Applied Intelligence (2021) 51:4408–4419
https://doi.org/10.1007/s10489-020-02095-3

Aspect-gated graph convolutional networks for aspect-based sentiment analysis

Qiang Lu[1] · Zhenfang Zhu[1] · Guangyuan Zhang[1] · Shiyong Kang[2] · Peiyu Liu[3]

Accepted: 24 November 2020 / Published online: 4 January 2021

Abstract

Aspect-based sentiment analysis aims to predict the sentiment polarity of each specific aspect term in a given sentence. However, the previous models ignore syntactical constraints and long-range sentiment dependencies and mistakenly identify irrelevant contextual words as clues for judging aspect sentiment. In addition, these models usually use aspect-independent encoders to encode sentences, which can lead to a lack of aspect information. In this paper, we propose an aspect-gated graph convolutional network (AGGCN), that includes a special aspect gate designed to guide the encoding of aspect-specific information from the outset and construct a graph convolution network on the sentence dependency tree to make full use of the syntactical information and sentiment dependencies. The experimental results on multiple SemEval datasets demonstrate the effectiveness of the proposed approach, and our model outperforms the strong baseline models.

Keywords Aspect-based sentiment analysis · Graph convolutional networks · Aspect gate · Aspect-specific

1 Introduction

Aspect-based sentiment analysis [6, 27, 28, 48] is a fine-grained task in sentiment analysis [2, 38, 41, 43] whose goal is to predict the sentiment polarity (e.g., positive, neutral or negative) toward each specific aspect term in a given sentence.

There are two subtasks in aspect-based sentiment analysis, including aspect-category sentiment analysis and aspect-term sentiment analysis [48]. An example in Fig. 1 presents a sample sentence. The aspect-category sentiment analysis implicitly describes the general entity category. For instance, in the sentence “The sushi is delicious but the waiter is very rude”, “sushi” describes the aspect category “food”, and “waiter” describes the aspect category “service”. The user expresses both positive and negative sentiments toward two aspect categories “food” and “service”, respectively. The aspect-term sentiment analysis characterizes specific entities that occur explicitly in a sentence. For the example sentence, the aspect terms are “sushi” and “waiter”, and the user expresses positive and negative sentiments toward them, respectively. In terms of the aspect granularity, the aspect category is coarse-grained, while the aspect term is fine-grained.

Earlier studies introduced recurrent neural network (RNN) [4] models into aspect-based sentiment analysis due to its ability to flexibly capture the semantic relations between an aspect and its context words. However, not all the information in the sequence is important; therefore, an attention mechanism [47] was introduced into the RNN model to cause the model to pay more attention to the more important parts of the sequence. Gu et al. [11] proposed a position-aware bidirectional attention network (PBAN) based on the Bi-GRU model. PBAN not only concentrates on the positional information of aspect terms but also mutually models the relation between an aspect term and the sentence by employing a bidirectional attention mechanism. With the development of word embedding technology, the convolutional neural network (CNN) [14] has been widely

✉ Zhenfang Zhu
zhuzf@sdjtu.edu.cn

✉ Guangyuan Zhang
xdzhanggy@163.com

Qiang Lu
luqiang@stu.sdjtu.edu.cn

[1] School of Information Science and Electrical Engineering, Shandong Jiao Tong University, Jinan 250357, China

[2] Chinese Lexicography Research Center, Lu Dong University, Yantai 264025, China

[3] School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China

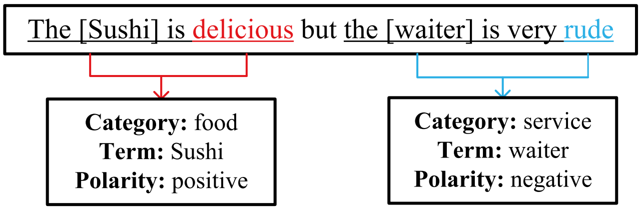

Fig. 1 An example sentence with two instances of aspect-based sentiment analysis. The user expresses both positive and negative sentiments toward two aspect categories “food” and “service”, and expresses positive and negative sentiments toward two aspect terms “sushi” and “waiter”

applied to aspect-based sentiment analysis. On the one hand, a CNN can utilize word embeddings to map a sentence into a lower-dimensional semantic representation while also maintaining the sequence information of the words. On the other hand, a CNN can extract a local text representation. Xue et al. [39] proposed a model based on convolutional neural networks with gating mechanisms. First, the novel gated tanh-ReLU units selectively outputs the sentiment features according to a specified aspect or entity. Second, the model computation was easy to parallelize during training.

However, the above models ignored both syntactical constraints [29] and long-range sentiment dependencies [7] while mistakenly identifying irrelevant contextual words as clues for judging aspect sentiment. For instance, in the sentence “Certainly not the best sushi in New York, however, it is always fresh, and the place is very clean and sterile”, the CNN model convolves only the information of consecutive words; thus it cannot judge the sentiment of nonadjacent words. Accordingly, it may mistakenly identify the phrase “not the best” as a clue for judging the aspect “sushi” but ignore the influence of the word “fresh” on the aspect “sushi”. In addition, these models usually use aspect-independent encoders to encode sentences, which could result in a lack of aspect information. In that same sentence, the words “sushi”, “best” and “fresh” are irrelevant for sentiment prediction when the considered aspect is “place”. The use of an aspect-independent encoder when encoding sentences can cause these words to be mistaken as clues for judging “place”, leading to an erroneous prediction. Therefore, we aim to use a graph convolutional network (GCN) [13] containing an aspect gate to address these shortages. A GCN has the ability to process data with generalized topological graph structure and extract spatial features; therefore, it can update the feature information by capturing the long-range sentiment dependencies between adjacent nodes. Zhang et al. [45] was the first to apply a GCN to aspect-based sentiment analysis. Their model exploits the syntactical dependency structures within a sentence and resolves the long-range multiword dependency issue for aspect-based sentiment classification, but it still exploits aspect-independent encoders to encode sentences, which can lead to a lack of aspect information.

In this paper, we propose the aspect-gated graph convolutional network (AGGCN) model for aspect-bas-ed sentiment analysis. First, we design an aspect gate based on a long short-term memory (LSTM) network that can guide the encoding of aspect-specific information from the outset while discarding aspect-indepen-dent information. Then, we generate a dependency tree based on aspect-specific information and construct a GCN on the dependency tree to fully capitalize on the syntactical information and long-range sentiment dependencies. Finally, we use a novel retrieval-based attention mechanism to obtain the hidden representation of the attention from GCN to predict aspect sentiment. Experimental results on multiple SemEval datasets de-monstrate the effectiveness of our proposed approach, and our model outperform the strong baseline models.

Our main contributions can be summarized as follows:

1. We propose an aspect-gate mechanism based on LSTM. The specific aspect gate can select an aspect-specific representation by controlling the token embedding transformation at each time step, which enables the LSTM to guide the encoding of aspect-specific from the outset and discard aspect-independent information. This mechanism solves the noise and bias problems caused by the weaker encoders used in previous models.
2. We propose a novel aspect-based sentiment analysis framework that employs a GCN to capture syntactical information and long-range sentiment dependencies. The proposed framework enables the model to perceive context through the syntactical information and long-range sentiment dependencies, and uses a novel attention mechanism to obtain the hidden representation of the attention from GCN. This framework can help identify irrelevant context words more accurately and avoid identifying them as clues for judging aspect sentiment.
3. We evaluate our method on multiple SemEval datasets. The experiments show that our model achieves higher accuracy than most of the baseline models and outperforms the strong baseline models.

The remainder of this paper is organized as follows. After introducing the related works in Section 2, we elaborate our proposed model in Section 3, and then conduct experiments in Section 4. Finally, we summarize our work and provide an outlook of future work in Section 5.

2 Related works

Aspect-based sentiment analysis has become one of the most active research fields in natural language processing

(NLP) and has spread from computer science to the social sciences [18, 20] and network sciences [16, 19], including marketing, finance, politics, communication, medical science and even history. Due to its clear commercial advantages, aspect-based sentiment analysis has aroused concerns throughout society. The previous aspect-based sentiment analysis models [10, 33] mostly used traditional methods based on dictionaries and machine learning methods. However, training those models was dependent on the quality of annotated data sets and obtaining high-quality datasets requires substantial investments of labor and high costs.

With the in-depth study of deep learning technology, neural networks are now widely used in aspect-based sentiment analysis. RNN models [23, 25, 31] have been applied to aspect-based sentiment analysis because they can flexibly capture the semantic relationship between an aspect and its context words. Tang et al. [31] were the first to introduce an LSTM into aspect-based sentiment analysis. They developed two target-dependent LSTM models that automatically considered target information. However, not all the information in a sequence is important to an RNN; thus, the attention mechanism [22, 35, 37] was introduced into the RNN model to cause it to pay more attention to the more important parts of the sequence. Wang et al. [36] proposed an attention-based LSTM for aspect-level sentiment classification. The attention mechanism concentrated on different parts of a sentence as different aspects were taken as input. Recursive neural network (RecNN) models [1, 26, 30] were applied to aspect-based sentiment analysis to replace RNNs because RNNs were unsuited for the tree and graph structures that information contains. Dong et al. [8] were the first to apply a RecNN model to aspect-based sentiment analysis. They proposed the adaptive recursive neural network (AdaRNN), which adaptively propagated the word sentiments to targets based on the context and syntactic relationships between words. However, RecNN may suffer from syntax parsing errors [34, 42]. With the development of word embedding technology, CNNs [5, 39, 40] have been widely employed in aspect-based sentiment analysis due to their ability to extract both local and global representations. Fan et al. [9] proposed a novel convolutional memory network that incorporated an attention mechanism. Their model sequentially computed the weights of multiple memory units corresponding to multiple words and can capture both word and multi-word expressions in sentences for use in aspect-based sentiment analysis.

Due to the unsatisfactory performance of neural networks such as RNN and CNN for processing graph data, related Graph algorithms [15] are introduced into NLP, especially GCNs have been widely used in aspect-based sentiment analysis. A GCN performs excellently for processing graph data containing rich relational information. A GCN possesses a multilayer architecture in which each layer encodes and updates the node representations in the graph using the features of the node's immediate neighbors. Zhao et al. [46] proposed a novel aspect-level sentiment classification model based on GCNs that effectively captures the sentiment dependencies between multiple aspects in one sentence. The model first introduced a bidirectional attention mechanism with position encoding to model the aspect-specific representations between each aspect and its context words and then employed a GCN over the attention mechanism to capture the sentiment dependencies between the different aspects in one sentence.

3 Our proposed model

In this section, we describe the proposed model AGGCN for aspect-based sentiment analysis in detail. The AGGCN is shown in Fig. 2. Specifically, we first define the model notations in Section 3.1. Then, we introduce an aspect-gated LSTM in Section 3.2 and construct the GCN based on the output of the aspect-gated LSTM in Section 3.3. Next, we describe the use a retrieval-based attention mechanism to generate an attention representation for sentiment analysis in Section 3.4. Finally, we present the model training process in Section 3.5.

3.1 Definition

First, we introduce some notations to facilitate the subsequent descriptions : $S = \left\{x_1^c, x_2^c, \cdots, x_n^c\right\}$ denotes an input sentence, which contains a corresponding aspect $X = \{x_{k+1}^a, x_{k+2}^a, \cdots, x_{k+m}^a\}$ starting from the $(k+1)$-th token. We embed each word token into a low-dimensional real-valued vector space [3] with an embedding matrix $E \in \mathbb{R}^{d_{emp} \times |V|}$, where d_{emp} denotes the dimension of word embedding, and V indicates the number of words involved in the corpus.

3.2 Aspect-gated LSTM

A conventional LSTM first discards irrelevant information via its forget gates, then it adds useful information through the input gate, and finally, it determines which information will be output through the output gate. Instead, we design an aspect gate in LSTM that selects aspect-specific representations by controlling the transformation of token embedding at each time step. At time step t, the hidden state h_t^c is formulated as follows:

$$h_t^c = o_t \times \tanh\left(C_t\right) \tag{1}$$

where $h_t^c \in \mathbb{R}^{2d_h}$ represents the hidden state vector at time step t from the bidirectional aspect-gated LSTM; d_h

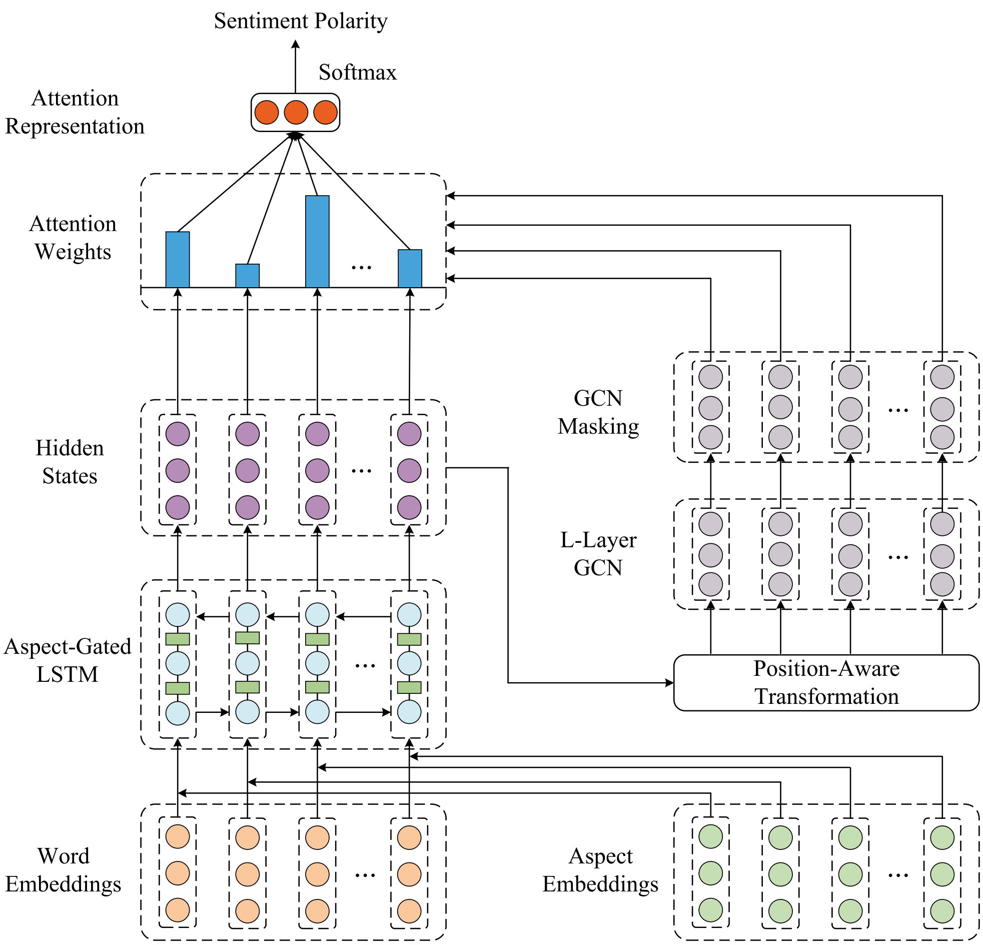

Fig. 2 The framework of our proposed model. A special aspect gate is designed to guide the encoding of aspect-specific information from the outset. The model constructs a graph convolution network on the dependency tree. In particular, we use position-aware transformation and GCN masking in the GCN to fully utilize the syntactical information and long-range sentiment dependencies. The position-aware transformation reduces noise and bias during graph convolution process, and GCN masking perceives contexts around the aspect in a way that considers both syntactical information and long-range sentiment dependencies

represents the dimension of the hidden state vector from the unidirectional aspect-gated LSTM; o_t is the output gate, which outputs the cell state through a sigmoid neural layer and a dot multiplication operation. Each element of the sigmoid layer output is a real number between 0 and 1 that represents the weight of the corresponding information passing through. For example, 0 means "no information" and 1 means "let all information pass". We process the cell state C_t through $tanh$ (obtaining a value between -1 and 1) and multiply it with the output of the output gate to obtain the hidden state h_t^c. The formula for the cell state C_t is as follows:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t^a \tag{2}$$

where f_t is the forget gate, which determines what information is discarded from the cell state. The forget gate outputs a value between 0 and 1 through h_{t-1}^c and x_t^a, where a 1 means "fully reserved", and a 0 means "completely abandoned". The i_t represents an input gate, which determines how much new information is added to the cell state. We multiply the previous cell state C_{t-1} by f_t to discard information determined a discardable. Then, we add the product of i_t and the candidate cell state C_{t-1} to obtain a new cell state C_t. In particular, we design the aspect gate to be located between the input gate i_t and the output gate o_t . The aspect gate controls the transformation of aspect information together with the $tanh$ function, and it plays a part in determining the candidate cell state $\tilde{C}_t^a$.

The $\tilde{C}_t^a$ is formulated as follows:

$$\begin{aligned}\tilde{C}_t^a &= \tanh\left(W_c h_{t-1}^c + g_t \cdot \left(W_g x_t^c\right)\right) + l_t \cdot H_t^c\left(x_t^c\right) \\ &\quad + g_t \cdot H_t^a\left(x_t^a\right)\end{aligned} \tag{3}$$

where g_t^a is the aspect gate, which is designed to guide the encoding of aspect-specific information from the outset. The aspect gate can control the transformation of aspect information together with the $tanh$ function, and it plays a part in the candidate cell state $\tilde{C}_t^a$. x_t^c and x_t^a represent the input word embedding and the aspect embedding at time step t. l_t [21] is the linear transformation gate for x_t^c, H_t^c and H_t^a represent the linear transformations of the input x_t^c and x_t^a, respectively, controlled by the linear transformation gate l_t and the aspect gate g_t^a. Equations (1) and (2) show that the hidden state h_t^c is controlled by the previous cell state C_{t-1}, $\tilde{C}_t^a$, l_t and g_t. The aspect gate structure alleviates the vanishing gradient problem because this approach provides a linear transformation path as a supplement between consecutive hidden states. W_c and W_g denote the weight matrix and bias, respectively. The remaining terms, o_t, i_t, f_t, l_t, g_t, $H_t^c\left(x_t^c\right)$ and $H_t^a\left(x_t^a\right)$ are formulated as follows:

$$\begin{aligned} o_t &= \sigma\left(W_o \cdot \left[h_{t-1}^c, x_t^c\right] + b_o\right)\\ i_t &= \sigma\left(W_i \cdot \left[h_{t-1}^c, x_t^c\right] + b_i\right)\\ f_t &= \sigma\left(W_f \cdot \left[h_{t-1}^c, x_t^c\right] + b_f\right)\\ l_t &= \sigma\left(W_l \cdot \left[h_{t-1}^c, x_t^c\right] + b_l\right)\\ g_t &= Relu\left(W_g \cdot \left[h_{t-1}^c, x_t^a\right] + b_g\right) \end{aligned} \tag{4}$$

$$\begin{aligned} H_t^c\left(x_t^c\right) &= W_{hc} x_t^c\\ H_t^a\left(x_t^a\right) &= W_{ha} x_t^a \end{aligned} \tag{5}$$

where σ represents the $sigmoid$ function, $ReLU$ is the $activation$ function. W_o, W_i, W_f, W_l, W_g, W_{hc}, W_{ha} are weight matrices and b_o, b_i, b_f, b_l, b_g are bias vectors to be learned during training.

In (3)–(5), the aspect gate g_t controls the nonlinear transformation of the input x_t^c under the guidance of the given aspect at time step t. According to the current input x_t^c, x_t^a and the previous hidden state h_{t-1}^c, we adopt the linear transformation gate l_t in the cooperative aspect gate g_t to control the linear transformation of input. Therefore, a specific aspect gate can select an aspect-specific representation by controlling the token embedding transformation at each time step, which enables the LSTM to guide the encoding of aspect-specific from the outset and discard aspect-independent information.

3.3 Graph convolutional network

In Section 3.2, we obtain the output $H^c = \{h_1^c, h_2^c, \cdots, h_{k+1}^c, \cdots, h_{k+m}^c, \cdots, h_n^c\}$ of the aspect-gated LSTM. Based on this H^c output, we first construct a syntactical dependency tree[1] and convert each tree into its corresponding adjacency matrix A, to make a GCN suitable for the modeling dependency tree. Then, the GCN is executed in an L-layer convolutional fashion on top of the aspect-gated LSTM output H^c, i.e., $H^l = H^c$ to create context-aware nodes. Finally, the hidden representation of each node is updated through a graph convolution operation with a normalization factor [13]. The graph convolution was inspired by contextualized Graph Convolutional Networks [44] as shown below:

$$h_i^l = Relu\left(\sum_{j=1}^{n} A_{ij} W_l\, g_j^{l-1}/d_i + b_l\right) \tag{6}$$

where $h_i^l \in \mathbb{R}^{2d_h}$ is the i-th token's hidden representation of the l-th GCN layer, and $g_j^{l-1} \in \mathbb{R}^{2d_h}$ is the j-th token's representation evolved from the $(l-1)$-th GCN layer. $A_{ij} \in \mathbb{R}^{n\times n}$ denotes the adjacency matrix. Specifically,based on the idea of self-looping, each word is manually set adjacent to itself, i.e., the diagonal values of A are all ones. W_l is the weight matrix and b_l is the bias vector to be learned during training. Then, $d_i = \sum_{j=1}^{n} A_{ij}$ represents the degree of the i-th token in the tree.

Specifically, to reduce the noise and bias during the graph convolution process, we conduct a position-aware transformation [11, 17, 36] before h_i^l is input into GCN.

$$g_i^l = p_i h_i^l \tag{7}$$

$$p_i = \begin{cases} \left|\frac{i-k-1}{n}\right| & 0 < i < k+1\\ 0 & k+1 \le i \le k+m\\ \left|\frac{k+m-i}{n}\right| & k+m < i \le n \end{cases} \tag{8}$$

where $p_i \in \mathbb{R}$ is the position weight of the i-th token. The final hidden representation of the L-layer GCN is $H_L = \left\{h_1^L, h_2^L, \cdots, h_{k+1}^L, \cdots, h_{k+m}^L, \cdots, h_n^L\right\}$, where $h_t^L \in \mathbb{R}^{2d_h}$. Table 1 describes the above process.

3.4 Retrieval-based attention

In this section, we use a retrieval-based attention mechanism to generate an attention representation. This idea was derived from the aspect-specific graph convolutional network [45]. The retrieval-based attention mechanism

Table 1 The formal pseudo-code for Graph Convolution is presented in Algorithm 1

```
Algorithm 1 Graph Convolution.
Input: The input sentence $S = \{x_1^c, x_2^c, \cdots, x_n^c\}$, which
contains the corresponding aspect $A = \{x_{k+1}^a, \cdots, x_{k+m}^a\}$
starting from $(k+1)$-th token. The hidden representation
$H^l = \{h_1^l, h_2^l, \cdots, h_{k+1}^l, \cdots, h_{k+m}^l, \cdots, h_n^l\}$.The position
-aware transformation relations $p$.
Output: The hidden representation $h_i^l$ of $l$-th GCN layer.
while (T <maximum number of iterations) do
  for i to n do
    for j to n do
      if (0 < j ≤ k+1) or (k+m < j ≤ n) then
        $h_i^l \xrightarrow{p} g_j^l$
        Update GCN hidden state $h_i^l$ by the (6)
      else
        $0 \xrightarrow{p} g_j^l$
        Update GCN hidden state $h_i^l$ by the (6)
      end if
    end for
  end for
end while
return $h_i^l$
```

[1] We use spaCy toolkit:https://spacy.io/

retrieves significant features that are semantically relevant to the aspect words from the hidden state vectors and sets a retrieval-based attention weight accordingly for each context word.

We first add a masking mechanism on top of the GCN to mask out nonaspect words. This operation enables the model to perceive context through syntactical information and long-range sentiment dependencies.

$$H^L_{Mask} = \begin{cases} 0 & 0 < t < k+1 \\ h^L_t & k+1 \le t \le k+m \\ 0 & k+m < t \le n \end{cases} \tag{9}$$

When the current word is not an aspect word, we set the value of H^L_{Mask} to 0. Conversely, when the current word is an aspect word, we use the value from (6). Table 2 describes the process of GCN Masking.

Then, we produce the retrieval-based attention representation based on H^c and H^L_{mask} and formulate it as follows:

$$h_R = \sum_{t=1}^{n} \alpha_t h^c_t \tag{10}$$

$$\alpha_t = \frac{exp\left(\beta_t\right)}{\sum_{i=1}^{n} exp\left(\beta_i\right)} \tag{11}$$

$$\beta_t = \sum_{i=1}^{n} \left(h^c_t\right)^\top h^L_i = \sum_{i=k+1}^{k+m} \left(h^c_t\right)^\top h^L_i \tag{12}$$

Table 2 The formal pseudo-code for GCN Masking is presented in Algorithm 2

Algorithm 2 GCN Masking.

```
Input: .The representation $g^l_i$ evolved from the $(l-1)-$th
GCN layer. The hidden representation of $L$-layers GCN $H^L$
$= \{h^L_1, h^L_2, \cdots, h^L_{k+1}, \cdots, h^L_{k+m}, \cdots, h^L_n\}$. The GCN Mask
relations $H_M$.
Output: The GCN Masking representation $H^L_{Mask}$.
while (T <maximum number of iterations) do
    for i to n do
        for j to n do
            if (0 < j ≤ k+1) or (k+m < j ≤ n) then
                $0 \xrightarrow{H_M} h^L_{Mask}$
            else
                $h^L_j \xrightarrow{p} h^L_{Mask}$
            end if
        end for
    end for
end while
return $H^L_{Mask}$
```

where h_R is the retrieval-based attention representation, α_t represents the attention weight, and β_t is the attention-aware function to obtain the semantic correlation between the aspect and context.

Finally, we input the attention representation h_R into the $softmax$ layer for aspect-based sentiment analysis:

$$y = softmax\left(W_p h_R + b_p\right) \tag{13}$$

where $y \in \mathbb{R}^{|C|}$ is the sentiment distribution prediction, and $W_p \in \mathbb{R}^{2d_h \times |C|}$ and $b_p \in \mathbb{R}^{|C|}$ are the trainable parameters. C is the dimension of the sentiment labels.

3.5 Training of model

The purpose of model training is to optimize all the parameters to minimize the loss function insofar as possible. Our model is trained using cross-entropy with the $L2$-regularization term and formulated as follows:

$$loss = -\sum_{i}^{N} y_i \log\left(\hat{y}_i\right) + \lambda \|\theta\|^2 \tag{14}$$

where N is the number of samples in the dataset, y_i is the ground truth probability, and $\hat{y}_i$ is the estimated probability of an aspect. λ is the $L2$-regularization factor and θ represents all the trainable parameters.

4 Experiments

In this section, we first describe the datasets and experimental settings in Section 4.1. Then, we describe the baseline models in Section 4.2 and the experimental results and analyses in Section 4.3. Next, we provide a discussion of AGGCN in Section 4.4, and we present an ablation study in Section 4.5. Finally, we describe a case study in Section 4.6

4.1 Datasets and experimental settings

To demonstrate the effectiveness of our proposed model, we conduct experiments on five datasets, namely Lap14, Rest14, Rest15, Rest16 and Twitter, which are originally from SemEval 2014 task 4,[2] SemEval 2015 task 12,[3] SemEval 2016 task 5[4] and Twitter[5] respectively. The SemEval datesets consist of data in two categories: $Restaurant$ and $Laptop$. The word embeddings that are fixed in the Twitter dataset consist of data in one category: $Twitter$, and the reviews include three sentiment polarity

[2] Available at: http://alt.qcri.org/semeval2014/task4/

[3] Available at: http://alt.qcri.org/semeval2015/task12/

[4] Available at: http://alt.qcri.org/semeval2016/task5/

[5] Available at: http://goo.gl/5Enpu7

labels: positive, negative, and neutral. The dataset statistics are shown in Table 3.

In our experiments, we apply the pretrained GloVe vectors with 300 dimensions to initialize the word embeddings. The dimension of the hidden state vectors is set to 300. All the weight matrices obtain their initial values from a uniform distributed U $(-0.1, 0.1)$. All the models are optimized using the Adam optimizer with the learning rate set to 0.001. The $L2$ regularization is set to 0.00001, and the batch size is set to 32. In addition, the number of GCN layers is set to 2, which was the best depth found during the experiment. To evaluate the performance, we obtain the experimental results by averaging the results of 20 runs with random initialization, and we adopt accuracy and the macro-averaged F1-score (Macro-F1) as the evaluation metrics. The Macro-F1 metric is more appropriate when the data set is unbalanced.

4.2 Baseline models

To evaluate the effectiveness of our model, we compare it with the following baseline models on all five datasets:

TD-LSTM constructs aspect-specific representation by the left context with aspect and the right context with aspect and then employs two LSTMs to model them [31].
ATAE-LSTM generates an attention vector by combining aspect embedding with hidden state, and it appends the aspect embedding into each word vector to better capitalize on the aspect information [36].
MenNet uses a deep memory network on the context word embeddings for sentence representation to capture the relevance between each context word and the aspect. Finally, the output of the last attention layer is used to infer aspect polarity [32].
IAN first learns attention from the contexts and aspect terms. Then, the representations for aspect terms and contexts are generated separately. Finally, it concatenates the aspect term representation and the context representation to predict the sentiment polarity of the aspect terms within its contexts [24].
AOA jointly learns the representations for aspects and sentences and automatically focuses on the important parts in sentences [12].
TNET proposed context-preserving transformation (C-PT) to preserve and strengthen the informative part of contexts [17].
AS-GCN exploits syntactical dependency structures within a sentence and resolves the long-range multiword dependency issue for aspect-based sentiment classification [45].
DMTL uses a shared layer to learn the common features of sentiment prediction (SP) and position prediction (PP). Then, it uses two task-specific layers to learn the features specific to the tasks and perform PP and SP in parallel [49].

Table 3 Dataset description

Dataset		Positive	Negative	Neutral
Lap14	Train	994	870	464
	Text	341	128	169
Rest14	Train	2164	807	637
	Text	728	193	196
Rest15	Train	978	307	36
	Text	326	182	34
Rest16	Train	1230	417	62
	Text	440	107	28
Twitter	Train	1561	1560	3127
	Text	173	1743	346

4.3 Experimental results and analyses

As shown in Table 4, we report the performance of all the baseline models and our proposed AGGCN model. From Table 4, we can make the following observations:

Compared with the baseline models, AGGCN achie-ves the best performances on the Rest14, Rest16 and Twitter datasets. On the Rest14 datasets, compared with the best baseline model DMTL, AGGCN achieves absolute increases of 2.14% and 1.33% in accuracy and Macro-F1, respectively. These results demonstrate the effectiveness of using the syntactic information and long-range sentiment dependencies. On the Rest16 and Twitter datasets, compared with the best baseline model AS-GCN, AGGCN achieves absolute increases of 1.54% and 1.07% in accuracy, respectively, and it also achieves absolute increases of 6.44% and 1.80% in Macro-F1. These results demonstrate that encoding the aspect-specific information from scratch can increase the model accuracy for sentiment analysis. The accuracy of AGGCN is slightly below that of AS-GCN on Lap14, and AS-GCN also achieves the best performance on the Lap14 datasets. AS-GCN makes full use of the syntactical dependency structures within a sentence and resolves the long-range multiword dependency issue. One possible reason for this discrepancy is that the Lap14 datasets are not as sensitive to aspect-specific information but they are more sensitive to syntactic information. In addition, the performance of AGGCN is lower than that of DMTL on Rest15, where DMTL also shows the best performance on Rest15. DMTL uses a shared layer to learn the common features of SP and PP. Then, it utilizes two task-specific layers to learn the features specific to the tasks and performs PP and SP in parallel. DMTL pays more attention to the influence of position information on the model. Its best results on the Rest15

Table 4 Average accuracy and macro-F1 score over 20 runs with random initialization. The two best results on each dataset are shown in bold font

Model	Lap14		Rest14		Rest15		Rest16		Twitter	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
TD-LSTM	71.80	68.46	78.00	68.43	76.39	58.70	82.16	54.21	69.89	66.21
ATAE-LSTM	68.88	63.93	78.60	67.02	78.48	62.84	83.77	61.71	70.14	66.03
MenNet	70.64	65.17	79.16	69.53	77.89	59.52	83.04	57.91	71.48	69.90
IAN	71.95	67.14	79.42	70.01	78.18	52.45	84.74	55.21	72.45	71.26
AOA	73.11	68.47	79.06	70.13	78.17	57.02	87.50	66.21	72.30	70.20
TNET	74.95	70.16	80.77	72.03	78.19	60.67	88.72	70.16	72.57	71.13
AS-GCN	**75.55**	**71.05**	80.77	72.02	79.89	61.89	88.99	67.48	72.15	70.40
DMTL	73.35	68.93	82.23	72.49	**82.72**	**66.94**	88.13	71.48	–	–
AGGCN	73.53	68.99	**84.37**	**73.82**	80.27	64.51	**90.53**	**73.92**	**73.64**	**72.20**

datasets may be because the Rest15 dataset is not as sensitive to syntactic information and long-range dependencies as it is to position information.

4.4 Discussion of AGGCN

One highly important parameter in AGGCN is the number of GCN layers because that value affects the performance of our model. To demonstrate the effectiveness of our proposed model, we investigate the effect of the layer number L on the final performance of AGGCN, and we conduct experiments with different numbers of GCN layers from 1 to 9. The performance results are shown in Figs. 3 and 4.

As the results in Figs. 3 and 4 show, the model achieves the best performances when the number of GCN layers is 2. When the number of GCN layers is larger than 2, the performance degrades as the number of GCN layers increases on both datasets. One possible reason for this performance drop phenomenon may be that as the number of model parameters increases, the model becomes more difficult to train and tends to overfit.

4.5 Ablation study

To investigate the impacts of each component on AGGCN, we conducted an ablation study. The results are shown in Table 5.

AGGCN w/o AG denotes a model with the aspect gate component removed. As the results show, when we remove the aspect gate, the performance of AGGCN degrades on the Rest14, Rest15, Rest16 and Twitter datasets but it improves on the Lap14 data sets. These results demonstrate that the aspect gate helps the model better identify and extract information on specific aspects. Specifically, recalling the result on the Lap14 datasets in Table 4, the reasons for the performance degradation of our proposed model may be that Lap14 datasets are less sensitive to aspect-specific information but are more sensitive to syntactic information. The notation *w/o AGGCN w/o GCN* denotes a model from

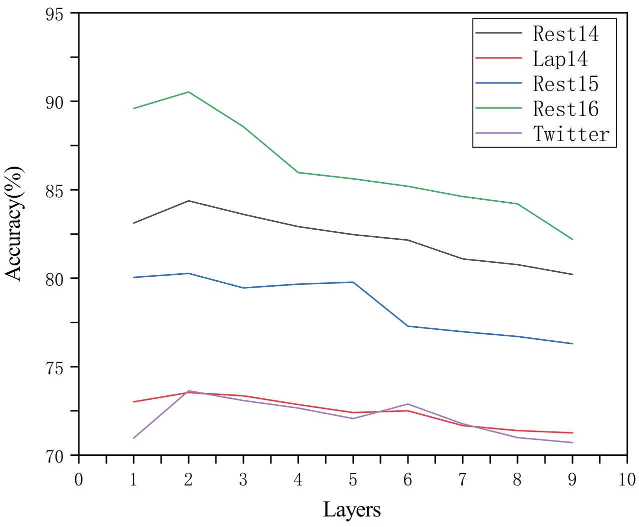

Fig. 3 The effect of the different layers number in Accuracy

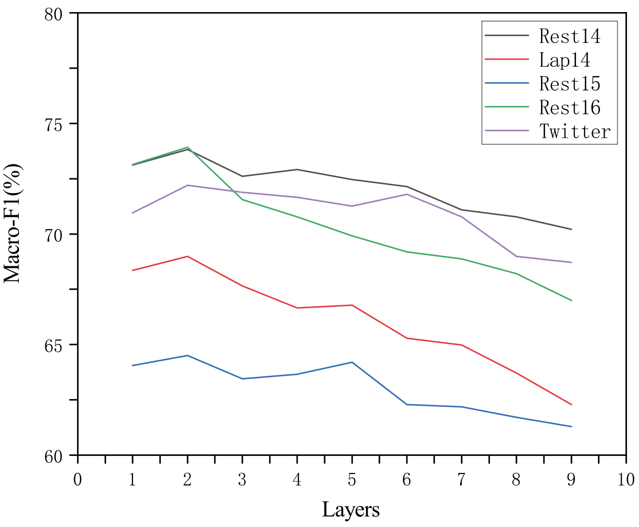

Fig. 4 The effect of the different layers number in Macro-F1

Table 5 Ablation study of the AGGCN on five datasets, AG means aspect gate

Model	Lap14		Rest14		Rest15		Rest16		Twitter	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
AGGCN	73.53	68.99	**84.37**	**73.82**	**80.27**	**64.51**	**90.53**	**73.92**	73.64	72.20
AGGCN w/o AG	**75.53**	**71.04**	80.76	72.01	79.66	62.01	89.04	67.52	72.12	70.36
AGGCN w/o GCN	72.33	68.41	81.15	72.42	78.24	63.95	88.70	73.47	**74.54**	**73.14**
AGGCN w/o pos.	73.07	68.70	82.37	73.12	75.99	63.36	90.36	73.73	73.22	71.19
AGGCN w/o mask	72.99	67.28	81.87	73.77	78.29	64.14	89.30	73.82	73.08	71.55

The two best results on each dataset are shown in bold font

which we removed the GCN mechanism. When we remove the GCN component, the performance of AGGCN drops on the Lap 14, Rest14, Rest15, and Rest16 datasets but improves on the Twitter dataset. This result demonstrates that the GCN simultaneously captures both the syntactic information and the long-range sentiment dependencies. One possible reason for the performance degradation is that the Twitter dataset is not sensitive to syntactic information and long-range sentiment dependencies. The notation *AGGCN w/o pos.* denotes a model from which we removed the position-aware transformation component. Compared with the complete AGGCN, the performance of *AGGCN w/o pos.* falls on all five datasets but especially on Rest15. Recall the performance of the baseline model DMTL in Table 4, which pays more attention to position information. We can conclude that the Rest15 datasets are more sensitive to position information. The notation *AGGCN w/o mask* denotes a model from which we removed the mask component. The performance of *AGGCN w/o mask* falls on all five datasets, demonstrating that the mask mechanism helps the AGGCN perceive contexts around the aspect in a way that considers both syntactical dependencies and long-range sentiment dependencies.

4.6 Case study

To provide an intuitive understanding of how the AGGCN works with different components, we adopted a case study as a test example for illustrative purposes. We constructed heat maps to visualize the attention weights on the words computed by the three models in which the color depth denotes the semantic relatedness level between the given

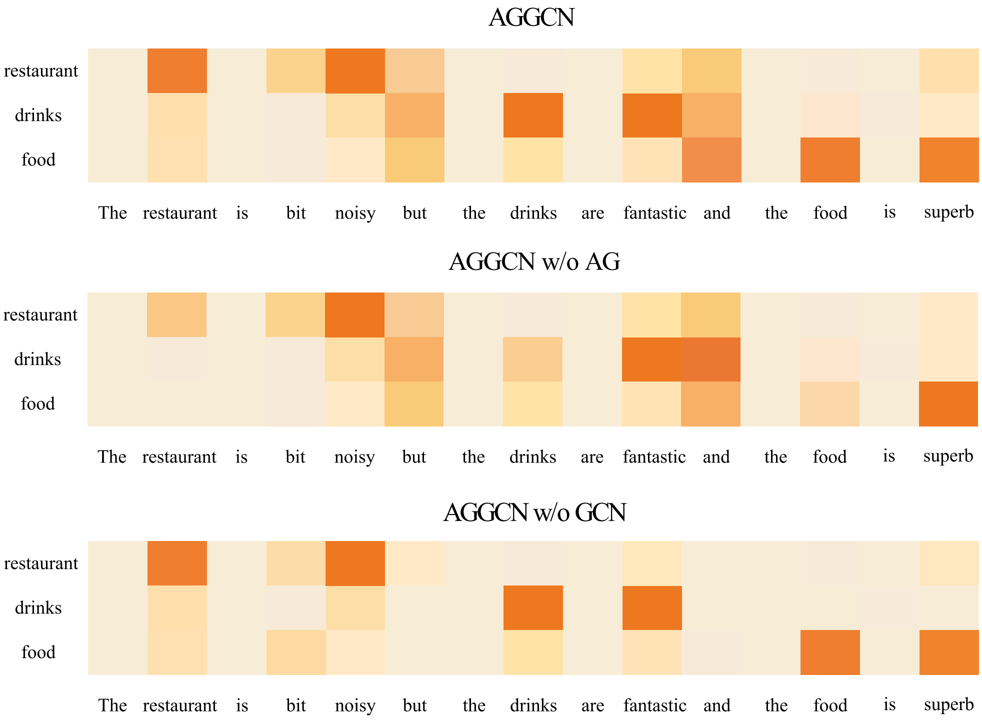

Fig. 5 Visualization of attention from AGGCN, AGGCN w/o AG and AGGCN w/o GCN on a testing example

aspect and each word. More depth indicates a stronger relation to the given aspect. The results are shown in Fig. 5.

In this sentence, the aspect word "restaurant" is the target of negative sentiment; the aspect words "drinks" and "food" are connected by the conjunction "and" to express positive sentiment; and the conjunction "but" reverses the previous negative sentiment. By comparing the heat maps of *AGGCN* and *AGGCN w/o AG*, we find that AGGCN accurately focuses on three aspects in the sentence: "restaurant", "drinks" and "food" and it also pays attention to the conjunctions "but" and "and". This phenomenon indicates that AGGCN not only identities aspect information but also perceives the context in a way that considers both syntactic information and long emotional dependency. When we remove the aspect gate, as can be seen in the heat maps in the second row, the color of the aspect words "restaurant", "drinks" and "food" becomes lighter. This phenomenon indicates that the ability of AGGCN to focus on aspects is weakened. When we remove the GCN, as seen from the third heat maps, AGGCN no longer focuses on the conjunctions "but" and "and". The *ASGCN w/o GCN* model predicts the polarity of aspect "drinks" by the word "fantastic" and the polarity of aspect "food" by the word "superb" in isolation, ignoring the relation between the two aspects.

From the above results, we can conclude that our proposed model can not only identify the aspect and address the lack of aspect information in prior models through the special aspect gate but also perceive the contexts around the aspect by considering both syntactical dependencies and long-range sentiment dependencies. These mechanisms make our model better for aspect-based sentiment analysis.

5 Conclusion and future work

In this paper, we proposed an Aspect-gated Graph Convolutional Network (AGGCN) for aspect-based sentiment analysis. The AGGCN not only guides the encoding of aspect-specific information from the outset and discards aspect-independent information but also perceives contexts around the aspect by considering both syntactical dependencies and long-range sentiment dependencies. The experimental results on multiple SemEval datasets demonstrate the effectiveness of our proposed approach, and our model outperforms the str-ong baseline models.

In future work, we plan to further improve the performance of the model from the following aspects. First, noise and biases may occur during the encoding of aspect-specific information; therefore, it is necessary to introduce a deep conversion transformation mechanism that can decode the aspect information to ensure that it is completely and accurately embedded into the model. Second, domain knowledge could be incorporated to improve model generalizability.

Acknowledgements This work was supported in part by the National Social Science Foundation under Award 19BYY076, in part Key R & D project of Shandong Province 2019 JZZY010129, and in part by the Shandong Provincial Social Science Planning Project under Award 19BJCJ51, Award 18CXWJ01, and Award 18BJYJ04.

References

1. Al-Smadi M, Qawasmeh O, Al-Ayyoub M, Jararweh Y, Gupta B (2018) Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. J Comput Sci 27:386–393
2. Bakshi RK, Kaur N, Kaur R, Kaur G (2016) Opinion mining and sentiment analysis. In: 2016 3rd International conference on computing for sustainable global development (INDIACom). IEEE, pp 452–455
3. Bengio Y, Ducharme R, Vincent P, Jauvin C (2003) A neural probabilistic language model. J Mach Learn Res 3:1137–1155
4. Chen P, Sun Z, Bing L, Yang W (2017) Recurrent attention network on memory for aspect sentiment analysis. In: EMNLP, pp 452–461
5. Chen T, Xu R, He Y, Wang X (2017) Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. Expert Syst Appl 72:221–230
6. Cheng J, Zhao S, Zhang J, King I, Zhang X, Wang H (2017) Aspect-level sentiment classification with heat (hierarchical attention) network. In: CIKM, pp 97–106
7. Dieng AB, Wang C, Gao J, Paisley J (2016) TopicRNN: a recurrent neural network with long-range semantic dependency. In: Proceedings of the 5th international conference on learning representations
8. Dong L, Wei F, Tan C, Tang D, Zhou M, Xu K (2014) Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, vol 2: short papers, pp 49–54
9. Fan C, Gao Q, Du J, Gui L, Xu R, Wong K-F (2018) Convolution-based memory network for aspect-based sentiment analysis. In: The 41st International ACM SIGIR conference on research & development in information retrieval, pp 1161–1164
10. Ghiassi M, Lee S (2018) A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach. Exp Syst Appl 106:197–216
11. Gu S, Zhang L, Hou Y, Song Y (2018) A position-aware bidirectional attention network for aspect-level sentiment analysis. In: Proceedings of the 27th international conference on computational linguistics, pp 774–784
12. Huang B, Ou Y, Carley KM (2018) Aspect level sentiment classification with attention-over-attention neural networks. In: International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation. Springer, pp 197–206
13. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th international conference on learning representations
14. Kumar R, Pannu HS, Malhi AK (2020) Aspect-based sentiment analysis using deep networks and stochastic optimization. Neural Comput Appl 32(8):3221–3235

15. Le N-T, Vo B, Nguyen LB, Fujita H, Le B (2020) Mining weighted subgraphs in a single large graph. Inf Sci 514:149–165
16. Li H-J, Wang L (2019) Multi-scale asynchronous belief percolation model on multiplex networks. New J Phys 21(1):015005
17. Li X, Bing L, Lam W, Shi B (2018) Transformation networks for target-oriented sentiment classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics, pp 946–956
18. Li H-J, Bu Z, Wang Z, Cao J (2019) Dynamical clustering in electronic commerce systems via optimization and leadership expansion. IEEE Trans Ind Inform 16(8):5327–5334
19. Li H-J, Wang Z, Pei J, Cao J, Shi Y (2020) Optimal estimation of low-rank factors via feature level data fusion of multiplex signal systems. IEEE Ann Hist Comput 01:1–1
20. Li H-J, Wang L, Zhang Y, Perc M (2020) Optimization of identifiability for efficient community detection. New J Phys 22(6):063035
21. Liang Y, Meng F, Zhang J, Xu J, Chen Y, Zhou J (2019) A novel aspect-guided deep transition model for aspect based sentiment analysis. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 5572–5584
22. Liu J, Zhang Y (2017) Attention modeling for targeted sentiment. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics, vol 2, short papers, pp 572–577
23. Lu Q, Zhu Z, Zhang D, Wu W, Guo Q (2020) Interactive rule attention network for aspect-level sentiment analysis. IEEE Access 8:52505–52516
24. Ma D, Li S, Zhang X, Wang H (2017) Interactive attention networks for aspect-level sentiment classification. In: Proceedings of the 26th international joint conference on artificial intelligence, pp 4068–4074
25. Ma Y, Peng H, Cambria E (2018) Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In: Thirty-second AAAI conference on artificial intelligence
26. Nguyen TH, Shirai K (2015) Phrasernn: phrase recursive neural network for aspect-based sentiment analysis. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 2509–2514
27. Pontiki M, Galanis D, Papageorgiou H, Manandhar S, Androutsopoulos I (2015) Semeval-2015 task 12: aspect based sentiment analysis. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp 486–495
28. Pontiki M, Galanis D, Papageorgiou H, Androutsopoulos I, Manandhar S, Al-Smadi M, Al-Ayyoub M, Zhao Y, Qin B, De Clercq O (2016) Semeval-2016 task 5: aspect based sentiment analysis. In: 10th International workshop on semantic evaluation (SemEval 2016)
29. Sayeed A, Boyd-Graber J, Rusk B, Weinberg A (2012) Grammatical structures for word-level sentiment detection. In: Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics, human language technologies, pp 667–676
30. Socher R, Lin CC, Manning CD, Ng AY (2011) Parsing natural scenes and natural language with recursive neural networks. In: International conference on machine learning, pp 129–136
31. Tang D, Qin B, Feng X, Liu T (2016) Effective lstms for target-dependent sentiment classification. In: COLING, pp 3298–3307
32. Tang D, Qin B, Liu T (2016) Aspect level sentiment classification with deep memory network. In: EMNLP, pp 214–224
33. Tang F, Fu L, Yao B, Xu W (2019) Aspect based fine-grained sentiment analysis for online reviews. Inf Sci 488:190–204
34. Vo D-T, Zhang Y (2015) Target-dependent twitter sentiment classification with rich automatic features. In: Twenty-fourth international joint conference on artificial intelligence
35. Wallaart O, Frasincar F (2019) A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and attentional neural models. In: European semantic web conference. Springer, pp 363–378
36. Wang Y, Huang M, Zhu X, Zhao L (2016) Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 606–615
37. Wang J, Li J, Li S, Kang Y, Zhang M, Si L, Zhou G (2018) Aspect sentiment classification with both word-level and clause-level attention networks. In: IJCAI, pp 4439–4445
38. Wen S, Wei H, Yang Y, Guo Z, Zeng Z, Huang T, Chen Y (2019) Memristive LSTM network for sentiment analysis. IEEE Trans Syst Man Cybern Syst 99:1–10
39. Xue W, Li T (2018) Aspect based sentiment analysis with gated convolutional networks. In: Proceedings of the 56th annual meeting of the association for computational linguistics, vol 2018, pp 2514–2523
40. Zeng D, Dai Y, Li F, Wang J, Sangaiah AK (2019) Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism. J Intell Fuzzy Syst 36(5):3971–3980
41. Zhang L, Liu B (2017) Sentiment analysis and opinion mining. In: Sammut C, Webb GI (eds) Encyclopedia of machine learning and data mining. Springer, Boston, pp 1152–1161
42. Zhang M, Zhang Y, Vo D-T (2016) Gated neural networks for targeted sentiment analysis. In: Thirtieth AAAI conference on artificial intelligence
43. Zhang L, Wang S, Liu B (2018) Deep learning for sentiment analysis: a survey. Wiley Interdiscip Rev: Data Min Knowl Discov 8(4):e1253
44. Zhang Y, Qi P, Manning CD (2018) Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 2205–2215
45. Zhang C, Li Q, Song D (2019) Aspect-based sentiment classification with aspect-specific graph convolutional networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 4560–4570
46. Zhao P, Hou L, Wu O (2020) Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. Knowl-Based Syst 193:105443
47. Zhou X, Wan X, Xiao J (2016) Attention-based LSTM network for cross-lingual sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 247–256
48. Zhou J, Huang JX, Chen Q, Hu QV, Wang T, He L (2019) Deep learning for aspect-level sentiment classification: survey, vision, and challenges. IEEE Access 7:78454–78483
49. Zhou J, Huang JX, Hu QV, He L (2020) Is position important? Deep multi-task learning for aspect-based sentiment analysis. Appl Intell 50:3367–3378

Qiang Lu received the B.E. degree in computer science and technology from Shandong Jiaotong University of Information Science and Electrical Engineering, Jinan, China, in 2016, where he is currently pursuing the M.E. degree. His research interests include sentiment analysis, text classification, and deep learning.

Zhenfang Zhu received the Ph.D. degree from Shandong Normal University in 2012, and he was a postdoctoral fellow at Shandong University between 2012 and 2015. He is currently a Professor and a Master's Supervisor with the School of information science and electrical engineering, Shandong Jiao Tong University, China. His research interests include network information security, natural language processing, and applied linguistics.

Guangyuan Zhang received the Ph.D. degree from Northeastern University, and he was a postdoctoral fellow at Tsinghua University. He is currently a Professor and a Master's Supervisor with the School of information science and electrical engineering, Shandong Jiao Tong University, China. His research interests include Digital Image Processing, Pattern recognition, and Detection technology.

Shiyong Kang received the M.A. degree from Lanzhou University. He is currently a Professor and a Master's Supervisor with the School of Chinese Language and Literature, Ludong University, China. He is the national outstanding science and technology worker, middle-aged and young expert with outstanding contribution in Shandong province, famous teacher of Shandong province. His research interests include Modern Chinese, Chinese information processing, and natural language processing.

Peiyu Liu received the Master degree from East China normal university in 1986. He is currently a Second-level Professor, doctoral supervisor, with the School of information science and engineering, Shandong Normal University, China. He is the national outstanding science and technology worker, middle-aged and young expert with outstanding contribution in Shandong province, famous teacher of Shandong province. His research interests include network information security, information retrieval, natural language processing, and artificial intelligence.