

# Multi-View Attention Syntactic Enhanced Graph Convolutional Network for Aspect-based Sentiment Analysis

Xiang Huang<sup>1</sup>, Hao Peng<sup>1,5,6,7,8\*</sup>, Shuo Sun<sup>1</sup>, Zhifeng Hao<sup>2</sup>, Hui Lin<sup>3</sup>, and Shuhai Wang<sup>4</sup>

<sup>1</sup> School of Cyber Science and Technology, Beihang University  
{huang.xiang, penghao, sun.shuo}@buaa.edu.cn

<sup>2</sup> University of Shantou zfhao@stu.edu.cn

<sup>3</sup> China Academic of Electronics and Information Technology linhui@cetc.com.cn

<sup>4</sup> Shijiazhuang Tiedao Univerisity wsh36302@126.com

<sup>5</sup> State Key Laboratory of Public Big Data, Guizhou University

<sup>6</sup> Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University

<sup>7</sup> Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology,

<sup>8</sup> Hangzhou Innovation Institute of Beihang University

**Abstract.** Aspect-based Sentiment Analysis (ABSA) is the task aimed at predicting the sentiment polarity of aspect words within sentences. Recently, incorporating graph neural networks (GNNs) to capture additional syntactic structure information in the dependency tree derived from syntactic dependency parsing has been proven to be an effective paradigm for boosting ABSA. Despite GNNs enhancing model capability by fusing more types of information, most works only utilize a single topology view of the dependency tree or simply conflate different perspectives of information without distinction, which limits the model performance. To address these challenges, in this paper, we propose a new multi-view attention syntactic enhanced graph convolutional network (MASGCN) that weighs different syntactic information of views using attention mechanisms. Specifically, we first construct distance mask matrices from the dependency tree to obtain multiple subgraph views for GNNs. To aggregate features from different views, we propose a multi-view attention mechanism to calculate the attention weights of views. Furthermore, to incorporate more syntactic information, we fuse the dependency type information matrix into the adjacency matrices and present a structural entropy loss to learn the dependency type adjacency matrix. Comprehensive experiments on four benchmark datasets demonstrate that our model outperforms state-of-the-art methods. The codes and datasets are available at <https://github.com/SELGroup/MASGCN>.

## 1 Introduction

Aspect-based sentiment analysis (ABSA) is a fine-grained task aimed at classifying the sentiment polarity of given aspect words in a sentence [48]. It has drawn significant

---

\* Corresponding author.

attention in recent years [22,39]. Unlike traditional sentence-level sentiment analysis, ABSA deals with sentences that may contain several different aspect terms in one sentence, each with different sentiment polarities and contexts. For example, in the sentence *Our waiter was friendly and it is a shame that he didn't have a supportive staff*, the aspect term *waiter* is positive, while *staff* is negative in sentiment. Consequently, compared with traditional sentiment analysis, ABSA faces the challenge of understanding the relevant context words for different aspects and fully utilizing the related information.

Early works on ABSA utilized various attention mechanisms [37,25,2,7,10,8,18] to model the relation between the aspect term and its context words. Although they have achieved good results, the attention-based weights also introduce substantial noise for aspect terms. For example, in the sentence *Our waiter was friendly and it is a shame that he didn't have a supportive staff*, for the aspect *waiter*, both opinion words *friendly* and *supportive* may be assigned large attention scores [50], which hampers performance improvement.

The rapid development of syntactic parsing offers a new paradigm for ABSA tasks. Recently, numerous works [46,9,47,1,34,36,19,49,21,50,29] leverage graph neural networks (GNNs) over the parsed dependency tree to exploit the syntactic structure, as the dependency tree provides explicit connections between aspect words and their related opinion words. These models could be divided into three classes based on distinct syntactic information within the adjacency matrix [50]: exploiting the topology of trees [46,9,34], exploiting the dependency types in trees [47,36], and exploiting the minimum distances in trees [49,50]. However, they still do not fully exploit the syntactic information of different types. [46] and [9] only exploit the simple topology information of dependency trees. [50] consider distance and type information via deep reinforced learning, but they generate only one view for distance information, limiting its expression capability. [49] take multiple views for distance information into account but fuse different views indiscriminately. This simple corporation method limits the model to capture as much view information, as it falls short in filtering the introduced noise from more views [49]. Hence, fully utilizing the diverse syntactic information in dependency trees remains challenging.

Recently, the theory of structural entropy [15] has demonstrated significant advantages in graph structural learning [52,23]. It converts the graph into a hierarchical tree, named the encoding tree, by minimizing the graph's structural entropy, which is the sum of the structural entropy of each node in the encoding tree. The structural entropy is calculated based on the adjacency matrix and the hierarchical partitions of the graph, where the weights assigned to the matrix are determined by the similarities of the graph node embeddings. Each type of graph partition has its optimal encoding tree, corresponding to the minimal structural entropy. Thus, structural entropy could serve as a graph structural learning target [38]. In the ABSA task, syntactic parsing offers an inherent dependency type partition within the dependency tree, which could be leveraged to learn the dependency type information matrix using the structural entropy theory.

In this paper, we propose a novel multi-view attention syntactic enhanced graph convolutional network (MASGCN) for ABSA, which effectively incorporates information from various syntactic views. We construct distance mask matrices from the dependency tree to obtain multiple distance information views. To fuse these distinct views, we

propose a multi-view attention mechanism that assigns different attention weights to various views, enhancing the important views and reducing noise from less relevant ones. Additionally, to incorporate the syntactic information of dependency types, we introduce a structural entropy loss derived from [15], exploiting the dependency type in the dependency tree to learn the dependency type matrix. Extensive experiments are conducted on four benchmark datasets. Comparative results and analysis demonstrate that the proposed model enjoys superior effectiveness compared to the state-of-the-art (SOTA) baselines.

In summary, the main contributions of our paper are as follows:

- We propose a novel multi-view attention mechanism syntactic enhanced graph convolutional network, which fully leverages extensive syntactic information and mitigates the noise introduced by multiple views.
- We present a multi-view attention mechanism to weigh different views, reinforce strongly correlated views, and diminish noise from weakly correlated views. Additionally, we introduce a structural entropy-based loss to learn the dependency type matrix by utilizing the dependency type information inherent in the dependency tree.
- We conduct extensive experiments on four benchmark datasets. The experimental results demonstrate the effectiveness of our model.

## 2 Related Works

### 2.1 Aspect-based Sentiment Analysis

In contrast to traditional sentiment tasks that are sentence-level or document-level, ABSA is entity-level oriented and more fine-grained for sentiment polarity analysis. Early works [12,13,4] extract sentiment features based on handcrafted rules and perform poorly in capturing rich sentiment information.

Recently, various context-based methods [37,25,2,7,10,8,18,33] propose to utilize attention mechanisms to model the contextual semantic information between the aspect term and its context words. ATAE-LSTM [37] proposes an attention-based LSTM model to concentrate on different parts of sentences to generate attention vectors for aspect sentiment classification. IAN [25] learns attention between the contexts and targets in an interactive manner and generates representations for targets and contexts separately. RAM [2] leverages the multiple-attention mechanism to capture sentiment features separated by a long distance. MGAN [7] proposes a fine-grained attention mechanism to capture the word-level interaction between aspects and context and then compose it with coarse-grained attention mechanisms. [18] design a hierarchical attention mechanism to fuse the information of the aspect terms and the contextual words. [33] propose a dual attention mechanism to address the problem of recognizing conflicting opinions. Despite these context-based methods achieving good results, they are unable to distinguish the relation between the aspect term and multiple opinion words. This hampers their performance in sentences having multiple aspects with different polarities.

Owing to the rapid development of syntactic parsing methods, another trend explicitly utilizes the parsed dependency tree to reveal the connection between aspect terms and opinion words and learn the syntactic features of aspect terms. These methods

could be categorized into three classes based on the syntactic information extracted from the dependency tree: exploiting topology information [46,9,34], exploiting dependency type information [47,36], and exploiting minimum distance information [49,50]. [46] is the first work to utilize the dependency tree of sentences and adopt GCNs to explore the topological information from the dependency trees. [34] jointly consider the flat representations learned from the Transformer and the graph-based representations learned from the corresponding dependency graph to diminish the noise induced by incorrect dependency trees. [47] utilize a global lexical graph to encode corpus-level word co-occurrence data and construct a concept hierarchy on both syntactic and lexical graphs, thereby incorporating dependency type information. [21] integrate the external knowledge from SenticNet to enhance the dependency graphs of sentences. [49] construct the syntactic mask matrices of different distances to learn structural distance information from local to global. [40] incorporates domain knowledge, dependency labels, and syntax path in the dependency tree to enhance the accuracy of the model. [50] employ deep reinforcement learning to guide the extraction of syntactic information from the dependency tree. However, as these approaches only incorporate part of these three syntactic information types, or indiscriminately fuse them, they don't fully leverage the diverse syntactic information in dependency trees.

## 2.2 Structural Entropy

Information entropy was proposed to meet the demand for measuring uncertainty in information transmitted through communication systems. Correspondingly, to measure the information uncertainty in graph-structured data, structural entropy was also proposed and used to evaluate the complexity of the hierarchical structure of a graph by defining the encoding tree and structural entropy [15]. The process of constructing and optimizing the encoding tree is also a natural vertices clustering method for graphs. Due to the theoretical completeness and interpretability of structural entropy theory, it has great potential for application in graph analyses such as graph hierarchical pooling [41] and graph structure learning [52,6]. Moreover, the two-dimensional and three-dimensional structural entropy, which measure the complexity of hierarchical structures at two and three dimensions, respectively, have found applications in fields such as medicine [16], bioinformatics [17], social bot detection [28,45,42], network security [14], natural language understanding [11] and reinforcement learning [44].

## 3 The proposed framework

This section outlines the overall architecture of MASGCN and details each component of our proposed model.

### 3.1 Overall Architecture

We illustrate the overall structure of MASGCN in Figure 1. First, following the approach of [49], we utilize the aspect-aware attention and self-attention mechanisms to obtain enhanced semantic matrices. Second, we parse the sentence syntactically to obtain a

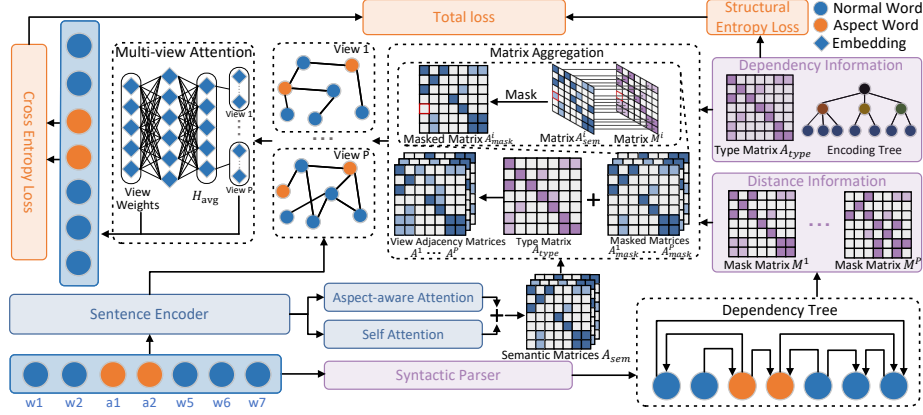


Fig. 1: Overall framework of MASGCN.

dependency tree, constructing distance mask matrices and a dependency type matrix to inform our model of the tree structure. We also design a structural entropy loss to capitalize on the dependency type information in the dependency tree. Third, we mask the semantic matrices with distance mask matrices and incorporate the dependency type matrix to create multi-view adjacency matrices for the GNN. Finally, we propose a multi-view attention mechanism to integrate features across these various views.

### 3.2 Semantic Feature

Given a sentence-aspect pair  $\mathcal{W} - \mathcal{A}$ , where  $\mathcal{A} = \{a_1, \dots, a_M\}$  represents the aspect words set and is also a sub-sequence of the sentence  $\mathcal{W} = \{w_1, \dots, w_N\}$ , with  $w_i$  and  $a_j$  denoting the  $i$ -th and  $j$ -th words in  $\mathcal{W}$  and  $\mathcal{A}$ , and  $N$  and  $M$  representing the lengths of the sentence and the aspect, respectively. We derive the contextual feature of sentences based on the sentence encoder like BERT [3] and obtain the low-dimensional embedding matrix  $H \in \mathbb{R}^{N \times D}$ , where the  $i$ -th word  $w_i$  corresponds to the  $i$ -th row feature  $h_i$  with dimension  $D$ . Aspect features  $H_a \in \mathbb{R}^{M \times D}$  are derived from  $H$ .

For enhanced semantic features, we follow [49] and apply the aspect-aware attention and self-attention mechanism to obtain aspect attention matrices  $\{A_{asp}^1, \dots, A_{asp}^P\}$  and self-attention matrices  $\{A_{self}^1, \dots, A_{self}^P\}$ , where  $P$  is the number of attention heads. The  $i$ -th matrices are calculated as follows:

$$A_{asp}^i = \tanh \left( \hat{H}_a W_a^i \times (H W_k^i)^T + b_a \right), \quad (1)$$

$$A_{self}^i = \frac{Q W_Q^i \times (K W_K^i)^T}{\sqrt{D}}. \quad (2)$$

Here,  $\hat{H}_a \in \mathbb{R}^{N \times D}$  represents the  $N$  times repeated mean of  $H_a$ ,  $W_a^i \in \mathbb{R}^{D \times D}$  and  $W_k^i \in \mathbb{R}^{D \times D}$  are learnable weights for aspect-aware attention of the  $i$ -th attention head, and  $b_a$  is the learnable bias. The query  $Q$  and the key  $K$  are equal to the feature

embedding  $H$ , and  $W_Q^i \in \mathbb{R}^{D \times D}$  and  $W_K^i \in \mathbb{R}^{D \times D}$  are learnable weights for the self-attention of the  $i$ -th attention head. Then, we integrate  $A_{\text{asp}}^i$  and  $A_{\text{self}}^i$  as follows:

$$A_{\text{sem}}^i = A_{\text{asp}}^i + A_{\text{self}}^i, \quad (3)$$

where  $A_{\text{sem}}^i \in \mathbb{R}^{N \times N}$  is the output enhanced semantic matrix. These  $P$  matrices  $\{A_{\text{sem}}^1, \dots, A_{\text{sem}}^P\}$  are used for later combination with syntactic matrices.

### 3.3 Syntactic Feature

In this section, we first utilize the Stanford Parser CoreNLP [26] to parse the sentence into a syntactic dependency tree, and then extract the distance information and dependency information from the dependency tree.

• **Distance information.** We treat the syntactic dependency tree as an undirected graph, with each word as a node. Then, we define the distance  $d(v_i, v_j)$  between nodes  $v_i$  and  $v_j$  as the number of hops between the two nodes in the graph. The shortest path distance between nodes  $v_i$  and  $v_j$  is calculated as follows:

$$D_{ij} = \min d(v_i, v_j). \quad (4)$$

We consider different scales of distance information by defining various distance mask matrices  $M^k$  as follows:

$$M_{ij}^k = \begin{cases} 0, & D_{ij} \leq k, \\ -\infty, & \text{otherwise,} \end{cases} \quad (5)$$

where  $k \in [1, P]$  and  $P$  is the number of attention heads mentioned in Section 3.2. As  $k \in [1, P]$  increases, the scope of distance information that the mask matrix  $M^k$  covers also increases.

• **Dependency information.** We define the initial dependency type adjacency matrix  $A_{\text{type}}^0 \in \mathbb{R}^{N \times N}$  as follows:

$$A_{\text{type}}^0[i, j] = \begin{cases} id_{\text{type}}, & D_{ij} = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where  $id_{\text{type}} \in [1, U]$  and  $U$  is the size of the dependency type vocabulary. To utilize the dependency information of the dependency tree, we initialize the dependency type feature matrix  $H_{\text{type}} \in \mathbb{R}^{U \times D}$ . The global attention mechanism [50] calculates the dependency type attention matrix as follows:

$$\alpha = \text{softmax}(H_{\text{type}} W_t), \quad (7)$$

$$A_{\text{type}}[i, j] = \begin{cases} \alpha[A_{\text{type}}^0[i, j]], & A_{\text{type}}^0[i, j] \neq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where  $W_t \in \mathbb{R}^{D \times 1}$  represents the learnable weights for global attention,  $\alpha \in \mathbb{R}^{U \times 1}$  denotes the attention score.  $A_{\text{type}} \in \mathbb{R}^{N \times N}$  is the dependency type information matrix output.

• **Matrix Aggregation.** To this end, we have obtained the semantic feature matrices  $\{A_{\text{sem}}^1, \dots, A_{\text{sem}}^P\}$ , the distance information mask matrices  $\{M^1, \dots, M^P\}$ , and the dependency type information matrix  $A_{\text{type}}$ . We then aggregate these semantic and syntactic information to construct the  $i$ -th view adjacency matrices for GNN as follows:

$$A^i = A_{\text{mask}}^i + A_{\text{type}}, \quad (9)$$

$$A_{\text{mask}}^i = \text{softmax}(A_{\text{sem}}^i + M^i). \quad (10)$$

Here, we utilize the softmax operator to mask the  $i$ -th semantic feature matrix  $A_{\text{sem}}^i$  with the  $i$ -th distance mask matrix  $M^i$ , as  $-\infty$  in  $M^i$  masks the corresponding place in  $A_{\text{sem}}^i$  to 0. As  $i$  increases,  $A_{\text{mask}}^i$  could attend more global information. Finally, we obtain  $P$  distinct view adjacency matrices  $\{A^1, \dots, A^P\}$ , each attending to different semantic and syntactic information.

### 3.4 GNN and Multi-View Attention Mechanism

Now we have obtained the sentence embedding  $H$  and  $P$  distinct adjacency matrices  $\{A^1, \dots, A^P\}$ . The output embedding of the  $l$ -th GNN layer is calculated as follows:

$$H_l = \sigma(AGG_{i=1}^P(A^i H_{l-1})W_l), H_0 = H, \quad (11)$$

where  $H_l \in \mathbb{R}^{N \times D}$  is the  $l$ -th layer output embedding,  $W_l \in \mathbb{R}^{D \times D}$  represents the  $l$ -th layer's learnable weights, and  $\sigma$  denotes an activation function. The operator  $AGG_{i=1}^P$  is the multi-view attention mechanism we propose, which is as follows:

$$H_{\text{avg}}^i = \frac{1}{N \times D} \sum_{r,s=1}^{N,D} (A^i H_{l-1})_{r,s}, \quad (12)$$

$$\alpha_{\text{view}} = W_2(\sigma(W_1 H_{\text{avg}})), \quad (13)$$

$$AGG_{i=1}^P(A^i H_{l-1}) = \sum_{i=1}^P \alpha_{\text{view}}^i A^i H_{l-1}. \quad (14)$$

Here,  $H_{\text{avg}}^i \in \mathbb{R}$  is  $i$ -th row of  $H_{\text{avg}} \in \mathbb{R}^{P \times 1}$ ,  $\alpha_{\text{view}} \in \mathbb{R}^{P \times 1}$  is the view weights of  $P$  views,  $W_1$  and  $W_2$  are the learnable weight,  $\alpha_{\text{view}}^i$  is the  $i$ -th row of  $\alpha_{\text{view}}$ . The multi-view attention mechanism adaptively extracts useful information from various views, and leverages the diversity and complementarity of information across different perspectives, thereby enhancing the representation capabilities for downstream tasks. Additionally, the multi-view attention mechanism reduces noise introduced by too many views [49].

### 3.5 Loss Function with Structural Entropy

Structural entropy theory [15] has demonstrated its capability in multiple graph-related works [41,52,6]. For a graph  $G = (X, E, W)$ ,  $X$  is the set of graph datapoints, and  $E$  and  $W$  are the edges and corresponding edge weights. We define a two-level encoding tree  $\mathcal{T}$ , where the intermediate tree nodes  $\alpha$  represent a partitioned subset of the graph

vertices  $X$ , denoted as  $X_\alpha \subset X$ . The leaf nodes are the graph vertices, and the top layer of  $\mathcal{T}$  is a single virtual node. The original two-dimensional structural entropy of  $\mathcal{T}$  is defined as follows:

$$H^\mathcal{T}(G) = \sum_{\alpha \in \mathcal{T}} H^\mathcal{T}(G; \alpha), \quad (15)$$

$$H^\mathcal{T}(G; \alpha) = -\frac{g_\alpha}{\text{vol}(G)} \log_2 \frac{\mathcal{V}_\alpha}{\text{vol}(G)}. \quad (16)$$

Here,  $\text{vol}(G)$  is the sum of the edge weights of all edges  $E$ ,  $\mathcal{V}_\alpha$  represents the sum of the edge weights of edges in  $X_\alpha$ , and  $g_\alpha$  is the sum of the edge weights of cut edges between  $X_\alpha$  and its complement set  $X_\alpha^c$ .

Within the ABSA task, syntactic parsing provides a natural segmentation of dependency types within the dependency tree. This partitioning can be utilized to develop a matrix of dependency type information based on the principles of structural entropy theory. To further enhance the utilization of syntactic information in the dependency tree, we design a structural entropy loss  $\mathcal{L}_{SE}$ . In detail, we first build the two-layer encoding tree  $\mathcal{T}$  with the dependency type as the immediate layer’s tree nodes. The encoding tree is an abstraction of the inherent dependency type partition  $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_U\}$  of words, where  $U$  is the size of the dependency type vocabulary. Denoting  $Y \in \{0, 1\}^{N \times U}$  as the one-hot encoding of  $\mathcal{P}$ , the two-dimensional structural entropy for this encoding tree can be described as follows:

$$\mathcal{L}_{SE} = \text{trace} \left\{ \frac{Y^T A_{\text{type}} Y}{2 \sum A_{\text{type}}} \otimes \log_2 \frac{\{1\}^{U \times N} A_{\text{type}} Y}{2 \sum A_{\text{type}}} \right\}, \quad (17)$$

where  $\otimes$  refers to matrix multiplication. As the dependency type partition  $\mathcal{P}$  is fixed, the optimized matrix  $A_{\text{type}}$  can be obtained as the minimal two-dimensional structural entropy [38]. Combined with cross-entropy loss, the final training loss is as follows:

$$\mathcal{L} = \mathcal{L}_{ce} + \gamma \mathcal{L}_{SE}, \quad (18)$$

$$\mathcal{L}_{ce} = - \sum_{a \in \mathcal{D}} \log (\text{softmax} (W_p H_a^l + b_p)), \quad (19)$$

where  $W_p$  and  $b_p$  are learnable weights and bias,  $H_a^l$  is the aspect embeddings of the final  $l$ -th GNN layer,  $\mathcal{D}$  is the set of training samples, and  $\gamma$  is a hyperparameter.

## 4 Experiment Setup

In this section, we detail the baselines, datasets, and implementation details of MASGCN.

### 4.1 Datasets

Following previous ABSA works, we evaluate MASGCN on four benchmark datasets: Restaurant14, Restaurant16, Laptop14, and Twitter. The Restaurant14 and Laptop14 datasets include reviews in the restaurant and laptop domains from SemEval-2014 [32]. The Restaurant16 dataset is from SemEval-2016 [31]. The Twitter dataset is collected from tweets by [5]. Each aspect is annotated with one of three polarities: positive, neutral, and negative. The statistics of these datasets are listed in Table 1.



Table 1: Statistics of experimental datasets.

Dataset	Division	Positive	Negative	Neutral
Restaurant14	Train	2164	807	637
	Test	727	196	196
Laptop14	Train	976	851	455
	Test	337	128	167
Twitter	Train	1507	1528	3016
	Test	172	169	336
Restaurant16	Train	1321	488	73
	Test	487	136	32

## 4.2 Baselines

To comprehensively evaluate the performance of MASGCN, we compare it with SOTA baselines, which are briefly described as follows:

### 1) Context-based methods.

- **ATAE-LSTM** [37] is an LSTM model with the attention mechanism on aspects.
- **IAN** [25] interactively learns attention scores for aspects with their context.
- **RAM** [2] proposes a recurrent attention memory network to capture the aspect-specific sentence representation.
- **MGAN** [7] applies a fine-grained attention mechanism to capture token-level interactions between aspects and contexts.
- **TNet** [20] utilizes a CNN model to extract salient features for sentiment analysis.

### 2) Syntax-based GNN methods.

- **ASGCN** [46] applies GCN on the raw topology of the dependency tree to extract syntactic information.
- **kumaGCN** [1] uses gating mechanisms to acquire syntactic features with latent semantic information.
- **DGEDT** [34] combines transformer and graph-based representations from the corresponding dependency graph to diminish the error induced by incorrect dependency trees.
- **BiGCN** [47] uses convolutions on hierarchical lexical and syntactic graphs to integrate token co-occurrence and dependency type information.
- **R-GAT** [36] uses a star-induced graph with minimum distances and dependency types as edges and applies a relational GAT for attention-based aggregation.
- **T-GCN** [35] uses attention to distinguish relation types and applies an attentive layer ensemble for feature learning from GCN layers.
- **DualGCN** [19] introduces syntactic and semantic information through SynGCN and SemGCN modules simultaneously.
- **SSEGCN** [49] combines aspect-aware attention, self-attention, and minimum tree distances to enhance sentiment information with syntactic information.
- **SenticGCN** [21] integrates external knowledge from SenticNet to enhance the dependency graphs of sentences.

- **SPGCN [24]** enhances dependencies and context-awareness with aspect focus and sentiment integration, using a multi-graph perception mechanism to reduce redundancy and captures unique dependencies.
- **KDGN [40]** integrates domain knowledge into the dependency tree and constructs the knowledge-aware dependency graph to capture latent sentiment polarity.
- **CSADGCN [43]** integrates context-guided, self, and aspect-level attention for information capture and boosts GCN with linguistic features and a biaffine module for word relationships.
- **RDGCN [50]** utilizes deep reinforced learning to extract syntactic information and improve the importance calculation of dependencies in both distance and type views.
- **BERT & Model+BERT [3]** represents the pre-trained language model BERT and the model with BERT as the sentence encoder.

### 4.3 Experimental Settings

In experiments, we initialize token embeddings with pre-trained 300-dimensional Glove vectors [30], combine them with 30-dimensional part-of-speech and position embeddings, and feed them into a BiLSTM model with a 0.7 dropout rate. The batch size is 16 and the number of GCN layers is 2. The learning rate of the Adam optimizer is 0.002. The view number and attention head number  $P$  are set to 10. The hyperparameter  $\gamma$  of the structural entropy loss is 0.01. Model+BERT utilizes the bert-base-uncased [3] English version. We utilize accuracy (Acc.) and macro-F1 (F1) to evaluate classification performance. The baseline results are from [50] and [51].

## 5 Experiments

In this section, we conduct comparative experiments with baselines for effectiveness comparison. Moreover, we carry out the ablation study, parameter sensitivity experiments, and investigations into adjacency matrices to explore each component of our model.

### 5.1 Performance Comparison

We conduct the experiments on four benchmark datasets and report the experiment results in Table 2. MASGCN achieves state-of-the-art results on all four datasets, except for accuracy on the Restaurant16 dataset when using Glove embeddings. Specifically, with Glove embeddings, MASGCN outperforms all baselines with improvements of at least 0.17–0.79 in accuracy and 0.48–0.99 in macro-F1 across the Restaurant14, Laptop14, and Twitter datasets. On the Restaurant16 dataset, MASGCN still excels in macro-F1, showing a 1.22 improvement over the SOTA SenticGCN model. Besides, MASGCN + BERT surpasses other Model+BERT combinations across all datasets, except for macro-F1 on the Twitter dataset and the Laptop14 dataset. These results demonstrate the effectiveness of our model. Compared to R-GAT, SPGCN, CSADGCN, and RDGCN, which also utilize three types of information, MASGCN still outperforms due to its enhanced utilization of this information through multi-view information aggregation.

Table 2: Classification results (%) of all models. The best results are bolded, and the second-best results are underlined. † denotes topology information, ‡ denotes type information, and § denotes distance information.

Embedding	Model	Restaurant14		Laptop14		Twitter		Restaurant16	
		Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
Glove	ATAE-LSTM	78.60	67.02	68.88	63.93	68.64	66.60	83.77	61.71
	IAN	78.60	-	72.10	-	-	-	-	-
	RAM	80.23	70.80	74.49	71.35	69.36	67.30	83.88	62.14
	MGAN	81.25	71.94	75.39	72.47	72.54	70.81	-	-
	TNet	80.69	71.27	76.54	71.75	74.90	73.60	89.07	70.43
	ASGCN <sup>†</sup>	80.77	72.02	75.55	71.05	72.15	70.40	88.69	66.64
	kumaGCN <sup>†</sup>	81.43	73.64	76.12	72.42	72.45	70.77	89.39	73.19
	DGEDT <sup>†</sup>	83.90	75.10	76.80	72.30	74.80	73.40	<u>90.80</u>	73.80
	BiGCN <sup>†‡</sup>	81.97	73.48	74.59	71.84	74.16	73.35	-	-
	R-GAT <sup>†‡§</sup>	83.30	76.08	77.42	73.76	75.57	73.82	88.92	70.89
	DualGCN <sup>†</sup>	84.27	<u>78.08</u>	78.48	74.74	75.92	74.29	-	-
	SSEGCN <sup>†§</sup>	<u>84.72</u>	77.51	79.43	76.49	76.51	75.32	89.55	75.62
	SenticGCN <sup>†‡</sup>	84.03	75.38	77.90	74.71	-	-	<b>90.88</b>	<u>75.91</u>
	SPGCN <sup>†‡§</sup>	83.16	74.91	77.90	73.86	74.86	72.95	90.75	75.20
	RDGCN <sup>†‡§</sup>	84.36	78.06	<u>79.59</u>	<u>76.75</u>	<u>76.66</u>	<u>75.37</u>	-	-
<b>MASGCN<sup>†‡§</sup></b>	<b>84.99</b>	<b>78.64</b>	<b>80.38</b>	<b>77.74</b>	<b>76.96</b>	<b>75.85</b>	89.01	<b>77.13</b>	
BERT	BERT	85.97	80.09	79.91	76.00	75.92	75.18	89.52	70.47
	DGEDT+BERT <sup>†</sup>	86.30	80.00	79.80	75.60	77.90	75.40	91.90	79.00
	R-GAT+BERT <sup>†‡§</sup>	86.60	81.35	78.21	74.07	76.15	74.88	89.71	76.62
	T-GCN+BERT <sup>†‡</sup>	86.16	79.95	80.88	77.03	76.45	75.25	<u>92.32</u>	77.29
	DualGCN+BERT <sup>†</sup>	87.13	81.16	81.80	78.10	77.40	76.02	-	-
	SSEGCN+BERT <sup>†§</sup>	87.31	81.09	81.01	77.96	77.40	76.02	90.99	78.78
	SenticGCN+BERT <sup>†‡</sup>	86.92	81.03	82.12	79.05	-	-	91.97	<u>79.56</u>
	KDGN+BERT <sup>†‡</sup>	87.01	<u>81.94</u>	81.32	77.59	77.64	75.55	-	-
	CSADGCN+BERT <sup>†‡§</sup>	87.40	81.56	82.12	<b>79.22</b>	76.81	75.67	-	-
	RDGCN+BERT <sup>†‡§</sup>	<u>87.49</u>	81.16	<u>82.12</u>	78.34	<u>78.29</u>	<b>77.14</b>	-	-
<b>MASGCN+BERT<sup>†‡§</sup></b>	<b>87.76</b>	<b>82.56</b>	<b>82.44</b>	<u>79.11</u>	<b>78.58</b>	<u>77.05</u>	<b>92.52</b>	<b>80.72</b>	

## 5.2 Ablation Study

We evaluate each component of MASGCN on four datasets and report the results in Table 3. The absence of structural entropy loss  $\mathcal{L}_{SE}$  and the multi-view attention mechanism results in a decline in model performance on most datasets, with an average reduction of 0.18% and 1.20% in accuracy, and 0.47% and 3.63% in macro-F1, respectively. This demonstrates the effectiveness of structural entropy loss in learning dependency type information and our proposed multi-view attention mechanism in fusing syntactic information.

Table 3: Ablation study results (%). The best results are bolded, and the second-best results are underlined. w/o refers to the model without the corresponding component.

Model	Restaurant14		Laptop14		Twitter		Restaurant16	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
MASGCN	<b>84.99</b>	<b>78.64</b>	<b>80.38</b>	<b>77.74</b>	<u>76.96</u>	<b>75.85</b>	<b>89.01</b>	<b>77.13</b>
- w/o structural entropy loss $\mathcal{L}_{SE}$	84.63	78.07	<u>79.73</u>	76.88	<b>77.25</b>	75.61	89.01	76.92
- w/o multi-view attention mechanism	83.56	75.98	79.11	75.01	75.63	74.43	88.23	69.44

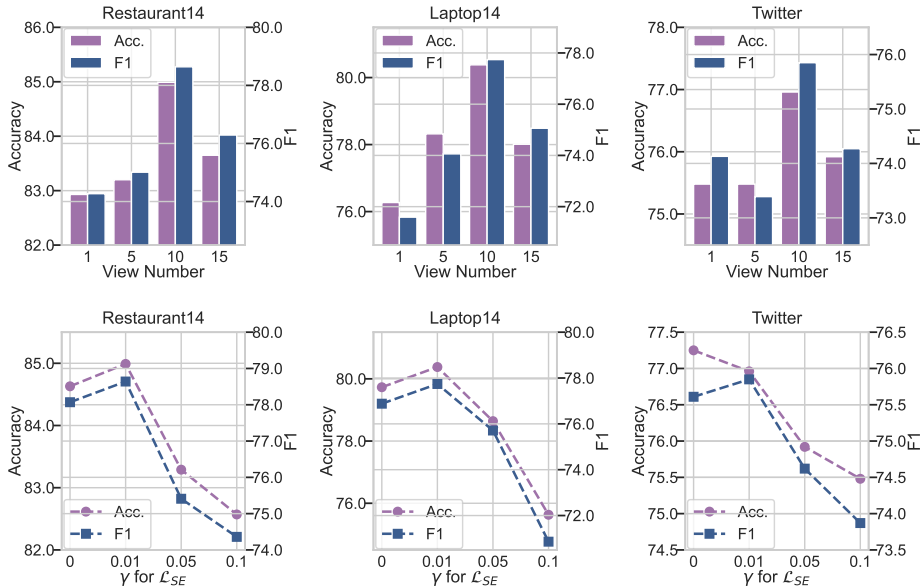
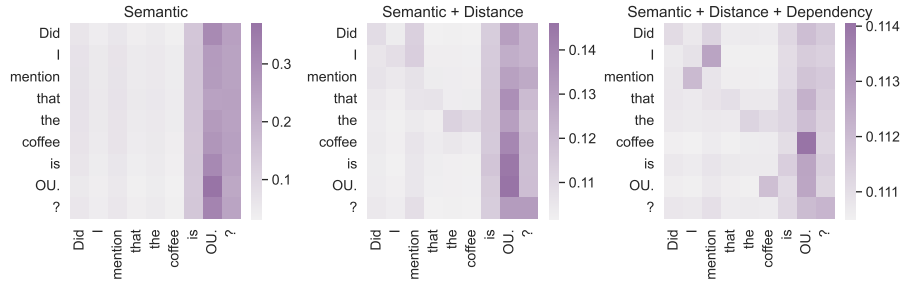


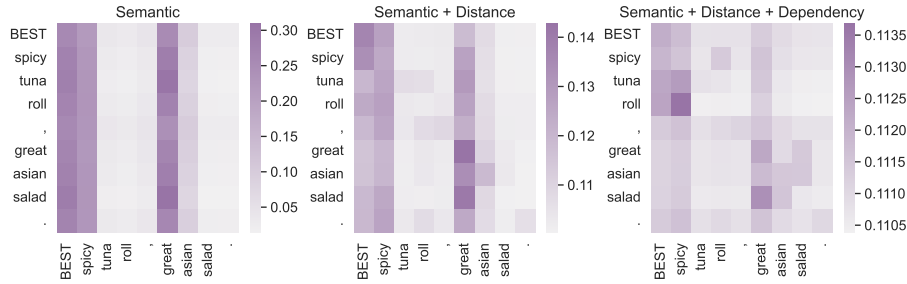
Fig. 2: Hyperparameter sensitivity analysis.

### 5.3 Hyperparameter Analysis

Two hyperparameters are introduced in our model: the view number  $P$  and the coefficient  $\gamma$  for the structural entropy loss  $\mathcal{L}_{SE}$ . We vary their values on three benchmark datasets and illustrate the results in Figure 2. When the view number  $P$  is not greater than 10, more views improve the model performance. However, the performance decreases when  $P$  is 15. This is because the number of views exceeds the maximum distance in the syntax tree, thereby introducing only additional noise. Despite this, compared to SSEGCN [49], which fuses different views of distance information indiscriminately and only integrates four different views when balancing the introduction of additional noise, MASGCN still improves the ability to fuse more information and reduce noise. Furthermore, we find that a smaller  $\gamma$  for structural entropy loss is preferred for the model. MASGCN achieves the best performance with  $\gamma$  being 0.01, except on Twitter, where  $\gamma$  is 0. A  $\gamma$  larger than 0.01 leads to a decrease in model performance.



(a) The aspect term is *coffee*.



(b) The aspect terms are *roll* and *salad*.

Fig. 3: Insights of matrices.

### 5.4 Insight of Matrices

To elucidate the effectiveness of each type of information, we visualize the semantic matrix  $A_{sem}$ , the masked matrix  $A_{mask}$  (incorporating semantic and distance information), and the final fused adjacency matrix  $A$  (incorporating semantic, distance, and dependency information) mentioned in Section 3. As depicted in Figure 3 (a), with the integration of distance information and dependency information, the correlation of the aspect term *coffee* with the polarity word *OUTSTANDING* is enhanced, and the attention to other uncorrelated words is gradually reduced. Figure 3 (b) shows that the integration of distance information and dependency information decreases the irrelevant attentions, enhances the attention of the aspect term *roll* to its polarity words *BEST spicy*, and reduces the attention to the polarity word *great*, which is connected with the other aspect term *salad*. The improvement can also be observed in the row of the aspect term *salad*. These two cases directly demonstrate the effectiveness of MASGCN.

## 6 Conclusion

In this paper, we propose a new multi-view attention syntactic enhanced graph convolutional network MASGCN. We propose a multi-view attention mechanism to aggregate

features from different views, which enhances the strongly correlated views while diminishing the noise introduced by too many views. Additionally, we design a structural entropy based loss to further leverage the dependency type information from the dependency tree. Extensive experiments on four benchmark datasets demonstrate that MASGCN outperforms the state-of-the-art baseline methods.

## 7 Limitations and Future Work

Our work incorporates various syntactic information from the dependency tree to improve the effectiveness of ABSA tasks. However, it is limited to the information from the parsed dependency tree and does not consider external knowledge. In future work, it would be valuable to exploit information from other models, such as external knowledge graphs or large language models like GPT-4 [27].

## Acknowledgments

This work is supported by the NSFC through grants 62322202, 62441612, and 62432006, Local Science and Technology Development Fund of Hebei Province Guided by the Central Government of China through grant 246Z0102G, the "Pioneer" and "Leading Goose" R&D Program of Zhejiang" through grant 2025C02044, the Hebei Natural Science Foundation through grant F2024210008, and the Guangdong Basic and Applied Basic Research Foundation through grant 2023B1515120020, Foundation of State Key Laboratory of Public Big Data through grant PBD2022-04, and Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (MIMS23-M-01).

## References

1. Chen, C., Teng, Z., Zhang, Y.: Inducing target-specific latent structures for aspect sentiment classification. In: EMNLP. pp. 5596–5607 (2020)
2. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: EMNLP. pp. 452–461 (2017)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT. pp. 4171–4186 (2019)
4. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: IJCAI (2015)
5. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent Twitter sentiment classification. In: ACL (Volume 2: Short Papers). pp. 49–54 (2014)
6. Duan, L., Chen, X., Liu, W., Liu, D., Yue, K., Li, A.: Structural entropy based graph structure learning for node classification. In: AAAI. vol. 38, pp. 8372–8379 (2024)
7. Fan, F., Feng, Y., Zhao, D.: Multi-grained attention network for aspect-level sentiment classification. In: EMNLP. pp. 3433–3442 (2018)
8. Gu, S., Zhang, L., Hou, Y., Song, Y.: A position-aware bidirectional attention network for aspect-level sentiment analysis. In: COLING. pp. 774–784 (2018)
9. Huang, B., Carley, K.: Syntax-aware aspect level sentiment classification with graph attention networks. In: EMNLP-IJCNLP. pp. 5469–5477 (2019)

10. Huang, B., Ou, Y., Carley, K.M.: Aspect level sentiment classification with attention-over-attention neural networks. In: *Social, Cultural, and Behavioral Modeling*. pp. 197–206 (2018)
11. Huang, X., Peng, H., Sun, L., Lin, H., Liu, C., Cao, J., Yu, P.S.: Structural entropy guided probabilistic coding. *arXiv:2412.08841* (2024)
12. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent Twitter sentiment classification. In: *ACL-HLT*. pp. 151–160 (2011)
13. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In: *SemEval 2014*. pp. 437–442 (2014)
14. Li, A., Hu, Q., Liu, J., Pan, Y.: Resistance and security index of networks: structural information perspective of network security. *Scientific reports* **6**(1), 1–24 (2016)
15. Li, A., Pan, Y.: Structural information and dynamical complexity of networks. *TIT* pp. 3290–3339 (2016)
16. Li, A., Yin, X., Pan, Y.: Three-dimensional gene map of cancer cell types: Structural entropy minimisation principle for defining tumour subtypes. *Scientific reports* **6**(1), 1–26 (2016)
17. Li, A., Yin, X., Xu, B., Wang, D., Han, J., Wei, Y., Deng, Y., Xiong, Y., Zhang, Z.: Decoding topologically associating domains with ultra-low resolution hi-c data by graph structural entropy. *Nature communications* **9**(1), 3265 (2018)
18. Li, L., Liu, Y., Zhou, A.: Hierarchical attention based position-aware network for aspect-level sentiment analysis. In: *CoNLL*. pp. 181–189 (2018)
19. Li, R., Chen, H., Feng, F., Ma, Z., Wang, X., Hovy, E.: Dual graph convolutional networks for aspect-based sentiment analysis. In: *ACL-IJCNLP*. pp. 6319–6329 (2021)
20. Li, X., Bing, L., Lam, W., Shi, B.: Transformation networks for target-oriented sentiment classification. In: *ACL*. pp. 946–956 (2018)
21. Liang, B., Su, H., Gui, L., Cambria, E., Xu, R.: Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowledge-Based Systems* **235**, 107643 (2022)
22. Liu, R., Cao, J., Sun, N., Jiang, L.: Aspect feature distillation and enhancement network for aspect-based sentiment analysis. In: *SIGIR*. p. 1577–1587 (2022)
23. Liu, Y., Liu, J., Zhang, Z., Zhu, L., Li, A.: Rem: From structural entropy to community structure deception. In: *NeurIPS*. p. 12938–12948 (2019)
24. Lu, Q., Sun, X., Sutcliffe, R., Xing, Y., Zhang, H.: Sentiment interaction and multi-graph perception with graph convolutional networks for aspect-based sentiment analysis. *Knowledge-Based Systems* **256**, 109840 (2022)
25. Ma, D., Li, S., Zhang, X., Wang, H.: Interactive attention networks for aspect-level sentiment classification. In: *IJCAI*. pp. 4068–4074 (2017)
26. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: *ACL*. pp. 55–60 (2014)
27. OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report (2024)
28. Peng, H., Zhang, J., Huang, X., Hao, Z., Li, A., Yu, Z., Yu, P.S.: Unsupervised social bot detection via structural information theory. *TOIS* **42**(6), 1–42 (2024)
29. Peng, K., Jiang, L., Peng, H., Liu, R., Yu, Z., Ren, J., Hao, Z., Yu, P.S.: Prompt based tri-channel graph convolution neural network for aspect sentiment triplet extraction. In: *SDM*. pp. 145–153. *SIAM* (2024)
30. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: *EMNLP*. pp. 1532–1543 (2014)
31. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S.M., Eryiğit, G.: SemEval-2016 task 5: Aspect based sentiment analysis. In: *SemEval 2016, ACL*. pp. 19–30 (2016)

32. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: SemEval-2014 task 4: Aspect based sentiment analysis. In: SemEval 2014, ACL. pp. 27–35 (2014)
33. Tan, X., Cai, Y., Zhu, C.: Recognizing conflict opinions in aspect-level sentiment classification with dual attention networks. In: EMNLP-IJCNLP. pp. 3426–3431 (2019)
34. Tang, H., Ji, D., Li, C., Zhou, Q.: Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In: ACL. pp. 6578–6588 (2020)
35. Tian, Y., Chen, G., Song, Y.: Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble. In: NAACL-HLT. pp. 2910–2922 (2021)
36. Wang, K., Shen, W., Yang, Y., Quan, X., Wang, R.: Relational graph attention network for aspect-based sentiment analysis. In: ACL. pp. 3229–3238 (2020)
37. Wang, Y., Huang, M., Xiaoyan, Z., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: EMNLP. pp. 606–615 (2016)
38. Wang, Y., Wang, Y., Zhang, Z., Yang, S., Zhao, K., Liu, J.: User: Unsupervised structural entropy-based robust graph neural network. In: AAAI. pp. 10235–10243 (2023)
39. Wang, Z., Xie, Q., Xia, R.: A simple yet effective framework for few-shot aspect-based sentiment analysis. In: SIGIR. p. 1765–1770 (2023)
40. Wu, H., Huang, C., Deng, S.: Improving aspect-based sentiment analysis with knowledge-aware dependency graph network. *Information Fusion* **92**, 289–299 (2023)
41. Wu, J., Chen, X., Xu, K., Li, S.: Structural entropy guided graph hierarchical pooling. In: ICML. pp. 24017–24030 (2022)
42. Yang, Y., Wu, Q., He, B., Peng, H., Yang, R., Hao, Z., Liao, Y.: Sebot: Structural entropy guided multi-view contrastive learning for social bot detection. In: KDD. pp. 3841–3852 (2024)
43. Yi, J., Wu, X., Liu, X.: Context-guided and syntactic augmented dual graph convolutional network for aspect-based sentiment analysis. In: ICASSP. pp. 12401–12405 (2024)
44. Zeng, X., Peng, H., Li, A.: Effective and stable role-based multi-agent collaboration by structural information principles. In: AAAI. pp. 11772–11780 (2023)
45. Zeng, X., Peng, H., Li, A.: Adversarial socialbots modeling based on structural information principles. In: AAAI. vol. 38, pp. 392–400 (2024)
46. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. In: EMNLP-IJCNLP. pp. 4568–4578 (2019)
47. Zhang, M., Qian, T.: Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In: EMNLP. pp. 3540–3549 (2020)
48. Zhang, W., Li, X., Deng, Y., Bing, L., Lam, W.: A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *TKDE* pp. 11019–11038 (2023)
49. Zhang, Z., Zhou, Z., Wang, Y.: SSEGCN: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In: NAACL-HLT. pp. 4916–4925 (2022)
50. Zhao, X., Peng, H., Dai, Q., Bai, X., Peng, H., Liu, Y., Guo, Q., Yu, P.S.: Rldgn: Reinforced dependency graph convolutional network for aspect-based sentiment analysis. In: WSDM. pp. 1–9 (2023)
51. Zhong, Q., Ding, L., Liu, J., Du, B., Jin, H., Tao, D.: Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. *TKDE* pp. 10098–10111 (2023)
52. Zou, D., Peng, H., Huang, X., Yang, R., Li, J., Wu, J., Liu, C., Yu, P.S.: Se-gsl: A general and effective graph structure learning framework through structural entropy optimization. In: Web Conference. p. 499–510 (2023)