



# Modeling multiple latent information graph structures via graph convolutional network for aspect-based sentiment analysis

Jiajun Wang<sup>1</sup> · Xiaoge Li<sup>1,2,3</sup> · Xiaochun An<sup>1,2</sup>

Received: 24 July 2022 / Accepted: 26 November 2022 / Published online: 16 December 2022  
© The Author(s) 2022

## Abstract

Aspect-based sentiment analysis (ABSA) aims to determine the sentiment polarity of aspects in a sentence. Recently, graph convolution network (GCN) model combined with attention mechanism has been used for ABSA task over graph structures, achieving promising results. However, these methods of modeling over graph structure fail to consider multiple latent information in the text, i.e., syntax, semantics, context, and so on. In addition, the attention mechanism is vulnerable to noise in sentences. To tackle these problems, in this paper, we construct an efficient text graph and propose a matrix fusion-based graph convolution network (MFLGCN) for ABSA. First, the graph structure is constructed by combining statistics, semantics, and part of speech. Then, we use the sequence model combined with the multi-head self-attention mechanism to obtain the feature representation of the context. Subsequently, the text graph structure and the feature representation of context are fed into GCN to aggregate the information around aspect nodes. The attention matrix is obtained by combining sequence model, GCN and the attention mechanism. Besides, we design a filter layer to alleviate the noise problem in the sentence introduced by the attention mechanism. Finally, in order to make the context representation more effective, attention and filtering matrices are integrated into the model. Experimental results on four public datasets show that our model is more effective than the previous models, demonstrating that using our text graph and matrix fusion can significantly empower ABSA models.

**Keywords** Aspect-based sentiment analysis · Graph convolution network · Attention mechanism · Matrix fusion

## Introduction

The aspect-based sentiment analysis (ABSA) task aims to determine the sentiment tendency of a particular aspect in a sentence. For example, “The environment of this restaurant is dirty, but the food is delicious.” there are two aspects “environment” and “food”, and the user expresses negative and positive sentiments over them, respectively. Thus, ABSA can precisely judge the sentiment tendency of a specific aspect, rather than simply judging sentiment polarity for a sentence.

In recent studies, various neural network models such as Recurrent Neural Networks (RNN) [1] and Convolutional

Neural Networks (CNN) [2,3] are widely used in aspect-based sentiment analysis tasks. With the popularity of the attention mechanism, more researchers [4–7] combine attention mechanism and neural network to reflect the degree of influence of each word on the aspect. However, the attention mechanism is susceptible to the noise in the sentence and mistakenly focuses on irrelevant words, and sequential models modeling context semantic relationships easily lose long-distance information.

Compared with sequential models, graph convolutional neural networks (GCN) [8] can handle complex data structures and model information to the global level. Because of the advantages of GCN, more researchers [9–12] combined dependency trees with GCN to make better use of syntactic dependency in sentences. However, the following problems arise when using dependency trees in ABSA tasks: (1) due to the establishment of the dependency tree structure is automatically created according to other tools, the dependencies will be inaccurate and uncontrollable. (2) Insensitive to domain-specific datasets [13]. To solve the problem of the dependency tree, researchers [13–18] makes improvements

✉ Xiaoge Li  
lixg@xupt.edu.cn

<sup>1</sup> Xi’an University of Posts and Telecommunications, Xi’an 710121, Shaanxi, China

<sup>2</sup> Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi’an 710121, Shaanxi, China

<sup>3</sup> Xi’an Key Laboratory of Big Data and Intelligent Computing, Xi’an 710121, Shaanxi, China

and enhancements based on dependency trees to solve ABSA tasks. However, these methods only consider the dependencies between words, but fail to consider more important information.

Information on the context, semantics, part of speech, and sentiment knowledge among words are essential when constructing GCN for aspect-based sentiment analysis. However, only a limited number of researchers model multiple hidden information via GCN for this task. For example, Yao et al. [19] proposed a text graph-based neural network (TextGCN). In their model, a text graph is first constructed based on the sequential contextual relationship between words. In [20], they combine syntax and knowledge via a GCN. More researchers [21,22] focus on building graph structures based on rich contextual information, such as semantic and syntactic contextual information. The effective graph learning is not only important in the ABSA field, but many scholars have also done a lot of meaningful work [23,24] in the graph learning in terms of multiple kernel graph-based clustering (MKGC). It can be seen that researchers have been paid more and more attention to effective graph learning.

Inspired by these studies, we model multiple latent information graph structures via GCN for aspect-based sentiment analysis. In the text graph structure building module, we integrate three different perspectives of statistics, semantics, and part of speech to build graph structures. The learned graph structure can fully extract the latent information in the text. In the ABSA module, we design a matrix fusion-based GCN for better context coding. In the ABSA field, most researchers directly combined the output of one type of neural network with the attention mechanism. They did not adequately combine the sequential neural network, GCN and attention mechanisms. Therefore, we first use the sequence model and the multi-head self-attention mechanism to generate the feature representation of the context. Second, the text graph and context feature matrix of graph structure are input into the graph structure model GCN. The aspect features that aggregate the information of adjacent nodes of the graph structure are obtained through multi-layer GCN. Then we calculate the attention matrix from the aspect features and the context features obtained from the sequence model. The filter matrix is obtained from the output of GCN through the aspect irrelevant information filter layer. Finally, the aspect-related information is enhanced using a matrix fusion layer. Finally, the aspect-related information is enhanced using a matrix fusion layer.

Our contributions are as follows:

- We propose a graph convolution network model with multiple latent information for ABSA tasks. Our model considers the statistics, semantics, and part of speech within a sentence. We combine the graph structure obtained by the statistical method and the graph struc-

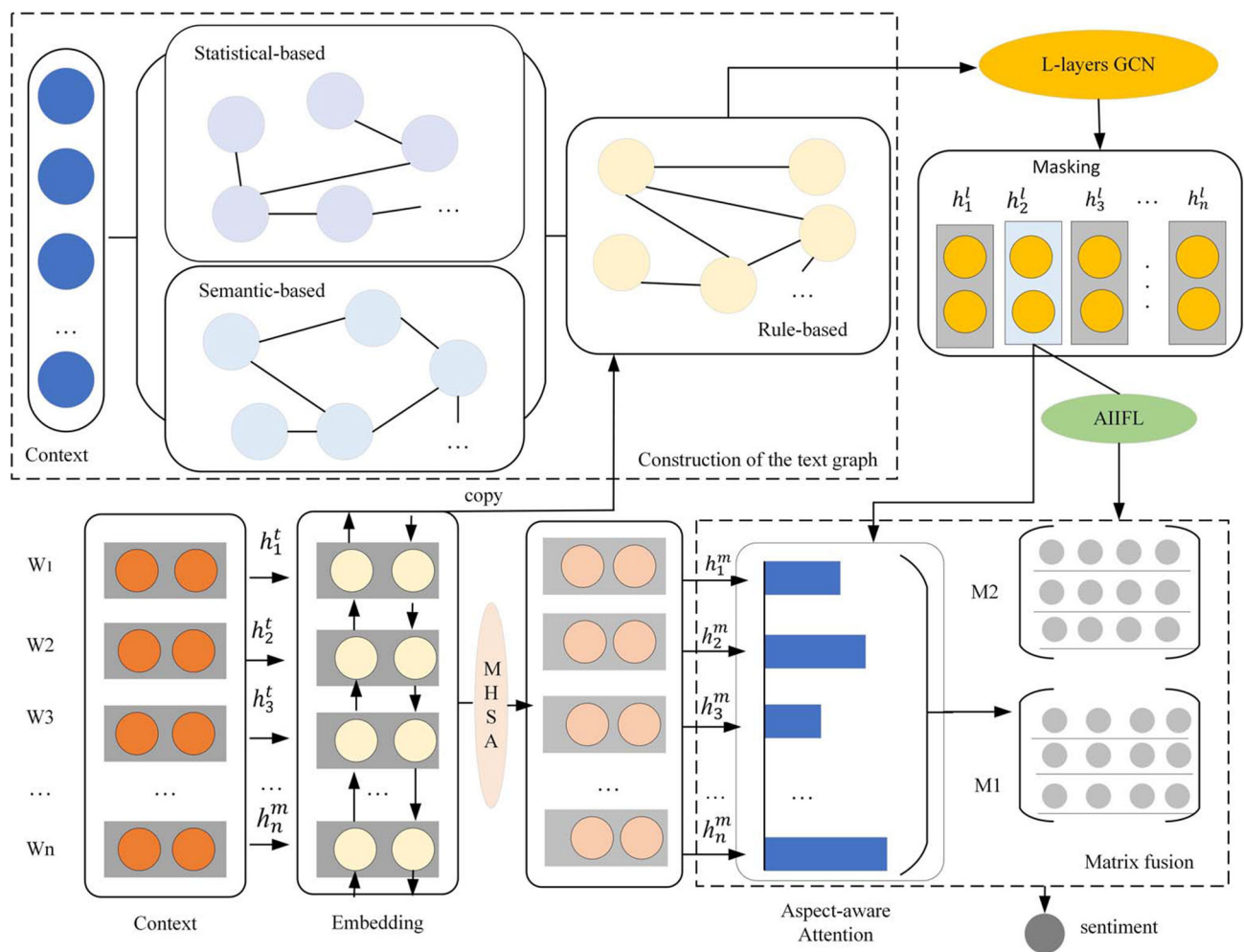
ture obtained by semantic similarity. Then the final graph structure is enhanced by part of speech rules.

- We use a matrix fusion-based GCN over the graph structure. First, the attention matrix is obtained by combining the aspect feature output of GCN, the text feature output of the sequence model LSTM and the attention mechanism. Then, we use the information filter layer after GCN to filter the irrelevant information of the aspect. Finally, we combine the attention matrix and the filter matrix to get the final matrix.
- Extensive experiments are conducted on four benchmark datasets to illustrate the effectiveness of our model for the ABSA task.

## Related work

Aspect-based sentiment analysis is a fine-grained sentiment analysis task. Recently, most research work on aspect-based sentiment analysis (ABSA) use neural network and attention to associate aspect and context to capture semantic information hidden in sentences. For instance, Wang et al. [1] and Ma et al. [6] propose a model to compute the attention of aspects and sentences directly. Tang et al. [4] uses multiple levels of attention to the model. Chen et al. [5] adds the weighted memory mechanism on the basis of multi-layer attention. Fan et al. [7] propose a model to learn the representation containing sentence and aspect-related information, integrate it into the multi-granularity sentence modeling process, and finally get a comprehensive sentence representation. Huang and Carley [2] propose a novel parameterized convolutional neural network for aspect-level sentiment classification. They use parameterized filters and parameterized gates on CNN to incorporate aspects of information. Tan et al. [25] introduced a dual attention network to recognize conflicting opinions. However, these studies ignore the syntactic dependence between words in sentences, which may lead to ambiguity in identifying the polarity of specific targets.

In order to solve this problem, dependency trees are introduced into the ABSA task. Dependency trees can capture dependencies between words and enhances the connection between aspect and related words. More researchers use graph convolution neural network (GCN) to model based on dependency trees. Sun et al. [10] present a convolution over dependency trees model which combines Bi-directional Long Short-Term Memory (Bi-LSTM) and GCN. Zhang et al. [9] propose a model to combine attention mechanism and GCN on dependency trees. Xiao et al. [12] improve the GCN and combine a GCN with a multi-head attention mechanism for aspect-based sentiment analysis. Although dependency trees perform well in ABSA tasks, it also has some defects. The methods based on dependency trees are inaccurate and insensitive to domain-specific datasets. In order to alleviate



**Fig. 1** The overall architecture of the MFLGCN. The upper part is built for the text graph structure. It includes graph structures for statistics, semantics, and parts of speech. The lower part is the aspect-based sentiment analysis part. It contains GCN and matrix fusion layer

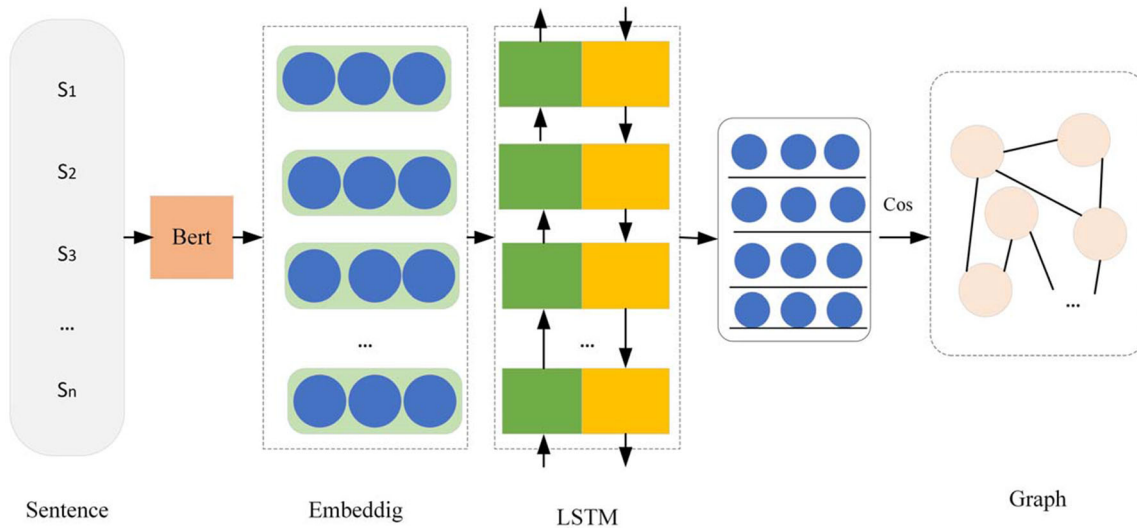
these problems, more researchers have enhanced the graph structure information on dependency trees. Chen et al. [15] propose a model to combine dependency trees and latent graphs which are generated by self-attention networks. Wang et al. [16] utilize reshaping and pruning methods to make the ordinary dependency trees focus on the target aspect. Zhou et al. [20] employ a new Syntax- and Knowledge-based Graph Convolutional Network (SK-GCN) model for aspect-level sentiment classification. Tang et al. [17] use dual graph convolutional networks to improve the insensitivity of dependency syntax trees to online reviews.

In this paper, the text graph structure is generated by combining different methods. The statistic-based graph can consider the co-occurrence frequency between words in the text. The semantic-based graph can consider the rich semantic relationship of context in text and well-build graph structure in different domain datasets. The part of speech rules-based graph can enrich the relationship of specific aspects and strengthen the importance of aspects in the graph.

We combine the aspect word feature output of GCN, the text feature output of the sequence model and the attention mechanism based on the constructed graph structure to extract attention matrix. We also use GCN and aspect information filter layer to extract aspect-related information matrix. Finally, we integrate the attention matrix considering the importance between aspect and other words and the matrix filtering irrelevant aspect information to obtain the final representation.

### Our model

Figure 1 gives an overview of our model. It contains the graph building part and the aspect-based sentiment analysis part. In the graph building part, we first use statistical methods to calculate the co-occurrence between words and use it as a basis for forming each edge in the statistic-based graph  $G_1$ . Second, we construct the semantic-based graph  $G_2$  using the semantic similarity between words as a basis. Thirdly, we



**Fig. 2** Semantic-based graph construction model

keep the common edges in  $G_1$  and  $G_2$  and delete the rest to get a graph. Finally, we process the nodes corresponding to the edges around the aspect and other edges corresponding to these nodes based on the part of speech rules. The edges that conform to the set part of speech rules are added and those that do not conform are deleted, thus obtaining the graph  $G_3$ . In the aspect-based sentiment analysis part, we first utilize Bi-LSTM to extract hidden contextual representations  $H^l$ . Then these hidden representations  $H^l$  are fed into GCN as features of the graph nodes. The vector representations  $H^l$  that aggregates relevant information after GCN is obtained. We use the multi-head self-attention mechanism to link words more effectively to get  $H^m$ . The attention matrix  $M_1$  is obtained by calculating the aspect vector in aspect and  $H^m$  for attention. The aspect-related information matrix  $M_2$  is obtained by passing the  $H^l$  and the aspect vector through the filtering information layer. Finally, the matrix fusion of  $M_1$  and  $M_2$  is performed to obtain the final text representations.

### Construction of the text graph

The statistic-based graph integrates the co-occurrence information between words. We combine the sliding window strategy and point-wise mutual information (PMI) to express the degree of association between words. The basis for the existence of edges between nodes in the graph can be formulated as

$$T_{st}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \quad (1)$$

$$p(w_i, w_j) = \frac{N(w_i, w_j)}{N_{total}} \quad (2)$$

$$p(w_i) = \frac{N(w_i)}{N_{total}} \quad (3)$$

$$p(w_j) = \frac{N(w_j)}{N_{total}}. \quad (4)$$

### Statistic-based graph

$p(w_i, w_j)$  is co-occurrence of  $w_i$  and  $w_j$  is the total number of the sliding windows over the whole dataset.  $N(w_i)$  is the number of occurrences that the word  $w_i$  occurs in the sliding windows over the whole dataset. The weight between the word nodes  $w_i$  and  $w_j$  is defined as

$$M_{st_{ij}} \begin{cases} 1, & T_{st}(w_i, w_j) > \text{threshold}_{st} \\ 1, & i = j \\ 0, & T_{st}(w_i, w_j) < \text{threshold}_{st} \end{cases} \quad (5)$$

where  $M_{st} \in \mathbb{R}^{n \times n}$  is the adjacency matrix representation of the statistic-based graph.  $M_{st_{ij}}$  represents whether the  $i$ th node is connected to the  $j$ th node in the graph. The  $\text{threshold}_{st}$  is the standard to judge whether nodes are connected.

### Semantic-based graph

Bert can fine-tune and act as a feature extractor according to the needs of the task to obtain high-quality word vector representation. LSTM can capture context semantic relationships. We use the Bert-LSTM model to represent the word vector of the text (shown in Fig. 2) and then calculate the cosine similarity between words to construct the edges in the semantic graph. The weight calculation formula of semantic

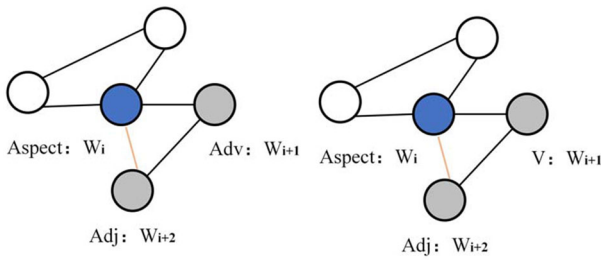


Fig. 3 Part of speech rules

relationship between words is as follows:

$$T_{se}(w_i, w_j) = \cos\langle v_i, v_j \rangle = \frac{v_i \cdot v_j}{\|v_i\| \times \|v_j\|} \tag{6}$$

where  $v_i$  and  $v_j$  are the vectors of words  $w_i$  and  $w_j$ , respectively. The weight between the word nodes  $w_i$  and  $w_j$  is defined as

$$M_{se_{ij}} \begin{cases} 1, & T_{se}(w_i, w_j) > \text{threshold}_{se} \\ 1, & i = j \\ 0, & T_{se}(w_i, w_j) < \text{threshold}_{se} \end{cases} \tag{7}$$

where  $M_{se} \in \mathbb{R}^{n \times n}$  is the adjacency matrix representation of the semantic-based graph.  $M_{se_{ij}}$  represents whether the  $i$ th node is connected to the  $j$ th node in the graph. The  $\text{threshold}_{se}$  is the standard to judge whether nodes are connected.

**Graph enhancement**

In order to integrate statistical and semantic information simultaneously, we build a graph that contains the common edges of the semantic-based graph and statistic-based graph. The weight between the word nodes  $w_i$  and  $w_j$  is defined as

$$M_{\text{fusion}_{ij}} \begin{cases} 1, & M_{st_{ij}} = 1 \text{ and } M_{se_{ij}} = 1 \\ 1, & i = j \\ 0, & M_{st_{ij}} = 0 \text{ or } M_{se_{ij}} = 0 \end{cases} \tag{8}$$

where  $M_{\text{fusion}} \in \mathbb{R}^{n \times n}$  is the adjacency fusion matrix representation of statistic-based graph and semantic-based graph. In order to enrich the information associated with aspect nodes in the graph, we use part of speech rules to enhance the graph structure, as shown in Fig. 3.

In the graph structure, if the part of speech of node  $w_{(i+1)}$  directly connected with aspect  $w_i$  is an adverb or verb, it will enter the next stage of part of speech judgment. if the part of speech of node  $w_{(i+2)}$  directly connected with aspect  $w_{(i+1)}$  is an adjective, add an edge between  $w_i$  and  $w_{(i+2)}$ .

Overall, the graph structure we built is a graph with properties, such as the following: (1) captures the co-occurrence

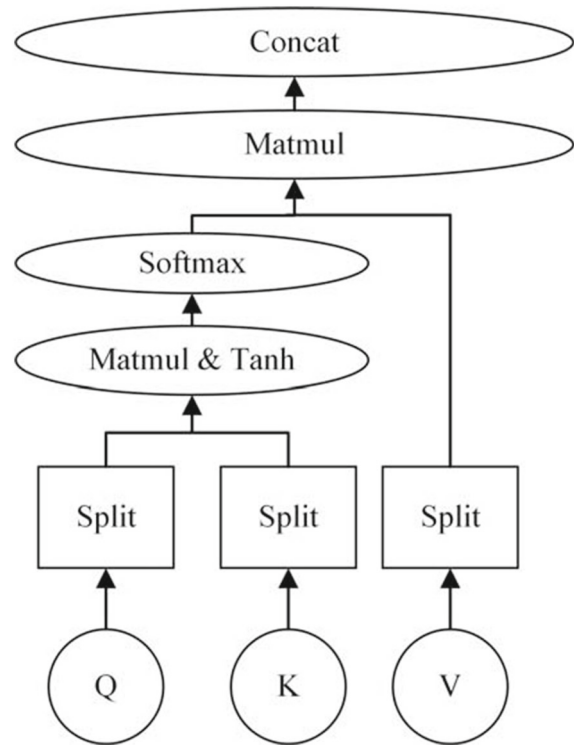


Fig. 4 Calculation process of MHSA

between words; (2) integrates rich contextual semantic relations; (3) considers the relationship between aspects and other words at the part of speech level.

**Contextualized word representation**

A sentence  $s = \{w_1^t, w_2^t, w_3^t, \dots, w_n^t\}$  containing aspects  $a = \{w_1^a, w_2^a, w_3^a, \dots, w_{m+1}^a\}$  is given, where  $n$  is the length of the sentence and  $m + 1$  is the length of the aspect. We use a Bi-LSTM encoder to capture contextual semantic relationships and obtain the contextualized word representations  $H^t = \{h_1^t, h_2^t, h_3^t, \dots, h_\beta^a, h_{\beta+1}^a, h_{\beta+2}^a, \dots, h_{\beta+m}^a, \dots, h_n^t\} \in \mathbb{R}^{n \times d_n}$ , where  $d_h$  is the dimension of the hidden state vectors.

**Multi-head self-attention (MHSA)**

Multi-head self-attention (MHSA) is an attention mechanism that can parallelly operate in space. Compared with self-attention, the MHSA can extract more abundant semantic features, as shown in Fig. 4. First, we initialize  $Q$ ,  $K$  and  $V$  to make them equal to the input  $H^t$ , respectively. The different weight matrices  $W$  is then used to map  $Q$ ,  $K$  and  $V$  to different subspaces. The subspace mapping formula is shown

in the following formula:

$$\begin{cases} Q_i = QW_i^Q \\ K_i = KW_i^K \\ V_i = VW_i^V \end{cases} \quad (i = 1, 2, 3, \dots, n) \quad (9)$$

where  $W_i^Q$ ,  $W_i^K$  and  $W_i^V$  are the learning parameter matrices of the  $i$ -th head, respectively.  $n$  is the number of the heads. In each subspace, the output is calculated according to the attention function with  $K$ ,  $Q$  and  $V$  as inputs. The calculation formula of attention is shown in the following formula:

$$\text{Attention}(Q, K, V) = \text{softmax}(\tanh(W_m \cdot [K; Q])) V \quad (10)$$

The formula (10) is essentially the weighted calculation of  $V$ . The weight is obtained by a softmax function. In the softmax, the tanh function is used to calculate the correlation between  $K$  and  $Q$ .  $W_m$  is the learning weight matrix. After obtaining the output of each subspace, they are concatenated to obtain the final output:

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i) \quad (i = 1, 2, 3, \dots, n) \quad (11)$$

$$H^m = \text{MHSA}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_n) \quad (12)$$

where  $\text{head}_i$  is the output of  $i$ -th head attention calculation and  $H^m = \{h_1^m, h_2^m, h_3^m, \dots, h_n^m\} \in \mathbb{R}^{n \times d_e}$  is the final output of the MAMS.  $d_e$  is the dimension of MHSA.

### Graph convolution network (GCN)

Graph convolutional network (GCN) [8] uses convolution operation to encode graph structure data. The output is the node representation that aggregates the information around the node. We use GCN to model our constructed text graph structure incorporating multiple latent information. The adjacency matrix representation of the graph and the feature matrix of the graph nodes are first passed into GCN. After multi-layer training, the obtained node feature representation then aggregates the surrounding information. Finally, the mask layer is used to extract the aspect-specific representation. A graph  $G = \{V, A\}$  is given, where  $V$  is the set of nodes in the graph, and  $A \in \mathbb{R}^{n \times n}$  is the adjacency matrix representation of a graph.  $n$  is the number of nodes.  $h_i^l$  is the node hidden state representation of  $l$ -layer. The node hidden state representation is updated by

$$h_i^l = \sigma \left( \sum_{j=1}^n A_{ij} W^l h_j^{l-1} + b^l \right) \quad (13)$$

where  $W^l$  is a learning weight matrix,  $b^l$  is a bias term,  $\sigma$  is a nonlinear function (e.g., ReLU).  $A_{ij}$  represents whether the  $i$ th node is connected to the  $j$ th node in the graph.  $A_{ij} = 1$  if node  $i$  is connected to node  $j$ , otherwise  $A_{ij} = 0$ .

In order to make the GCN learn aspect-specific representations. We use aspect mask on the output of GCN  $H_l = \{h_1^l, h_2^l, h_3^l, \dots, h_n^l\}$ . The aspect  $a = \{w_\beta^\alpha, w_{\beta+1}^\alpha, w_{\beta+2}^\alpha, \dots, w_{\beta+m}^\alpha\}$  is given. The output weight matrix is calculated as follows:

$$o_i = \begin{cases} 1 - \frac{\beta-i}{n}, & 1 \leq i < \beta \\ 0, & \beta \leq i < m \\ 1 - \frac{i-m}{n}, & m < i \leq n \end{cases} \quad (14)$$

where  $o_i$  is the weight of the  $h_i^l$ . The weight and GCN output are calculated as follows:

$$h_i^{lo} = o_i h_i^l \quad (15)$$

where  $h_i^{lo} \in \mathbb{R}^d$  is a hidden representation of the  $i$ th node and  $H^{lo} = \{h_1^{lo}, h_2^{lo}, h_3^{lo}, \dots, h_n^{lo}\}$  denotes hidden representations.

### Aspect-aware attention

GCN aggregates the information of nodes around aspect, which enriches the aspect representation. In order to fully associate aspect with other words, we combine the feature output of the aspect after GCN aggregation extracted by the masking layer, the text feature output of the sequence neural network model LSTM and the attention mechanism. This operation allows the fusion of the rich features hidden by aspect in the text graph structure with the text features learned from the sequence model for attention calculation, so as to fully integrate the vital information in aspect into the attention matrix obtained. We extract the aspect representation  $a^{lo} = \{0, \dots, h_\beta^{lo}, h_{\beta+1}^{lo}, h_{\beta+2}^{lo}, \dots, h_{\beta+m}^{lo}, \dots, 0\}$  from the sentence representation output by GCN, and then calculate the attention mechanism with the sentence feature representation  $H^m$  by

$$\alpha_t = \sum_{i=\beta}^{\beta+m} h_i^{mT} h_i^{lo} \quad (16)$$

$$M_1 = \sum_{t=1}^n \frac{\exp(\alpha_t) h_t^m}{\sum_{i=1}^n \exp(\alpha_i)} \quad (17)$$

where  $M_1$  is the output of aspect-aware attention. We use the dot product to calculate the semantic similarity between aspect and other words.

### Aspect irrelevant information filter layer (AIIFL)

The attention mechanism may introduce noise (irrelevant information) in the ABSA task. It may cause the model to capture irrelevant sentimental information, thus reducing the accuracy of the analysis. In order to alleviate this problem, we designed an aspect irrelevant information filter layer (AIIFL). The calculation formula is as follows:

$$h_i^f = f_i h_i^{lo} \tag{18}$$

$$f_i = \tanh \left( W_s \cdot h_i^{lo} + W_a \cdot a^{lo} + b_f \right) \tag{19}$$

where  $W_s$  and  $W_a$  are the weight matrix.  $b_f$  is a bias term.  $M_2 = \{h_1^f, h_2^f, h_3^f, \dots, h_n^f\}$  is the output matrix of the filter layer.

### Matrix fusion layer (MFL)

In order to effectively integrate the attention matrix  $M_1$  and the aspect information correlation matrix  $M_2$ , we designed a matrix fusion layer to improve the feature representation of sentences. The fusion formula is as follows:

$$M'_1 = \text{softmax} \left( M_1 W_1 (M_2)^T \right) M_2 \tag{20}$$

$$M'_2 = \text{softmax} \left( M_2 W_2 (M_1)^T \right) M_1 \tag{21}$$

$$M = \left[ M'_1, M'_2 \right] \tag{22}$$

where  $W_1$  and  $W_2$  are learning weight matrix.

Finally, the obtained representation  $M$  is input to a linear layer and the sentiment probability distribution is assigned using the softmax function:

$$p(e) = \text{softmax}(W_p M + b_p) \tag{23}$$

where  $W_p$  and  $b_p$  are the learning weight matrix and the bias.

### Loss function

The loss function of MFLGCN uses cross entropy and L2-regularization: The input format for the above table is as follows:

$$\text{Loss} = - \sum_{(d,e) \in D} \log p(e) + \lambda \|\theta\|_2 \tag{24}$$

where  $D$  is the training dataset,  $e$  is the true label and  $p(e)$  is the label of model prediction.  $\theta$  represents all trainable parameters, and  $\lambda$  is the coefficient of the regularization term.

**Table 1** The details of the experimental datasets

Datasets	Positive		Neutral		Negative	
	Train	Test	Train	Test	Train	Test
Twitter	1561	173	3127	346	1560	173
Rest14	2164	728	637	196	807	196
Lap14	994	341	464	169	870	128
MAMS	3380	400	5042	329	2761	607

## Experiments

### Datasets

We conduct experiments on four public datasets in different domains: Twitter [26], Restaurant, Laptop [27] and MAMS [28]. These datasets have three sentiment polarities: positive, negative, and neutral. The details of the experimental datasets are shown in the Table 1.

### Evaluation metrics

We use accuracy and  $F1$ -score to evaluate the performance of each model. The evaluation metrics are defined as follows:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \tag{25}$$

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}} \tag{26}$$

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{27}$$

$$F\text{-measure} = \frac{2 (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{28}$$

where TP, TN, FP and FN denote true positive, true negative, false positive and false negative, respectively.

### Implementation details

For our text graph construction, we set the number of words in the sliding window to 5. In the semantic-based graph, the initial word embeddings are pre-trained with Bert, and the dimension is 768. Our experiment uses 300-dimensional Glove vectors to initialize the word embeddings. The Bi-LSTM hidden size is set to 300. We set the parameter of the regulation to 0.00001. The batch size is set to 32. To alleviate overfitting, we apply dropout at a rate of 0.5. We set the number of GCN layers for the GCN model to 2. The number of multi-head self-attention heads was 8. Adam optimizer is used.

**Table 2** Comparison of different experimental results on public datasets

Model	Twitter		Lap14		Rest14		MAMS	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
ATAE-LSTM	69.65	67.40	69.14	63.18	78.60	67.02	77.56	73.03
MEMNET	71.48	69.88	69.90	65.71	79.61	69.64	64.57	62.70
IAN	72.50	70.81	72.05	67.38	79.26	70.09	76.60	71.17
RAM	69.36	67.30	74.49	71.35	80.23	70.80	–	–
GCAE	–	–	71.98	68.71	75.74	62.45	77.59	73.25
MGAN	72.54	70.81	75.39	72.47	81.25	71.94	–	–
AOA	–	–	72.62	67.52	79.97	70.42	77.26	72.29
TD-GAT	72.20	70.45	75.63	70.74	81.32	71.72	–	–
ASGCN-DT	71.53	69.68	74.14	69.24	80.86	72.19	78.12	76.87
ASGCN-DG	72.15	70.40	75.55	71.05	80.77	72.02	78.84	77.23
kumaGCN	72.45	70.77	76.12	72.42	81.43	73.64	–	–
BiGCN	74.16	<b>73.35</b>	74.59	71.84	81.97	73.48	–	–
AEGCN	73.16	71.82	75.91	71.63	81.04	71.32	79.72	77.93
MFLGCN	<b>74.42</b>	73.33	<b>76.33</b>	<b>72.57</b>	<b>83.25</b>	<b>74.54</b>	<b>80.98</b>	<b>79.93</b>

The bold value is the highest value in this column

## Comparison with the state-of-the-art

### Comparison models

We use ASGCN and AEGCN as our main baseline models, and we also compare our proposed model (MFLGCN) with the following methods:

- ATAE-LSTM [1] combines aspect embedding and attention mechanism to ABSA.
- MEMNET [4] employs multi-hops attention level to represent the features of context.
- IAN [6] uses the combination of LSTM and interactive attention mechanism to express the context and aspect.
- RAM [5] designs a model combining multiple attention and memory networks to learn the sentence representation.
- GCAE [29] utilizes the gating units to combine the output of CNN with two convolutional layers.
- MGAN [7] proposes a multi-grained attention mechanism to capture the relationship between context and aspect.
- AOA [2] obtains the corresponding representation of the context and aspect through the interactive learning of attention.
- TD-GAT [11] proposes a graph attention network over the dependency tree to solve ABSA tasks.
- ASGCN [9] propose a model to combine attention mechanism and GCN on the dependency tree.
- kumaGCN [14] designs a latent graph structure to capture aspect representations with syntactic information.

- BiGCN [13] propose a novel architecture which convolutes over hierarchical syntactic and lexical graphs.
- AEGCN [12] utilizes a variety of attention mechanisms and GCN to ABSA on the dependency tree structure.

### Experimental results

Table 2 shows the comparison results on four benchmark datasets, which demonstrate that the proposed MFLGCN model is better than all comparison models. Accuracy and *F1*-score are used to evaluate these models. Compared to the traditional sequential model combined with attention mechanism such as ATAE-LSTM, MEMNET, IAN, RAM, GCAE, MGAN, and AOA, graph convolution network (GCN) combined with dependency trees such as ASGCN and AEGCN are making use of the rich dependencies between words and can avoid noises introduced by the attention mechanism, so its performance has been improved. However, the methods based on dependency trees are inaccurate and insensitive to domain-specific datasets. Compared with the method of directly using dependency trees, the performance of GCN combined with the improved dependency tree structure such as kumaGCN and BiGCN are higher than ASGCN and AEGCN. In order to model multiple latent information, we combined statistics, semantics, and part of speech to construct the graph structure and built our sentiment classification model. Compared with the previous optimal model, our proposed MFLGCN model achieves the best results in terms of accuracy on all datasets and obtains the best macro-averaged *F1*-score on Lap14, Rest14 and MAMS datasets. In order to more vividly and concretely show the performance of our model, we average the accuracy of the sequence model



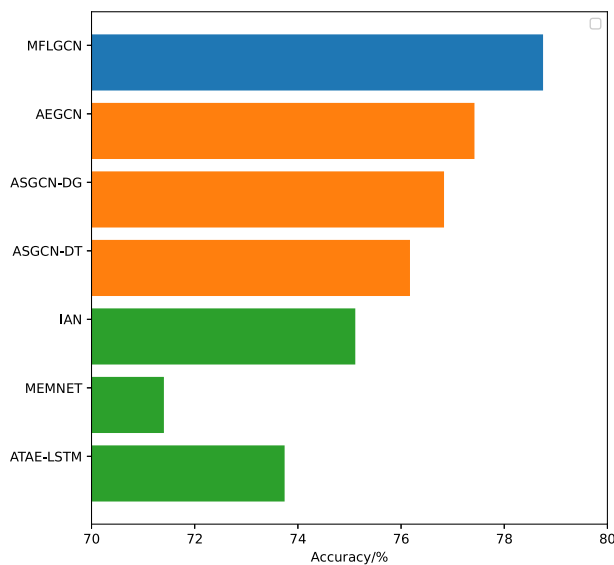


Fig. 5 Comparison of average accuracy of different models

(ATAE-LSTM, MEMNET, IAN), the dependency tree-based GCN model (ASGCN-DT, ASGCN-DG, AEGCN) and our model (MFLGCN) on the four datasets. The results are shown in Fig. 5. The green in the figure represents the sequence model, the orange represents the dependency tree-based model and the blue is our model. From the Fig. 5, we can find that the proposed model obtains higher accuracy than other models.

### Ablation study

To further investigate the impact of each component of the model on the experimental results, we make some ablation experiments. The results are shown in Table 3.

In order to prove the effectiveness of part of speech graph structure enhancement, MFLGCN w/o GE experiment is set up. MFLGCN w/o GE means that we remove the graph enhancement (GE) in the composition stage. It can be seen from the results that the experimental results on the four datasets have decreased after removing GE. Compared with the complete model on the four datasets, when the GE was ablated, the accuracy of the model decreased by 0.9%, 0.19%,

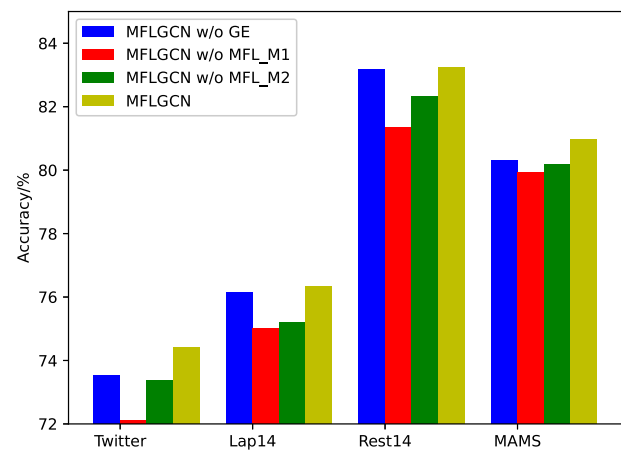


Fig. 6 Component performance comparison

0.07% and 0.66%, respectively. The attention matrix  $M_1$  obtained by combining the sequence model, GCN and attention mechanism can make the feature representation of aspect more relevant to the context in the model. The aspect-related information matrix  $M_2$  obtained through the combination of GCN output and aspect irrelevant information filter layer (AIIFL) not only aggregates the rich information around aspect nodes, but also alleviates the impact of noise in sentences on aspect feature generation. In order to reflect the effectiveness of our matrix fusion layer in fusing the attention matrix and the information filtering matrix, we set experiment w/o MFL\_M1, which means ablation of  $M_1$  only uses  $M_2$  for testing, and experiment w/o MFL\_M2, which means ablation of  $M_2$  only uses  $M_1$  for testing. Compared with the complete model on the four datasets, when the attention matrix  $M_1$  was ablated, the accuracy of the model decreased by 2.3%, 1.31%, 1.9% and 1.05%, respectively. When the aspect-related information matrix  $M_2$  was ablated, the accuracy of the model decreased by 1.03%, 1.12%, 0.93% and 0.81%, respectively. Based on the results, we can find that the interaction of GE, Attention, and AIIFL is significant in our model.

From Fig. 6, we can find that different components have different effects on the performance of the model. The influence degree of each component on the model ranges from high to low: Attention, AIIFL, and GE.

Table 3 Experimental results of ablation study

Model	Twitter		Lap14		Rest14		MAMS	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
MFLGCN w/o GE	73.52	71.90	76.14	72.48	83.18	74.47	80.32	79.84
MFLGCN w/o MFL_M1	72.12	70.13	75.02	72.12	81.35	72.32	79.93	78.15
MFLGCN w/o MFL_M2	73.36	71.87	75.21	72.33	82.32	73.36	80.17	78.82
MFLGCN	<b>74.42</b>	<b>73.33</b>	<b>76.33</b>	<b>72.57</b>	<b>83.25</b>	<b>74.54</b>	<b>80.98</b>	<b>79.93</b>

The bold value is the highest value in this column

**Table 4** Experimental results of different graph structures

Model	Twitter		Lap14		Rest14		MAMS	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
ASGCN-DG—our graph	72.03	71.91	75.61	71.23	81.16	73.52	79.07	77.82
ASGCN-DG—dependency tree	72.15	72.04	75.55	71.05	80.77	72.02	78.84	77.23
AEGCN—our graph	73.21	71.84	76.05	72.32	82.14	74.26	79.87	78.03
AEGCN—dependency tree	73.16	71.82	75.91	71.63	81.04	71.32	79.72	77.93
MFLGCN—dependency tree	73.35	71.88	76.07	72.45	82.20	73.83	79.96	78.72
MFLGCN	<b>74.42</b>	<b>73.33</b>	<b>76.33</b>	<b>72.57</b>	<b>83.25</b>	<b>74.54</b>	<b>80.98</b>	<b>79.93</b>

The bold value is the highest value in this column

### Impact of different graph structures

In order to verify the effectiveness of our graph structure, we use our graph to replace the dependency tree on ASGCN and AEGCN, and use the dependency tree to do experiments on our model. Table 4 shows the results of different graph structures.

As can be seen from Table 4, when the baseline models (ASGCN and AEGCN) use our graph, the performance is higher than that using the dependency tree. On Lap14, Rest14 and MAMS datasets, the performance of ASGCN model was improved by 0.06%, 0.39% and 0.23%, respectively. On the Twitter dataset, our graph is inferior to the dependency tree. After our research and analysis, the reasons are as follows: (1) the design of the ASGCN model is initially aimed at the dependency tree structure, so its data processing and model components are more suitable for the dependency tree structure and less adaptable to our graph structure. (2) Twitter is an online social comment data, its grammatical structure and verbal expression are relatively irregular. On Twitter, Lap14, Rest14 and MAMS datasets, the performance of AEGCN model was improved by 0.05%, 0.02%, 1.1% and 0.15%, respectively. We also use our model to experiment on the dependency tree. The experimental results show that our graph combined with our model has better performance. On Twitter, Lap14, Rest14 and MAMS datasets, the performance of our model was improved by 1.07%, 0.26%, 1.05% and 1.02%, respectively. From these comparative experiments, it can be seen that the accuracy of sentiment analysis is improved using our graph structure.

### Impact of GCN layer number

To investigate the impact of number of GCN layer, we set the number of GCN layer from one to five to evaluate our model on four datasets, respectively.

From Figs. 7, 8, 9 and 10, we can see with the increase of GCN layers, the performance of the model increases first and then decreases. Thus, the performance of the model does not always get improved with the increasing number of layers.

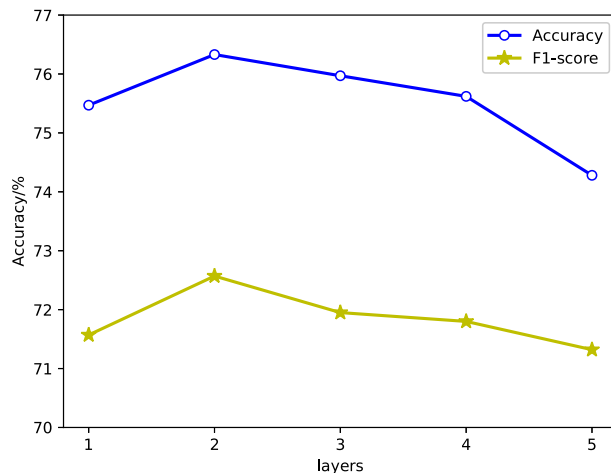


Fig. 7 Impact of GCN layer number on the Lap14

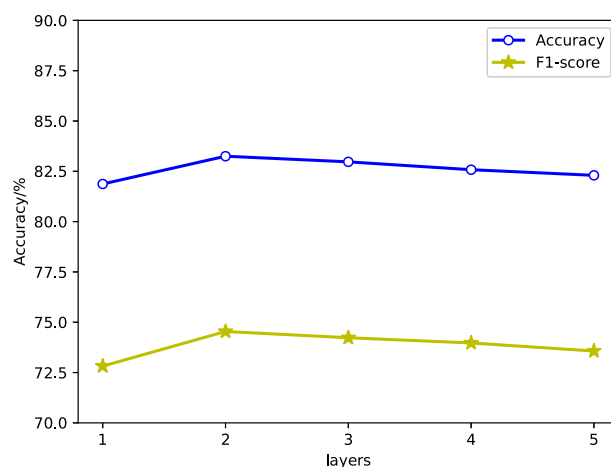
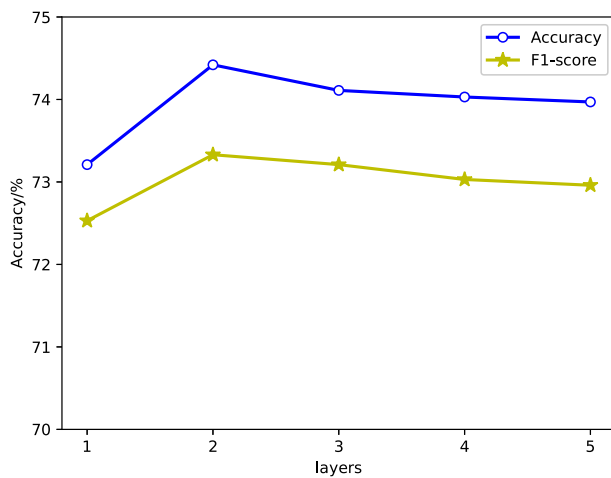
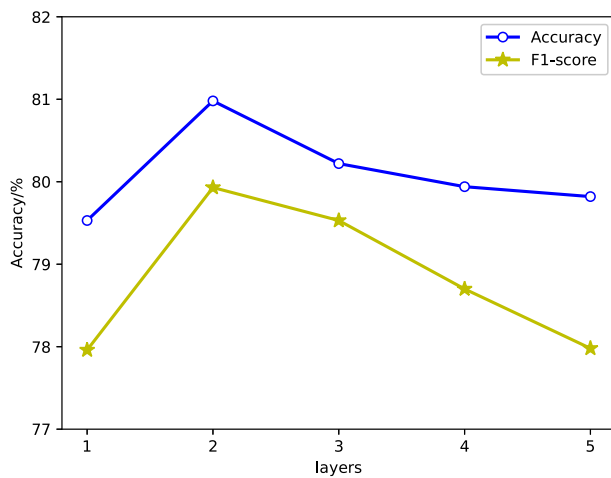


Fig. 8 Impact of GCN layer number on the Rest14

This is because a large layer number makes it hard to train the model. Moreover, a larger layer number introduces more parameters and results in a less generalizable model. To avoid problems caused by large layers, the best GCN with two layers is applied to train the model.



**Fig. 9** Impact of GCN layer number on the Twitter



**Fig. 10** Impact of GCN layer number on the MAMS

## Conclusion

In this paper, we propose a matrix fusion-based graph convolution network (MFLGCN) over multiple latent information graph structures to solve aspect-based sentiment analysis tasks. The learned graph structure combines semantic, statistic, and part of speech information to incorporate more latent information into the graph structure. MFLGCN can generate efficient and informative word coding. Experiments show that our graph structure leads to a more efficient node representation. Comprehensive experiments illustrate the effectiveness of our model. The proposed model outperforms the baseline models on four public datasets: Twitter, Lap14, Rest14, and MAMS. In addition, we performed ablation experiments on our model to prove the indispensable and effectiveness of each component of our model.

**Acknowledgements** Funding was provided by Key Research and Development Projects of Shaanxi Province (Grant Nos. 2020ZDLGY09-05).

## Declarations

**Conflict of interest** On behalf of all the authors, the corresponding author states that there is no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. 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
2. 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, London, pp 197–206
3. Li X, Bing L, Lam W, Shi B (2018) Transformation networks for target-oriented sentiment classification. Preprint [arXiv:1805.01086](https://arxiv.org/abs/1805.01086)
4. Tang D, Qin B, Liu T (2016) Aspect level sentiment classification with deep memory network. Preprint [arXiv:1605.08900](https://arxiv.org/abs/1605.08900)
5. Chen P, Sun Z, Bing L, Yang W (2017) Recurrent attention network on memory for aspect sentiment analysis. In: Proceedings of the 2017 conference on empirical methods in natural language processing, pp 452–461
6. Ma D, Li S, Zhang X, Wang H (2017) Interactive attention networks for aspect-level sentiment classification. Preprint [arXiv:1709.00893](https://arxiv.org/abs/1709.00893)
7. Fan F, Feng Y, Zhao D (2018) Multi-grained attention network for aspect-level sentiment classification. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 3433–3442
8. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. Preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)
9. Zhang C, Li Q, Song D (2019) Aspect-based sentiment classification with aspect-specific graph convolutional networks. Preprint [arXiv:1909.03477](https://arxiv.org/abs/1909.03477)
10. Sun K, Zhang R, Mensah S, Mao Y, Liu X (2019) Aspect-level sentiment analysis via convolution over dependency tree. 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 5679–5688
11. Huang B, Carley KM (2019) Syntax-aware aspect level sentiment classification with graph attention networks. Preprint [arXiv:1909.02606](https://arxiv.org/abs/1909.02606)

12. Xiao L, Xiaohui H, Chen Y, Xue Y, Donghong G, Chen B, Zhang T (2020) Targeted sentiment classification based on attentional encoding and graph convolutional networks. *Appl Sci* 10(3):957
13. Li R, Chen H, Feng F, Ma Z, Wang X, Hovy E (2021) Dual graph convolutional networks for aspect-based sentiment analysis. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers), pp 6319–6329
14. Zhang M, Qian T (2020) Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 3540–3549
15. Chen C, Teng Z, Zhang Y (2020) Inducing target-specific latent structures for aspect sentiment classification. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 5596–5607
16. Wang K, Shen W, Yang Y, Quan X, Wang R (2020) Relational graph attention network for aspect-based sentiment analysis. Preprint [arXiv:2004.12362](https://arxiv.org/abs/2004.12362)
17. Tang H, Ji D, Li C, Zhou Q (2020) Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 6578–6588
18. Tian Y, Chen G, Song Y (2021) Enhancing aspect-level sentiment analysis with word dependencies. In: Proceedings of the 16th conference of the European chapter of the association for computational linguistics: main volume, pp 3726–3739
19. Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 7370–7377
20. Zhou J, Huang JX, Qinmin Vivian H, He L (2020) SK-GCN: modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification. *Knowl Based Syst* 205:106292
21. Vashishth S, Bhandari M, Yadav P, Rai P, Bhattacharyya C, Talukdar P (2018) Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. Preprint [arXiv:1809.04283](https://arxiv.org/abs/1809.04283)
22. Liu X, You X, Zhang X, Ji W, Lv P (2020) Tensor graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8409–8416
23. Ren Z, Sun Q (2020) Simultaneous global and local graph structure preserving for multiple kernel clustering. *IEEE Trans Neural Netw Learn Syst* 32(5):1839–1851
24. Ren Z, Yang SX, Sun Q, Wang T (2020) Consensus affinity graph learning for multiple kernel clustering. *IEEE Trans Cybern* 51(6):3273–3284
25. Tan X, Cai Y, Zhu C (2019) Recognizing conflict opinions in aspect-level sentiment classification with dual attention 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 3426–3431
26. 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 (volume 2: short papers), pp 49–54
27. Pontiki DGM, Pavlopoulos HPJ, Androutsopoulos SMI (2014) Semeval-2014 task 4: Semeval-2014 task 4: aspect based sentiment analysis. In: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), Dublin, Ireland, pp 23–24
28. Jiang Q, Chen L, Xu R, Ao X, Yang M (2019) A challenge dataset and effective models 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)
29. Xue W, Li T (2018) Aspect based sentiment analysis with gated convolutional networks. Preprint [arXiv:1805.07043](https://arxiv.org/abs/1805.07043)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.