

Aspect-Guided Multi-Graph Convolutional Networks for Aspect-based Sentiment Analysis

Yong Wang¹, Ningchuang Yang^{1†}, Duoqian Miao², Qiuyi Chen¹

¹School of Artificial Intelligence, Chongqing University of Technology, 401135, Chongqing, China

²Department of Computer Science and Technology, Tongji University, 201804, Shanghai, China

Keywords: Graph convolutional networks; Aspect-based sentiment analysis; Multi-headed attention; BERT encoder

Citation: Wang Y., Yang N.C., Miao D.Q., et al.: Aspect-Guided Multi-Graph Convolutional Networks for Aspect-based Sentiment Analysis. *Data Intelligence* 6(3), 771-791 (2024). doi: 10.3724/2096-7004.di.2024.0052

Submitted: January 10, 2023; Revised: June 15, 2023; Accepted: December 12, 2023

ABSTRACT

The Aspect-Based Sentiment Analysis (ABSA) task is designed to judge the sentiment polarity of a particular aspect in a review. Recent studies have proved that GCN can capture syntactic and semantic features from dependency graphs generated by dependency trees and semantic graphs generated by Multi-headed self-attention (MHSA). However, these approaches do not highlight the sentiment information associated with aspect in the syntactic and semantic graphs. We propose the Aspect-Guided Multi-Graph Convolutional Networks (AGGCN) for Aspect-Based Sentiment Classification. Specifically, we reconstruct two kinds of graphs, changing the weight of the dependency graph by distance from aspect and improving the semantic graph by Aspect-guided MHSA. For interactive learning of syntax and semantics, we dynamically fuse syntactic and semantic diagrams to generate syntactic-semantic graphs to learn emotional features jointly. In addition, Multi-dropout is added to solve the overfitting of AGGCN in training. The experimental results on extensive datasets show that our model AGGCN achieves particularly advanced results and validates the effectiveness of the model.

1. INTRODUCTION

Sentiment analysis has become a hot topic in natural language processing. ABSA is a fine-grained sentiment classification task that aiming to determine the sentiment polarity of a particular aspect in a review. The classification labels are POSITIVE, NEUTRAL and NEGATIVE. For example, in the sentence,

[†] Corresponding author: Ningchuang Yang (E-mail: yangningchuang@163.com; ORCID: 0000-0001-9796-2801).

“The price is reasonable although the service is poor.” the sentiment polarities for the two aspect terms “price” and “service” are positive and negative, respectively.

Early work on this task was generally rule-based and statistical methods [1-3]. Some predefined rules are matched to determine the polarity of aspect. For traditional machine learning methods, feature engineering such as sentiment table, bag-of-words is established, and then traditional machine algorithms are used to train classifiers such as SVM and the Naive Bayes model. This approach requires a significant amount of time to collect and analyze data, and then design features. Therefore, traditional machine learning methods have poor permeability, and feature engineering is labor-intensive, almost reaching a bottleneck.

Intuitively, the primary purpose of ABSA is to determine the relationship between aspects and other emotional words. With the emergence of the attention mechanism, it seems to be able to solve this problem. Recent studies [4-9] have exploited various attention mechanisms to extract the relationship between aspect and sentiment words, and achieved good results. However, due to the complexity of sentences, the attention mechanism does not always capture the desired results. For example, “it has a bad memory but a great battery life.” in which the words “battery life” focus more on “bad” rather than “great”, so the syntactic information is important for the task.

Now more and more researchers try to add syntactic dependency information to the model so that the model can learn syntactic features [10, 11]. At first people designed syntactic dependency information by hand, and then used parse dependency trees that to provide more comprehensive information. The parsing dependency tree is usually converted into a neighbor matrix, and then a graph convolutional networks (GCN) [12-14] and a graph attention network (GAT) [15-18] are used to train to better encode the relationship between words. These methods can better encode word-word relationships based on syntactic dependency trees. but the limitations cannot be ignored. Because of the instability of the dependency tree, which is influenced by parsing dependents with varying quality, it introduces some noise to the model. So some models propose to combine MHSA, Transformers [19] or BERT [29] with learning together [20, 21] to assist syntactic learning and weaken the noise.

Most existing methods use syntactic and semantic information for ABSA, but ignore the important of aspect, mainly in the following two problems. The first one is that the dependency graph has only two cases of values 0 and 1, which does not reflect the difference in importance between words. The other one is not paying much attention to aspect in MHSA. Our AGGCN is designed to extract syntactic and semantic information that is more suitable for ABSA. Firstly, in order to strengthen the focus on words that are close to the aspect, we modify the values in the dependency graph, when the closer the distance to the aspect the larger the value, and conversely, the smaller the value. Secondly, Previous studies went through MHSA to solve the noise inherent in dependency trees. We enriched the MHSA by adding aspect information to the attention calculation, which helps to capture context relevant information. Then, for the interaction of syntax and semantics, we dynamically fuse syntactic and semantic graphs. Finally, the features in the graph are captured by GCN. In addition, we find that AGGCN has overfitting on the training set, so we add the Multi-dropout [22] module to solve it. Extensive experiments are performed

on SemEavl 2014, Twitter datasets, and the experimental results demonstrate that our model AGGCN achieves excellent performance.

The main contributions of this paper are as follows:

- We propose to improve the values in the dependency graph based on the distance relationship between aspect and other. The closer the distance to the aspect word gets more attention and, on the contrary, gets less attention.
- We propose an Aspect-guided MHSA module that embeds aspect information in the attention computation so that the sentiment information associated with the aspect receives a larger attention score.
- To address the instability of dependency trees and the shortcomings of attention mechanisms, we dynamically integrate syntactic and semantic graphs so that they assist each other in interactive learning.
- For the overfitting of AGGCN on the training set, which in turn leads to poor results on the validation set, we introduce the Multi-dropout technique of computer vision to reduce the overfitting by Multi-dropout.

2. RELATION WORK

In this section, deep learning approaches for aspect-based sentiment classification will be introduced. It mainly includes the following parts: 1) Based on the attention mechanism. 2) Based on graph neural network. 3) Pretrained model.

2.1 Attention-Based Neural Network

The core of the ABSA task is to capture emotional features related to aspect, so the emergence of attention makes this task a good achievement. Wang et al [4]. first proposed to combine LSTM and attention mechanism to extract emotional features, while using each word and aspect concatenation as input. Ma et al [5]. first proposed that aspect need to be modeled separately, using an interactive attention network to generate representations of aspects and contexts separately. Chen et al [7]. adopted multiple-attention to strengthen expressive power, adding memory weights to learn more points. Fan et al [6]. exploited fine-grained and coarse-grained attention mechanisms for the interaction between context and aspect. Li et al [8]. used LSTM to embed aspect into each word, and finally uses convolutional neural networks (CNN) to extract feature representations. Song et al [9]. exploited an attention-encoding network to encode aspect and context, and then uses attention to interact to obtain representations.

2.2 Syntactic-Based Neural Network

In the early days, some researchers used the artificial definition of syntactic rules to add syntactic information to improve the model's ability to capture syntax [10, 11]. With the emergence of the

dependency tree, Dong et al [23]. used the syntactic relationship of the sentence to adaptively propagate the emotion of the word to the specific aspect. Nguyen et al [24]. used a simpler method, using a combination of dependency tree and composition tree for further coding. He et al [25]. proposed to use the tree node distance to calculate the attention weight.

Zhang et al [12]. used GCN [26] to extract syntactic information through a dependency graph generated by a dependency tree. Sun et al [13]. proposed to use GCN model to extract aspect-specific features learned from Bi-directional LSTM(BI-LSTM) through the dependency tree of sentences. Huang et al [18]. proposed to capture syntactic features using GAT and LSTM. Huang et al [17]. combined GAT and BERT to encode aspect and context together. Wang et al [15]. proposed a new RGAT to extract the syntactic information of sentences. Tian et al [14]. proposed to embed relation information in word vectors, and then use GCN to extract features from self-attention graph generated from dependency graph. Tang et al [20]. used the Dule-transform structure to separate the syntax and semantics of the dependency tree and learn information from each other. Li et al [21]. proposed to use two GCNs to deconvolve the dependency graph and the semantic graph generated by MHSA respectively.

2.3 Pretrained model

Traditional deep learning obtains word vector embedding through Word2vec [27] and Glove [28]. Recently Pretrained models BERT [29], ALBERT [30], Roberta [31] and DeBERTa [32] have achieved good accuracy on NLP classification tasks. Song et al [9]. utilized BERT to encode context and aspect separately, and then uses attention to interact. Song et al [33]. used [CLS]+Context+[SEP]+Aspect+[SEP] as BERT input and [CLS] as final emotion representation. Xu et al [34]. trained BERT using a corpus-specific semantic library and uses the resulting model for sentiment classification.

3. PROPOSED METHODOLOGY

This section will introduce the details of the AGGCN model in detail. The main framework of the AGGCN model is shown in Figure 1. The core idea of AGGCN is to change the syntactic and semantic graphs to help GCN capture the accurate features to help sentiment classification. Firstly, the semantic graph is obtained by adding the distance information of aspect to other words in the dependency graph. Secondly, by incorporating aspect word information in Q and K , the aspect-based score matrix is obtained as a semantic graph. Thirdly, in order to make the syntactic and semantic information learn interactively, the semantic and syntactic graphs are dynamically fused to obtain the syntactic-semantic graph.

3.1 Input embedding layer

In the ABSA task, a sentence $w_c = \{w_c^1, w_c^2, \dots, w_c^n\}$ sequence and aspect $w_a = \{w_a^1, w_a^2, \dots, w_a^m\}$ sequence are given, where n and m are sentence length and aspect length, and W_a is a subsequence of W_c . We use

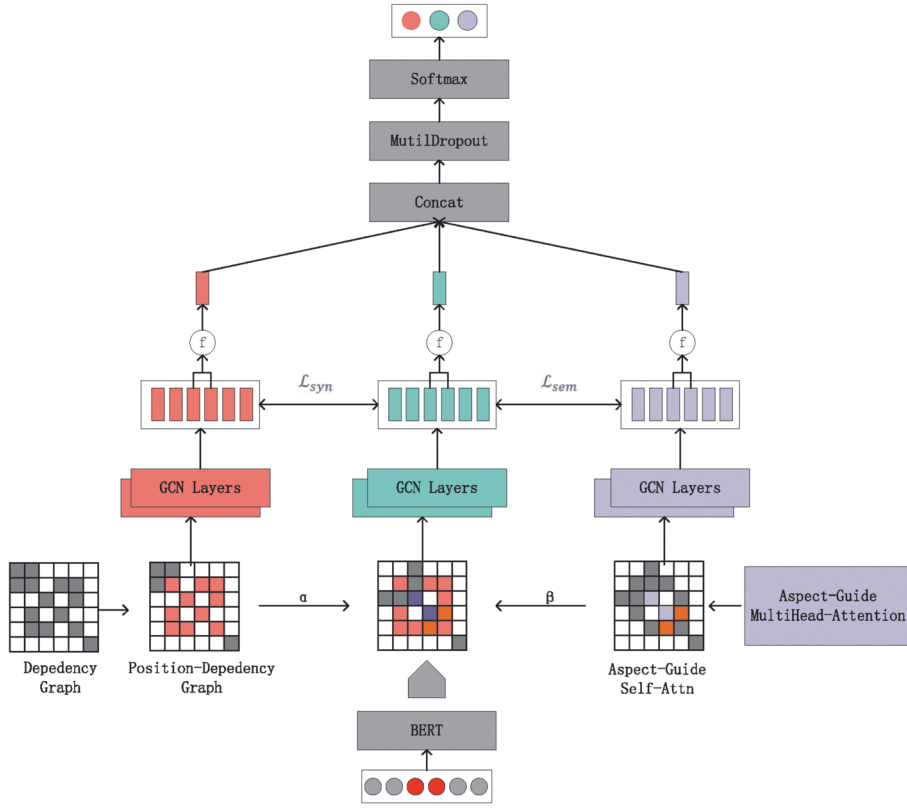


Figure 1. Framework of Aspect-Guided Multi-Graph Convolutional Networks.

BERT to project a high-latitude dimension to a low-latitude space. BERT is a Pretrained model based on Transformers. Most NLP tasks have proved that BERT is effective in improving model effect. To accord with the input format of BERT, we reconstruct the sequence to follow: “[CLS]” + Context + “[SEP]” + Aspect + “[SEP]”. It is expressed as follows:

$$W = \{[CLS], w_c^1, w_c^2, \dots, w_c^n, [SEP], w_a^1, w_a^2, \dots, w_a^m, [SEP]\} \quad (1)$$

Where “[CLS]” and “[SEP]” are special characters for BERT. A low-latitude representation learned by BERT is as follows:

$$H^{BERT} = \{h_{cls}, h_1, \dots, h_n, h_{sep}, h_{n+2}, \dots, h_{n+m+1}, h_{sep}\} \quad (2)$$

Where $H^{BERT} \in R^{n \cdot d}$, d is the dimension of the word embedding, $H = \{h_1, h_2, \dots, h_n\}$ as a vector representation of the context.

3.2 Aspect-guided syntactic graph

In most of the existing graph neural network methods, the syntactic information is extracted based on the dependency graph. The original dependency graph is generated from a dependency tree with the following generation process:

$$D_{i,j} = \begin{cases} 1 & \text{if } i = j \text{ or } w_i, w_j \text{ in the dependency tree} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

According to the Equation 3, it can be seen that $D_{i,j} = 1$ for the dependent relationship, otherwise $D_{i,j} = 0$. However, the dependency graph in this way can only represent two values of 0 or 1 and cannot highlight the important relationships in the graph that are close to the aspect. We construct the dependency graph according to the distance relationship, and the main idea is to emphasize words that are close to the aspect, shown in Figure 2. The reconstruction method is as follows:

$$A_{i,j}^{syn} = \begin{cases} D_{i,j} + 1, & \text{if } w_i \in a^s \text{ and } w_j \in W_a \\ D_{i,j} + \frac{1}{|p_i - p_j| + 1}, & \text{if } w_i \in W_a \\ D_{i,j} + \frac{1}{|p_j - p_i| + 1}, & \text{if } w_j \in W_a \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

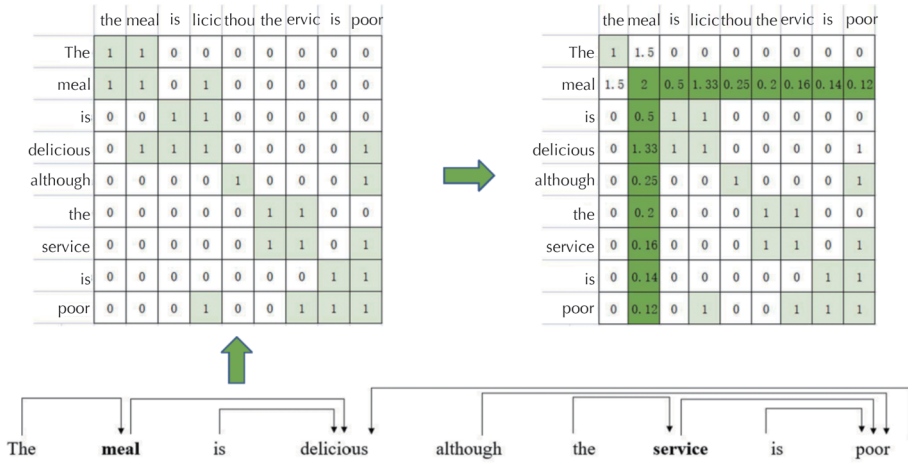


Figure 2. The figure illustrates how the dependency tree becomes an aspect guided syntactic graph.

Where $||$ is an absolute value function, W_a is a set of aspect, P_i and P_j are the position of the word. By reconstructing the dependency graph, we can highlight the importance of aspect in context.

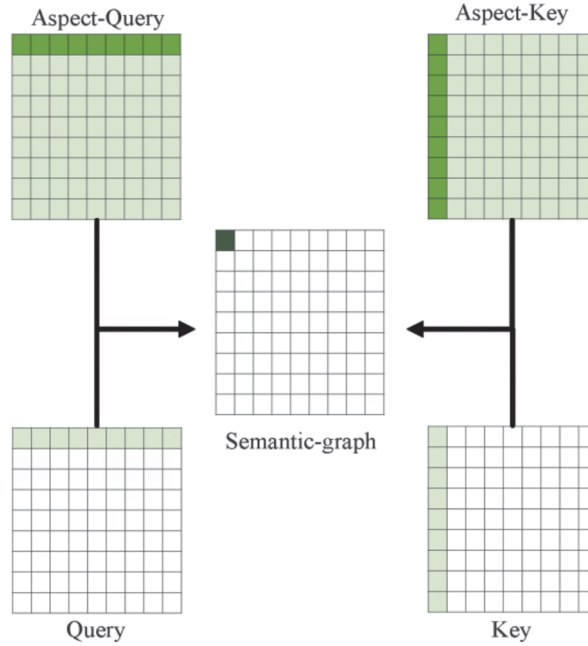


Figure 3. The figure illustrates how the attention mechanism becomes an aspect-guided semantic graph.

3.3 Aspect-guided semantic graph

In the first part, we learned the syntactic relations by aspect guided syntactic graph, but the instability of the dependency tree will bring noise to the model. Most of the existing studies use other semantic information to assist learning in order to weaken the noise of the dependency tree. For example, the attention score matrix generated by MHSA as the semantic graph and use GCN to learn the semantic information. The original attention score matrix is obtained from Q and K by operation, as shown in Equation 6. The original Q and K are determined by the context, as shown Equation 5.

$$\begin{bmatrix} Q^h \\ K^h \end{bmatrix} = H \begin{bmatrix} W_Q^h \\ W_K^h \end{bmatrix} \quad (5)$$

$$A^h = \text{softmax} \left(\frac{Q^h \times K^{hT}}{\sqrt{d/h}} \right) \quad (6)$$

Where Q^h and K^h are the query and key matrices by h head, d and h are word dimension and head count.

Although A can learn certain semantic relations, it does not consider the importance of aspect in sentences. In this section we propose an Aspect-guided MHSA that is more suitable for ABSA tasks than MHSA. The $\hat{Q}^h$ and $\hat{K}^h$ of Aspect-guided MHSA by using a gate control to retain important context

information and add aspect information. The attention score of Aspect-guided MHSA gives greater scores for information related to aspect. For Aspect-guided MHSA, firstly, the maximum pooling operation is performed on aspect and then embedded into each word of the context, as shown in Equation 7.

$$C = (W^C [H; H_{pool}^a] + b^C) + H \quad (7)$$

Where $C \in R^{n \times d}$ is a context vector incorporating aspect information, $H_{pool}^a \in R^{n \times d}$ is the average pooling of aspect and then flattened to sentence length. Secondly, a gated scalar $\{\lambda_Q^h, \lambda_K^h\}$ is set to control the contribution of each expression and aspect to the attention weight, as shown in Equation. 8.

$$\begin{bmatrix} \lambda_Q^h \\ \lambda_K^h \end{bmatrix} = \sigma \left(\begin{bmatrix} Q^h \\ K^h \end{bmatrix} \begin{bmatrix} V_Q^h \\ V_K^h \end{bmatrix} + C^h \begin{bmatrix} U_Q \\ U_K \end{bmatrix} \begin{bmatrix} V_Q^C \\ V_K^C \end{bmatrix} \right) \quad (8)$$

Where σ denotes the logistic *Sigmoid* function. $\{U_Q, U_K\} \in R^{d/h \times 1}$ are learnable transition matrices. $\{V_Q^h, V_K^h\} \in R^{d/h \times 1}$ and $\{V_Q^C, V_K^C\} \in R^{d/h \times 1}$ are trainable parameters.

$$\begin{bmatrix} \hat{Q}^h \\ \hat{K}^h \end{bmatrix} = \left(1 - \begin{bmatrix} \lambda_Q^h \\ \lambda_K^h \end{bmatrix} \right) \begin{bmatrix} Q^h \\ K^h \end{bmatrix} + \begin{bmatrix} \lambda_Q^h \\ \lambda_K^h \end{bmatrix} \left(C^h \begin{bmatrix} U_Q \\ U_K \end{bmatrix} \right) \quad (9)$$

Thirdly, the $\{\hat{Q}^h, \hat{K}^h\}$ that incorporates aspect information considers aspect information when performing attention calculations. This approach highlights the status of the aspect and gives greater score to information related to the aspect. The improved Aspect-guided MHSA is as follows:

$$A_h^{sem} = softmax \left(\frac{\hat{Q}_h^a \times (\hat{K}_h^a)^T}{\sqrt{\frac{d}{h}}} \right) \quad (10)$$

3.4 Fuse semantic-syntactic graph

Both dependency tree based aspect-guided syntactic graph and attention based aspect-guided semantic graph has their inherent drawbacks. Therefore, in order to learn accurate sentiment information, we want these two parts to complement each other's learning and enhance their stability. So we set a set of dynamic parameters α and β to control the degree of integration of the aspect-guided syntactic graph and the aspect-guided semantic graph.

$$A^{com} = \frac{\alpha * A^{sem} + \beta * A^{syn}}{2} \quad (11)$$

3.5 Graph convolutional network

The GCN can capture features in graph data. In this section, we use GCN to perform feature extraction on the graphs obtained in the first three sections. Thus the three graphs $SYNGCN(A^{syn})$, $SEMGCN(A^{sem})$ and $COMGCN(A^{com})$ are extracted by GCN in the graph dependencies.

$$H^{syn} = SYNGCN(A^{syn}, H, W^{syn}) \quad (12)$$

$$H^{sem} = SEMGCN(A^{sem}, H, W^{sem}) \quad (13)$$

$$H^{com} = COMGCN(A^{com}, H, W^{com}) \quad (14)$$

Where H represents the context vector representation obtained by the embedding layer, W^{syn} , W^{sem} and W^{com} are trainable parameters.

3.6 Multi-dropout with classification

Finally we use average pooling to process aspect in *SYNGCN*, *SEMGCN* and *COMGCN* outputs. To better integrate the three parts to learn the most critical features, we concatenate the three parts and then input the MLP and activate it to obtain the final expression.

$$H_a^{syn} = Avgpool(mask * H^{syn}) \quad (15)$$

$$H_a^{sem} = Avgpool(mask * H^{sem}) \quad (16)$$

$$H_a^{com} = Avgpool(mask * H^{com}) \quad (17)$$

$$mask = \begin{cases} 0, & 1 \leq t < \tau + 1, \tau + m < t < n \\ 1, & \tau + 1 \leq t \leq \tau + m \end{cases} \quad (18)$$

Where *Avgpool* is average pooling and *mask* is an aspect sequence. $\tau + 1 \leq t \leq \tau + m$ is the index of the aspect.

$$H^c = [H_a^{syn}, H_a^{sem}, H_a^{com}] \quad (19)$$

$$H^f = Relu(H^c W^f + b^f) \quad (20)$$

W^f and b^f are learnable parameters and bias.

Multi-dropout is also a variant of dropout technology as Figure 4. Traditional dropout randomly selects a set of samples from the input at each training epoch. The multi-dropout creates multiple samples, and each sample uses different dropout values to generate different losses. In which we adopt parameter sharing at the fully connected layer. Finally, the average sum of each loss will be lower than the loss obtained by any single dropout.

$$H_d^f = Dropout_d(H^f) \quad (21)$$

We get the final representation H_d^f , which is mapped to the most probable sentiment polarity by a fully connected layer network.

$$P_d(a) = \text{softmax}(H_d^f W^f + b^f) \quad (22)$$

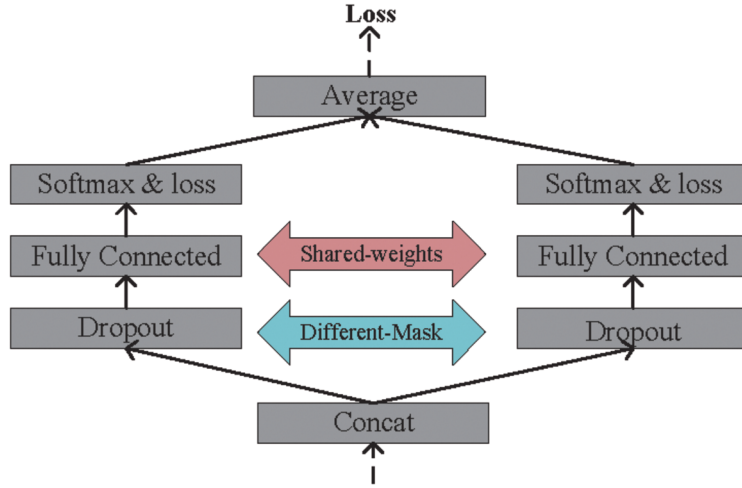


Figure 4. Framework of multi-dropout. The text is processed with different dropout values, and four values 0.1, 0.2, 0.3, 0.4 are chosen in this paper. The full connection of all paths is weight sharing. Calculate the average sum of all losses at the end.

Where W^f and b^f are learnable weights and biases, which are shared by several other dropout channels. Several different classification results are obtained by pooling different dropouts. We sum and average the *softmax* results and losses of different dropout channels to obtain the final result and loss.

$$P(a) = \frac{\sum_d^{n_d} p_d(a)}{n_d} \quad (23)$$

$$\ell_c = \frac{-\sum_{d=1}^{n_d} \sum_{(s,a) \in D} \sum_{c \in C} \log p(a)_d}{n_d} \quad (24)$$

Where D represents all sentence and aspect pairs in the training datasets, C stands for emotional polarity set. ℓ_c is a standard cross-entropy loss. N represents the number of different dropouts.

3.7 Loss Function

In order for H^{syn} , H^{sem} and H^{com} learn a more accurate feature representation and there should not be too much difference, we use ℓ_{syn} and ℓ_{sem} to enhance their similarity.

$$\ell_{syn} = \|H^{com} - H^{syn}\|_{l_2}^2 \quad (25)$$

$$\ell_{sem} = \|H^{com} - H^{sem}\|_{l_2}^2 \quad (26)$$

The overall loss function is as follows:

$$\ell_T = \ell_C + \lambda_1 \ell_{syn} + \lambda_2 \ell_{sem} + \lambda_3 \|\Theta\|_2 \quad (27)$$

Where ℓ_C is the cross-entropy loss function obtained by different dropout as Formula 24; Θ represents all the trainable parameters of the model.

4. EXPERIMENT

4.1 Datasets

Our extensive experiments are conducted on three datasets, including Restaurant and Laptop datasets from SemEval2014 [35], and another Twitter datasets which is initially constructed by (Adaptive recursive neural network for target-dependent twitter sentiment classification [36]. The datasets labels are divided into three types (POSITIVE, NEUTRAL, NEGATIVE). The specific information of each dataset is shown in the Table 1.

Table 1. Detailed statistics of the three datasets used in our experiment.

Dataset	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	727	807	196	637	196
Laptop	976	337	851	128	455	167
Twitter	1507	172	1528	169	3016	336

4.2 Setting

For our experiment, the batch size is set to 16. We use the Pre-trained model BERT (Bert-base-uncased) to fine-tune, which contains 12 attention heads and each hidden layer is 768 dimensions. The BERT dropout is 0.1, and the regularization term is 10^{-5} . The dependency tree is generated by Deep Biaffine Parser [37] to get syntactic information. For Aspect-guided MHSA, we set 2 heads, and the dropout rate is 0.1. We uniformly set the three GCN layers to 2 layers and dropout rate of 0.4. Multi-dropout sets 4 values (0.1, 0.2, 0.3, 0.4). The Adam optimizer [38] is used to train the model with a learning rate of 2×10^{-5} .

4.3. Baseline Methods

We compare our model with models without syntactic information and with syntactic dependencies:

- IAN [5] proposes to encode context and aspect separately and use attention to interact with each other to get aspect context representation to classify.

- RAM [7] uses gating units to control the multi-attention mechanism and capture the relation of aspect to each word. Then the sentence representations are learned for classification.
- TNet [8] uses BiLSTM to transform specific aspects into each word, and finally uses CNN to extract features for classification.
- MGAN [6] uses BiLSTM to encode context information and devise a multi-granularity attention mechanism to capture the interaction of aspect and context.
- AEN [9] uses attention encoding to encode aspects and contexts separately and learn representations by interacting aspects and contexts with attention.
- AdaRNN [23] uses RNNs to learn dependency tree-based feature representations.
- TD-GAT [18] uses GAT to capture the syntax structure and uses LSTM to improve the syntax structure to model cross-layer relationships.
- ASGCN [12] proposes to use a graph convolutional network to extract the syntactic relationship of dependency trees and combine the attention mechanism for sentiment classification.
- CDT [13] uses graph convolutional networks to extract syntactic information of dependency trees, and then pooled aspect for sentiment classification.
- RGAT [15] uses GAT to extract the syntactic information of the dependency tree while considering the dependencies between words and pruning the useless information in the tree.
- DGEDT [20] uses the dual-transformer structure to learn feature representations using self-attention encoding and GCN networks respectively.
- TGCN [14] proposes a graph convolutional network that explicitly uses dependency types, T-GCN uses an attention mechanism to distinguish different edges in the graph, and proposes attention layer integration to synthetically learn different layers of T-GCN.
- LD2G [39] Proposed domain-adaptive Pre-training using Multi-task learning, combined with Biaffine Attention and Mask Language Model.
- MWGCN [40] proposes a multi-weight graph convolutional network that extracts context features that integrate syntactic information and captures aspect features that focus on local context.
- DMFGAT [41] proposes to use BERT to mitigate the damage of dependency tree information on the model and to integrate syntactic information learned by GAT using an attention mechanism.
- AG-VSR [42] design of an attention-assisted graph representation and variational sentence representation for classification.
- GCN-BERT uses GCN to extract the syntactic relations of the dependency tree and uses BERT as the word embedding method of the model.

4.4 Main results

To verify the effect of AGGCN, the experiments are mainly carried out on three datasets Res14, Lap14 and Twitter, and the evaluation standards are the recognized accuracy and Macro-F1. the main experimental results are shown in Table 2. The baseline models are divided into attention mechanism, syntactic dependency, and BERT. Our AGGCN-BERT achieved the best results on Restaurant14, with an increase of 4.4 and 0.61 compared to the least effective TD-GAT-BERT and to the best effective AKM-IGCN-BERT, respectively. On the Laptop14 dataset, there is an increase of 2.73 and 0.05 compared to the

least effective RGAT-BERT and the best TGCN-BERT, respectively. On Twitter dataset, AGGCN achieved the second best results, after DGEDT-BERT, but somewhat ahead of the other models.

Table 2. Overall performance of accuracy and F1 on three datasets.

Category Model		Restaurant		Laptop		Twitter	
		Acc	F1	Acc	F1	Acc	F1
Attention	IAN	78.60		72.10			
	RAM	78.60		72.10			
	TNet	80.79		76.54		73.12	
	MGAN	81.25	71.94	75.39	72.47	72.54	70.81
	AEN	80.98	72.14	73.15	69.04	72.83	69.81
Syntax	AdaRNN	-	-		-	60.29	65.90
	ASGCN	80.77	72.02	75.55	71.05	72.15	70.40
	CDT	82.30	74.02	77.19	72.99	74.66	73.66
	TDGAT	81.20	-	74.0	-		
	RGAT	83.30	76.08	77.42	73.76	75.57	73.82
	DGEDT	83.90	75.10	76.80	72.30	74.80	73.30
BERT+Syntax	TD-GAT-BERT	83.00		80.10			
	RGAT-BERT	86.60	81.35	78.21	74.07	76.15	74.88
	DGEDT-BERT	86.30	80.00	79.80	75.60	77.90	75.40
	LD2G-BERT	85.98	79.45	80.56	76.61		
	MWGCN-BERT	86.36	80.54	79.78	76.68	75.00	74.03
	TGCN-BERT	86.16	79.95	80.88	77.03	76.45	75.25
	AG-VSR-BERT	86.34	80.88	79.92	75.85	76.45	75.04
	DMF-GAT-BERT	86.10	80.17	80.38	77.20	76.22	75.10
Ours	GCN-BERT	86.04	79.07	78.59	76.07	75.26	74.11
	AGGCN-BERT	87.40	81.39	80.94	77.43	77.16	76.00

As seen in Table 2, compared to the attention-based model, our AGGCN designs an Aspect-guided MHSA that makes the attention scores more biased towards words related to aspect. In addition, the score matrix is used as a semantic graph and its features are learned using GCN. However, due to the complexity of the sentences and the distance relationship, the attention mechanism in many cases is unable to capture the correct sentiment information, leading to classification failure. With the ability to generate dependency trees based on parsing dependents, all GNN-based models have a substantial improvement in metrics on all datasets. GNN can capture sentiment information based on the syntactic relationships provided by the dependency trees.

Compared to models based on syntactic dependencies, AGGCN designed the aspect-guided syntactic graph based on the dependency tree, which can highlight the importance of aspect, centered on aspect, the closer the more important. But the dependency tree is not always valid, and its quality is influenced by the parser. The AGGCN design aspect-guided semantic graph can provide semantic information to effectively weaken the errors of the dependency tree. In addition, the com-graph can learn the interaction information of semantic and syntactic information effectively, so our AGGCN can learn the accurate syntactic and semantic information.

It is worth noting that all models have a substantial increase in ACC and F1 after utilizing BERT as an embedding layer. This is due to the fact that BERT is able to provide richer semantic representations and it has been shown that BERT is also able to provide implicit syntactic information. Our AGGCN can obtain rich semantic representations using BERT and can weaken the errors of dependency trees, achieving the best results in Restaurant14 and Laptop14.

4.5 Classification statistics

To judge how well the model classified each sentiment category, we count the number of successful classifications for each category in the validation set. It can be seen from Figure 5 that Positive has the most success in classification, while the results of Negative and Neutral are not very satisfactory.

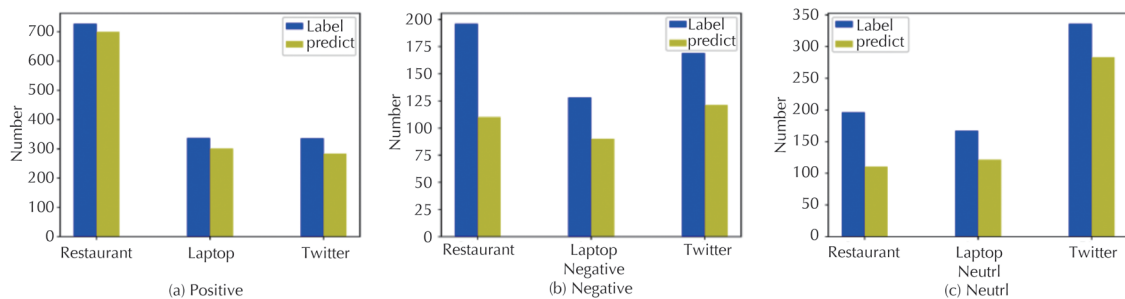


Figure 5. Three sentiment polarity classification results demonstrates that our model is more successful for positive while the effect of Negative and neural is less.

4.6 Ablation study

We will further explore the effect of each module in the model on the experimental results. Without changing all module parameters, the effect is discussed by removing only one module at a time. The experiments are divided into the following:

- AGGCN w/o Multi-dropout is to remove Multi-dropout from AGGCN and directly perform classification and loss calculation.
- AGGCN w/o ℓ_{syn} and ℓ_{sem} is the removal of L and L from the original model.
- AGGCN w/o Aspect-guided MHSA is the original MHSA to calculate the score graph.
- AGGCN w/o H^{syn} is the removal of H^{syn} from the original model.
- AGGCN w/o H^{sem} is the removal of H^{sem} from the original model.
- AGGCN w/o H^{com} is the removal of H^{com} from the original model.

The results of the ablation experiment are shown in Table 3. As a whole, the effectiveness metrics of the model decreased to some extent when any module was removed. In experiment 1, the Multi-dropout is used to weaken the training overfitting. After the Multi-dropout module is removed, the accuracy and F1 drop by more than 1 percent in three datasets, which indicates that the Multi-dropout is effective. In experiment 2, the effect of removing ℓ_{syn} and ℓ_{sem} is obvious, We speculate that it is because the gap between the values of A_h^{sem} and A_h^{syn} is too large, resulting in a gap in the learned features, so adding ℓ_{syn} and ℓ_{sem} can effectively enhance their similarity. In experiment 3, the original MHSA does not consider aspect information when computing information between words, and experiments demonstrate that aspect can enhance the ability to model the relationship between aspect and context, so the modeling effect decreases when the Aspect-guided MHSA is removed. H^{syn} , H^{sem} and H^{com} are to learn the accurate sentiment representation, where H^{syn} is based on syntactic information, H^{sem} is based on semantic information and H^{com} is based on semantic-syntactic interactions. In removing any of them, there is a clear decrease in the effectiveness of the model, which is due to the fact that H^{syn} is able to provide important syntactic relations to help learn the accurate sentiment word information. However, the dependency tree is noisy, and H^{sem} and H^{com} are able to learn semantic information which can effectively reduce the noise of the dependency tree.

4.7 Effect of GCN Depth

To investigate the effect of the number of GCN layers on the model, we set the layers 1-10 to evaluate the model on three datasets. The experimental results are as Figure 6. When the number of layers is 2, the effect of the model is the best. As expected, when the number of layers is 1, the GCN is not sufficient to learn the exact syntactic dependencies of specific aspects. However, it is not that the deeper the depth, the better the effect. The effect is worse when starting from the third layer as the depth deepens. This means that as the depth increases, the number of parameters increases, and the model becomes more difficult to learn and overfitting.

Table 3. Results of ablation experiments.

Model	Restaurant		Laptop		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
AGGCN	87.40	81.39	80.94	77.43	77.16	76.00
AGGCN w/o Multi-dropout	85.43	77.68	79.69	76.36	75.55	72.65
AGGCN w/o ℓ_{syn} , ℓ_{sem}	85.88	80.13	80.29	76.69	74.24	71.10
AGGCN w/o aspect-guided MHSA	84.44	76.6	80.78	77.25	76.42	75.23
AGGCN w/o H^{com}	84.54	76.14	79.06	75.95	74.97	74.46
AGGCN w/o H^{syn}	86.68	81.6	79.21	75.83	75.55	74.87
AGGCN w/o H^{sem}	84.09	77.88	79.38	75.67	74.53	74.02

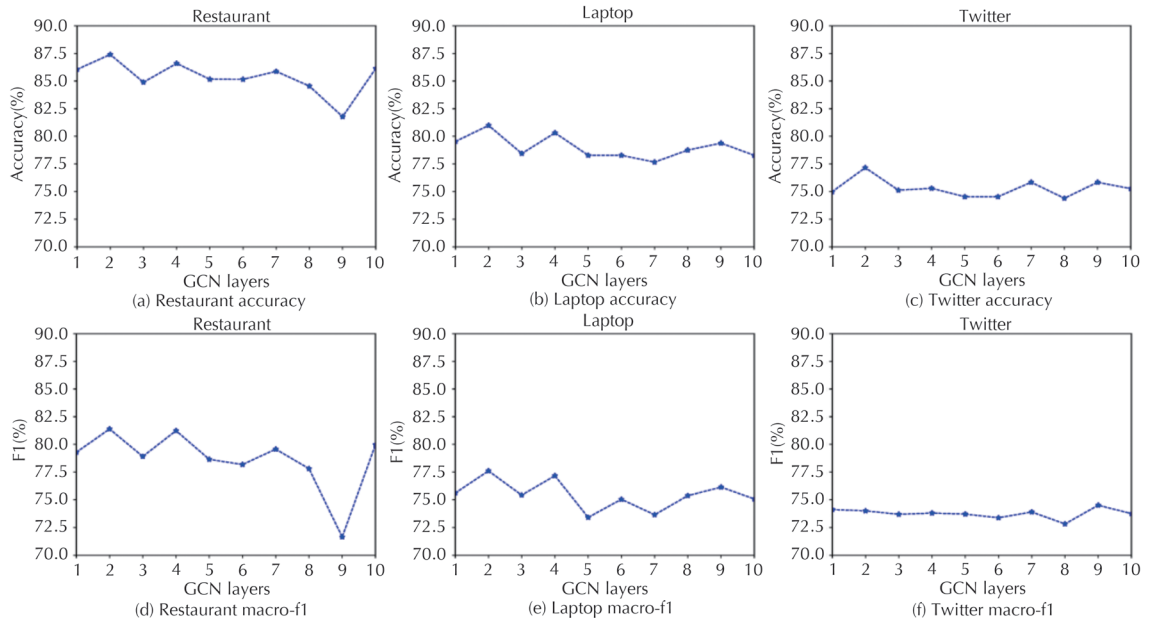


Figure 6. The influence of the number of GCN layers on the model accuracy and F1 value, mainly considering layers 1-10.

4.8 Effect of Attention Head

The number of heads in attention is an important parameter that affects the model's performance. The essence of the multi-headed is to divide it into multiple independent calculations and integrate them, so that more features can be learned, and overfitting can be prevented. This article explores the effects of 1-6 heads on the model. The experimental results are shown in Figure 7, and the best effect is achieved when it is equal to 2 heads. This also indicates that it is not that the more attention heads, the better. The attention heads, the greater the cumulative error, which will lead to worse model performance.

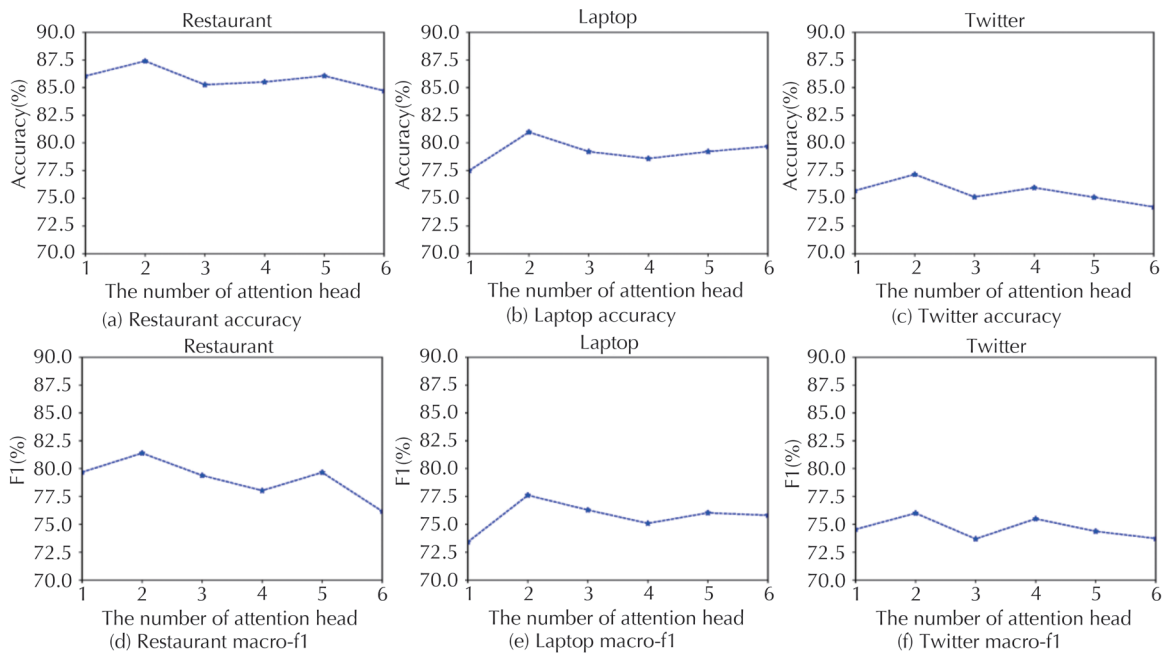


Figure 7. The influence of the number of attention heads on the accuracy of the model and F1, mainly considering 1-6 heads.

4.9 Error Analysis

We further investigate the errors from AGGCN, which can be divided into 3 types: 1) The sentiment polarity of aspect in sentences is unclear, and even annotators cannot determine the sentiment polarity. 2) Very few sentences have sentiment words far away from the aspect, causing our syntactic graph modified according to the distance to fail. 3) The captured word itself has multiple meanings, causing the model to fail to understand their meaning, e.g. "like", which has favor and similar meanings.

4.10 Case Study and Attention Distribution Exploration

To demonstrate how AGGCN can improve the performance of aspect-level sentiment analysis tasks, we analyze the attention visualization weights by three examples, as shown in Figure 8. In case1, “it is super fast and has outstanding graphics,” the aspect word “graphics” gets the highest attention weight, and its associated sentiment information “outstanding” and “fast” also have greater attention weight. In case2, “it has so much more speed and the screen is very sharp.” the aspect “screen” gets the most attention score. In case3, “biggest complaint is window 8.” the aspect “window 8” received the greatest weight, and its associated sentiment information “complaint” also received the second highest attention score after the aspect. This verifies that AGGCN can effectively capture aspects and their associated sentiment information to improve the accuracy of classification.

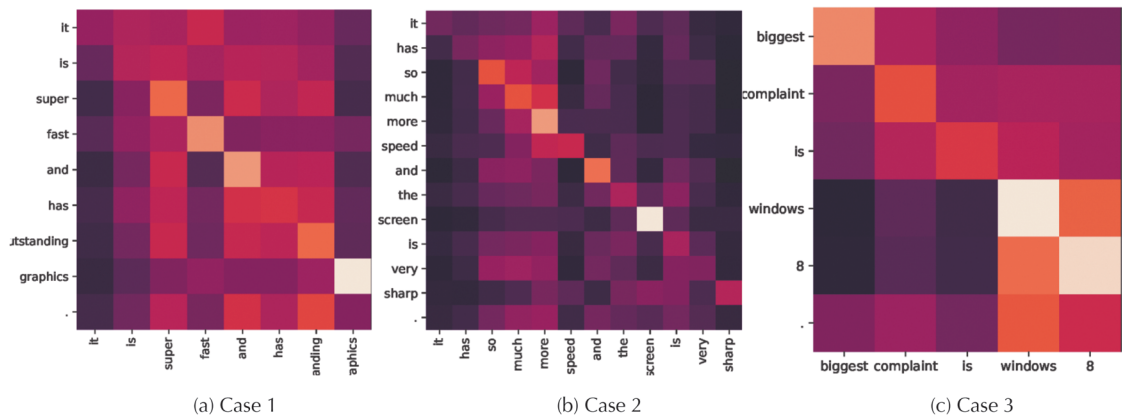


Figure 8. The attention score matrix plots for the three validation sets demonstrate that our AGGCN can accurately focus on aspect and the sentiment words associated with them.

5. CONCLUSION AND FUTURE WORKS

With the development of graph neural networks, many NLPs have made good progress. In ABSA, some researchers use graph convolution to deconvolve dependency graphs and semantic graphs generated by MHSA, respectively, and achieve good results. In this paper, motivated by [43] and [44], we reconstruct the dependency graph and semantic graph through aspects to be more suitable for the ABSA task. In addition, in order to take into account the information of the two parts, we use a set of dynamic parameters to fuse semantic graph and syntactic graph to generate semantic-syntactic graph. Finally, in the process of experiments, we found that the model is easy to overfitting. Therefore, we used multi-dropout to mitigate overfitting with different dropout models. Excellent experimental results on three datasets demonstrate the syntactic and semantic learning ability of our model.

To verify the effectiveness of the model, we conduct ablation experiments on the model itself to demonstrate the effectiveness of each module. We also conduct several additional experiments to explore the effect of model depth and attention heads.

In the future, we will try to introduce an external knowledge base and embed it into the feature representation of aspect to provide more accurate semantic information of aspect, because some classification errors are caused by the inability to understand the meaning of aspect accurately.

ACKNOWLEDGMENT

The authors would like to thank Editor-in-Chief, editor, and anonymous reviewers for their valuable comments and helpful suggestions. This work was supported by the National Natural Science Foundation of China under Grant 61976158 and Grant 61673301.

REFERENCES

- [1] Nasukawa, T., Yi, J.: Sentiment analysis: Capturing favorability using natural language processing. In: Proceedings of the 2nd international conference on Knowledge capture, pp. 70–77 (2003)
- [2] Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 168–177 (2004)
- [3] Ding, X., Liu, B.: The utility of linguistic rules in opinion mining. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 811–812 (2007)
- [4] Wang, Y., Huang, M., Zhu, X., et al.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 606–615 (2016)
- [5] Ma, D., Li, S., Zhang, X., et al.: Interactive attention networks for aspect-level sentiment classification. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp. 19–25 (2017)
- [6] Fan, F., Feng, Y., Zhao, D.: 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 (2018)
- [7] Chen, P., Sun, Z., Bing, L., et al.: 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 (2017)
- [8] Li, X., Bing, L., Lam, W., et al.: Transformation networks for target-oriented sentiment classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume1: Long Papers), pp. 946–956 (2018)
- [9] Song, Y., Wang, J., Jiang, T., et al.: Targeted sentiment classification with attentional encoder network. In: Proceedings of the 28th International Conference on Artificial Neural Networks, Part IV 28. Springer International Publishing, pp. 93–103 (2019)
- [10] Qiu, G., Liu, B., Bu, J., et al.: Opinion word expansion and target extraction through double propagation. *Computational linguistics* 37(1), 9–27 (2011)

- [11] Liu, K., Xu, H.L., Liu, Y., et al.: Opinion target extraction using partially-supervised word alignment model. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 2134–2140 (2013)
- [12] Zhang, C., Li, Q., Song, D.: 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, pp. 4567–4577 (2019)
- [13] Sun, K., Zhang, R., Mensah, S., et al.: 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, pp. 5679–5688 (2019)
- [14] Tian, Y., Chen, G., Song, Y.: Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 2910–2922 (2021)
- [15] Wang, K., Shen, W., Yang, Y., et al.: Relational graph attention network for aspect-based sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3229–3238 (2020)
- [16] Bai, X., Liu, P., Zhang, Y.: Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, 503–514 (2020)
- [17] Huang, L., Sun, X., Li, S., et al.: Syntax-aware graph attention network for aspect-level sentiment classification. In: Proceedings of the 28th international conference on computational linguistics, pp. 799–810 (2020)
- [18] Huang, B., Carley, K M.: Syntax-aware aspect level sentiment classification with graph 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, pp. 5468–5476 (2019)
- [19] Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in neural information processing systems 30: Annual Conference on Neural Information Processing Systems, pp. 5998–6008 (2017)
- [20] Tang, H., Ji, D., Li, C., et al.: 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 (2020)
- [21] Li, R., Chen, H., Feng, F., et al.: 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 (2021)
- [22] Inoue, H.: Multi-sample dropout for accelerated training and better generalization[J]. *arXiv preprint arXiv:1905.09788* (2019)
- [23] Dong, L., Wei, F., Tan, C., et al.: 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 (2014)
- [24] Nguyen, T. H., Shirai, K.: 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 (2015)
- [25] Nguyen, H. T., Le Nguyen, M.: Effective attention networks for aspect-level sentiment classification. In: 2018 10th International Conference on Knowledge and Systems Engineering. IEEE, pp. 25–30 (2018)
- [26] Bruna, J., Zaremba, W., Szlam, A., et al.: Spectral networks and locally connected networks on graphs. In: Proceedings of the 2nd International Conference on Learning Representations (2014)

- [27] Mikolov, T., Chen, K., Corrado, G., et al.: Efficient estimation of word representations in vector space. In: Proceedings of the 1st International Conference on Learning Representations (2013)
- [28] Pennington, J., Socher, R., Manning, C. D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp. 1532–1543 (2014)
- [29] Devlin, J., Chang, M. W., Lee, K., et al.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)
- [30] Lan, Z., Chen, M., Goodman, S., et al.: Albert: A lite bert for self-supervised learning of language representations. In: 8th International Conference on Learning Representations (2020)
- [31] Liu, Y., Ott, M., Goyal, N., et al.: Roberta: A robustly optimized bert pretraining approach[J]. arXiv preprint arXiv:1907.11692 (2019)
- [32] He, P., Liu, X., Gao, J., et al.: Deberta: Decoding-enhanced bert with disentangled attention. In: 9th International Conference on Learning Representations (2021)
- [33] Song, Y., Wang, J., Jiang, T., et al.: Targeted sentiment classification with attentional encoder network. In: Proceedings of the 28th International Conference on Artificial Neural Networks, pp. 93–103 (2019)
- [34] Xu, H., Liu, B., Shu, L., et al.: BERT post-training for review reading comprehension and aspect-based sentiment analysis. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2324–2335 (2019)
- [35] Pontiki, M., Galanis, D., Pavlopoulos, J., et al.: SemEval-2014 task 4: Aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Association for Computational Linguistics, pp. 27–35 (2014)
- [36] Dong, L., Wei, F., Tan, C., et al.: Adaptive recursive neural network for target-dependent twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, pp. 49–54 (2014)
- [37] Dozat, T., Manning, C. D.: Deep biaffine attention for neural dependency parsing. In: Proceedings of the 5th International Conference on Learning Representations (2017)
- [38] Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (2015)
- [39] Ma, Y., Pang, Y.: Learnable dependency-based double graph structure for aspect-based sentiment analysis. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 7086–7092 (2022)
- [40] Yu, B., Zhang, S.: A novel weight-oriented graph convolutional network for aspect-based sentiment analysis. The Journal of Supercomputing 79(1), 947–972 (2023)
- [41] Zhou, X., Zhang, T., Cheng, C., et al.: Dynamic multichannel fusion mechanism based on a graph attention network and BERT for aspect-based sentiment classification. Applied Intelligence 53(6), 6800–6813 (2023)
- [42] Feng, S., Wang, B., Yang, Z., et al.: Aspect-based sentiment analysis with attention-assisted graph and variational sentence representation. Knowledge-Based Systems 258, 109975 (2022)
- [43] Liang, B., Yin, R., Gui, L., et al.: Jointly learning aspect-focused and inter-aspect relations with graph convolutional networks for aspect sentiment analysis. In: Proceedings of the 28th international conference on computational linguistic, pp. 150–161 (2020)
- [44] Wu, Z., Ong, D. C.: Context-guided bert for targeted aspect-based sentiment analysis. In: Proceedings of the AAAI conference on artificial intelligence 35(16), pp. 14094–14102 (2021)