

Article

Syntactic Structure-Enhanced Dual Graph Convolutional Network for Aspect-Level Sentiment Classification

Jiehai Chen , Zhixun Qiu , Junxi Liu, Yun Xue and Qianhua Cai *

School of Electronics and Information Engineering, South China Normal University, Foshan 528225, China; cjh_scnu@m.scnu.edu.cn (J.C.); qiu zhixun@m.scnu.edu.cn (Z.Q.); liujunxi@m.scnu.edu.cn (J.L.); xueyun@m.scnu.edu.cn (Y.X.)

* Correspondence: caiqianhua@m.scnu.edu.cn

Abstract: Aspect-level sentiment classification (ALSC) is a fine-grained sentiment analysis task that aims to predict the sentiment of the given aspect in a sentence. Recent studies mainly focus on using the Graph Convolutional Networks (GCN) to deal with both the semantics and the syntax of a sentence. However, the improvement is limited since the syntax dependency trees are not aspect-oriented and the exploitation of syntax structure information is inadequate. In this paper, we propose a Syntactic Structure-Enhanced Dual Graph Convolutional Network (SSEDGCN) model for an ALSC task. Firstly, to enhance the relation between aspect and its opinion words, we propose an aspect-wise dependency tree by reconstructing the basic syntax dependency tree. Then, we propose a syntax-aware GCN to encode the new tree. For semantics information learning, a semantic-aware GCN is established. In order to exploit syntactic structure information, we design a syntax-guided contrastive learning objective that makes the model aware of syntactic structure and improves the quality of the feature representation of the aspect. The experimental results on three benchmark datasets show that our model significantly outperforms the baseline models and verifies the effectiveness of our model.

Keywords: aspect-level sentiment classification; contrastive learning; graph convolutional networks

MSC: 68T07



Citation: Chen, J.; Qiu, Z.; Liu, J.; Xue, Y.; Cai, Q. Syntactic Structure-Enhanced Dual Graph Convolutional Network for Aspect-Level Sentiment Classification. *Mathematics* **2023**, *11*, 3877. <https://doi.org/10.3390/math11183877>

Academic Editor: Giuseppe Pirro

Received: 5 August 2023

Revised: 29 August 2023

Accepted: 10 September 2023

Published: 11 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Aspect-level sentiment classification (ALSC) is a fine-grained task in the field of sentiment analysis. The main purpose of ALSC is to identify the user-expressed sentiment polarity at aspect-level [1]. In practical use, ALSC not only focuses on analyzing opinions in a given text but also looks into the aspects and its sentiment, and thus, giving a much clearer understanding. For example, in the sentence “I am never disappointed at the system and software”, the sentiment toward aspects “system” and “software” are both positive (Figure 1).

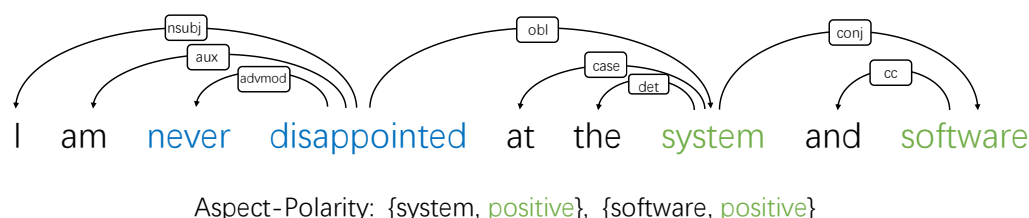


Figure 1. An example sentence with its dependency tree where aspects are highlighted in green and opinion words are highlighted in blue.

A key issue for addressing ALSC lies in modeling the relationship between the aspect and its opinion words. Early studies [2–8] principally focus on integrate attention mechanisms into recurrent neural networks (RNNs) to capture the opinion words that

semantically relate to the aspect. The devising of pre-trained model BERT allows to exploit abundant semantic information in ALSC tasks. Song et al. proposed an attention-based encoder and a BERT-SPC model to learn features of the aspect and its context words [9]. By contrast, the syntax-based models also pave a way for resolving ALSC issues for sentences of complex syntactic structures. The application of graph neural networks to syntax dependency tree encoding is most pronounced [10–17]. Both GCN and GAT are capable of learning syntactic-based feature representations by aggregating neighboring node features through node connection on a syntax dependency tree. Zhang and Song [10,11] took GCN to encode the syntax dependency tree and achieve satisfying results, indicating the introduction of sentence syntax benefits the relation modeling. Note that syntax-based models have deficiencies in syntactic-insensitive sentences; thus, the integration of syntax and semantics is developed [18,19]. Zhang et al. [20] defined a syntactic mask matrix and enhance the interaction between semantics and syntax by multiplying the syntactic mask matrix with the semantic-based attention matrix.

Despite the improvement in ALSC, existing GCN-based methods have two limitations. On the one hand, for syntax-based models, the syntax dependency trees generated by the parser are not aspect-oriented. As such, the dependency tree may miss important connections to the aspect. As presented in Figure 1, if the context toward aspect “software” is processed using one-layer GCN, only the information from “system” and “and” can be aggregated. By contrast, no connection is built between the aspect “software” with its opinion words “never disappointed”. Although an aspect-based dependency tree is proposed [16], the sentiment is still mis-identified due to its neglecting the information flow in the basic syntax dependency tree. Taking the words “never disappointed” as an example, if the syntactic connection between “never” and “disappointed” is disregarded, the sentiment polarity can be classified as negative. The main reason is that neither “never” nor “disappointed” expresses a positive sentiment. On the other hand, GCN aggregates node feature merely via the relation in the graph. Such GCN-based models mainly focus on node features and do not fully leverage syntactic structure, which may affect the performance of GCN-based models in ALSC.

Aiming to address the issues mentioned above, we propose Syntactic Structure-Enhanced Dual Graph Convolutional Network for ALSC task. To start with, an aspect-wise dependency tree is developed, which preserves the basic syntax and enhances the connection toward the aspect. The aspect-wise dependency tree is encoded via GCN with the integration of dependency relation aware attention mechanism. Likewise, the sentence semantics is also processed by GCN and self-attention mechanism. To fully consider the syntactic structure, we design a contrastive learning objective with a contrastive coefficient based on the syntactic structure similarity which is characterized by the distribution of anonymous walks. Training with this objective encourages the model to generate closer representations for those aspects with the same labels and similar syntactic structure, thereby further improving the classification performance.

The contributions of our work are summarized as follows:

- A novel aspect-wise dependency tree is established to overcome the deficiencies of a classical syntax dependency tree. The relation between aspect and its opinion words is enhanced, while the basic syntax is preserved.
- On the task of ALSC, the SSEDGCN model is proposed, which deals with both the syntax and the semantics. Moreover, a contrastive learning module is exploited to learn the feature representation of the aspect.
- Experimental results on three benchmark datasets reveal that our model is a competitive alternative comparable with the state of the art.

The paper is mainly divided into five sections. In the introduction, we present an overview of the article and propose our solution to address the limitations of current ALSC methods. Section 2 provides a summary of research related to our work. In Section 3, we offer a detailed description of the proposed model and its individual modules. In Section 4,

experiments and result analysis are performed on three public datasets. Finally, concluding remarks are given in Section 5.

2. Related Work

2.1. Aspect-Level Sentiment Analysis

As pointed out in the introduction, ALSC involves with classified the sentiment towards a given aspect according to a predefined set of sentiment polarities. Early research focuses on employing RNN-based method, together with the integration of attention mechanisms or knowledge distillation [21,22], to extract aspect-related information. In recent years, advances in GCN-based algorithms significantly improve the working performance in ALSC tasks. The ALSC methods can be loosely classified into the three categories, i.e., semantic-based models, syntactic-based models, and their integration. For semantic-based models, the most widely-used models are developed based on deep neural networks and attention mechanisms, which are capable of modeling the semantic relation between aspect and its contexts [2–8]. Another main focus is the exploiting of syntax [10–17], because the syntactic information can be applied to set the connection of aspect to its opinion words. On the task of ALSC, the GCNs are first employed in syntax-based models to encode the syntax dependency tree of the given sentence, and thus, model the syntactic relation between aspect and the contexts [10,11]. Wang et al. reconstructed the syntax dependency tree by pruning the irrelevant edges and encode the revised dependency tree using GAT [16]. Considering the interaction between syntactic and semantic information, multi-channel GCN-based approaches has been proposed to extract syntactic and semantic information [18,19,23]. Zhang et al. [20] generated an attention matrix and a syntactic mask matrix based on word-syntactic distances, which are processed to enhance the interaction between semantics and syntax.

2.2. Random Anonymous Walks

Given a graph $G = (V, E)$, with $V = \{v_1, \dots, v_{|V|}\}$ and $E = \{(v_i, v_j)\}$ separately representing the node set and the edge set, the random anonymous walk aims to capture the structural pattern of the graph [24]. Let $w = (v_1, v_2, \dots, v_l)$ be a random walk with length l and $(v_i, v_{i+1}) \in E$. We shall define the random anonymous walk with respect to w as:

$$aw(w) = (DIS(w, v_1), DIS(w, v_2), \dots, DIS(w, v_l))$$

where $DIS(w, v_i)$ represents the number of distinct nodes in w when v_i first appears in w . Specifically, the anonymous walks of length l can be written as $\omega_1^l, \omega_2^l, \dots$ in line with their lexicographical order. For example, the four-length anonymous walks can be given as $\omega_1^4 = (1, 2, 3, 1)$, $\omega_2^4 = (1, 2, 3, 2)$, $\omega_3^4 = (1, 2, 1, 3)$, etc.

On this occasion, an intuitive description of anonymous walks is also provided. Distinguishing from random walks, the essence of anonymous walks lies in denoting the underlying pattern of random walks, regardless of the specific node being visited. According to Figure 2, all three random walks can be mapped into one anonymous walk referring to certain structure, such as triadic closure.

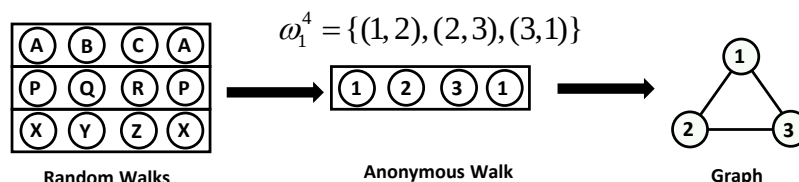


Figure 2. An example of anonymous walks.

2.3. Graph Neural Networks

Graph Neural Networks (GNNs) [25–27] have emerged as a powerful framework for learning representations of graph data. GNNs aim to learn expressive node representations by leveraging the graph structure and node features. Subsequently, researchers

have explored the combination of self-supervised learning and GNNs [28–30]. For instance, You [28] proposed a graph-contrastive learning framework that enhances graph representations through data augmentation. Wei [29] utilized contrastive learning and the information bottleneck principle to enhance recommendation systems based on graphs. In the field of Aspect-Level Sentiment Classification (ALSC), GNNs have demonstrated significant potential. Numerous studies [10–20] have employed GNNs to capture the syntactic information of sentences, where nodes represent words and edges indicate dependencies. This enables the generation of representation vectors for nodes based on their neighboring features. With the increasing performance of GNN-based models, it has been observed that they effectively enhance contextual and aspectual dependencies in ABSA tasks through the utilization of GNNs.

2.4. Contrastive Learning

In recent years, advances in contrastive learning algorithms significantly improve the working performance in a variety of tasks. The main purpose of contrastive learning is to pull an anchor and a positive sample closer in the feature space while pushing apart the anchor with negative samples. Motivated by its success in computer vision tasks [31–33], research in contrastive learning-based methods for NLP primarily focuses on the learning of representations. Gao et al. [34] proposed a contrastive learning framework to derive the sentence embeddings, aiming at improving the precision of sentence representations through contrastive learning. Wang et al. [35] applied a contrastive learning-enhanced KNN mechanism to Multi-Label Text Classification (MLTC) task, based on which the representation quality of the retrieved neighbors can be upgraded. For the ALSC task, Liang et al. [36] took contrastive learning to extract aspect-invariant and aspect-dependent features to discriminate the sentiment features among the sentiment pattern and polarity perspectives.

3. Methodology

Figure 3 shows the architecture of the SSEDGCN model. The proposed model consists of five major components, i.e., a Sentence Encoder, a Syntax-aware GCN, a Semantic-aware GCN, a Syntax-guided contrastive learning module, and a Sentiment classifier. More details of each component are described as follows. We start with an aspect-oriented dependency tree constructed in our model.

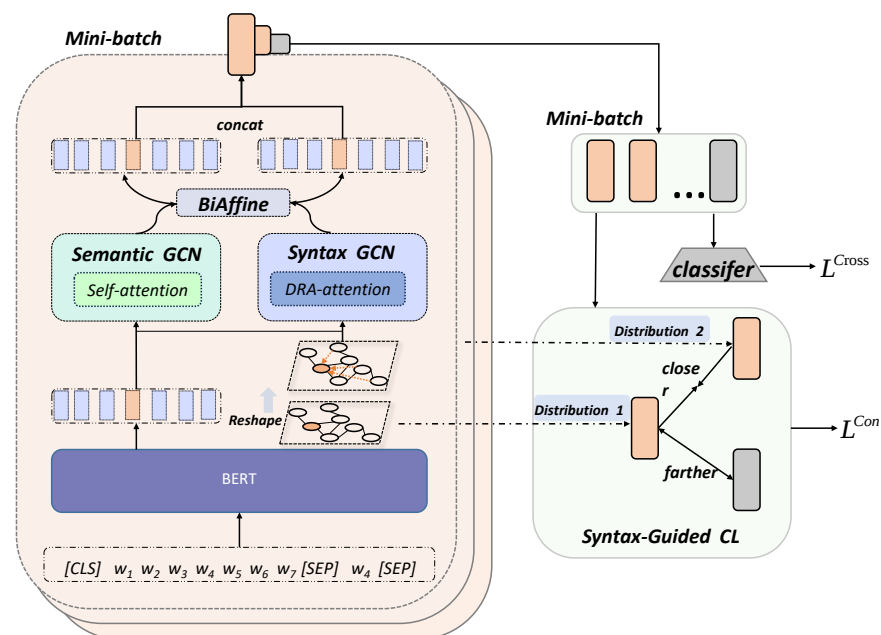


Figure 3. The overall architecture of SSEDGCN.

3.1. Aspect-Wise Dependency Tree

In syntax-based ALSC methods, most existing models [11,17,37] tend to derive the sentence syntactic information by processing the dependency relation. Typically, the syntax dependency tree that depicts the sentence dependency relation is directly developed via dependency parsers. Nevertheless, such dependency trees are not aspect-oriented, in which case the aspect word may not connect to its opinion words in line with the sentence syntax dependency. Notably, the ALSC task aims to assign the predominant focus to the aspect instead of the root nodes of the syntax dependency tree. For this reason, an aspect-wise dependency tree is devised based on the classical syntax dependency tree. Specifically, we designate the aspect as the root node and connect all other nodes to it. Concurrently, we assign syntactic dependency relationships to these connecting edges, based on their respective syntactic distances.

Figure 4 shows the basic syntax dependency tree of the sentence and the aspect-wise dependency tree. Comprehensively, we start with obtaining the classical sentence syntax dependency tree via syntactic parser. Then, each unidirectional connection is revised to a bidirectional relation. The aspect-wise dependency tree is established by setting the aspect as the root node of the basic syntax dependency tree. With respect to an aspect composed of multiple words, all these words are considered as one root node. For nodes that have no direct syntactic connection with the aspect in the basic syntax dependency tree, a directional relationship to the aspect is developed. According to Figure 4, the directional relation between the node and the aspect is denoted as [n: con] where n represents the shortest distance between the two nodes in the original syntactic tree. For example, in the sentence “I love the system and the software”, the relation between “software” and “and” is defined as [2: con]. The reason is that these words are not exactly connected to each other in the basic syntax dependency tree, but the word “software” reaches “and” through the path of *software-system-and*. That is, the distance between “software” and “and” in the basic syntactic dependency tree is 2, so the connection between them is written as [2: con].

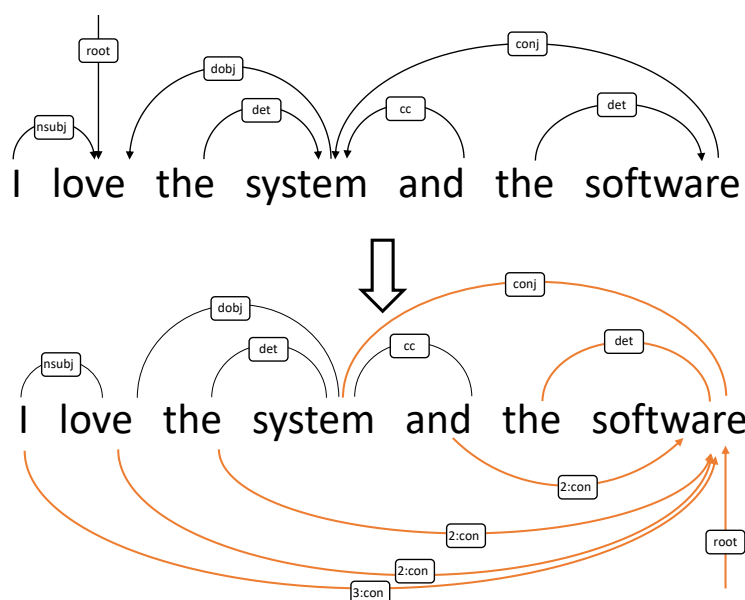


Figure 4. Aspect-wise dependency tree.

Two primary superiorities can be observed from the aspect-wise dependency tree. For one thing, each aspect retains a distinctive dependency tree, avoiding the condition that multiple aspects in one sentence possess the same syntax dependency tree. For another, every single node directionally relates to the aspect, which enhances the connection between the opinion word and the aspect word.

3.2. Sentence Encoder

Let $s = \{w_1, w_2, \dots, w_n\}$ be a n -word sentence with aspect $a = \{w_{start}, w_{start+1}, \dots, w_{end}\}$, where $start$ and end represent the starting index and the ending index of a , respectively. In light of the recent success of BERT [38] in word embedding, we employ a pre-trained BERT model to obtain the hidden representation of the sentence. The sequence “[CLS] s [SEP] a [SEP]” is generated and sent to BERT encoder. Thus, we obtain the contextual feature vector $H = \{h_1, h_2, \dots, h_n\}$:

$$H = \text{BERT}\{[\text{CLS}]s[\text{SEP}]a[\text{SEP}]\} \quad (1)$$

where $H \in R^{n \times d}$ and d represent the hidden layer dimension. Notably, the subvector $h_a = \{h_{a1}, h_{a2}, \dots, h_{am}\}$ indicates the aspect representation.

3.3. Syntax-Aware GCN with DRA Attention

As previously described [10–12], we transform the aspect-wise dependency tree into a syntactic graph $G_{syn}(A^{syn}, H)$, where $A^{syn} \in R^{n \times n}$ refers to the adjacency matrix of the graph and $H = \{h_1, h_2, \dots, h_n\}$ is the hidden representation of the sentence. We take $H_{syn}^{(0)}$ as the initialized feature representation of nodes and feed it into the syntax-aware GCN.

Normally, A GCN iteratively updates a node representation by averagely aggregating neighboring nodes, which is carried out based on the element in the adjacency matrix. Specifically, the element A_{ij}^{syn} stands for the relation between node i and node j . Only if $A_{ij}^{syn} = 1$ are the two nodes regarded as connected, whereas since the dependencies of different neighboring nodes toward the aspect vary, the specific dependency relation and its effect have to be considered during aggregating. Thereby, dependency relation aware attention (DRA attention) is performed in the process of aspect node updating. In such a manner, the dependency weights are assigned to the aspect-neighboring nodes for information aggregation. We map the dependency relations of the aspect node and its neighboring nodes into vector representations. The dependency relation between nodes is, thus, incorporated into the graph convolution. The attentive weights are taken to control the information flow from neighboring nodes to the aspect. Computation with DRA attention is conducted using the following equations:

$$H_{syn}^{(0)} = \{h_1^{syn,0}, h_2^{syn,0}, \dots, h_n^{syn,0}\} = H \quad (2)$$

$$h_i^{syn,l} = \sigma\left(\sum_{j=1}^n A_{ij}^{syn'} W_{syn}^l h_j^{syn,l-1} + b_{syn}^l\right) \quad (3)$$

$$A_{ij}^{syn'} = \begin{cases} A_{ij}^{syn} & i \notin (start, end) \\ \delta_{ij} & i \in (start, end) \end{cases} \quad (4)$$

$$\delta_{ij} = \frac{\exp(g_{ij})}{\sum_{j=1}^{N_i} \exp(g_{ij})} \quad (5)$$

$$g_{ij} = \sigma(\text{relu}(r_{ij}W_1 + b_1)W_2 + b_2) \quad (6)$$

where r_{ij} indicates the dependency relation between words w_i and w_j , $h_i^{syn,l} \in R^d$ is the embedding of node i in the l -th GCN layer, δ_{ij} is a normalized attention coefficient for the DRA attention mechanism, W_{syn}^l and b_{syn}^l are the trainable parametric matrix and bias in the l -th GCN layer, respectively, and W_1 , W_2 and b_1 , b_2 are the trainable parametric matrices and biases for computing the attention weights.

3.4. Semantic-Aware GCN with Self-Attention

The working performance of ALSC methods is not just affected by the syntax, but also the semantics [20]. In this way, the semantic-aware module is introduced. Similar to the syntax-aware GCN, we define $G_{sem}(A^{sem}, H)$ as the semantic graph, with $A^{sem} \in R^{n \times n}$ as

the semantic graph adjacency matrix. $H = \{h_1, h_2, \dots, h_n\}$ is also taken as the initialized feature representations $H_{sem}^{(0)}$ of nodes in the semantic-aware GCN. With respect to semantic information learning, the self-attention mechanism is exploited to build the semantic adjacency matrix $A^{sem} \in R^{n \times n}$. The idea behind this approach is that the attention matrix generated by self-attention can represent semantic correlations between words.

$$H_{sem}^{(0)} = \{h_1^{sem,0}, h_2^{sem,0}, \dots, h_n^{sem,0}\} = H \quad (7)$$

$$A^{sem} = \text{softmax}\left(\frac{(H_{sem}^{(0)} W_{sem,k})(H_{sem}^{(0)} W_{sem,q})^T}{\sqrt{d}}\right) \quad (8)$$

$$h_i^{sem,l} = \sigma\left(\sum_{j=1}^n A_{ij}^{sem} W_{sem}^l h_j^{sem,l-1} + b_{sem}^l\right) \quad (9)$$

where $h_i^{sem,l} \in R^d$ is the embedding of node i in the l -th GCN layer, d is the dimension of the node feature representation, $W_{sem,k}$, $W_{sem,q}$ are trainable parametric matrices for self-attention mechanism, and W_{sem}^l and b_{sem}^l are trainable parametric matrix and bias of the l -th layer in GCN, respectively.

3.5. Sentiment Classifier

For the purpose of information interaction, the BiAffine mechanism [17,18] is exploited to integrate the syntactic and semantic information in advance to sentiment classification. The processing of outputs from syntax-aware GCN and semantic-aware GCN are presented in Equations (10) and (11):

$$H_{sem}^{(l)'} = \text{softmax}(H_{sem}^{(l)} W_3 (H_{syn}^{(l)})^T) H_{syn}^{(l)} \quad (10)$$

$$H_{syn}^{(l)'} = \text{softmax}(H_{syn}^{(l)} W_4 (H_{sem}^{(l)})^T) H_{sem}^{(l)} \quad (11)$$

where W_3 and W_4 are trainable parametric matrices.

The pooling operation $f(\cdot)$ is performed to derive the semantic representation h_a^{sem} and the syntactic representation h_a^{syn} of the aspect. Then, h_a^{sem} and h_a^{syn} are concatenated to obtain the final aspect representation r , which is fed into the linear layer and the softmax classifier to identify the sentiment polarity of the aspect.

$$h_a^{sem} = f(\{h_{a1}^{sem}, h_{a2}^{sem}, \dots, h_{am}^{sem}\}) \quad (12)$$

$$h_a^{syn} = f(\{h_{a1}^{syn}, h_{a2}^{syn}, \dots, h_{am}^{syn}\}) \quad (13)$$

$$r = [h_a^{sem}, h_a^{syn}] \quad (14)$$

$$\hat{y}(a) = \text{softmax}(W_y r + b_y) \quad (15)$$

where W_y and b_y are the trainable parametric matrix and the bias, respectively, and $\{h_{a1}^{sem}, h_{a2}^{sem}, \dots, h_{am}^{sem}\} \in R^{m \times d}$ and $\{h_{a1}^{syn}, h_{a2}^{syn}, \dots, h_{am}^{syn}\} \in R^{m \times d}$ stand for aspect sequence derived from $H_{sem}^{(L)'} \in R^{n \times d}$ and $H_{syn}^{(L)'} \in R^{n \times d}$.

3.6. Syntax-Guided Contrastive Learning

Inspired by [32,35], the syntax-guided contrastive learning is designed to learn the feature representation of the aspect. Specifically, we design a contrastive learning objective with a contrastive coefficient based on the syntactic structure similarity for positive instance pair. For each input aspect, the remaining aspects with the same sentiment polarity within the same batch are considered as the positive examples, otherwise as negative examples.

Then, we use the distribution of anonymous walks to characterize syntactic graph structures since the anonymous walk is a powerful tool that represents graph structural patterns. Subsequently, we employ KL divergence to compute the differences in the anonymous walk distribution of positive samples, such as $f_G(i)$ and $f_G(j)$ in Figure 5. The resulting KL divergence is then used as the contrast coefficient β_{ij} in contrastive learning to measure the similarity of syntactic structure between two positive samples.

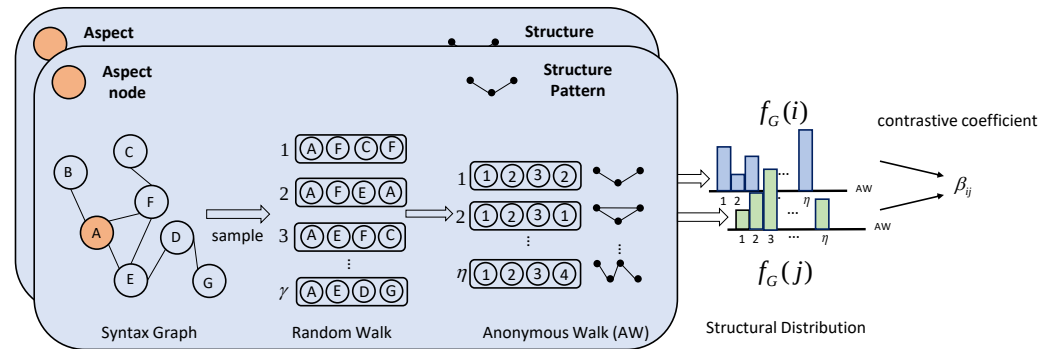


Figure 5. The local syntactic structure distribution.

As reported by [39], the number of anonymous walks with length l in a given graph is fixed. For instance, there are only two kinds of anonymous walks ω^2 with length 2, i.e., $\omega_1^2 = (1, 1)$ and $\omega_2^2 = (1, 2)$. In our model, we take the aspect node as the starting node of the sentence s in the syntactic graph. As Figure 5 show, a set of γ random walks RW with the length of l is sampled, which corresponds to a set of η different anonymous walks ($\omega_1^l, \omega_2^l, \dots, \omega_\eta^l$). The empirical distribution of these γ random walks over the η anonymous walks is thus determined, which is:

$$f_G(s) = (p_s(\omega_1^l), p_s(\omega_2^l), \dots, p_s(\omega_\eta^l)) \quad (16)$$

$$p_s(\omega_i^l) = \frac{\sum_{w \in RW} \mathbb{I}(aw(w) = \omega_i^l)}{\gamma}, \quad i = 0, 1, \dots, \eta \quad (17)$$

where $p_s(\omega_i^l)$ refers to the probability of an anonymous walk ω_i^l with the starting point of the aspect node, $\mathbb{I}$ is the indicator function, and $f_G(s)$ denotes the distribution of γ random walks over η anonymous walks.

Given an anchor aspect r_i , we designate other aspect with the same sentiment $P = \{j \in B, y_j = y_i, j \neq i\}$ as the positive example set of r_i , and otherwise as a negative example set. The syntax-guided contrastive loss is written as:

$$L^{Con} = \frac{1}{B} \sum_{i \in B} \frac{1}{|P|} \sum_{j \in P} \mathcal{L}(r_i, r_j) \quad (18)$$

$$\mathcal{L}(r_i, r_j) = -\beta_{ij} \log \frac{\exp(\text{sim}(r_i, r_j)/\tau)}{\sum_{k \in B, k \neq i} \exp(\text{sim}(r_i, r_k)/\tau)} \quad (19)$$

$$\beta_{ij} = \frac{C(i, j)}{\sum_{m \in P} C(i, m)} \quad (20)$$

$$C(i, j) = \frac{1}{KL(f_G(i) \parallel f_G(j))} \quad (21)$$

where B indicates the total number of samples in the mini-batch, τ is the temperature coefficient in contrastive learning, and $\text{sim}(r_i, r_j)$ is taken to calculate the cosine similarity of the vectors r_i and r_j . For the positive example pair (r_i, r_j) , the more similar the local structure of the aspect nodes is, the smaller the KL divergence can be calculated, and hence, the larger $C(i, j)$ is determined. Furthermore, a larger $C(i, j)$ results in a larger contrastive

coefficient β_{ij} , which enlarges the loss term $\mathcal{L}(r_i, r_j)$ of (r_i, r_j) . In this way, the distance between their representation vectors will be optimized to be closer.

3.7. Model Training

The loss function in our model is given by:

$$L(\theta) = L^{Cross} + \lambda L^{Con} \quad (22)$$

$$L^{Cross} = - \sum_{(s,a) \in \mathcal{D}} \sum_{c \in \mathcal{C}} \log \hat{y}(a) \quad (23)$$

where θ refers to the set of trainable parameters, λ is the contrastive learning loss weight, L^{Cross} is the cross entropy loss function for ALSC task, $\mathcal{D}$ is the set of all sentence-aspect pairs, and $\mathcal{C}$ is the class of sentiment polarity.

4. Experiment

4.1. Datasets

We evaluate our model on three publicly available benchmark datasets, i.e., Restaurant and Laptop from SemEval 2014 Task 4 [40] and Twitter [41]. Each sample contains a sentence, with the aspect (single or multiple words) and its corresponding sentiment polarity (positive, negative or neutral). The statistics of the datasets are shown in Table 1.

Table 1. Statistics of the experimental datasets.

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

4.2. Implementation Details

In this experiment, the Stanford parser (<https://stanfordnlp.github.io/CoreNLP/>, accessed on 2 November 2022) is employed to build the basic syntax-dependent tree, based on which the aspect-wise dependency tree is re-constructed. The pre-trained bert-base-uncased model is used for sentence encoding. Notably, the dimension of the relation embedding between aspect nodes and the remaining nodes is set to 100 while the hidden layer dimension for both GCN modules is 384. In addition, the layer number for SynGCN and SemGCN is set to 2 to prevent overfitting. The dropout function, with a dropout rate of 0.1, is used in both modules. With respect to anonymous walks, the length l is 6 and the sampling number γ is 100. The weight λ of contrastive loss for contrastive learning scheme is set to 0.4, 0.6, and 0.1 for the three datasets. Besides, the Adam optimizer is adopted with a learning rate of 0.002. The SSEDGCN model is trained for 20 epochs with a batch size of 16. We use accuracy (Acc) and Macro-F1 (F1) to evaluate the model performance, which are the primary evaluation measures used in ALSC models.

4.3. Baseline Methods

The following methods are taken as the baselines in the experiment:

- **CDT+BERT** [11]: The first model that employs GCN to learn syntactic information based on dependency trees and generate aspect node representation. The model is trained for 20 epochs with a batch size of 16 and a learning rate of 0.02.
- **BERT-SPC** [9]: The sentence-aspect pair is fed into BERT model, whose token output is used for sentiment classification.

- **R-GAT+BERT** [16]: An aspect-specific dependency tree is constructed, which is further encoded using the relational graph attention network. The model is trained for 30 epochs with a batch size of 16 and a learning rate of 0.00005.
- **DGEDT+BERT** [17]: A dual-transformer structure is devised, which learns the flat representations via the Transformer and the graph-based representations via the corresponding dependency graph in an iterative interaction manner. The model is trained for 50 epochs with a batch size of 32 and a learning rate of 0.001.
- **T-GCN+BERT** [37]: A type-aware graph convolutional network is proposed to encode distinguishing dependency types, while the attentive layer ensemble is used to learn the contextual information within GCN layers. The model is trained for 30 epochs with a batch size of 16 and a learning rate of 0.00002.
- **DualGCN+BERT** [18]: A dual-channel GCN is developed to tackle both the syntax and the semantics. The orthogonal regularization and differential regularization are also applied. The model is trained for 20 epochs with a batch size of 16 and a learning rate of 0.002.
- **DMGCN+BERT** [19]: A dynamic and multi-channel graph convolution network is proposed to encode the syntactic and semantic information for aspect-based sentiment analysis. The model is trained for 50 epochs with a batch size of 32 and a learning rate of 0.001.
- **TCL+BERT** [42]: A triple contrastive learning network, focusing on achieving the alignment of syntactic and semantic features for sentiment classification. The model is trained for 15 epochs with a batch size of 16 and a learning rate of 0.00002.
- **SSEGCN+BERT** [20]: Both the semantic information and the syntactic information is comprehensively learned while the word representations are enhanced via GCN. The model is trained for 15 epochs with a batch size of 16 and a learning rate of 0.00002.
- **DR-BERT** [43]: A dynamic re-weighting BERT model is built, which tends to learn the dynamic aspect-oriented semantic information.
- **SSK-GAT+BERT** [23]: A novel graph attention network model is proposed to incorporate syntactic, semantic, and knowledge-based features. The model is trained for 15 epochs with a batch size of 16 and a learning rate of 0.000022.
- **HD-GCN+BERT** [22]: A hierarchical graph convolutional network is proposed to fuse the outputs of multiple GCN layers as the final representation for prediction. The model is trained for 20 epochs with a batch size of 16 and a learning rate of 0.002.

4.4. Comparison of Results

The overall results on three benchmark datasets are recorded in Table 2. We provide accuracy and Macro-F1 score as evaluation metrics. Among all the methods, SSEGCN obtains the best and most consistent results in sentiment classification accuracy. In contrast with classical syntax dependency tree-based methods (DGEDT and T-GCN), our model consistently dominates these baselines. The main reason is that the relation between aspect and its opinion words is enhanced in the aspect-wise dependency tree, which facilitates the sentiment classification. Comparing with R-GAT+BERT, which also concerns the encoding of aspect-oriented dependency tree, our model shows its superiority in all evaluation settings. Furthermore, the proposed model outperforms the state-of-the-art methods that deal with semantics and syntax (i.e., DualGCN and SSEGCN). In addition, our model also exceeds the contrastive-learning network that tackles the overall alignment (i.e., TCL+BERT), indicating the significance of explicitly considering syntactic structural information. Therefore, SSEGCN presents a more precise way to extract the aspect-related information and model both the semantic and syntactic relation between aspect and its corresponding opinion words, which validates the effectiveness of our model.

Table 2. Experimental results comparison on three publicly available datasets.

Models	Restaurant		Laptop		Twitter	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
CDT+BERT [11]	86.24	80.66	80.06	76.17	77.10	75.90
BERT-SPC [9]	84.46	76.98	78.99	75.03	73.55	72.14
R-GAT+BERT [16]	86.60	81.35	78.21	74.07	76.15	74.88
DGEDT+BERT [17]	86.30	81.35	79.80	75.60	77.90	75.40
T-GCN+BERT [37]	86.16	79.95	80.88	77.96	76.45	75.25
DualCGN+BERT [18]	87.13	81.16	81.80	78.10	77.40	76.02
DMCGN+BERT [19]	87.66	82.79	80.22	78.10	78.06	77.06
TCL+BERT [42]	87.40	82.12	81.80	78.96	77.55	76.57
SSEGCN+BERT [20]	87.31	81.09	81.01	77.96	77.40	76.02
DR-BERT [43]	87.72	82.31	81.45	78.16	77.24	76.10
SSK-GAT+BERT [23]	87.41	81.65	80.25	75.85	75.72	74.44
HD-GCN+BERT [22]	87.13	81.40	81.80	78.88	77.34	76.12
SSEDGCN(Ours)	87.85	82.08	82.91	80.12	78.14	77.08

4.5. Ablation Study

We carry out an ablation study to further analyze the impact of different components in SSEDGCN; see Table 3. The results show that the most significant module for our method is SynGCN. The removal of SynGCN results in the largest accuracy for the datasets Restaurant and Laptop, which indicates the effectiveness of syntactic feature extraction in ALSC. Moreover, the ablating of dependency relation aware attention (DRA attention) in SynGCN also degrades the performance, especially for Twitter. A possible explanation is that the colloquial expression in Twitter introduces an amount noise, which confuses the model in relation setting without the DRA attention. In addition, removing SemGCN lead to different degrees of model performance degradation, which suggests that the working performance of ALSC methods is not just affects by the syntactic information, but also the semantic information. As for the “w/o SGCL” model, the accuracy decreases 0.83%, 1.18%, and 0.9% on Restaurant, Laptop, and Twitter, respectively, demonstrating the capability of contrastive learning in obtaining the aspect representation.

Table 3. Ablation study results.

Models	Restaurant		Laptop		Twitter	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
SSEDGCN	87.85	82.08	82.91	80.12	78.14	77.08
w/o SGCL	87.02	81.46	81.73	78.61	77.24	76.12
w/o SemGCN	86.56	80.50	81.24	77.23	77.10	76.34
w/o DRA attention	86.33	79.92	81.63	77.77	76.69	75.56
w/o SynGCN	85.88	79.25	80.94	77.45	76.54	75.42

To validate the effectiveness of the aspect-wise dependency tree, we conducted comparative experiments based on the two types of dependency trees. The implementation results are shown in Table 4. From Table 4, it can be observed that the model based on the aspect-wise dependency tree outperform the model based on the classical dependency tree on all three datasets. This indicates the superiority of the aspect-wise dependency tree proposed in our work.

Table 4. Results of SSEDGCN based on two different dependency trees.

Dependency Tree	Models	Restaurant		Laptop		Twitter	
		Accuracy	F1	Accuracy	F1	Accuracy	F1
Aspect-wise	SSEDGCN	87.85	82.08	82.91	80.12	78.14	77.08
Classic	SSEDGCN	86.06	79.78	81.48	78.14	77.59	76.13

4.6. Attention Visualization

To validate whether the DRA attention mechanism effectively helps aspect nodes capture important nodes, we visualized the attention weight distribution. The experimental results are shown in Figure 6, where brighter colors indicate higher weights and vice versa. Additionally, to compare the effectiveness of the DRA attention mechanism, we also compared it with the classical attention mechanism. In this experiment, the classical attention mechanism refers to using the aspect as a query to calculate attention scores based on the interaction between the aspect and other tokens.

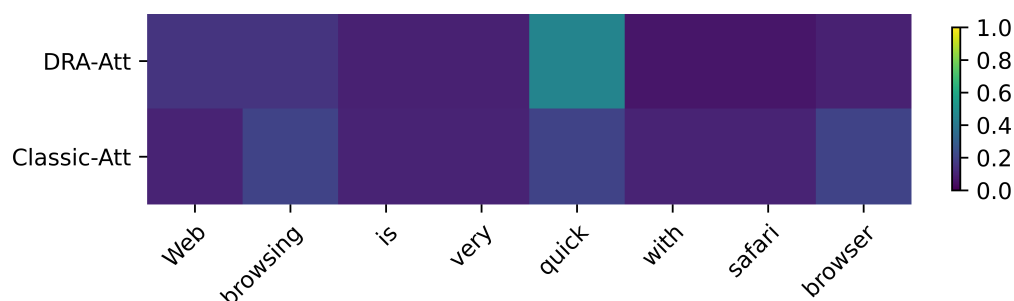
**Figure 6.** Attention visualization for DRA Attention (DRA-Att) and Classic Attention (Classic-Att) for the aspect word ‘web browsing’.

Figure 6 shows the attention weight distribution. It can be observed that, after using the DRA attention mechanism, the opinion word “quick” receives a higher attention score. This indicates that the introduction of syntactic dependency relationships helps aspect nodes capture important nodes. Furthermore, it can be seen that the weight distribution of the classical attention mechanism is more uniform, and the attention score for the opinion word “quick” is only slightly higher than other tokens. We believe that this is because the semantic correlation between “web browsing” and “browser” is high, which leads to an increased attention score for “browser” and reduces the focus on “quick”.

4.7. Impact of the GCN Layer Number

To examine the effect of varying the number of GCN layers, we conducted experiments with the number of GCN layers ranging from 1 to 8, and present the results in Figure 7. The results indicate that our model achieves best performance with two GCN layers, which we adopt in our model. It is noteworthy that a single-layer GCN yields suboptimal accuracy and F1, suggesting that a single layer is insufficient to effectively leverage syntactic and semantic information. Furthermore, we observe a decline trend in performance when the number of GCN layers is excessively large, indicating that an indiscriminate increase in the number of GCN layers may hinder the model’s learning ability due to the sharp increase parameters.

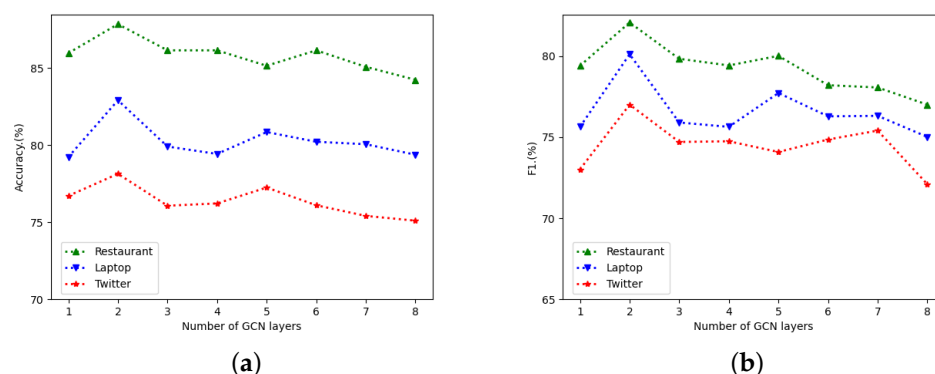


Figure 7. Effect of the number of GCN layers (a) Accuracy performance based on different numbers of GCN layers. (b) F1-score performance based on different numbers of GCN layers.

4.8. Impact of the Training/Testing Data Ratio

To investigate the impact of the training/testing data ratio the accuracy, we have included a supplementary experiment using the Restaurant dataset. The initial ratio in the Restaurant dataset is 3.22/1, meaning that the training set contains 3.22 times more samples than the testing set. To better understand the influence of the training/testing data ratio on accuracy, we gradually reduced this ratio by 20% increments. The purpose of this approach is to observe whether there are any changes in model accuracy as the ratio decreases. The experimental results are shown in the Table 5.

Table 5. Impact of the training/testing data ratio on accuracy.

Train:Test	Accuracy (%)
3.22:1	87.85
2.58:1	84.72
1.93:1	84.53
1.29:1	73.02
0.64:1	64.96

Based on our experimental results, we observed a consistent decrease in accuracy as the ratio decreased. We believe that this trend can be attributed to the reduced size and diversity of the training set when the ratio decreases. Consequently, the model receives fewer samples during the training process, which may limit its ability to fully capture the complexity and variability of the data. As a result, the performance of the model is compromised.

4.9. Case Study

The impacts of aspect-wise dependency tree and the proposed model for ALSA are investigated on case examples in the Table 6. To this end, we visualize the aspect and the predicted sentiment polarity. Specifically, the CDT model, which is devised based on classical syntax dependency tree, is taken for comparison. In the first sentence, both CDT and our model set the connection between the aspect “food” and its opinion word “enjoyed”. For the second sentence, the distance between aspect “sandwiches” and the sentiment word “good” is as long as four hops, as indicated by the red line in Figure 8a. Therefore, the two-layer GCN model (i.e., CDT) fails to capture this valuable word and misidentifies the sentiment as neutral. By contrast, our model is capable of establishing a direct syntactic relation of the sentiment word toward the aspect, as indicated by the red line in Figure 8b. In the third sample, more importance is attached to the noise word “tiny” instead of the opinion word “fast”, which causes the misclassification. In the proposed aspect-wise dependency tree, both “tiny” and “fast” relate to the aspect word “service”. The exploiting of DRA attention assigns more attention weights to the word “fast”, based on which the sentiment is classified as positive. Lastly, no explicit sentiment word is provided in the last sentence. Under this condition, both syntactic and semantic information

has to taken into account. It is clear that our model identifies the positive polarity to the aspect with the enhancing of sentence semantics.

Table 6. Case study. ALSC results of CDT and SSEDGCN on testing examples, along with their predictions. The marker ✓ and ✗ indicate the correct classification and incorrect classification, respectively.

#	Testing Instance	CDT	Ours
1	I definitely enjoyed the food as well	pos ✓	pos ✓
2	Most of the sandwiches are made with soy mayonaisse which is actually pretty good	neu ✗	pos ✓
3	tiny restaurant with very fast service	neg ✗	pos ✓
4	and the food, well the food will keep you coming back	neg ✗	pos ✓

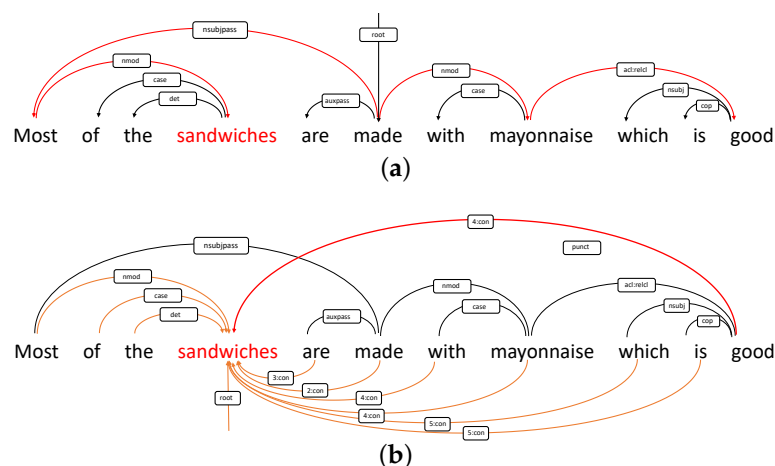


Figure 8. Visualization of dependency trees in the case study. (a) Classical dependency tree of the second sentence. (b) Aspect-wise dependency tree of the second sentence.

4.10. Analysis of the Contrastive Coefficient

The contrastive coefficients in SGCL are studied to clarify its significance. The working performance on ALSC with and without contrastive coefficient is presented in Table 7. One can easily see that the proposed model overperforms the one using simply contrastive learning without contrastive coefficient to a large extent. Accordingly, the contrastive coefficient plays a pivot role in learning the word syntactic structure of ALSC tasks.

Table 7. Experimental results with and without the contrastive coefficient.

Models	Restaurant		Laptop		Twitter	
	Accuracy	F1	Accuracy	F1	Accuracy	F1
SSEDGCN (w/ ω)	87.85	82.08	82.92	80.12	78.14	77.08
SSEDGCN (w/o ω)	87.13	81.54	82.07	78.76	77.34	76.89

The way of enhancing sentiment feature to the aspect using SGCL is presented as well. We visualize the obtained aspect vectors generated on the Restaurant test set using t-SNE [44] for SSEDGCN (w/o CL&SGCL), SSEDGCN (w/ CL), and SSEDGCN (w/ SGCL), respectively. The sentiment classification results of three categories of methods are shown in Figure 9. The dots in red, green, and blue correspond to the sentiment polarities of positive, negative, and neutral, respectively. Without contrastive learning, a certain proportion of the aspect nodes of different sentiment polarities overlap with each other, as presented in Figure 9a. With the application of basic contrastive learning scheme, the aspect vectors are partially distinguished, as presented in Figure 9b. Furthermore, it can be observed that the separations of representations derived from SGCL are significantly clearer than that

generated by basic contrastive learning scheme among different sentiment polarities. The experimental result verifies not just the importance of contrastive coefficient in SGCL, but also the superiority of SGCL in learning aspect representations.

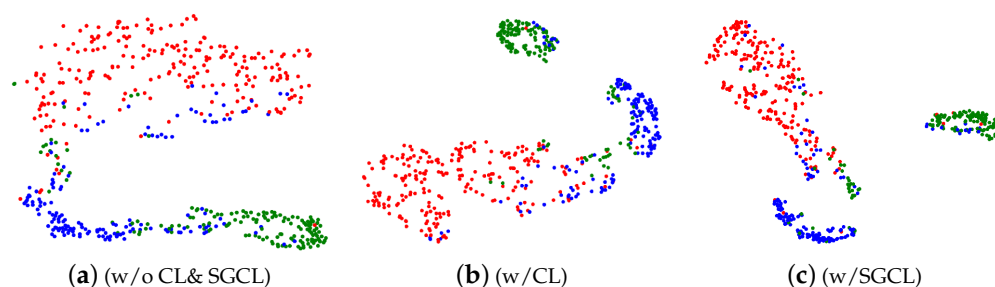


Figure 9. Visualization of sentiment classification results. SSEDGCN (w/o CL& SGCL) refers to the model without contrast learning and syntactic guidance, SSEDGCN (w/CL) refers to the use of contrastive learning, and SSEDGCN is (w/SGCL) is the model with SGCL.

5. Conclusions

In this work, we propose a Syntactic Structure-Enhanced Dual Graph Convolutional Network which focuses on effectively encoding comprehensive syntax information for ALSC. To start with, a novel aspect-wise dependency tree is proposed by reconstructing the basic syntax dependency tree. Then, the sentence syntax and semantics are encoded via two specific GCNs: a self-attention mechanism is adopted by SemGCN and a DRA-attention mechanism is adopted by SynGCN. Moreover, to fully consider the syntactic structural information, the syntax-guided contrastive learning is designed to learn the feature representation of the aspect. The experimental results on three benchmark datasets demonstrate the effectiveness of our proposed method, achieving state-of-the-art performance in terms of accuracy and macro-F1. Moreover, the ablation study and visual analysis have validated the role of the new tree structure and each component in our proposed model.

Author Contributions: Conceptualization, J.C. and Y.X.; methodology, J.C.; formal analysis, J.C. and Z.Q.; writing—original draft preparation, J.C. and Z.Q.; writing—review and editing, J.C. and J.L.; supervision, Q.C. and Y.X.; funding acquisition Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011370, the Characteristic Innovation Projects of Guangdong Colleges and Universities (No. 2018KTSCX049), and the Science and Technology Plan Project of Guangzhou under Grant No. 202102080258.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Al-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2016; pp. 19–30.
2. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 606–615.
3. Tang, D.; Qin, B.; Liu, T. Aspect Level Sentiment Classification with Deep Memory Network. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 214–224.
4. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 4068–4074.

5. Chen, P.; Sun, Z.; Bing, L.; Yang, W. Recurrent attention network on memory for aspect sentiment analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 452–461.
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, Brussels, Belgium, 31 October 2018–4 November 2018; pp. 3433–3442.
7. Huang, B.; Ou, Y.; Carley, K.M. Aspect level sentiment classification with attention-over-attention neural networks. In Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation, Washington DC, USA, 10–13 July 2018; pp. 197–206.
8. Gu, S.; Zhang, L.; Hou, Y.; Song, Y. A position-aware bidirectional attention network for aspect-level sentiment analysis. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 774–784.
9. Song, Y.; Wang, J.; Jiang, T.; Liu, Z.; Rao, Y. Attentional encoder network for targeted sentiment classification. *arXiv* **2019**, arXiv:1902.09314.
10. Zhang, C.; Li, Q.; Song, D. Aspect-based sentiment classification with aspect-specific graph convolutional networks. *arXiv* **2019**, arXiv:1909.03477.
11. Sun, K.; Zhang, R.; Mensah, S.; Mao, Y.; Liu, X. Aspect-level sentiment analysis via convolution over dependency tree. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5679–5688.
12. 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 (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5469–5477.
13. Zhang, M.; Qian, T. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Punta Cana, Dominican Republic, 8–12 November 2020; pp. 3540–3549.
14. Chen, C.; Teng, Z.; Zhang, Y. Inducing target-specific latent structures for aspect sentiment classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Punta Cana, Dominican Republic, 8–12 November 2020; pp. 5596–5607.
15. Liang, B.; Yin, R.; Gui, L.; Du, J.; Xu, R. 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 Linguistics, Barcelona, Spain, Online, 8–13 December 2020; pp. 150–161.
16. Wang, K.; Shen, W.; Yang, Y.; Quan, X.; Wang, R. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3229–3238.
17. Tang, H.; Ji, D.; Li, C.; Zhou, Q. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6578–6588.
18. Li, R.; Chen, H.; Feng, F.; Ma, Z.; Wang, X.; Hovy, E. 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), Bangkok, Thailand, 1–6 August 2021; pp. 6319–6329.
19. Pang, S.; Xue, Y.; Yan, Z.; Huang, W.; Feng, J. Dynamic and multi-channel graph convolutional networks for aspect-based sentiment analysis. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 2627–2636.
20. Zhang, Z.; Zhou, Z.; Wang, Y. SSEGCN: Syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, WA, USA, 10–15 July 2022; pp. 4916–4925.
21. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. [[CrossRef](#)]
22. Zhou, T.; Shen, Y.; Chen, K.; Cao, Q. Hierarchical dual graph convolutional network for aspect-based sentiment analysis. *Knowl.-Based Syst.* **2023**, *276*, 110740. [[CrossRef](#)]
23. Zhang, S.; Gong, H.; She, L. An aspect sentiment classification model for graph attention networks incorporating syntactic, semantic, and knowledge. *Knowl.-Based Syst.* **2023**, *275*, 110662. [[CrossRef](#)]
24. Micali, S.; Zhu, Z.A. Reconstructing markov processes from independent and anonymous experiments. *Discret. Appl. Math.* **2016**, *200*, 108–122. [[CrossRef](#)]
25. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations, Palais des Congrès Neptune, Toulon, France, 24–26 April 2017.
26. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
27. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? In Proceedings of the International Conference on Learning Representations, New Orleans, Louisiana, MI, USA, 6–9 May 2019.

28. You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph contrastive learning with augmentations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5812–5823.
29. Wei, C.; Liang, J.; Liu, D.; Wang, F. Contrastive Graph Structure Learning via Information Bottleneck for Recommendation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 20407–20420.
30. Xia, L.; Huang, C.; Huang, C.; Lin, K.; Yu, T.; Kao, B. Automated Self-Supervised Learning for Recommendation. In Proceedings of the ACM Web Conference, New York, NY, USA, 30 April–4 May 2023; pp. 992–1002.
31. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 12–18 July 2020; pp. 1597–1607.
32. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised contrastive learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18661–18673.
33. Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; Hinton, G.E. Big self-supervised models are strong semi-supervised learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 22243–22255.
34. Gao, T.; Yao, X.; Chen, D. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021. Association for Computational Linguistics (ACL), Punta Cana, Dominican Republic, Online, 7–11 November 2021; pp. 6894–6910.
35. Wang, R.; Dai, X. Contrastive Learning-Enhanced Nearest Neighbor Mechanism for Multi-Label Text Classification. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Dublin, Ireland, 22–27 May 2022; pp. 672–679.
36. Liang, B.; Luo, W.; Li, X.; Gui, L.; Yang, M.; Yu, X.; Xu, R. Enhancing aspect-based sentiment analysis with supervised contrastive learning. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 1–5 November 2021; pp. 3242–3247.
37. 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, Online, 6–11 June 2021; pp. 2910–2922.
38. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. 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, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
39. Ivanov, S.; Burnaev, E. Anonymous walk embeddings. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 2186–2195.
40. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 Task 4: Aspect based sentiment analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014.
41. Dong, L.; Wei, F.; Tan, C.; Tang, D.; Zhou, M.; Xu, K. 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), Baltimore, MD, USA, 22–27 June 2014; pp. 49–54.
42. Xiong, H.; Yan, Z.; Zhao, H.; Huang, Z.; Xue, Y. Triplet Contrastive Learning for Aspect Level Sentiment Classification. *Mathematics* **2022**, *10*, 4099. [[CrossRef](#)]
43. Zhang, K.; Zhang, K.; Zhang, M.; Zhao, H.; Liu, Q.; Wu, W.; Chen, E. Incorporating Dynamic Semantics into Pre-Trained Language Model for Aspect-based Sentiment Analysis. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 3599–3610.
44. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.