. (7).

$$Y_{\mathrm{j}} = \sum (F_j \odot H_{[:,i:i+m-1]}) \tag{7}$$

where j is the filter size, and $H_{[:,i:i+m-1]}$ is the matrix segment. To obtain the simplified expression of the lower dimension, we use maximum pooling to reduce the dimension. The equation is shown in Eq. (8).

$$C_{\mathrm{j}} = Max(Y_{j\in L}) \tag{8}$$

where L is the collection of different filter sizes.

After extracting local features, we input these features into Bi-LSTM. The forward direction of Bi-LSTM can focus on past information, and the backward direction can focus on future information. In the single transmission, a forward map transfer from a previous state to a subsequent state is performed as described in 3.1.

Therefore, output vectors in two directions are generated, and the final bidirectional output d is the sequential connection of the forward h_l and backward h_r, as shown in Eq. (9).

$$d = h_l \oplus h_r (h_l, h_r \in R^{l\times w}) \tag{9}$$

Here, w is the number of hidden units.

Eventually, we fuse the different levels of feature abstraction, as shown in Eq. (10).

$$D = d_1 \oplus d_2 ... \oplus d_j \tag{10}$$

where D is an abstract feature with multi-level stance information.

3.3 Applying sparse coding

We employ sparse coding to remove noise during feature extraction. The implementation of sparse coding does not depend on the nature of the input data but rather the statistical characteristics of the natural environment. Sparse coding aims to reconstruct data with a few components. Feature extraction mostly contains redundant components in the original model, i.e., noise, which has a specific influence on the study. Furthermore, the features of short social media texts always contain some noise. In this study, we obtain high-relevance features through the application of sparse coding, that is, the significant features of the stance text.

Using sparse coding satisfies condition Eq. (11).

$$x_n = \sum_{k=1}^{K} a_{nk} \phi_k (n, k \in R) \tag{11}$$

where x_n is a set of given vectors, and ϕ_k is a dictionary of bases. The sparse coding training process is shown in Eq. (12).

$$\min_{a,\phi} \sum_{n=1}^{N} ||x_n - \sum_{k=1}^{K} a_{n,k} \phi_k ||^2 + \lambda \sum_{n=1}^{N} \sum_{k=1}^{K} | a_{n,k} | \tag{12}$$

where $\min_{a,\phi} \sum_{n=1}^{N} ||x_n - \sum_{k=1}^{K} a_{n,k} \phi_k ||^2$ is the reconstruction error, $\lambda \sum_{n=1}^{N} \sum_{k=1}^{K} | a_{n,k} |$ is the sparsity penalty, where λ is a constant, and $| a_{n,k} |$ is the L_1-norm regularization on $a_{n,k} (a_{n,k} \geq 0)$.

Sparse coding can be completed by optimizing the former part of Eq. (12) because the coefficient vector is sparser and the value of the objective equation is smaller. Thus far, in the process of optimization, the coefficient vector can develop in a sparser direction.

4 Experiment

We evaluated the proposed method for stance detection by conducting experiments.

4.1 Dataset collection and preparation

We chose a public tweets dataset used in the SemEval-2016Task 6-A (Mohammad et al. [Mohammad, Kiritchenko, Sobhani et al. (2016)]) to conduct our experiments. The training set consists of 2914 tweets, and the test set consists of 1249 tweets relevant to five targets:

"Atheism," "Climate Change is a Real Concern" ("Climate"), "Feminist Movement" ("Feminist"), "Hillary Clinton" ("Hillary"), and "Legalization of Abortion" ("Abortion"). Each tweet was annotated as Favor, Against, or None toward the specific target. Specific data statistics are reported in Tab. 1.

Table 1: Distribution of instances in the stance training and test sets (the total number of training and test tweets are shown in bold)

		% of instances in training set				% of instances in test set			
Target	**#total**	**#train**	favor	against	neither	**#test**	favor	against	neither
Hillary Clinton	984	689	17.1	57.0	25.8	295	15.3	58.3	26.4
Feminist Movement	949	664	31.6	49.4	19.0	285	20.4	64.2	15.4
Legal Abortion	933	653	18.5	54.4	27.1	280	16.4	67.5	16.1
Atheism	733	513	17.9	59.3	22.8	220	14.5	72.7	12.7
Climate Change	564	395	53.7	3.8	42.5	169	72.8	6.5	20.7
All	**4163**	**2914**	25.8	47.9	26.3	**1249**	24.3	57.3	18.4

Apart from tweets that express an opinion toward the target, the dataset also includes tweets in which the target of opinion is not directly given. On the SemEval2016 website, an interactive visualization of the stance dataset shows various statistics about the data, including the target of opinion annotations. The stance dataset divided opinion toward the target, other, and no opinion. We collect statistics of the instances of opinion toward the stance labels. Whether the opinion is expressed directly about a given target is shown in Tab. 2.

Table 2: Distribution of instances in the opinion toward (opinion's relevance about the given target is shown in bold)

	Opinion Toward		
Stance	Target	Other	No Opinion
FAVOR	94.23%	5.11%	0.66%
AGAINST	72.75%	26.54%	0.71%
NEITHER	0.90%	79.52%	19.58%
ALL	**36.18%**	**58.81%**	**5.01%**

We prepared the text by removing irrelevant symbols and performing case conversion. There are a lot of consecutive texts such as "WhyInotVotingForHillary," which is rewritten using regular expressions and processed as "why I not voting for Hillary."

4.2 Experimental settings

During preprocessing, the 150-dimensional word2vec model was trained on the dataset in the embedding layer. The maximum length of the text after preprocessing was 30.

A 30×150-dimensional attention matrix was generated through 30 LSTM neurons. The activation function of LSTM was *tanh*.

We then employed four kernel sizes (2, 3, 4, 5) to extract local features. the number of filters was 32. Then, we spliced the local features as a higher-level feature. In the

convolution layer, we used the *ReLU* activation function, selected the maximum pooling, and used a dropout size of 0.2 to avoid over-fitting. The activation function of Bi-LSTM was *tanh*.

L_1 regularization with 10e-5 was applied to the sparse coding. L_2 regularization with 10e-6 was applied to avoid over-fitting in the *Dense* layer. The learning rate was 0.01. We used the mean *F1* score as the evaluation index. Parameter statistics are shown in Tab. 3.

Table 3: Parameters of our model

Parameter	Value
Word embedding size	150
LSTM neurons	30
Bi-LSTM neurons	64
kernel sizes	2, 3, 4, 5
filters number	32
dropout	0.2
learning rate	0.01
L_1-norm	1.00E-05
L_2-norm	1.00E-06

4.3 Results and discussion

In Tab. 4, the results of our proposed method are compared to those of the benchmark methods SVM-unigrams and SVM-grams [Mohammad, Kiritchenko, Sobhani et al. (2016)] and to those of the top-three participating teams, MITRE [Zarrella and Marsh (2016)], pkudblab [Wei, Zhang, Liu et al. (2016)], and Takelab [Tutek, Sekulić, Gombar et al. (2016)]. A-ensemble is an ensemble method based on attention, and AS-ensemble is an ensemble method based on attention and sparse coding. A-ensemble obtained a better result, while AS-ensemble had the best performance and could surpass the baseline. Moreover, utilizing sparse coding led to a 1.8% improvement over A-ensemble.

The results of the experiment demonstrate that our proposed method has the following advantages. First, our approach minimizes the steps of linguistic processing, and use nothing of the external corpus. Previous methods mainly rely on linguistic processing or external lexicons, which increases the number of processing tasks and the possibility of error due to interference from external data. Second, we consider target information in the application of the attention matrix. The attention matrix serves as the weight matrix of the text, and concerns the target-related information. Third, we obtain multi-level features by the extraction of local features and global features. Through comparison with other methods, we found that the performance of SVM-ngrams is much higher than the performance of SVM-unigrams; this demonstrates that employing higher-order n-grams features is much better than using unigrams features. It also illustrates that the extraction performance of the multi-kernel CNN is better than that of the single-kernel CNN. Furthermore, we optimize the performance of stance detection through sparse coding. Sparse coding can remove noise by utilizing the statistical characteristics of the natural environment.

We found that the extraction performance of the against stance is better than that of the favor stance. This may be because the number of tweets toward against is more than that toward favor. Furthermore, the baseline training of a separate classifier for each target is better than training a single classifier for all targets, and thus the best baseline method has higher performance than the methods of participating teams. Training separate classifiers for overall targets is likely to more easily balance 'favor' and 'against' than only one classifier. The best experimental result of our method is better than that of the best baseline method. However, we found that the improvement toward against is better than that toward favor. We consider that the reason for this may be that the percentage of target-related tweets toward favor is higher than those against. According to Tab. 2, 94.23% of target-related tweets are toward favor while 72.75% are against. Because the opinion toward others also has a stance, the employment of the attention matrix makes target-unrelated tweets concern the target. Therefore, the higher the proportion of other opinions, is more significant the improvement. Tab. 2 shows that 5.11% of other opinions are toward favor while 26.54% are against. Another reason for this may be that the sparse coding is randomly sparse, and the number of tweets with against stances is higher than that with favor stances. Thus, the effect of removing noise from the against stance is more stable than for the favor stance.

Table 4: Results of different methods (the highest scores in each column are shown in bold)

Method	F_{favor}	$F_{against}$	F_{avg}
Baselines			
SVM-unigrams	54.49	72.13	63.31
SVM-ngrams	**62.98**	74.98	**68.98**
Participating Teams			
MITRE	59.32	76.33	**67.82**
pkudblab	61.98	72.67	67.33
Takelab	60.93	72.73	66.83
We propose			
A-ensemble	56.8	77.9	67.95
AS-ensemble	59.2	**79.2**	**69.20**

5 Conclusion

This paper showed that the ensemble model based on attention and sparse coding is a successful approach to stance detection. The target attention matrix concerns target-related information, and sparse coding can remove the noise of features. When applying both an attention mechanism and sparse coding, our method surpasses the baseline method on the SemEval-2016Task 6-A Twitter Stance Detection corpus. Our future work will focus on methods of extracting characteristic features from multiple targets and fake news detection based on stance detection.

Funding Statement: This work is supported by the Fundamental Research Funds for the Central Universities (Grant No. 2572019BH03).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Anand, P.; Walker, M.; Abbott, R.; Tree, J.; Bowmani, R. et al. (2011): Cats rule and dogs drool! Classifying stance in online debate. *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*. *Association for Computational Linguistics*, pp. 1-9.

Augenstein, I.; Rocktäschel, T.; Vlachos, A.; Bontcheva, K. (2016): Stance detection with bidirectional conditional encoding. *Association for Computational Linguistics*, pp. 876-885.

Bourgonje, P.; Schneider, J. M.; Rehm, G. (2017): From clickbait to fake news detection: an approach based on detecting the stance of headlines to articles. *Proceedings of the EMNLP Workshop: Natural Language Processing Meets Journalism*, pp. 84-89.

Chen, H.; Sun, M.; Tu, C.; Lin, Y.; Liu, Z. (2016): Neural sentiment classification with user and product attention. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1650-1659.

Darwish, K.; Stefanov, P; Michaël, A.; Nakov, P. (2019): Unsupervised user stance detection on twitter. *The Computing Research Repository*.

Dey, K.; Shrivastava, R.; Kaushik, S. (2018): Topical stance detection for twitter: a two-phase LSTM model using attention. *European Conference on Information Retrieval*, pp. 529-536.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735-1780.

Kim, Y. (2014): Convolutional neural networks for sentence classification. *Empirical Methods in Natural Language Processing*, pp. 1746-1751.

Li, Q.; Hu, Q.; Lu, Y.; Yang, Y.; Chen, J. (2020): Multi-level word features based on CNN for fake news detection in cultural communication. *Personal and Ubiquitous Computing*, vol. 24, no. 2, pp. 259-272.

Liu, J.; Yang, Y. H.; Lv, S. Q.; Wang, J.; Chen, H. (2019): Attention-based BiGRU-CNN for Chinese question classification. *Journal of Ambient Intelligence and Humanized Computing*.

Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X. D.; Cherry, C. (2016): Semeval-2016 task 6: detecting stance in tweets. *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 31-41.

Mohammad, S.; Sobhani, P.; Kiritchenko, S. (2017): Stance and sentiment in tweets. *ACM Transactions on Internet Technology*, vol. 17, no. 3, pp. 1-23.

Mohtarami, M.; Baly, R.; Glass, J.; Nakov, P.; Màrquez, L. et al. (2018): Automatic stance detection using end-to-end memory networks. *Association for Computational Linguistics*, pp. 767-776.

Rajadesingan, A.; Liu, H. (2014): Identifying users with opposing opinions in twitter debates. *Lecture Notes in Computer Science*, vol. 8393, pp. 153-160.

Ren, Y.; Zhang, L.; Suganthan, P. N. (2016): Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computation Intelligence Magazine*, vol. 11, no. 1, pp. 41-53.

Siddiqua, U. A.; Chy, A, N.; Aono, M. (2019): Tweet stance detection using multi-kernel convolution and attentive LSTM variants. *IEICE Transactions on Information and Systems*, vol. 102-D, no. 12, pp. 2493-2503.

Sobhani, P.; Inkpen, D.; Zhu, X. (2019): Exploring deep neural networks for multitarget stance detection. *Compurtational Intelligence*, vol. 35, no. 1, pp. 82-97.

Sobhani, P.; Mohammad, S.; Kiritchenko, S. (2016): Detecting stance in tweets and analyzing its interaction with sentiment. *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pp. 159-169.

Tutek, M.; Sekulić, I.; Gombar, P.; Paljak, I.; Najder, J. (2016): Takelab at semeval-2016 task 6: stance classification in tweets using a genetic algorithm based ensemble. *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 464-468.

Vijayaraghavan, P.; Sysoev, I.; Vosoughi, S.; Roy, D. (2016): Deep stance at semeval-2016 task 6: detecting stance in tweets using character and word-level CNNs. *The Association for Computer Linguistics*, pp. 413-419.

Vychezhanin, S. V.; Kotelnikov, E. V. (2019): Stance detection based on ensembles of classifiers. *Programming and Computer Software*, vol. 45, no. 5, pp. 228-240.

Wang, Z.; Sun, Q.; Li, S.; Zhu, Q.; Zhou, G. (2020): Neural stance detection with hierarchical linguistic representations. *IEEE-ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 635-645.

Wei, W.; Zhang, X.; Liu, X.; Chen, W.; Wang, T. (2016): Pkudblab at semeval-2016 task 6: a specific convolutional neural network system for effective stance detection. *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 384-388.

Xie, J.; Hou, Y.; Wang, Y.; Wang, Q; Li, B. et al. (2020): Chinese text classification based on attention mechanism and feature-enhanced fusion neural network. *Computing*, vol. 102, no. 3, pp. 683-700.

Xu, J.; Zheng, S.; Shi, J.; Yao, Y.; Xu, B. et al. (2016): Ensemble of feature sets and classification methods for stance detection. *Natural Language Understanding and Intelligent Applications*, pp. 679-688.

Xu, R.; Zhou, Y.; Wu, D.; Lin, G.; Du, J. et al. (2016): Overview of NLPCC shared task 4: stance detection in Chinese microblogs. *Natural Language Understanding and Intelligent Applications*, pp. 907-916.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A. J. et al. (2016): Hierarchical attention networks for document classification. *Proceedings of the 2016 conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480-1489.

Zarrella, G.; Marsh, A. (2016): Mitre at Semeval-2016 task 6: transfer learning for stance detection. *The Association for Computer Linguistics*, pp. 458-463.

Zhu, L.; He, Y.; Zhou, D. (2020): Neural opinion dynamics model for the prediction of user-level stance dynamics. *Information Processing & Management*, vol. 57 no. 2, pp. 102031.