IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 9, 2020, accepted July 15, 2020, date of publication July 21, 2020, date of current version July 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010802

ADeCNN: An Improved Model for Aspect-Level Sentiment Analysis Based on Deformable CNN and Attention

JIE ZHOU[1], SIQI JIN[1], AND XINLI HUANG[1,2], (Member, IEEE)
[1]School of Computer Science and Technology, East China Normal University, Shanghai 200062, China
[2]Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200062, China

Corresponding author: Xinli Huang (xlhuang@cs.ecnu.edu.cn)

This work was supported in part by the National Key Research and Development Plan of China under Grant 2019YFB2012803, in part by the Key Project of Shanghai Science and Technology Innovation Action Plan under Grant 19DZ1100400 and Grant 18511103302, in part by the Key Program of Shanghai Artificial Intelligence Innovation Development Plan under Grant 2018-RGZN-02060, and in part by the Key Project of the ''Intelligence plus'' Advanced Research Fund of East China Normal University.

ABSTRACT Aspect-level sentiment analysis aims at identifying the sentiment polarity of target in the context. In most of the previous sentiment analysis models, there usually exists the problem of insufficient extraction capability of local features and long-distance dependency features. To solve the above problem, in this paper, we propose an improved model (called ADeCNN) for aspect-level sentiment analysis, by incorporating the attention mechanism into the deformable CNN model. In ADeCNN, we use deformable convolutional layers and bi-directional long short-term memory network (Bi-LSTM), combined with sentence-level attention, to extract sentiment features, and to break through the limitations of the model's long-distance dependency feature extraction capability. We then use a gated end-to-end memory network (GMemN2N) to integrate the target into the sentiment feature extraction process, so as to obtain sentiment features. And finally, we obtain the corresponding sentiment analysis results through the classifier. In addition, in order to solve the problem that the same words have large differences in the polarity of sentiments expressed in different targets, the model is also constructed with the ability to generate different attention weights based on target to assist sentiment analysis, with the aim of further enhancing the correlation between the target and the words in the sentence. We setup experiments to demonstrate the functionality effectiveness and performance gains of ADeCNN, based on the SemEval 2014 Task4 and SemEval 2017 Task4 datasets. Extensive experimental results show that ADeCNN outperforms its competitors, producing an arresting increase of the classification accuracy on all the three datasets of Laptop, Restaurant, and Twitter.

INDEX TERMS Aspect-level sentiment analysis, deformable CNN, attention mechanism, gated end-to-end memory networks.

I. INTRODUCTION

With the rapid development of information technology, the Internet has become an important way for people to obtain information and express their opinions, and a large amount of text data has accumulated on the Internet. The opinions and sentiments extracted from these text data by using technologies like big data analysis and natural language processing can help people make better decisions [1], [2].

Sentiment analysis is the process of analyzing the meaning of the text and the sentiment hidden in the text by using natural language processing technology to disassemble, model, extract, reason, and classify the text. Usually, the purpose of sentiment analysis is to find out the attitudes of bipolar views on certain targets in a sentence [2].

Traditional approaches to sentiment analysis, which consider the polarity of text, are too rough. Many companies prefer to know these sentiments of the product in different aspects. Therefore, a fine-grained aspect level sentiment analysis needs to be proposed, which analyzes different sentiments of different aspects of the same product [3]. For example, ''The headphone price is very high, but its quality is very good''. The sentiment of aspect ''quality'' is positive, while the sentiment of the given aspect ''price'' is negative.

The associate editor coordinating the review of this manuscript and approving it for publication was Biju Issac.

Aspect-level sentiment analysis is a fine-grained subtask in the field of sentiment analysis [3]. Currently, representative methods include feature-based SVM sentiment classifiers [4], neural network-based supervised learning methods. Tang *et al.* [5] proposed target-dependent long short-term memory networks (TD-LSTM), which is an improved LSTM model based on target dependence. Hu and Liang [6] and Ma *et al.* [7] use attention mechanism for Aspect-level analysis. Ma *et al.* [8] proposed a hierarchical attention model that explicitly attends to first the targets and then the whole sentence, extended the classic LSTM cell with components accounting for integration with external knowledge, and incorporated affective commonsense knowledge into a deep neural network.

Supervised learning usually requires a large amount of annotated data. Some studies use semi-supervised learning to perform text classification tasks. Since the variational autoencoder has been verified to extract the global features of text (such as sentiment, targets and styles) [9], it has also been successfully applied to semi-supervised text classification tasks [10]. Fu *et al.* [11] propose a Semi-supervised Aspect Level Sentiment Classification Model based on Variational Autoencoder (AL-SSVAE) for semi-supervised learning in the aspect-level sentiment classification. The AL-SSVAE model inputs a given aspect to an encoder a decoder based on a variational autoencoder (VAE), and it also has an aspect level sentiment classifier. It enables the attention mechanism to deal with different parts of a text when different aspects are taken as input as previous methods.

In aspect-level sentiment analysis, we use the word vector to represent a word. We usually use Continuous Bag-of-Words (CBOW) and Skip-Gram [12] to obtain the word vector of a word in the context. But the same word has different semantics in different contexts, a single vector of a word cannot distinguish different sentiment corresponding to different aspects. The sentiment polarity of words is usually sensitive to a given aspect. Therefore, researchers have proposed methods for word-word vector semantic disambiguation, such as the topic embedding model (TWE) [13], which assigns topics to each word in the text corpus, and learns topic embedding based on the words and their topics. In this way, the model can flexibly obtain context word embeddings, detect different topics in different contexts and capture different semantics.

In the field of aspect-level sentiment analysis, although a lot of efforts have been made, there still exist some outstanding problems to be solved, such as, how to improve the extraction capability of aspect-level sentiment analysis models, especially for accurate extraction of both local features and long-distance dependency features. In this paper, we focus on methodology, modeling and verification of aspect-level sentiment analysis. The work and main contributions are summarized as follows:

- Firstly, we propose an improved model (called ADeCNN) for aspect-level sentiment analysis, by incorporating sentence-level attention into the deformable CNN model, with the main purpose to solve the problem of insufficient extraction capability of local features and long-distance dependency features.
- Secondly, in ADeCNN, we introduce the deformable CNN that is mainly applied in the image processing domain into the field of sentiment analysis, and adapt deformable convolutional layers with Bi-LSTM, so that it is suitable for extracting sentiment features from one-dimensional text, thereby obtaining stronger feature extraction capability than the standard CNN.
- Third, to break through the limitations of the model's long-distance dependency feature extraction capability, we incorporate into and use the sentence-level attention to consider the overall structural characteristics of the sentences, so that the model has a better degree of attention to important information in the process of capturing the relationship between word pairs in the sentences. In addition, we also use the GMemN2N module to generate different attention weights based on the target, which is expected to help to solve the problem that the same text words have different sentiment polarities when describing different targets.
- And finally, to demonstrate the functionality effectiveness and performance gains of ADeCNN, we setup experiments and conduct comprehensive verifications and evaluations based on the SemEval 2014 Task4 and SemEval 2017 Task4 datasets. Extensive experimental results show that ADeCNN outperforms its competitors, producing an arresting increase of the classification accuracy on all the three datasets of Laptop, Restaurant, and Twitter. We also validate and analyze the impact of different structures, mechanisms and parameters on the performance of the model.

The rest of the paper is structured as follows. We first review the related work in Section II and then present the detailed approach in Section III. The experiments, dataset descriptions, results and analysis are reported in Section IV, followed by the conclusion and future work in Section V.

II. RELATED WORK

A. ASPECT-LEVEL SENTIMENT ANALYSIS

Aspect-level sentiment analysis is a fine-grained sentiment classification task. The goal of this task is to infer the sentiment polarity of the topics that appear in a given sentence.

Tang *et al.* [5] proposed target-dependent long short-term memory networks (TD-LSTM), which is an improved LSTM model based on target dependence. TD-LSTM can make full use of the semantic information of target by using target as the last input unit of LSTM, so as to better judge the sentiment polarity of the topic. Sentence splitting is performed in the LSTM sentence coding process. The target is used as the split point for forward and backward modeling.

TD-LSTM then concatenates the vectors of the forward and backward LSTMs and uses softmax [14] as the activation function to achieve sentiment classification. TD-LSTM obtained better sentiment analysis results than traditional

LSTMs. Further, Tang *et al.* [5] proposed target-connection long short-term memory networks (TC-LSTM). TC-LSTM concatenates the word vectors with the average value of all target vectors and input them into the model. This model fully combines the information on target and the sentence for feature representation.

Some researchers integrated the attention mechanism into the existing sentiment analysis models, and used target as an important factor to influence the attention weight to realize sentiment analysis of specific target. Hu and Liang [6] incorporated a deep attention mechanism into the LSTM model, and obtained aspect-level sentiment analysis result by focusing attention on word vectors related to the target. Ma *et al.* [7] proposed interactive attention networks for aspect-level sentiment classification (IAN). IAN interactively learns the attention in context and target, and respectively generate representations of the target and context. Therefore, the IAN model can represent the target and context well.

B. DEFORMABLE CONVOLUTIONAL NEURAL

Dai *et al.* [15] proposed deformable convolutional neural network (DeCNN) for better extraction of local features of pictures. DeCNN improves the ability of convolutional neural network (CNN) [16] to extract local features by improving key modules to make the scope of convolutional layer and pooling layer adaptive to transform shape. Specifically, DeCNN has improved two key modules on CNN. DeCNN adds a new convolution layer to the traditional convolution step to extract the geometric deformation offset of the convolution scope. After changing the shape, the scope is no longer a standard rectangle, but an adaptive shape of the same size that expands outward. The deformable RoI (Region of Interest) adds an additional pooling layer based on the standard pooling layer to extract and generate an offset matrix. DeCNN introduces offset matrices into the traditional convolutional layer and pooling layer respectively, so that the model can adaptively extract more valuable feature information, which greatly improves the local feature extraction ability of CNN. Therefore, this paper introduces DeCNN for local feature extraction of text sentiment, so that more valuable features can be input into subsequent models for sentiment analysis to obtain better results.

Most of the existing aspect-level sentiment analysis models used CNN to perform the first round of sentiment features extraction, and further extract the contextual features of the text by connecting the LSTM layer [17]–[19]. The data information used to represent pixels in the image features contains the color and contour of the image and other features information. There exists a correlation between the near data points. Therefore, the features of similar data points are important. By expanding the scope of convolution kernels and RoI pooling, DeCNN enables the model to adaptively locate data points that are closely related to the current data point in feature extraction, thereby more effectively capturing important information features to achieve better extraction effect.

C. ATTENTION MECHANISM

Attention [28] considers the overall structural features of the context, so that in the process of capturing the relationship for the word pairs in the sentence, the model will not be limited to a small range of information or blindly focus on all word pairs. Attention makes the model pay more attention to important information and improves the accuracy of the model. The attention mechanism is used to capture the association between different word pairs in context. It enables the model to pay more attention to words with strong sentiment features in sentiment analysis, thereby improving the effect of sentiment analysis.

D. GATED END-TO-END MEMORY NETWORKS

Traditional neural network models such as RNN, GRU, etc. will forget a lot of context features during the continuous conduction of neurons, and even the gradient disappears. The LSTM [20] is improved based on RNN [21], and the memory gate and forgotten gate to solve the memory problem of the model to a certain extent. However, the amount of memory in this way is not enough to store all the important features expressed in the text, so the researchers proposed the concept of the memory network model to store the important memory of the model [22]. Sukhbaatar *et al.* [23] proposed end-to-end memory networks(MemN2N). MemN2N processes the input discrete variables and stores them in memory to form a memory space, and outputs the corresponding prediction results according to the input keyword vector. MemN2N uses an attention mechanism to control the importance of storing information. Perez and Liu [24] proposed gated end-to-end memory networks (GMemN2N) which are based on the MemN2N network. GMemN2N combines the advantages of information interaction in residual network and highway network, and proposes a new method to deal with the disappearance of gradient. Ma *et al.* [25] added location, part-of-speech and other features to the dynamic memory network [26], and proposed the feature-based compositing memory network(FCMN) model to perform sentiment analysis on text based on specific words. Yang *et al.* [27] proposed the coattention-MemNet network, which considers the relationship between targets and other words with multiple angles to calculate the attention weight of the model.

III. ADeCNN

A. PROBLEM DEFINITION

Given a sentence $\boldsymbol{s} = [w_s^1, w_s^2, w_s^3, \ldots, w_s^n]$ consisting of n words and a target $\boldsymbol{t} = [w_t^1, w_t^2, w_t^3, \ldots, w_t^m]$ consisting of m targets occurring in sentence $\boldsymbol{s}$, the aspect-based sentiment analysis task aims at analyzing the sentiment of sentence $\boldsymbol{s}$ towards the target $\boldsymbol{t}$. For example, the sentiment polarity of sentence "The switchable graphic card is pretty sweet when you want gaming on the laptop." towards "switchable graphic card" is positive, while the polarity towards"gaming" is negative.

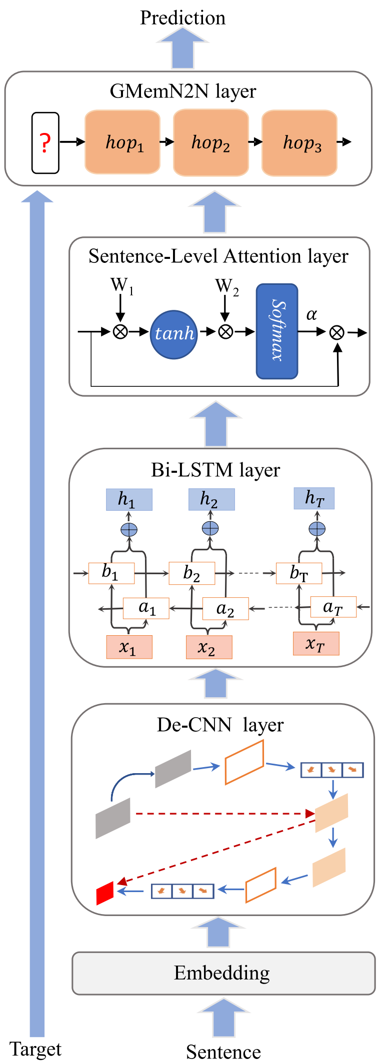


FIGURE 1. The framework of the ADeCNN.

B. MODEL STRUCTURE

The framework of the ADeCNN, as shown in Fig 1, is composed of the following parts.

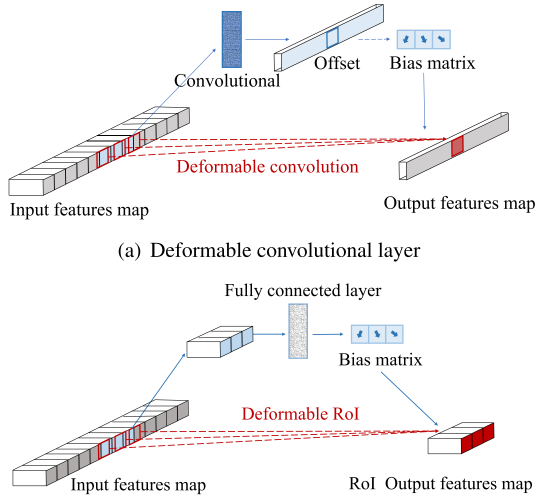
1) EMBEDDING LAYER

It uses the pre-trained word embedding matrix to convert the words in the context into word vectors.

2) DEFORMABLE CNN LAYER

The word vector generated by the embedding layer is input to the deformable CNN layer for local feature extraction. The deformable CNN layer is composed of a deformable convolution layer and a deformable ROI layer. The process is shown in Fig 2.

- Deformable convolutional layer: The deformable convolutional layer adds a new convolutional layer to extract the geometrical deformation offset of the convolution scope. After changing the shape, the scope is no longer a standard rectangle, but an adaptive shape of the same size that expands outward (see Fig 3). The migration



(a) Deformable convolutional layer

(b) Deformable ROI layer

FIGURE 2. Process of deformable convolutional layer(a) and deformable ROI layer(b).

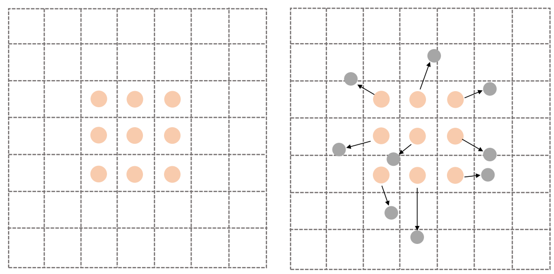


FIGURE 3. 3 × 3 convolution sampling position. (right)deformable convolution, (left)convolution.

matrix defines the size and extent of the scope $\mathcal{R}$ of the convolution kernel. The closer the offset sampling point is to the *key information point*, the more valuable the extracted features. The deformable convolution process can be expressed as follows:

$$y(\boldsymbol{p_0}) = \sum_{\boldsymbol{p_n} \in \mathcal{R}} \boldsymbol{w}(\boldsymbol{p_n})\boldsymbol{x}(\boldsymbol{p_0} + \boldsymbol{p_n} + \Delta \boldsymbol{p_n}), \tag{1}$$

where, $\boldsymbol{p_0}$ represents the position of the point on the input feature maps, $\boldsymbol{p_n}$ is the position of the point in the scope $\mathcal{R}$, and $\Delta \boldsymbol{p_n}$ is the offset value of $\boldsymbol{p_n}$. $y(\boldsymbol{p_0})$ is the output feature of each input $\boldsymbol{p_0}$ after going through the offset matrix. $\boldsymbol{w}(.)$ is the weight of the corresponding sampling position. $\boldsymbol{x}(.)$ is a discrete function. The offset value $\boldsymbol{x}(\boldsymbol{p_0}+\boldsymbol{p_n}+\Delta \boldsymbol{p_n})$ is not the actual point. Therefore, we use the bilinear difference method proposed by Jaderberg *et al.* [29] in the one-dimensional feature maps to calculate the value of the discrete position of the feature. $\boldsymbol{x}(.)$ is defined as follows:

$$\boldsymbol{x}(\boldsymbol{p}) = \sum_{\boldsymbol{q}} \max(\boldsymbol{0}, \boldsymbol{1} - |\boldsymbol{p} - \boldsymbol{q}|)\boldsymbol{x}(\boldsymbol{q}) \tag{2}$$

$$\boldsymbol{p} = \boldsymbol{p_0} + \boldsymbol{p_n} + \Delta \boldsymbol{p_n}, \tag{3}$$

where $\boldsymbol{q}$ is all integer positions on the input feature maps.

- Deformable ROI: The deformable ROI (Region of Interest) pooling layer adds an additional pooling layer based on the standard pooling layer to extract and generate an offset matrix, and then the offset matrix and the standard pooling layer work together to get a deformable ROI feature information after pooling. The deformable RoI pooling layer can be expressed as follows:

$$y(p_0) = \sum_{p_n \in \mathcal{R}} x(p_0 + p_n + \Delta p_n)/n_p, \tag{4}$$

where n_p represents the number of points in the scope. The process of obtaining Δp_n is as follows. Firstly, RoI pooling generates pooled feature maps. From the maps, a fc layer generates the normalized offsets $\Delta\widehat{p_n}$. However, the size of $\Delta\widehat{p_n}$ is not the same as the size of each scope (ω, h), so we use the following formula to calculate Δp_n:

$$\Delta p_n = \gamma \Delta\widehat{p_n} \circ (\omega, h), \tag{5}$$

where γ is the gain scalar, the default $\gamma = 0.1$; $\circ$ represents the product of elements in the matrix. The input feature maps output the feature maps under the action of the generated offset matrix and the pooling layer to realize feature extraction and dimensionality reduction.

3) BI-LSTM LAYER

The feature vectors X obtained from the sentence by the deformable CNN layer are the input of the Bi-LSTM layer [20], [21], [30]. In the Bi-LSTM layer, X is fed into the forward and backward LSTM to extract the forward hidden vectors and backward hidden vectors. Then the forward hidden vector $a_t \in R^d$ and the backward hidden vector $b_t \in R^d$ are concatenated to obtain the hidden vector $h_t \in R^{2d}$ of the Bi-LSTM layer.

For the forward layer, hidden state a_t is given by the following equations:

$$i_t^{(a)} = \sigma(U_i^{(a)} x_t + W_i^{(a)} a_{t-1} + b_i^{(a)}), \tag{6}$$
$$f_t^{(a)} = \sigma(U_f^{(a)} x_t + W_f^{(a)} a_{t-1} + b_f^{(a)}), \tag{7}$$
$$o_t^{(a)} = \sigma(U_o^{(a)} x_t + W_o^{(a)} a_{t-1} + b_o^{(a)}), \tag{8}$$
$$u_t^{(a)} = \tanh(U_u^{(a)} x_t + W_u^{(a)} a_{t-1} + b_u^{(a)}), \tag{9}$$
$$C_t^{(a)} = i_t^{(a)} \odot u_t^{(a)} + f_t^{(a)} \odot C_{t-1}^{(a)}, \tag{10}$$
$$a_t = o_t^{(a)} \odot \tanh(C_t^{(a)}). \tag{11}$$

For the backward layer, hidden state b_t is calculated by the following equations:

$$i_t^{(b)} = \sigma(U_i^{(b)} x_t + W_i^{(b)} b_{tC1} + b_i^{(b)}), \tag{12}$$
$$f_t^{(b)} = \sigma(U_f^{(b)} x_t + W_f^{(b)} b_{tC1} + b_f^{(b)}), \tag{13}$$
$$o_t^{(b)} = \sigma(U_o^{(b)} x_t + W_o^{(b)} b_{tC1} + b_o^{(b)}), \tag{14}$$
$$u_t^{(b)} = \tanh(U_u^{(b)} x_t + W_u^{(b)} b_{tC1} + b_u^{(b)}), \tag{15}$$

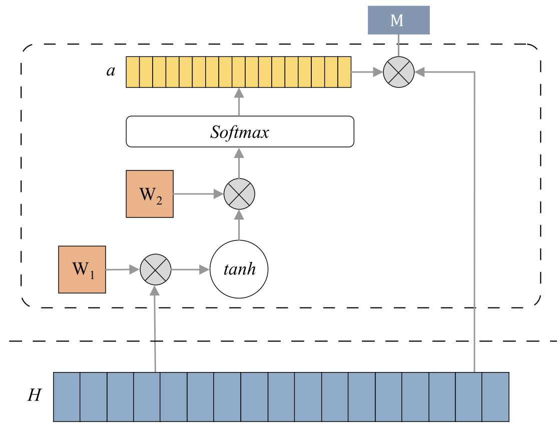


FIGURE 4. Sentence-level attention layer weight calculation process.

$$C_t^{(b)} = i_t^{(b)} \odot u_t^{(b)} + f_t^{(b)} \odot C_{tC1}^{(b)}, \tag{16}$$
$$b_t = o_t^{(b)} \odot \tanh(C_t^{(b)}). \tag{17}$$

The hidden vectors are calculated by the following equations:

$$h_t = a_t \oplus b_t. \tag{18}$$

4) SENTENCE-LEVEL ATTENTION LAYER

Further, we add a sentence-level attention layer to encode the hidden layer vectors. The hidden layer weight calculation process is shown in Fig 4, the weight α calculation formula is as follows:

$$\alpha = softmax(W_2 \tanh(W_1 H^T)), \tag{19}$$

where, $H = [h_1, h_2, \ldots, h_n]$, the weight matrix $W_1 \in R^{n \times 2d}$, the weight matrix $W_2 \in R^{1 \times 2d}$.

The weight α represents the model's attention to different words in the context. So, we can get the semantic encoding $M \in R^{n \times 2d}$ according to the following formula:

$$M = \alpha \times H. \tag{20}$$

5) GMemN2N LAYER

Then, we use the GMemN2N layer to capture the internal correlation between the semantic encoding and the subject word, and obtain the polarity of the target. The internal structure of the GMemN2N layer is shown in Fig 5. The inputs of the GMemN2N layer are semantic encoding M and topic words vector $u^1 \in R^{d \times 1}$.

GMemN2N is composed of t hops, as shown in Fig. 5. For the k-th hop, the process is defined as follows:

$$p^k = \sigma(ReLU(W_1^k [m_k; u^k]^T) + b_1^k), \tag{21}$$
$$o^k = \sum_{i=1}^{t} p_i^k \times m_i, \tag{22}$$
$$T^k(u^k) = \sigma(W_2^k u^k + b_2{}^k), \tag{23}$$
$$u^{k+1} = o^k \odot T^k(u^k) + u^k \odot (1 - T^k(u^k)), \tag{24}$$

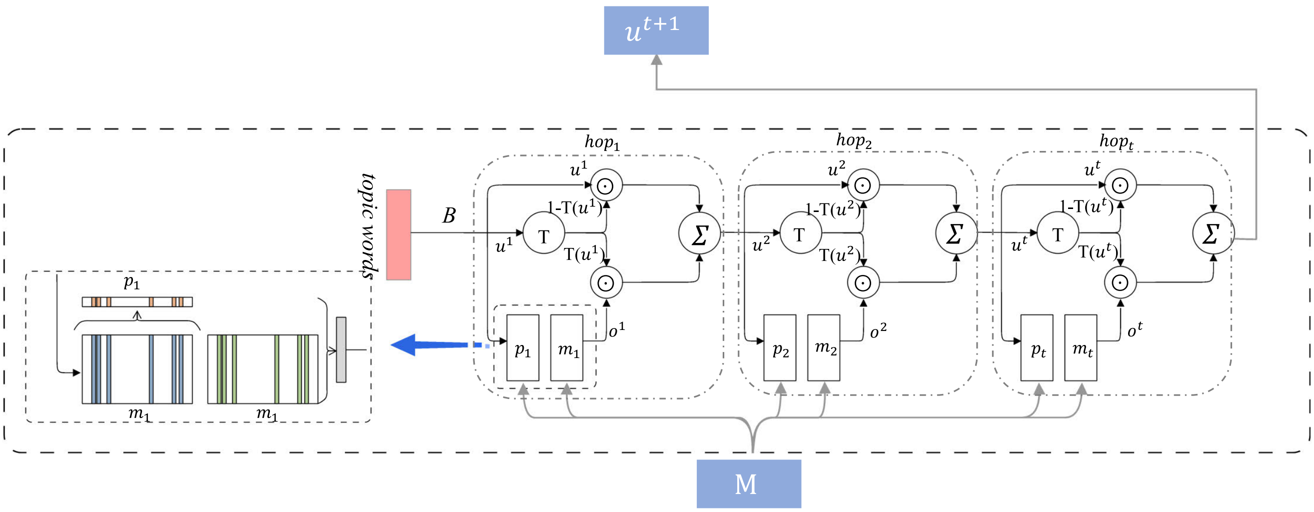


FIGURE 5. Internal structure diagram of GMemN2N layer.

where, $\boldsymbol{m}_k \in \boldsymbol{R}^{1\times 2d}$ represents the k-th vector in the semantic encoding $\boldsymbol{M}$. After concatenating $\boldsymbol{m}_k$ and $\boldsymbol{u}^k$, a non-linear transformation is performed, and then normalized using the softmax function to obtain the importance weight $\boldsymbol{p}_k$. $\boldsymbol{m}_k$ times $\boldsymbol{p}_k$ to obtain the output vector $\boldsymbol{o}^k$, $\boldsymbol{W}_1^k \in \boldsymbol{R}^{n\times 3d}$, $\boldsymbol{b}_1^k \in \boldsymbol{R}^{n\times 1}$, $\boldsymbol{W}_2^k \in \boldsymbol{R}^{2d\times 1}$, $\boldsymbol{b}_2^k \in \boldsymbol{R}^{2d\times d}$.

Finally, we obtain the sentiment classification of the target by the following formula:

$$\widehat{\boldsymbol{u}} = ReLU(\boldsymbol{W}_{reu}\boldsymbol{u}^{t+1}) + \boldsymbol{b}_{reu}, \tag{25}$$

$$\widehat{\boldsymbol{y}} = softmax(\boldsymbol{W}^{t+1}(\widehat{\boldsymbol{u}} + \boldsymbol{o}^t)), \tag{26}$$

where, $\boldsymbol{W}_{reu} \in \boldsymbol{R}^{n\times d}$, $\boldsymbol{b}_{reu} \in \boldsymbol{R}^{n\times 1}$, $\boldsymbol{W}^{t\mathcal{C}1} \in \boldsymbol{R}^{3\times n}$

IV. EXPERIMENT

A. EXPERIMENTAL ENVIRONMENT AND PARAMETER SETTINGS

The experimental environment is a 64-bit macOS system, Intel 2.6GHz 8-core i7 CPU, 16GB 2400MHz DDR4 memory, Radeon Pro 560X 4GB GPU. The compilation environment is Python 3.6, and Keras based on TensorFlow is selected as the deep learning framework.

In the experiment, the word vector is set to 300 dimensions, and the model uses a uniform distribution of (−0.05, 0.05) to randomly generate initial values of all parameters. The learning rate of the training process is set to 0.001. The Adam optimizer and classification cross entropy are used as the loss function of the experiment, and the decay rate is 0.95. This model uses 5-fold cross-validation, the size of the training batch size is 6, and the epoch is 100. We use dropout in LSTM layer and GMemN2N layer and L2 regularization to prevent overfitting. The dropout rate set to 0.75 and the L2 coefficient is set to 0.0001.

The deformable convolution layer uses 2 deformable convolutions, followed by a deformable pooling. The convolution kernel is a one-dimensional convolution kernel with a window size of 3 and a step size of 1. Deformable pooling layer has a window size of 2 and a step size of 1.

B. DATASETS AND INDICATOR

The SemEval dataset is a sentiment analysis dataset based on Twitter, which was built in 2013. Initially, the dataset contains only a simple text and the sentiment analysis results of the text (positive, negative, neutral). In 2015, with the refinement of sentiment analysis research, the dataset added a new topic-based sentiment analysis, through input text and topic, inferred the results of sentiment analysis based on the target. The SemEval dataset has now become the mainstream dataset in sentiment analysis. This experiment uses the Restaurant and Laptop datasets in SemEval 2014 Task4 and the Twitter dataset in SemEval 2017 Task4 to conduct the model training and testing process.

In this paper, *Accuracy* is used to evaluate the sentiment analysis effect of ADeCNN model and other models. The accuracy rate represents the percentage of the total number of correct results predicted by the model. The higher the value of accuracy, the better the model sentiment analysis. The calculation formula of the accuracy rate is as follows:

$$\boldsymbol{accuracy} = \frac{\boldsymbol{n_{correct}}}{\boldsymbol{n_{total}}}, \tag{27}$$

where $\boldsymbol{n_{correct}}$ represents the number of correct results predicted by the model, and $\boldsymbol{n_{total}}$ represents the total number of model prediction results.

All three datasets contain positive, negative and neutral sentiment polarities, and each target tag in the dataset is extracted from the corresponding text. The data statistics of the training and test sets used in the experiments in this paper are shown in Table 1.

TABLE 1. Train and test set statistics.

DataSet statistics	restaurant		laptop		Twitter	
	train	test	train	test	train	test
positive	2170	657	990	341	4897	2458
neutral	500	94	464	165	3127	346
negative	839	220	870	128	3997	3722

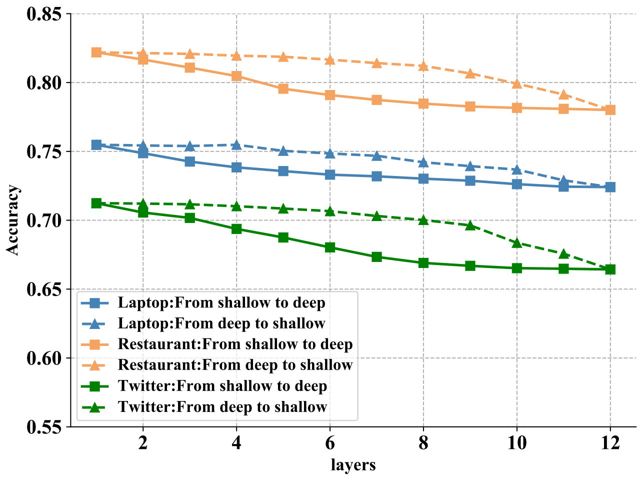


FIGURE 6. The effect of the cumulative number of replacement layers on the accuracy rate. The solid line shows that the deformable layer is replaced with the standard layer in order from the shallow layer to the deep layer. The dotted line shows that the deformable layer is replaced with the standard layer in order from the deep layer to the shallow layer.

C. COMPARISON WITH BASE MODELS

1) BASE MODELS

We compare the ADeCNN with the baseline models to show the effectiveness of the deformable CNN layer. Baseline methods include:

- Bi-LSTM: It takes each sentence sequence as input. The hidden unit state of the last layer after concatenating is sent to the softmax classification layer, and finally it outputs sentiment classification results.
- TD-LSTM: It uses the target as a boundary, and divides the entire sentence into two paragraphs, which are processed using forward and backward LSTM. In this way, the topic words are located at the beginning or ending of the LSTM to extract more sentimental information.
- TC-LSTM: It is an improvement of TD-LSTM, concatenates the average of all word vectors and target vectors in a sentence, and inputs the word vectors into the TD-LSTM model. Compared with TD-LSTM, in addition to considering the structural information of the topic words, it also considers the content information of the topic words and integrates the target with each word in the text to obtain more excellent sentiment analysis results for the target.
- IAN: It uses the attention mechanism to interact the target and context and then gets the context and target codes to obtain more accurate sentiment analysis results.

2) EXPERIMENT RESULTS

In this experiment, for the sentiment classification task, the above five models are trained and tested on the datasets of three different fields, and the classification accuracy of the models on the test set is compared.

TABLE 2. Model accuracy comparison results.

Model	Laptop	Restaurant	Twitter
Bi-LSTM	0.6708	0.7572	0.6216
TD-LSTM	0.7023	0.7637	0.6362
TC-LSTM	0.7032	0.7709	0.6729
IAN	0.7349	0.7968	0.6993
ADeCNN	**0.7726**	**0.8403**	**0.7294**

As shown in Table 2, TD-LSTM and TC-LSTM further improve the accuracy rate by considering the content information of target. Compared with the above models, IAN adds an attention mechanism between topic words and text, which significantly improves the classification effect. Compared to the IAN, the accuracy of ADeCNN on the three datasets of Laptop, Restaurant, and Twitter increases by 3.77%, 4.35%, and 2.87%, respectively.

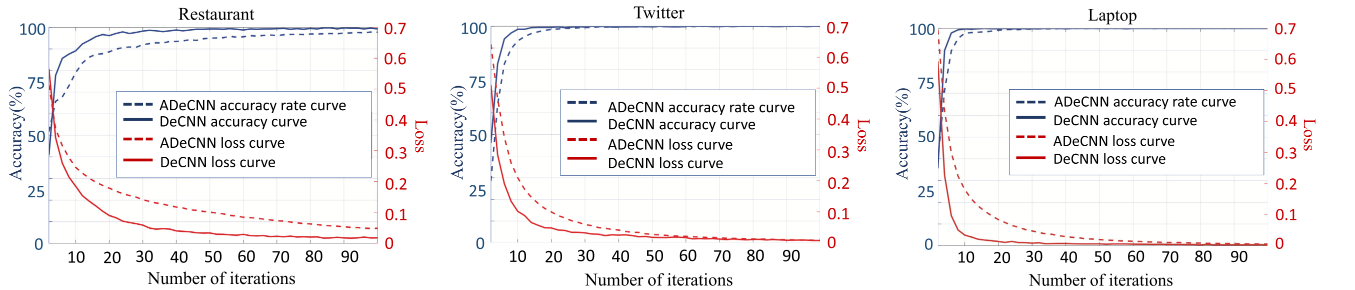
In addition, as shown in Fig 7, the accuracy rate of the ADeCNN model in the training process reached 100% on the verification set, which indicates that ADeCNN has a high probability of being overfitting, which reminds us that we can try to reduce the model parameters to further improve accuracy rate. We regard it as our future work.

D. EFFECTIVENESS OF DEFORMABLE CNN LAYER

To explore the effectiveness of the two modules of the deformable CNN layer (the deformable convolution layer and the deformable RoI layer), we conduct a comparative experiment analysis on the models of each layer with or without *deformable* elements. The core idea is to remove the *deformable* elements in order from high to low and replace them with standard CNN convolutional layers and pooling layers, and then observe the changes in the accuracy of the model on each dataset and infer the *deformable* results impact.

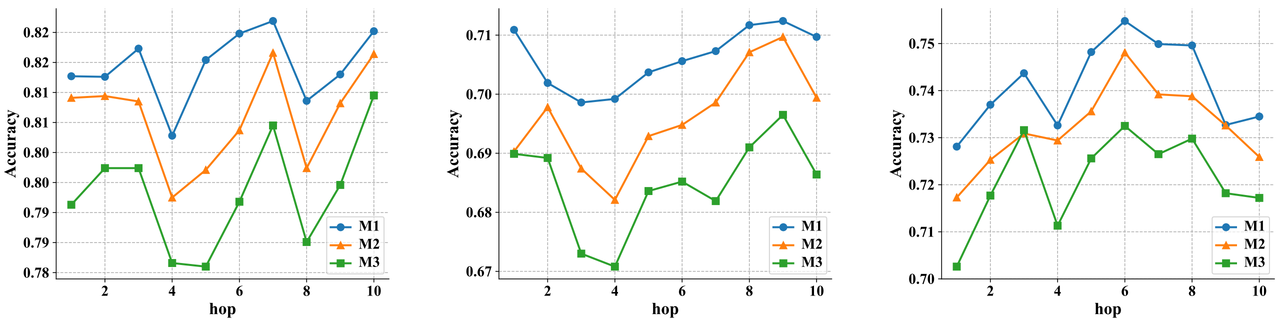
We use deformable CNN layer and Bi-LSTM layer to build the basic model. The deformable CNN layer includes 8 deformable convolutional layers and 3 deformable RoI layers, and the sequence numbers are 1 to 11 from low to high, respectively. Every 3 deformable convolutional layers are followed by a deformable ROI layer. Because the deep model extracts general features in the shallow layer and extracts more abstract advanced features in the deep layer, this experiment replaces the deformable layer with the standard layer from shallow layer to deep layer and from deep layer to shallow layer to explore whether the deformable elements are related to the advanced level of the extracted features.

As shown in Fig 6, as the number of cumulative replacement layers increases, the accuracy rate tends to decrease, which proves that the deformable layer has a stronger feature extraction capability than the ordinary CNN layer, and



(a) Restaurant (b) Twitter (c) Laptop

FIGURE 7. Accuracy and Loss change curve of ADeCNN model and DeCNN model during training.



(a) Restaurant (b) Twitter (c) Laptop

FIGURE 8. Influence of end-to-end memory network hop number and attention mechanism on accuracy. M1 represents the ADeCNN model with sentence-level attention. M2 represents the ADeCNN model with general attention. M3 represents the ADeCNN without the attention layer.

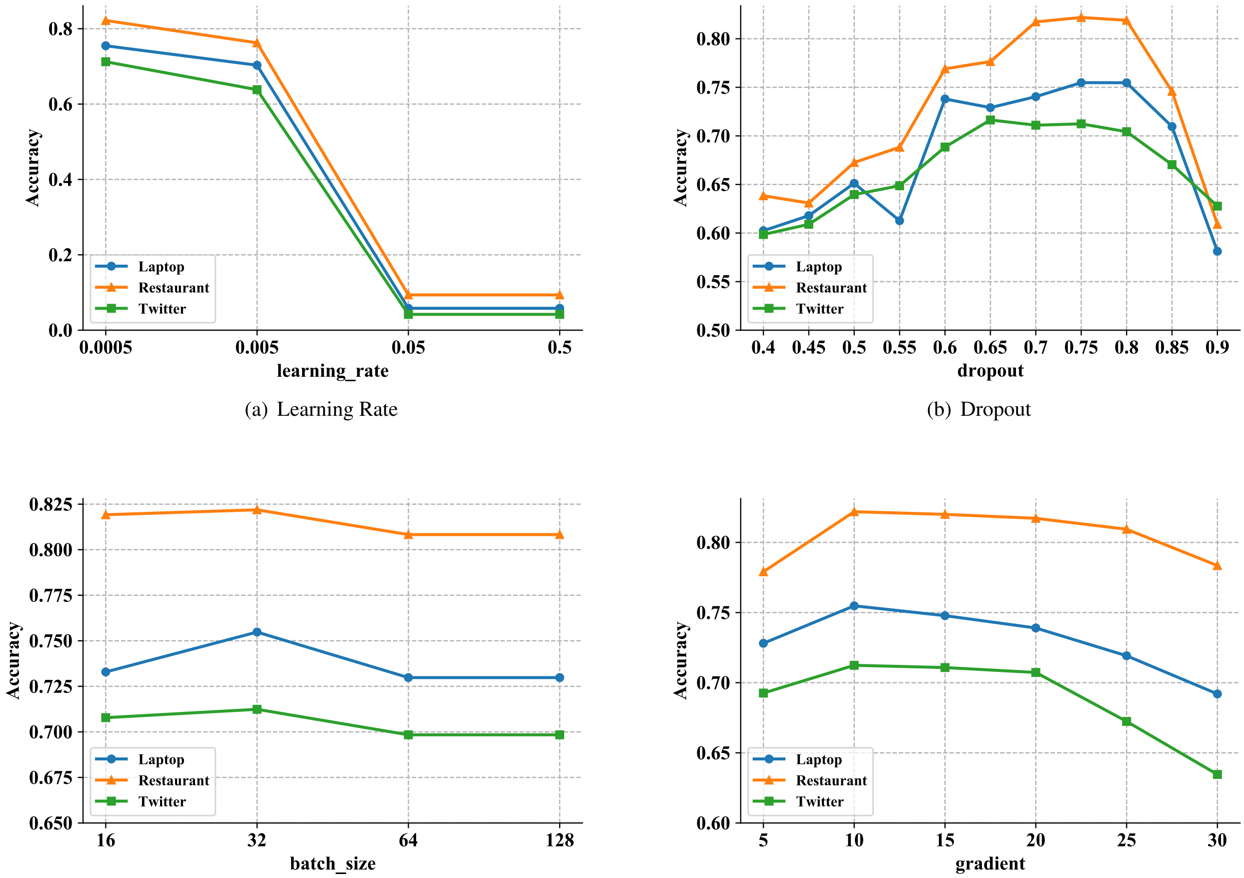
can learn more representative features. More specifically, the declining speed of the blue fold line is basically high and then low, and the declining speed of the red fold line is basically low and then high. This phenomenon shows that the deformable layer improves the model's ability to extract shallow features more significantly. This may be because the shallow layer usually contains a lot of redundant information, and the filtering of this information often puts forward higher requirements on the local feature extraction ability of the model. If the local feature capability of the model is stronger, it will be more obvious in the hidden layer.

E. INFLUENCE OF SENTENCE-LEVEL ATTENTION

We explore the contribution of the sentence-level attention in the ADeCNN to the sentiment classification accuracy of each dataset. We compare the performance of sentence-level attention and general attention. We label the ADeCNN model with the sentence-level attention as M1, the ADeCNN model with the general attention layer instead of the sentence-level attention layer as M2, and the ADeCNN without the attention layer as M3. We define general attention as multiplying the feature vector with a normalized weight. We adjust the hop of the end-to-end network from 1 to 10 in order to classify the sentiment on each dataset, and compare the accuracy. As shown in Fig 8.

First, the three models in the Laptop dataset all show a trend of increasing first and then decreasing with the increase of hop. The three models also reach the highest accuracy rate at 6 hops. Among them, the accuracy of M1 is always better than the accuracy of M2, and the two models achieve similar results at 9 hops. Except for the accuracy of 3 hops, M2 is slightly lower than M3, but it is an overall advantage. Second, in the Restaurant dataset, the three models show a large fluctuation trend overall, M1 and M2 reach the optimal result at the 7th hop, and M3 reaches the highest accuracy rate at the 10th hop. The overall accuracy ranking of the three models is still M1 greater than M2, and M2 greater than M3. In the Twitter dataset, the relative change of M1 is relatively stable, and the other two models fluctuate significantly. The numerical accuracy is still M1 greater than M2 greater than M3.

Overall, the effect of the sentence-level attention layer is better than the attention layer, and it can perform well in the topic-specific sentiment classification task. The effect of the model with attention layer is better than that of the model without attention layer. The experimental results verify the effectiveness of the sentence-level attention model in this task. It is probably since that the attention mechanism can make the model dynamically adjust the weights between different input features and lay more weights on more important information. The sentence-level attention further improves the attention mechanism, reduces the dependence on external information, and focuses more on capturing the correlation between the internal information of the sentence.



(a) Learning Rate (b) Dropout (c) Batch Size (d) Gradient

FIGURE 9. Influence of hyperparameters on accuracy.

F. IMPACT OF HYPERPARAMETERS

Finally, we investigate the effect of some hyperparameters used in the ADeCNN model on accuracy. By sequentially adjusting the learning rate, batch size, dropout value and gradient clipping coefficient within a certain range, the accuracy of the classification results is observed.

As shown in Fig 9, the accuracy rate gradually decreases as the learning rate increases. Increasing the learning rate causes the model to fall into the local optimal solution. Adjusting the learning rate appropriately can help the model achieve the global optimum. The choice of learning rate is a trade-off between the accuracy of results and the speed of program convergence. The accuracy of the model increases first and then decreases as the batch size increases, however the overall change is not obvious. The accuracy of the model increases first and then decreases with the increase of dropout from 0 to 1. The performance is the best in the range of about 0.6 to 0.8.

The accuracy rate is highest at a value of about 0.75. When the dropout value is very small, the structure of the model does not change significantly during the training process, and it plays a small role. When the dropout value is too large, the large changes in the model structure in each training cycle play the opposite role, making the model difficult to converge. Similarly, the accuracy of the model increases first and then decreases as the gradient clipping coefficient increases.

V. CONCLUSION

In this paper, we investigate the current problems and challenges facing the field of sentiment analysis, and propose an improved model (ADeCNN) for aspect-level sentiment analysis, by incorporating sentence-level attention into the deformable CNN model, with the main purpose to enhance the existing models' extraction capability of local features and long-distance dependency features. Considering that the same text words have different sentiment polarities when describing different targets, we propose, in the ADeCNN model, to use the GMemN2N module to generate different attention weights based on the target. We demonstrate the functionality effectiveness and performance gains of ADeCNN through comprehensive experiments and evaluations based on the SemEval 2014 Task4 and SemEval 2017 Task4 datasets, and the results show that ADeCNN outperforms its competitors, producing an arresting increase of the classification accuracy on all the three datasets of Laptop, Restaurant, and Twitter. We also discuss the impact of different structures, mechanisms and parameters on the performance of the model.

Text sentiment analysis has always been an important and hot research area for many years. However, quite a few online review texts do not express their sentiment polarities straightforward, but ironically. Ironic comments convey negative polarity using positive expressions. For example, "What a great phone! It stopped working in 2 days". The existing models, to our best knowledge, cannot effectively achieve ironic text sentiment analysis. It is quite a big challenge to explore models able to identify ironic sentences and accurately identify their sentiment polarities.

REFERENCES

[1] B. Liu, "Sentiment analysis and opinion mining," *Synth. Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[2] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.

[3] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 27–35.

[4] J. Wagner, P. Arora, S. Cortes, U. Barman, D. Bogdanova, J. Foster, and L. Tounsi, "DCU: Aspect-based polarity classification for SemEval task 4," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 223–229.

[5] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2016, pp. 3298–3307.

[6] H. Chaoju and L. Ning, "Deeper attention-based LSTM for aspect sentiment analysis," (in Chinese), *Appl. Res. Comput.*, vol. 36, no. 4, pp. 121–125, 2019.

[7] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2017, pp. 4068–4074.

[8] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 5876–5883.

[9] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, 2016, pp. 10–21.

[10] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *Proc. AAAI*, 2016, pp. 3358–3364.

[11] X. Fu, Y. Wei, F. Xu, T. Wang, Y. Lu, J. Li, and J. Z. Huang, "Semi-supervised aspect-level sentiment classification model based on variational autoencoder," *Knowl.-Based Syst.*, vol. 171, pp. 81–92, May 2019.

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR*, 2013, pp. 1–12.

[13] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2418–2424.

[14] J. Goodman, "Classes for fast maximum entropy training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, May 2001, pp. 561–564.

[15] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[16] Y. Le Cun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1995, pp. 255–258.

[17] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017.

[18] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Double embeddings and CNN-based sequence labeling for aspect extraction," 2018, *arXiv:1805.04601*. [Online]. Available: http://arxiv.org/abs/1805.04601

[19] W. Xue and T. Li, "Aspect based sentiment analysis with gated convolutional networks," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 2514–2523.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[22] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[23] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," 2015, *arXiv:1503.08895*. [Online]. Available: http://arxiv.org/abs/1503.08895

[24] F. Liu and J. Perez, "Gated end-to-end memory networks," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1–10.

[25] R. Ma, K. Wang, T. Qiu, A. K. Sangaiah, D. Lin, and H. B. Liaqat, "Feature-based compositing memory networks for aspect-based sentiment classification in social Internet of Things," *Future Gener. Comput. Syst.*, vol. 92, pp. 879–888, Mar. 2019.

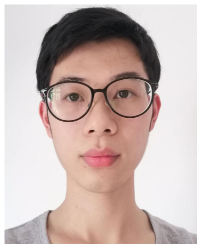
[26] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher "Ask me anything: Dynamic memory networks for natural language processing," in *Proc. 33rd Int. Nat. Conf. Int. Conf. Mach. Learn. (ICML)*, vol. 48, 2016, pp. 1378–1387.

[27] C. Yang, H. Zhang, B. Jiang, and K. Li, "Aspect-based sentiment analysis with alternating coattention networks," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 463–478, May 2019.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[29] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 2017–2025.

[30] J. Zhou, Y. Lu, H.-N. Dai, H. Wang, and H. Xiao, "Sentiment analysis of Chinese microblog based on stacked bidirectional LSTM," *IEEE Access*, vol. 7, pp. 38856–38866, 2019.



JIE ZHOU received the B.S. degree from the Department of Computer Science and Technology, Xiangtan University, Hunan, China, in 2019. He is currently pursuing the master's degree with the School of Computer Science and Technology, East China Normal University, Shanghai, China. His current research interests are in the areas of deep learning and knowledge graph reasoning.

SIQI JIN received the B.S. degree from the Department of Computer Science and Technology, East China Normal University, Shanghai, China, in 2017, where she is currently pursuing the master's degree with the School of Computer Technology. Her current research interest are in the areas of natural language processing.

XINLI HUANG (Member, IEEE) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2007. He is currently an Associate Professor with the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His research interests include the areas of internetworking, software-defined networking, cloud computing, future networks, and network security.