Learning Stance Classification with Recurrent Neural Capsule Network

Lianjie Sun[1], Xutao Li[1(✉)], Bowen Zhang[2], Yunming Ye[1], and Baoxun Xu[3]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
[2] School of Computer Science and Technology, Harbin Institute of Technology,Harbin, China
[3] Information Management Department Shenzhen Stock Exchange, 518038, China
lixutao@hit.edu.cn

Abstract. Stance classification is a natural language processing (NLP) task to detect author's stance when give a specific target and context, which can be applied in online debating forum, e.g., Twitter, Weibo, etc. In this paper, we present a novel target orientation recurrent neural capsule network, called TRNN-Capsule to solve the problem. In TRNN-Capsule, the target and context are both encoded by leveraging a bidirectional LSTM model. Then, capsule blocks are appended to produce the final classification outcome. Experiments on two benchmark data sets are conducted and the results show that the proposed TRNN-Capsule outperforms state-of-the-art competitors for the stance classification task.

Keywords: Stance Classification · RNN Capsule Network.

1 Introduction

With the growth of the extensive collection of stance-rich resources, much attention has been given to stance classification. Essentially, stance classification is a prediction problem about people's attitude towards a specific topic, and many conventional machine learning methods have been utilized or adapted to solve the problem. For instance, climate change is very serious problem and it is essential to understand whether the public is concerned about the problem. To this end, we need to conduct the stance classification. In this case, " climate change is a real concern" is regarded as the target, and the goal of stance classification is to classify people's stance given the target.

Existing methods for stance classification can be divided into two classes, which are feature-based and corpus-based approaches. Feature-based method mainly focus on how to design rich features [2, 3], e.g., arguing lexicon. In contrast, corpus-based approaches use machine learning models to train a classifier [19]. As feature engineering is often labor-intensive, in this paper, we mainly focus on corpus-based approaches.

With the success of deep learning in natural language processing (NLP) task, many researchers leverage deep learning method to carry out stance classification. Augenstein et al. [4] utilized a conditional long short-term memory network to represent the target dependent context. [5, 6] adopted the attention mechanism to extract target-related information for stance detection.

Although these methods have achieved excellent performance in stance classification tasks, there are still some defects. First, most conventional algorithms cannot effectively identify the relationship between target and text. However, target information plays a key role in stance classification. Therefore, it is important to find the dependency relationship between target and text. Secondly, existing models focus and depend heavily on the quality of instance representation. However, an instance can be a sentence, a paragraph or a document. It is very limited to use a vector to represent stance information because stance information can be subtle and sophisticated.

To alleviate the above shortcomings, we apply the RNN-Capsule network and develop a novel target oriented RNN capsule network for stance classification. RNN-Capsule network is initially introduced by [1] for sentimental analysis, where each capsule is composed of multiple neurons and the neurons form a presentation of the original text, summarizing the semantic information of words, n-gram information, etc. In other words, RNN-Capsule network can model the abundant feature information in the original text. Hence, we propose to adapt RNN Capsule network and develop a novel target oriented RNN-Capsule network (TRNN-Capsule) for stance classification.

The proposed TRNN-Capsule is mainly composed of three layers, which are an embedding layer, an encoding layer and a capsule layer. In the embeding layer, word2vec representation is utilized to represent each word. Then the representations of target and context are encoded by a bidirectional LSTM, respectively. Finally, the capsule layer is constructed. For each stance category, a carefully designed capsule block is embedded, which can produce the output probability of corresponding category. To better capture the information in target and context, we develop a useful attention mechanism in the capsule part. Experimental results on two benchmark data sets are reported, which show that the developed TRNN-Capsule method outperforms state-of-the-art competitors in stance classification problem. The main contributions of the paper can be summarized as follows:

1. To the best of our knowledge, we are the first to introduce RNN-Capsule into stance classification. We develop a TRNN-Capsule model, which extracts the rich stance tendencies features with multiple vectors, instead of one vector.
2. A useful attention mechanism is developed, which can effectively identify the relationship between target and context.
3. Experimental results on H&N14 and SemEval16 datasets demonstrate the proposed method is superior to state-of-the-art stance classification competitors.

2 Related Work

2.1 Stance Classification

Early methods for stance classification adopted typical linguistic features. For example, Somasundaran and Weibo [2] constructed an arguing lexicon and employed sentiment-based and arguing-based features. In addition to linguistic features, previous work also utilized all kinds of extra information, such as citation structure information [9], dialogic structure information [7]. Sridhar et al. [8] used probabilistic soft logic [11] to model the post stance by leveraging both local linguistic features as well as the observed network structure of the posts.

With the popularity and success of deep learning techniques in natural language processing, many researchers begun to applied such techniques into stance classification task. Augenstein et al. [4] encoded context and target using bidirectional LSTM respectively, and then combined them with conditional encoding. Du et al. [5] emphasized the importance of target and applied the attention mechanism into the stance classification for the first time. Sun et al. [6] employed linguistic factors (i.e., sentiment, argument, dependency) into the neural model for stance classification.

2.2 Capsule Networks

As CNN and RNN both form a vector representation given an input, which may fail to preserve the rich information, The concept of "capsules" is proposed by Hinton et al. [14] to solve the limitations. The capsule network show great capacity through achieving a state-of-the-art result on MNIST data. Zhang et al. [16] introduced a Capsule network for sentiment analysis in Domain Adaptation scenario with semantic Rules. Wang et al. [1] proposed RNN-Capsule model and applied it into primary sentiment classification.

To date, no study has ever utilized RNN-Capsule network in stance classification task.

3 The Proposed Method

The overall architecture of TRNN-Capsule model is shown in Figure 1. The TRNN-Capsule consists of three layers, an embedding layer, an encoding layer and a capsule layer. We describe the details of three layers in the following sub-sections.

3.1 Embedding Layer

The first layer is the embedding layer. Given a descriptive target and the context, we use the word embedding [12] which is a dense vector to represent each word in the target and text. As shown in Fig.1, the output of this layer are two sequences of vectors $T = [w_t^1, w_t^2, ..., w_t^m]$ and $C = [w_c^1, w_c^2, ..., w_c^n]$, where m, n are the number of word vectors of target and context respectively.

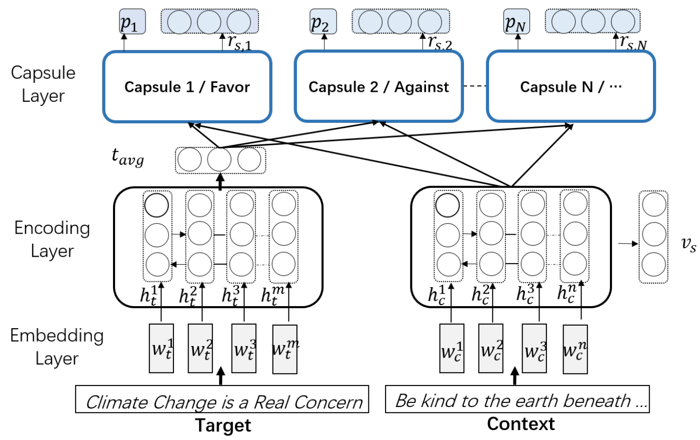

Fig. 1. The overall architecture of TRNN-Capsule.

3.2 Encoding Layer

In this layer, Like [4], we use bidirectional LSTM to encode target and context respectively. Formally, in LSTM, given the current input word embedding w^k, previous cell state c^{k-1} and previous hidden state h^{k-1}, the current cell state c^k and current hidden state h^k are calculated by the following formulae.

$$i^k = \sigma(W_i w^k + U_i h^{k-1} + V_i c^{k-1}) \tag{1}$$

$$f^k = \sigma(W_f w^k + U_f h^{k-1} + V_i c^{k-1}) \tag{2}$$

$$o^k = \sigma(W_o w^k + U_o h^{k-1} + V_o c^{k-1}) \tag{3}$$

$$\tilde{c} = tanh(W_c w^k + U_c h^{k-1}) \tag{4}$$

$$c^k = f^k \odot c^{k-1} + i^k \odot \tilde{c} \tag{5}$$

$$h^k = o^k \odot tanh(c^k) \tag{6}$$

where i^k, f^k and o^k are input gate, forget gate and output gate respectively. They are all vectors in $\mathbb{R}^d$. $W_{\{i,f,o,c\}}$, $U_{\{i,f,o,c\}}$, $V_{\{i,f,o\}}$ are all the weight parameters to be learned. σ is the sigmoid function and $\odot$ is element-wise multiplication. And then, we can get the a series hidden states $[h_c^1, h_c^2, ..., h_c^n]$ which is the final word representation for context and a series hidden states $[h_t^1, h_t^2, ..., h_t^m]$ which is the final word representation for target.

Target Representation Target information is essential to determine the stance for a given context. To extract the important target-related words with stance tendencies from context, we make target representation as the query and then utilize the attention mechanism to get the important target-related words with stance tendencies from context (Details will be covered in the Capsule layer). In our study, we use the average vector t_{avg} as the target representation:

$$t_{avg} = \frac{1}{m} \sum_{i=1}^{m} h_t^i \tag{7}$$

Context Representation As shown in Figure 1, the context representation v_s is the average of the hidden state vectors obtained from the LSTM:

$$v_s = \frac{1}{n} \sum_{i=1}^{n} h_c^i \tag{8}$$

where n is the length of context, and h_i is the i_{th} hidden state for context.

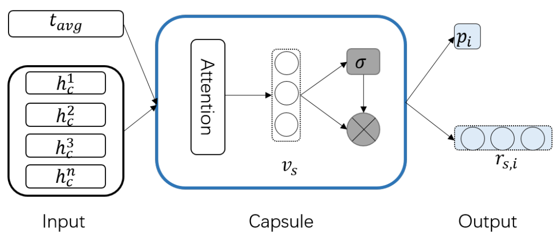

Fig. 2. The overall architecture of a single capsule.

3.3 Capsule Layer

The structure of a single capsule is shown in Figure 2. The number of capsule is consistent with the number of stance categories, and we make the target representation t_{avg} and context hidden state $[h_c^1, h_c^2, ..., h_c^n]$ as input for each capsule. In other words, all the capsule blocks have the same input.

A capsule contains three modules: representation module, probability module, reconstruction module, where the probability module and reconstruction module are consistent basically with [1]. To fit the stance classification task, we proposed our representation module. Since the internal structure of each capsule is the same, let's take a capsule as an example to illustrate the three modules.

Representation Module Given the target representation t_{avg} and context hidden states $[h_c^1, h_c^2, ..., h_c^n]$, we make t_{avg} as the query and utilize the attention mechanism to generate the capsule representation.

$$e_t^i = h_c^t w_a^i t_{avg}, \quad a_t^i = \frac{exp(e_t^i)}{\sum_{k=1}^{n} exp(e_k^i)} \tag{9}$$

where h_c^t is the t_{th} hidden state in context, w_a^i is the attention parameter for i_{th} capsule. By multiplying the h_c^t, w_a^i, t_{avg}, and then normalizing the result into a probability over all hidden states in context. we can get the a_t^i which is the importance score of h_c^t in the context. After calculating importance score a_i for all hidden states in context, $a_i = [a_1^i, a_2^i, ..., a_n^i]$, we can get the capsule representation v_c^i by:

$$v_c^i = \sum_{t=1}^{n} a_t^i h_t^i \tag{10}$$

The capsule representation in each capsule is used to compute the state probability and reconstruction representation.

Probability Module After obtaining the corresponding capsule representation v_c^i, we can get the state probability p_i by:

$$p_i = \sigma(w_p^i v_c^i + b_p^i) \tag{11}$$

where w_p^i and b_p^i are weight matrix and bias respectively for probability module of i_{th} capsule, and p_i is state probability.

The capsule with the highest state probability will be activated, and the final predicted stance category is consistent with the activated capsule category.

Reconstruction Module After obtaining the corresponding capsule representation v_c^i and state probability p_i, we can obtain the reconstruction representation r_s^i by:

$$r_s^i = p_i v_c^i \tag{12}$$

Since the reconstruction representation is calculated from capsule representation and state probability, the reconstruction representation whose state is active can represent the full input context.

3.4 Model Training

On the one hand, because the final predicted category is consistent with the category of the activated capsule, only one capsule can be activated. Therefore, one of our goals is to maximize the active state probability and minimize the inactive state probabilities. On the other hand, since the reconstruction representation whose state is activated can represent the full input context, the other one goal is to maximize the reconstruction error for inactive capsules and minimize the reconstruction error for the active capsule.

Probability Objective To maximize the active state probability and minimize the inactive state probabilities, the objective J with hinge loss can be calculated as:

$$J(\Theta) = max(0, 1 + \sum_{i=1}^{N} y_i p_i) \tag{13}$$

where Θ stands for all the parameters to be learned, and N is the number of the capsule. The value of y is related to the state of the capsule. For a given training instance, the corresponding y of the activated capsule is set to -1, and the y corresponding to all remaining capsules are set to 1.

Reconstruction Objective To maximize the reconstruction error for inactive capsules and minimize the reconstruction error for active capsule, the objective U with hinge loss can be calculated as:

$$U(\Theta) = max(0, 1 + \sum_{i=1}^{N} y_i v_s r_s^i) \tag{14}$$

Θ, y and N has been defined in the probability objective part. v_s is the context representation, and r_s^i is the reconstruction representation for i_{th} capsule.

The final objective function L is defined obtained by adding the above two parts and L_2 regularization together:

$$L(\Theta) = J(\Theta) + U(\Theta) + \lambda_r(\sum_{\theta \in \Theta} \theta^2) \tag{15}$$

λ_r is the coefficient for L_2 regularization.

4 Experiments

4.1 Experiment Preparation

Dataset In this study, we conduct experiments on two benchmark datasets to validate the effectiveness of our proposed model.

H&N14 Dataset. [10] collected H&N14 dataset and utilized it for stance classification and reason classification. In the dataset, there are more than 4000 debate posts which are collected from an online debate forum. The debate posts contain four popular domains, Abortion, Gay Rights, Obama, and Marijuana. Every debate post contains two stance label, *favor* and *against*. We use five-fold cross-validation on this dataset. The distribution of the dataset is shown in Table 1.

SemEval16 Dataset. This dataset is released by [3] for stance from English Tweets. The tweets contains five targets: "Atheism", "Climate Change is a Real Concern", "Feminist Movement", "Hillary Clintion", and "Legalization of Abortion". Each tweet has a specific target and is annotated by *favor*, *against* and *none*. The distribution of the dataset is shown in Table 2.

Evaluation Metric Like [3] and [5], we utilize the average value (F_{avg}) of the F1-score for *favor* category and *against* category as the evaluation metrics. In addition, we calculated the F_{avg} across all targets to obtain the micro-average F1-score ($MicF_{avg}$).

Table 1. Distribution of H&N14 dataset

Target	Favor (%)	Against (%)	Total
Abortion	54.9	45.1	1741
GayRights	63.4	36.6	1376
Obama	53.9	46.1	985
Marijuana	69.5	30.5	626

Table 2. Distribution of SemEval16 dataset

Target	Favor (%)	Against (%)	None (%)	Total
Atheism	16.9	63.3	19.8	733
Climate Change is Concern	59.4	4.6	36.0	564
Feminist Movement	28.2	53.9	17.9	949
Hillary Clinton	16.6	57.4	26.0	984
Legalization of Abortion	17.9	58.3	23.8	933

Hyperparameters Setting In our experiments, all word vectors are initialized by word2vec [12]. For SemEval16, word embedding is pre-trained on unlabelled corpora which are crawled from Twitter. For H&N14, word embedding is pre-trained on training data. The dimension of the word is 300 and fine-tuning during the training process. We use bi-directional LSTM, and the size of units of LSTM is 300 and 512. The dropout rate is 0.4 and we use Adam [13] as our optimization method. The two-parameter β_1 and β_2 are 0.9 and 0.999. The other hyper-parameters and learning rate are fine-tuned on the validation data which is obtained by extracting 10% from the training data.

4.2 Model Comparisons

To validate the effectiveness of our proposed model, we compare TRNN-Capusle with several state-of-the-art baselines for stance classification.

Baseline Methods

- **Neural Bag-of-Words(NBOW)** is a basic baseline. [5] leverages it as a baseline model and it sums the word vectors within the context and applies a softmax classifier.
- **LSTM** only uses context embedding, and learns the context representation through LSTM network.
- **LSTM**$_E$ utilizes the target information. Specifically, LSTM$_E$ appends the average of target word embedding to the embedding of each word in origin context.

State-of-the-art Methods

- **AT-biGRU** [15] utilizes two BiGRUs to represent the target and tweet respectively. Moreover token-level attention mechanisms is adopted to find important words in tweets.

– **AS-biGRU-CNN** [15] extends the attention used in AT-biGRU through a gating structure and stacks CNNs at the top.
– **TAN** is proposed by [5] and utilize both target information and context information. TAN model proposed a target-specific attention extractor to extract the important information which is highly related to the corresponding target.
– **HAN** is proposed by [6]. HAN fully employs the linguistic factors, such as sentiment, argument, and dependency, and then utilizes the mutual attention between context and the linguistic factors to learn the final context representation for stance classification.

In our experiment, the micro-average F1-score ($MicF_{avg}$) across targets is adopted as the final metrics. We summarize the experimental results in Table 3 and Table 4. From Table 3 and Table 4, we can observe that, on both datasets, NBOW and LSTM are the worst. On H&N14 and SemEval16, LSTM is **1.03** and **3.03** lower than LSTM$_E$ respectively. Because NBOW and LSTM don't make use of target information, they only extract some simple information and cannot highlight important target-related information for stance classification.

Table 3. Comparison with baselines on H&N14 dataset.

Model	Abortion	GayRights	Obama	Marijuana	$MicF_{avg}$
NBOW	60.56	55.50	58.86	54.09	59.39
LSTM	60.72	56.07	60.14	55.58	59.45
LSTM$_E$	62.24	56.94	60.54	56.38	60.48
TAN	63.96	58.13	63.00	56.88	62.35
HAN	63.66	57.36	65.67	62.03	63.25
TRNN-Capsule	**67.15**	**58.55**	**65.71**	**65.29**	**64.63**

Table 4. Comparison with baselines on SemEval16 dataset.

Model	Atheism	Climate	Feminism	Hillary	Abortion	$MicF_{avg}$
NBOW	55.12	39.93	50.21	55.98	55.07	60.19
LSTM	58.18	40.05	49.06	61.84	51.03	63.21
LSTM$_E$	59.77	48.98	52.04	56.89	60.34	66.24
AT-biGRU	62.32	43.89	54.15	57.94	64.05	67.97
AS-biGRU-CNN	66.76	43.40	**58.83**	57.12	65.45	69.42
TAN	59.33	53.59	55.77	**65.38**	63.72	68.79
HAN	**70.53**	49.56	57.50	61.23	66.16	**69.79**
TRNN-Capsule	66.10	**60.03**	58.24	62.76	**67.04**	69.44

Though LSTM$_E$ outperforms LSTM and NBOW, it is inferior to TAN which is developed from LSTM$_E$ with attention mechanism, showing that attention mechanism is beneficial to extract important target-related information for stance classification. Further, HAN model considers more linguistic features, and does some work in advance to extract linguistic factors, such as sentiment, argument, dependency and so on, and then utilizes attention mechanism to combine context and linguistic to produce the final context representation. Because HAN uses external language knowledge, it is slightly better than TAN.

Our TRNN-Capsule model outperforms state-of-the-art competitors on both datasets. Compared with TAN, our model improves the performance about **2.28** and **0.65** on H&N14 and SemEval16. The main reason may be that building a capsule for each stance category is effective, and each capsule can identify important target-related words with stance tendencies reflecting capsules' category. Compared with HAN, our model improves the overall performance up to **1.38** on H&14 and shows very competitive performance on SemEval16. However, HAN needs external knowledge as input, e.g., sentimental words, argument sentence and dependency pair.

4.3 Analysis of TRNN-Capsule

In this section, we design and analyze several variants of our model. First, we create a No-Target model which ignores the target and only uses the context representation, In this case, we adopt only one bidirectional LSTM network to encode the context, and self-attention mechanism is utilized to combine the encoding results into one vector representation for final classification. Upon No-Target model, we then develop the second variant Target-Embedding-Attention (TEA). In TEA, we use the average of each word embedding in the target as a query to calculate the attention weight on each context word. Different from TEA, the third variant Target-LSTM-Attention (TLA) encodes both target and context with a bidirectional LSTM, respectively. Then, TLA utilizes the average of encoding results of target as a query to form the attention weights w.r.t. context. The difference between TLA and TRNN-Capsule is that TRNN-Capsule has a capsule layer to produce the classification. The performance of all the variants is shown in Table 5.

Table 5. Analysis of TRNN-Capsule Networks.

Model	Abortion	GayRights	Obama	Marijuana	overall
No-Target	61.00	56.58	59.97	55.85	60.45
TEA	61.34	56.93	60.98	55.94	60.59
TLA	62.78	57.15	61.15	56.49	61.29
TRNN-Capsule	67.15	58.55	65.71	65.29	64.63

We can see from Table 5 that No-Target model performs the worst. The observation indicates that target information plays an important role in stance classification and should not be neglected. Both the TEA and TLA models outperform NO-Target, and TLA is more promising. The observation suggests that encoding by LSTM is better than computing the average of word embeddings. Finally, we find that the proposed TRNN-Capsule delivers the best result, because TRNN-Capsule is more powerful to extract rich features and model the relationship between target and context.

4.4 Case Study

Table 6. Visualization of Attention Weights for Abortion

Favor	**Against**
Abortion should be legal, because abortions are legal, because if abortions should not be legal, then they would be illegal, but they are not illegal, which is why they should be legal.	I realize that adoption affects the parents lives as well, but would it not be better than killing it? Won't killing the fetus have a potential emotional side-effect on the parent? They would go through life knowing that they killed their own child.

Here we present a case study on H&N14 to show that our model can extract important target-related words with stance tendencies in a given context. Two examples are given in Table 6, where important words (identified by attention weights) are marked with red (for favor samples) or blue (for against samples). And the lighter the color is, the smaller weight it indicates. We can see from the table that our model indeed identifies the important words for stance classification. For instance, "legal" are selected for favor contexts and "killing" are selected for against contexts.

5 Conclusion

In this paper, we propose a novel target orientation RNN-Capsule network for stance classification (TRNN-Capsule). The TRNN-Capsule is composed of three layers, namely an embedding layer, an encoding layer and a capsule layer. In embedding layer, conventional word2vec representations are used. In the encoding layer, a bidirectional LSTM is adopted to form the representations of target and context respectively. Finally, capsule blocks with attention mechanism are designed and appended to produce the stance classification. Experimental results on two data sets demonstrate that the proposed TRNN-Capsule outperforms state-of-the-art competitors.

6 Acknowledgment

This work was supported by the National Key R&D Program of China, 2018YFB2 101100, 2018YFB2101101 and NSFC under Grant No. 61602132, and Guangdong Province Joint Project of Research and Industry under Grant No. 2017B090901022.

References

1. Wang, Y., Sun, A., Han, J., Liu, Y., Zhu, X.: Sentiment analysis by capsules. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp. 1165-1174. AAAI (2018).
2. Somasundaran, S., Wiebe, J.: 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, pp. 116-124. ACL (2010).
3. Mohammad, S.M., Kiritchenko, S., Sobhani, P., Zhu, X., Cherry, C.: SemEval-2016 task 6: detecting stance in tweets. In: Proceedings of 10th International Workshop on Semantic Evaluation, pp. 31-41. (2016).
4. Augenstein, I., Rockt aschel, T., Vlachos, A., Bontcheva, K.: Stance detection with bidirectional conditional encoding. In: Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing, pp. 876-885. ACL (2016).
5. Du, J., Xu, R., He, Y., Gui, L.: Stance classification with target-specific neural attention networks. In: International Joint Conferences on Artificial Intelligence, pp. 3988-3994. IJCAI (2017).
6. Sun, Q., Wang, Z., Zhu, Q., Zhou, G.: Stance Detection with Hierarchical Attention Network. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2399-2409. COLING (2018).
7. Walker, M. A., Anand, P., Abbott, R., Grant, R.: Stance classification using dialogic properties of persuasion. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 592-596. ACL (2012).
8. Sridhar, D., Getoor, L., Walker, M.: Collective stance classification of posts in online debate forums. In: Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media, pp. 109-117. (2014).
9. Burfoot, C., Bird, S., Baldwin, T.: Collective classification of congressional floor-debate transcripts. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 1506-1515. ACL (2011).
10. Hasan, K. S., Ng, V.: Why are you taking this stance? identifying and classifying reasons in ideological debates. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 751-762. EMNLP (2014).
11. Bach, S., Huang, B., London, B., Getoor, L.: Hinge-loss Markov random fields: Convex inference for structured prediction. In: arXiv preprint arXiv:1309.6813 (2013).
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: arXiv preprint arXiv:1301.3781 (2013).
13. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. In: arXiv preprint arXiv:1412.6980 (2014).
14. Hinton, G. E., Krizhevsky, A., Wang, S. D.: Transforming auto-encoders. In: International Conference on Artificial Neural Networks, pp. 44-51. Springer, Berlin, Heidelberg (2011).
15. Zhou, Y., Cristea, A. I., Shi, L.: Connecting targets to tweets: Semantic attention-based model for target-specific stance detection. In: International Conference on Web Information Systems Engineering, pp. 18-32. Springer, Cham (2017).
16. Zhang, B., Xu, X., Yang, M., Chen, X., Ye, Y.: Cross-Domain Sentiment Classification by Capsule Network With Semantic Rules. In: IEEE Access 6, pp. 58284-58294. (2018).