Syntactic Edge-Enhanced Graph Convolutional Networks for Aspect-Level Sentiment Classification With Interactive Attention

YAO XIAO¹ AND GUANGYOU ZHOU²

¹Central China Normal University Wollongong Joint Institute, Central China Normal University, Wuhan 430079, China
²School of Computer Science, Central China Normal University, Wuhan 430079, China

Corresponding author: Guangyou Zhou (gyzhou@mail.ccnu.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant KJ02072019-0383.

ABSTRACT
Aspect-level sentiment classification is a hot research topic in natural language processing (NLP). One of the key challenges is that how to develop effective algorithms to model the relationships between aspects and opinion words appeared in a sentence. Among the various methods proposed in the literature, the graph convolutional networks (GCNs) achieve the promising results due to their good ability to capture the long distance between the aspects and the opinion words. However, the existing methods cannot effectively leverage the edge information of dependency parsing tree, resulting in the sub-optimal results. In this article, we propose a syntactic edge-enhanced graph convolutional network (ASEGCN) for aspect-level sentiment classification with interactive attention. Our proposed method can effectively learn better representations of aspects and the opinion words by considering the different types of neighborhoods with the edge constraint. To evaluate the effectiveness of our proposed method, we conduct the experiments on five standard sentiment classification results. Our results demonstrate that our proposed method obtains the better performance than the state-of-the-art models on four datasets, and achieves a comparative performance on Rest16.

INDEX TERMS
Natural language processing, sentiment analysis, text mining, graph convolutional networks.

I. INTRODUCTION
Sentiment analysis has a long history and is still considered a challenging research topic in natural language processing (NLP) [1]–[3]. The aspect-level sentiment classification (ALSC) [4], [5] is a fine-grained subtask in sentiment analysis. ALSC aims to automatically identify the sentiment polarity of one or more aspects appearing in a sentence. Consider the example shown in Figure 1, with the sentence “The food is so good and so popular that waiting can really be a nightmare.” as well as the two aspects “food” and “waiting”, toward which the sentiment polarities are positive and negative, respectively.

One of the key challenges for ALSC is that how to design effective algorithms to model the aspects with their corresponding opinion words. As it is challenging to model semantic relevance between the context words and the aspects has been proposed [6], recently many efforts have been made in learning the low-dimensional representations for aspects [7], [8] or in learning the aspect-specific representations by using the attention mechanism [9]–[14]. Among them, attention-based models have demonstrated promising results. However, the limitation is still obvious that attention mechanisms are not considered ideal in capturing the dependencies between the context words and various aspects of sentences, so the attention mechanisms may incorrectly locate specific targets. Besides, though some works combine RNN [15]–[17], long-range semantic information may be lost. As shown in Figure 1, the existing attention-based models may assign the higher weight to “popular” while identifying the polarity of the aspect “waiting”. However, the opinion word “popular” is found to be closer to the aspect “food” than “waiting” in the real case.

In order to better model the relationship between the opinion words and the aspects, various graph convolutional networks (GCNs) over the parsing tree have been employed.
Y. Xiao, G. Zhou: Syntactic Edge-Enhanced GCNs for ALSC With Interactive Attention

The dependency parsing tree can capture the long distance between the aspects and the opinion words, thereby providing a discriminative syntactic path for the propagation of the information over the tree. Moreover, the dependency parsing tree has a graph-like structure, which assists the recent GCN to successfully learn the representation of the nodes and capture the rich neighborhood information in the graph [18]–[22]. Among them, Zhang et al. [22] obtained the promising results and proposed a GCN over the dependency parsing tree in order to capture the syntactic information and word dependencies. However, the above existing methods cannot effectively leverage the edge information of the syntactic dependency parsing tree. In fact, the edges over the dependency parsing tree usually indicate the relationship between the aspects and the opinion words. The edge information may provide additional knowledge about neighborhood propagation, thereby resulting in the better performance for aspect-level sentiment classification.

In this paper, we propose a syntactic edge-enhanced graph convolutional network (ASEGCN) for aspect-level sentiment classification with interactive attention. First, we build the syntactic dependency parsing tree to gather the features from its neighbors. Specifically, the dependency parsing tree is a directed labeled graph, and the proposed ASEGCN can effectively gather the node information from four different types (along, reverse, undirected, self-loop, we will describe the details in Section 3.3) of neighborhoods. Second, we use the multi-head self-attention mechanism [23], [24] to concurrently capture the long-range contextual features and the specific aspect features. As shown in Figure 1, the label for (good, so) is advmod, therefore we can more effectively represent the effective representations of aspects and the opinion words by considering the various types of neighborhoods with the edge constraint. Moreover, the representation of the aspect food can be directly affected by the opinion word popular, even though they are not immediate neighbors.

To verify the effectiveness of the proposed method, we conduct extensive experiments on five standard benchmark datasets (Twitter, Lap14, Rest14, Rest15, Rest16). Our results show that the proposed method significantly outperforms various baselines on four datasets, and obtains the second best results on Rest16.

The main contributions are as follows:

- We propose a syntactic edge-enhanced graph convolutional network (ASEGCN) to effectively aggregate the node features from different types of neighborhoods, which is done by considering the direction and syntactic labels of edges in the dependency parsing tree as the constraints.
- We also propose to concurrently capture the long range contextual features and the specific aspect features by utilizing the multi-head self-attention.
- We conducted extensive experiments on the five benchmark datasets. The results show that our proposed method outperforms state-of-the-art on four datasets (Twitter, Lap14, Rest14, and Rest15), and achieves the comparative performance on Rest16.

The remainder of this paper is organized as follows. Section II presents an overview of the related works. In Section III, we describe the details of our approach. Section IV presents the details of the experimental results. Finally, in Section V, we conclude this paper with the future plans.

II. RELATED WORKS

To address the proposed framework, we briefly divide the related work into two categories: aspect-level sentiment classification and graph convolutional networks.

A. ASPECT-LEVEL SENTIMENT CLASSIFICATION

Aspect-level sentiment classification (ALSC) is a fine-grained sub-task in sentiment analysis [25], [26], which has gained widely interest from both academic and industrial communities [27], [28]. Compared to the well known sentence-level sentiment classification and the document-level sentiment classification, ALSC considers the aspects and the opinion words.

Given an aspect appearing in the sentence, ALSC attempts to automatically predict the sentiment polarity of the given aspect instead the whole sentence. Based on the techniques,...
used in the literature, we divided these previous work into five sub-categories: traditional shallow models for ALSC, recurrent neural networks (RNNs) based models for ALSC, convolutional neural networks (CNNs) based models for ALSC, memory networks (MNs) based models for ALSC, and Graph convolutional networks (GCNs) based models for ALSC. For more information, please refer to the recent survey [25].

1) TRADITIONAL SHALLOW MODELS FOR ALSC
This kind of methods employ the shallow machine learning models to extract features, such as bag-of-word features, lexical features and syntactic features [29]–[31]. However, these methods mainly rely on the hand-craft features and resources, which cannot easily access in many real applications. Moreover, when transferring to a different domain, the performance of these models may significantly degraded.

2) RNN BASED MODEL FOR ALSC
This kind of methods regard the ALSC as a sequence learning problems. Li et al. [32] proposed a model called Bi-RNN to model the interaction between the aspects and the their surrounding contextual words. Ruder et al. [33] proposed a HRNN to better learn the intra- and inter-sentence relations. Dong et al. [34] proposed a recursive neural network (RecNN) to learn the tree structure for ALSC. Recently, many researchers leverage the attention-based RNN for ALS [7], [35]–[39]. Wang et al. [7] proposed to concentrate on different parts regarding to the different aspects using the attention mechanism. Ma et al. [35] designed the attention mechanism to connect the specific aspects and the context representation. Yang et al. [36] designed attention-based bidirectional RNN approach to improve the target dependent sentiment classification.

3) CNN BASED MODEL FOR ALSC
This kind of methods are able to capture the local representations for the specific aspects and the corresponding opinion words. Some researchers attempt to adopt CNN for ALSC [8], [40]–[43]. For example, Xue an Li [8] proposed a CNN with the gate mechanisms to learn the representations for aspects and the opinion words. Huang and Carley [40] proposed to take the aspect information into account and adopt CNN for ALSC. Xing et al. [41] incorporated attention-based input layers into CNN to introduce context words information. Li et al. [42] considered the positional relevance between the specific aspect and the corresponding opinion words with the an input convolutional layer.

4) MN BASED MODEL FOR ALSC
This kind of methods have the ability to read and write with the aim of using the memory for prediction. MN has played important role in ALSC and many researchers have adopted MN for ALSC [16], [44]–[49]. For example, Chen et al. [16] proposed to use memory network under the framework of a recurrent attention mechanism for ALSC. Tang et al. [44] proposed an end-to-end memory network to capture the important information towards the given aspects. Fan et al. [45] proposed to capture the multi-words expressions with a convolutional MN for ALSC. Wang et al. [46] designed the target-sensitive memory networks to infer the sentiment polarity of a given target from the context.

5) GCN BASED MODEL FOR ALSC
This kind of methods attempt to convert the sentence into graph structure to model the relationship between the aspects and the opinion words. For example, Kipf et al. [18] learned hidden layer representations that encode both local graph structure and features of nodes. Zhang et al. [22] builded GCN over the dependency tree to capture the long-range dependencies as well as the relevant syntactic information. Sun et al. [50] propagated both contextual and dependency information from opinion words to aspect words, offering discriminative properties for ALSC. Zhao et al. [51] attempted to learn the sentiment dependencies among different aspects appearing in a sentence using the attention-based GCN. Hou et al. [52] designed a selective attention based GCN block to find the most important context words and directly aggregate these information into the aspect term representation for ALSC. Zuo et al. [53] proposed a new context-specific heterogeneous graph convolutional network framework that can combine all context representations to obtain more accurate semantic acquisition. Tang et al. [54] proposed to interact the flat representations learnt from Transformer and graph based representations learnt from the corresponding dependency graph for ALSC. Liang et al. [55] proposed a novel dependency syntactic knowledge augmented interactive architecture with multi-task learning.

Although the above GCN based models have obtained promising results for ALSC, they ignore the syntax edge information during the neighborhood propagation, resulting in the sub-optimal results. On the contrary, we have two differences compared to these existing methods: (1) our proposed ASEGCGN can effectively aggregate the node features from different types of neighborhoods; (2) our proposed solution can concurrently capture the long range contextual features and the specific aspect features by utilizing multi-head self-attention.

B. GRAPH CONVOLUTIONAL NETWORKS
Gori et al. [56] and Scarselli et al. [57] introduced the early work of trying to extend the neural network to handle arbitrary structured graphs, where the state of the node is updated according to the state of its neighbors. Then Bruner [58] applied the convolutional operation on graph Laplacians to build an effective architecture. And subsequent work improves computational efficiency through local spectrum convolutional technology [18]. GCN is actually a multi-layer graph convolutional neural network. Each convolutional layer only processes the first-order neighborhood information, multi-order neighborhood information transmission can be obtained by superimposing several convolutional layers.
FIGURE 2. The overall framework of the proposed model. Input Embedding: pre-trained GloVe embedding or BERT embedding; Hidden States: the bidirectional LSTM; MHA: multi-head attention encoder layer; SEGCN: syntactic edge-enhanced GCN layer. After initially using Bi-LSTM to model semantics information of context and the specific aspects, MHA encodes the hidden state of the specific aspects and context to obtain rich semantic information. Meanwhile, the combination of SEGCN and MHA is used to encode the syntactic information. Then, contextual semantic encoding and aspect-specific semantic encoding fully interacts with syntactic information encoding by using MHIA. Finally, the average pool is used to concatenate the final representation for predicting the polarity of aspects in output layer.

Therefore, the GCN approach can ensure the effective coding of the sentence structure expressed by the dependency tree, and thereby the representation of the node encodes the target word and the local position of the opinion word in the dependency tree.

III. OUR APPROACH

Given an $n$-word sentence $c = \{x_1^c, x_2^c, \ldots, x_t^c, \ldots, x_n^c\}$ containing $m$-word aspect term that begins from the $\tau$-th token, and the target sentence $t = \{x_1^t, x_2^t, x_3^t, \ldots, x_m^t\}$.

The objective of our model is to predict the sentiment polarities $y \in \{-1, 0, 1\}$ of the multiple aspect terms, where $-1$, $0$ and $1$ denote positive, neutral and negative respectively. The notations used in this paper are listed in Table 1.

Figure 2 shows the overall architecture of the our proposed ASEGCN model, which consists of five modules: input embedding layers, multi-head attention mechanism, position weight encoding, syntactic edge-enhanced GCN (SEGCN) layer and the output layer. (a) The embedding module contains pre-trained Glove embedding and BERT embedding. (b) The multi-head attention module captures long-range semantic dependencies of the context and the aspect. (c) The position encoding module reduces the noise and deviation caused by different dependency parsing processes. (d) The
TABLE 1. Notations used in this paper.

| Notations | Explanation |
|-----------|-------------|
| $H^c$, $H^t$ | The collection of context and target hidden words. |
| $H^{ac}$, $H^{at}$ | The collection of encoding context and target words. |
| $H^r$ | The collection of syntactic-aware words. |
| $H^{ic}$, $H^{it}$ | The collection of final concatenation words. |
| $c$, $t$ | The sentences and aspects. |
| $n$, $m$ | The length of the sentences $c$ and aspects $t$. |
| $x_i$ | A word $x_i \in c$ |
| $h_i$ | The embedding of word $x_i$. |
| $k$, $q$, $o$ | The key, query and output sentence of attention mechanism. |
| $W$ | The weight matrix. |
| $W^l_i$ | The different type transformation matrices. |
| $F(\cdot)$ | The function to assign position weights. |

SEGCGN module leverages the syntactic dependency tree in order to better capture the syntactic information. (e) The output module obtains the final representation.

A. INPUT EMBEDDING LAYERS

We employ two types of embedding layers to obtain the word embedding of each word. The pretrained GloVe embedding [59], maps each word to its corresponding embedding vector $e^i \in \mathbb{R}^{d_{emb} \times 1}$, where $d_{emb}$ is the dimension of word vectors. In the pretrained BERT embedding [60], we refactor the given context and the target as “[CLS]” + context + “[SEP]” and “[CLS]” + target + “[SEP]” respectively [10]. Thereafter, Bi-LSTM networks are applied to capture the contextual representation of each word:

$$
\overrightarrow{h}_i = \text{LSTM}(x_i^l, \overrightarrow{h}_{i-1})
$$

$$
\overleftarrow{h}_i = \text{LSTM}(x_i^l, \overleftarrow{h}_{i-1})
$$

$$
\overrightarrow{h}_i = (\overrightarrow{h}_i, \overleftarrow{h}_i)
$$

(1)

where $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ represent the forward hidden state and the backward hidden state, respectively, $h_i^l \in \mathbb{R}^{2d_{hid}}$ is the hidden state vector at the i-step after concatenating both the forward and backward hidden states, and $d_{hid}$ represents the dimensionality of a hidden state vector output by a unidirectional LSTM. Therefore, we can obtain the context hidden representation $H^c = \{h_1^c, h_2^c, h_3^c, \cdots, h_n^c\}$ and the target $H^t = \{h_1^t, h_2^t, h_3^t, \cdots, h_m^t\}$.

B. MULTI-HEAD ATTENTION MECHANISM

The multi-head attention mechanism (MHA) uses multiple queries to calculate multiple information from the input information in parallel [10], and each attention focuses on different parts of the input information, and then combines them together. Our paper employs the multi-head attention to capture long-range semantic dependencies for the context and the target respectively, and infer semantic encoding. Given the hidden representation $H^c$ and the target representation $H^t$, we can obtain the representation of the encoding context $H^{ac} = \{h_1^{ac}, h_2^{ac}, \cdots, h_n^{ac}\} \in \mathbb{R}^{d_{hid} \times n}$ and the target words $H^{at} = \{h_1^{at}, h_2^{at}, \cdots, h_m^{at}\} \in \mathbb{R}^{d_{hid} \times m}$ as follows:

$$
H^{ac} = \text{MHA}(H^c, H^c)
$$

(2)

$$
H^{at} = \text{MHA}(H^t, H^t)
$$

(3)

$$
\text{MHA}(k, q) = W^O[O^{head_1} \odot O^{head_2} \cdots \odot O^{head_n}]
$$

(4)

where $[\odot]$ represents the vector concatenation, $W^O \in \mathbb{R}^{d_{hid}}$ is a linear variable weight matrix, $O^{head_i}$ represents the output of the i-th head attention, and $i \in [head_1, head_n]$.

$$
\text{Attention}(k, q) = \text{softmax}(\frac{f(k, q)}{\sqrt{d_k}})k
$$

(5)

$$
f(k, q) = \text{tanh}([k_l; q_i] \ast W^a)
$$

(6)

where $d_k$ is the dimension of the regulator, $W^a \in \mathbb{R}^{2d_{hid}}$ denotes the weight matrix, and $f$ is the function that calculates and evaluates the semantic relevance of $k_i$ and $q_i$ respectively.

C. SYNTACTIC EDGE-ENHANCED GCN

Although Bi-LSTM networks have encoded the temporal context of word sequences, they still ignore the syntactic information, which indicates that capturing longer-range dependencies between words in a sentence may be less effective. In order to better capture the information of syntactic structure, we use graph convolutional networks to leverage the syntactic dependency tree. We denote the graph $G = (V, E)$ by $V$ nodes, and the edge set $E$ contains all the directed syntactic dependencies of the node pairs. In order to capture rich neighborhood information at a larger depth, we employ the deep connection GCN to replace the traditional GCN mechanism that cannot effectively use non-local interactive information. We define the concatenation of the initial node and the node from the continuous GCN layer as the $\phi_j^l$, which can be expressed as follows:

$$
\phi_j^l = \{h_0^j; u_1^j; \cdots; u_{l-1}^j\}
$$

(7)

As compared to the traditional way of obtaining information only from the previous layer $u_{l-1}^j$, the deep connection can capture rich information from all the preceding layers, including local and non-local information. Then, the representation of each node is updated as follows:

$$
\overrightarrow{h}_i = \text{ReLU} \left( \sum_{j \in N(i)} W^l \phi_j^l + b^l \right)
$$

(8)

where $\text{ReLU}$ represents the non-linear function [61], $W^l$ denotes the transformation matrix, $b^l$ denotes the bias vector, $\phi_j^l$ and $\overrightarrow{h}_i$ denote the representation of $j$-th token from all

1We employ spaCy to construct: https://spacy.io/
preceeding GCN layer and the output in the current layer, respectively.

However, the deep connection GCN only encodes neighbors to enhance the semantic representation of words, and cannot effectively use the edge direction and label information. Considering these shortcomings, we denote the edge from node $i$ to node $j$ with label $(i, j)$ as $direction(i, j)$, and then we use syntactic edge-enhanced GCN (SEGCN) to utilize the directional and labeled dependency edges between nodes to provide a more complete syntactic constraint for one aspect of the sentence:

$$ h_i^l = ReLU \left( \sum_{j \in N(i)} W_{direction(i,j)}^l \phi_j^l + b_{label(i,j)}^l \right) $$  \tag{9}

Due to the possibility of transmitting syntactic information in the reverse direction of the dependency edge [62], thereby we define four states by $direction(i,j)$ which include along, reverse, undirected and self-loop edges [63]:

$$ direction(i,j) = \{(i,j, label(i,j)) | (i,j, label^{-1}(i,j))$$
$$ (i,j, label(i,j) \cdot label^{-1}(i,j)) \}$$  \tag{10}

where $label^{-1}(i,j)$ represents the reverse edge label corresponding to the original $label(i,j)$, $label(i,j) \cdot label^{-1}(i,j)$ denotes the undirected edge label that is the union of the former two, and $\perp$ is a special relationship symbol that is used to represent the self-loop edge. The dissemination of these four different types of information passing corresponds to different transformation matrices $[W_1^l; W_2^l; W_3^l; W_4^l]$ respectively:

$$ \begin{cases} 
W_1^l = W_{direction(i,j)}^l & \text{if the along edge} \\
W_2^l = W_{direction(i,j)}^l & \text{if the reverse edge} \\
W_3^l = W_{direction(i,j)}^l & \text{if the undirected edge} \\
W_4^l = W_{direction(i,j)}^l & \text{if the self loop edge with } i = j 
\end{cases} $$  \tag{11}

Besides, $label(i,j)$ represents the label of $edge(i,j)$, selecting the unequal bias for unequal-type dependencies. At last, we apply the residual connection [64] $[\oplus]$ to obtain the following result:

$$ r_i^l = h_i^l \oplus \phi_i^{l-1} $$  \tag{12}

In addition, we need to follow the multi-layer GCN for obtaining the syntactic-aware representation $r = \{r_1^l, r_2^l, \cdots, r_m^l\}$:

$$ r_i^l = h_i^l \oplus r_i^{l-1} $$

where $h_i^l = SEGCN(r_i^{l-1})$  \tag{13}

### D. POSITION ENCODING

In order to reduce the noise and deviation caused by different dependency parsing process, we use the position-aware transformation to process $h_i^l$ before feeding it into the continuous GCN layer, where $l \in [1, 2, \cdots, L]$ for $L$ layers GCN and $h_i^l$ means the final state of the node $i$:

$$ u_i^l = F(h_i^l) $$  \tag{14}

where $F(\cdot)$ is a function that is used to assign position weights [16], [42]. The objective is to enhance the importance and relevance of the context to the target word, which is denoted by the following expression:

$$ d_i = \begin{cases} 
\frac{n - \tau - 1 + i}{n} & \text{if } 1 \leq i < \tau + 1 \\
\frac{n + \tau + \tau - i}{n} & \text{if } \tau + m < i \leq n \\
0 & \text{if } \tau + 1 \leq i \leq \tau + m 
\end{cases} $$  \tag{15}

$$ F(h_i^l) = d_i h_i^l $$  \tag{16}

where $d_i \in \mathbb{R}$ represents the position weight to the $i$-th token.

### E. MULTI-HEAD INTERACTIVE ATTENTION

The multi-head interactive attention mechanism (MHIA) is a common form of the key different from the query [10], [23]. The interactive information obtained from the respective semantic information and syntactic information is more abundant, which benefits the aspect sentiment classification. Therefore, we employ MHIA to calculate and construct the final concatenation, so that the semantic information and syntactic information can be fully interacted as shown below:

$$ H^{ic} = MHIA(H^{mc}, H^{rs}) $$  \tag{17}

$$ H^{it} = MHIA(H^{at}, H^{rs}) $$  \tag{18}

In this formula, $H^{ic} \in \mathbb{R}^{d_{hid} \times n}$ and $H^{it} \in \mathbb{R}^{d_{hid} \times m}$ where $d_{hid}$ is the dimension of MHIA, $H^{mc}$ and $H^{at}$ are the context semantic encoding and the target semantic encoding, respectively, and $H^{rs}$ is the syntactic information using the MHA encoding of the final SEGCN layer.

### F. SENTIMENT CLASSIFICATION

After obtaining the final representation of the previous output by using the average pooling, we concatenate these representations and obtain the unified representation $g$ as shown below:

$$ g = [h_i^{mc}; h_i^{it}; h_i^{rs}] $$  \tag{19}

It is noted that the above unified representation $g$ is fed into the softmax layer to obtain the probability distribution of different target semantic polarities:

$$ z = softmax(W^s g + b^s) $$  \tag{20}

where $z \in \mathbb{R}^k$ denotes the probability of the sentiment polarity, $W^s$ and bias $b^s$ represent the trainable parameters, respectively, and $k$ represents the categories of the classification.

### G. TRAINING

The objective function is defined by minimizing the cross-entropy with the $L_2$ regularization, which is shown as follows:

$$ loss = - \sum_{i=1}^{C} \sum_{j=1}^{m} z_{ij} \log(\hat{z}_{ij}) + \lambda \| \theta \|^2 $$  \tag{21}
TABLE 2. Details description of the datasets used in this article.

| Dataset     | Twitter | Lap14 | Rest14 | Rest15 | Rest16 |
|-------------|---------|-------|--------|--------|--------|
|             | Train   | Test  | Train  | Test   | Train  | Test   |
| #positive   | 1561    | 173   | 987    | 341    | 2164   | 728    |
| #neutral    | 3127    | 346   | 460    | 169    | 633    | 196    |
| #negative   | 1560    | 173   | 866    | 128    | 805    | 196    |

TABLE 3. The final values of main hyper-parameters for each dataset. L.R, B.S, H.D, N.E, L₂.R represent the learning rate, batch size, hidden dimension, the number of epochs and L₂ regularization.

| Dataset     | L.R   | B.S   | H.D   | Layers | N.E  | Heads | L₂.R | Drop |
|-------------|-------|-------|-------|--------|------|-------|------|------|
| Twitter     | 1e-3  | 32    | 300   | 2      | 100  | 3     | 1e-5 | 0.4  |
| Lap14       | 1e-3  | 32    | 300   | 2      | 100  | 3     | 1e-3 | 0.3  |
| Rest14      | 1e-3  | 32    | 300   | 2      | 100  | 3     | 1e-5 | 0.3  |
| Rest15      | 1e-3  | 32    | 300   | 2      | 100  | 3     | 1e-7 | 0.4  |
| Rest16      | 1e-3  | 32    | 300   | 2      | 100  | 3     | 1e-7 | 0.3  |

where zᵢⱼ represents the element of the i-th aspect for the j-th category, λ is the coefficient of the L₂ regularization, C is the classification category, and θ represents all trainable parameters, such as learning rate, dropout and so on. We will describe these parameters in Section IV.

IV. EXPERIMENTS

In this Section, we first describe the datasets used in this article. Then, we present the experimental settings in IV-B and the evaluation metrics in IV-C. Finally, we describe the details of results in the rest parts.

A. DATASETS DESCRIPTION

We conduct the experiments on five benchmark datasets: Twitter dataset used in [34], Lap14 and Rest14 datasets from SemEval 2014 task [65], and Rest15 and Rest16 datasets from SemEval 2015 task and SemEval 2016 task [66], [67], respectively. The sentiment polarities in these datasets are divided into positive, neural and negative. The detailed statistics of the five datasets are described in Table 2.

B. EXPERIMENTAL SETTINGS

In our experiment, we initialize the word embedding with a dimension of 300 for the pre-trained Glove [59] and with a dimension of 768 for pre-trained BERT.² During the training stage, as performed in [22], we randomly select 25% from each training data as the development set, and the remaining 75% are used for training. We use Adam optimizer [68] with the default configuration and set the batch size to 32. We fine-tune the pre-trained BERT, and set the L₂-regularization to 2 × 10⁻⁵, and the dropout rate is 0.2. All the weights are initialized with a uniform distribution in the model. The number of the multi-heads and the GCN layers are two important hyper-parameters used in the paper, and we tune these two parameters on the development set by using the grid search. Finally, we set the number of multi-heads to 3 and the GCN layers to 2 due to the best results on the development set. The main hyper-parameter setting is shown in Table 3.

All experiments reported in this article are conducted on NVIDIA GTX 1080Ti GPUs using PyTorch. We implement the proposed method using Python programming language.

C. EVALUATION METRICS

In order to make a fair comparison, we adopt the evaluation metrics of Accuracy (Acc.), Macro-F₁ (F₁), Macro-Precision (Prec.) and Macro-Recall (Recall) which have been widely used for ALSC [17], [42], [45], [46], [69].

Accuracy is the basic and most popular metric for ALSC. This evaluation metric represents the proportion of the correctly predicted samples of all examples, which can be computed as:

\[
\text{Acc} = \frac{\text{correctly predicted samples}}{\text{All examples in the dataset}} \quad (22)
\]

Macro-F₁ metric is frequently used to evaluate binary, multi-class and multi-label classification problems considering ALSC generally has more than three possible values for sentiment polarity. This evaluation calculates metrics for each label, and find their unweighted mean, which can be computed as:

\[
F₁ = \frac{2 \times \text{Macro-Precision} \times \text{Macro-Recall}}{\text{Macro-Precision} + \text{Macro-Recall}} \quad (23)
\]

where, Macro-Precision means the average proportion of correct predictions among all predictions with the positive label, Macro-Recall is the average proportion of correct predictions among all positive instances [25].

²We use uncased BERT-base: https://github.com/huggingface/transformers
D. COMPARED METHODS AND RESULTS ANALYSIS
We compare our proposed method with various baselines as follows:

- **Long Short-term Memory (LSTM)** [6]: They employed the last hidden state vector of LSTM to predict polarity.
- **Deep Memory Networks (MemNet)** [44]: They proposed to use external memory to model the context and make the model enjoy the the multi-hop information.
- **Attention-Based LSTM (ATAE-LSTM)** [7]: They proposed an attention-based model using the aspect embedding and word embedding to provide the solution of ALSC.
- **Recurrent Attention Network on Memory (RAM)** [16]: They proposed a attention-based aggregation model to learn the sentence representation with a multi-layer structure.
- **Interactive Attention Networks (IAN)** [35]: They proposed to achieve the sentiment analysis task using CNN and Bi-RNN to learn the representations with the attention mechanism.
- **Gated convolutional Networks (GCAE)** [8]: They proposed to use the gate mechanism to combine the outputs of two convolutional layers.
- **Position-Aware Bidirectional Attention Network (PBAN)** [70]: They proposed a position-aware BAN with the framework of Bi-GRU. The advantage of PBAN is that it can effectively consider the influence of the context words in different positions on the emotional polarity of the target words.
- **Multi-Granularity Alignment Network (MGAN)** [71]: They proposed a coarse2fine attention model to model the AC and AT tasks jointly.
- **Aspect-Specific Graph Convolutional Networks (ASGCN)** [22]: They proposed to learn the aspect-specific feature representation using GCN with the aim of solving the long-range multi-word dependency problem.
- **Attentional Encoder Network (AEN)** [10]: They employed attention-based encoders to model the relationship between the contexts and the specific targets to avoid duplication.
- **Target-Dependent Graph Attention Network (TD-GAT)** [40]: They proposed a novel target-dependent graph attention network for the aspect level sentiment classification, which explicitly uses the dependency relationship among words.
- **Sentiment Dependencies Graph convolutional Networks (SDGCN)** [51]: They proposed to model the aspect-specific representations with the position encoding and employed attention-based GCN to capture the sentiment polarity.

Table 4 and Table 5 shows the results with the input word embedding obtained from Glove and BERT, respectively. From the above Table 4 and Table 5, we have inferred several observations:

1) Compared to the traditional RNN-based and CNN-based models, the GCN-based models can obtain more better performance (e.g., LSTM vs. ASEGCN; LSTM vs. GCAE). For example, ASEGCN achieves absolute increases of 7.29% and 9.55% regarding Acc and F1, respectively. The results verify our claim that modeling the relationship between the aspects and the opinion words with the graph structure can significantly improve the performance.

2) Compared to AEN, which does not consider the bi-directional LSTM networks, our proposed ASEGCN can obtain better performance (AEN vs. ASEGCN). Taking the Lap14 dataset for example, ASEGCN achieves absolute increases of 2.98% and 3.6% in Acc and F1, respectively. The results indicate that the multi-head attention mechanism with Bi-LSTM can extract more abundant semantic information and better capture the long-range context semantics.

3) Compared to ASGCN, which does not use the edge information for ALSC, our proposed ASEGCN can obtain better performance (ASGCN-DG vs. ASEGCN). For example on Twitter dataset, ASEGCN achieves absolute increases of 1.69% and 2.02% in Acc and F1, respectively. The results again demonstrate that the syntactic edge information (direction and label information) is indeed useful for better learning the representations of aspects and the opinion words.

4) Taking the attention into account, the performance can be further improved (MemNet vs. ASEGCN). For example, ASEGCN achieves absolute increases of 5.85% and 7.47% regarding Acc and F1, respectively. The results show that multi-head self-attention can be indeed helpful to capture the long-range contextual features and the specific aspect features.

5) When using the initial word embedding BERT, ASEGCN achieves the comparable accuracy with SDGCN-BERT on the dataset of Lap14, the improvement is slightly higher using the F1 score (0.86% improvement). Nevertheless, we also note that the performances of ASEGCN-BERT is overall better than all baselines. It obtains the remarkable performance gain on the dataset of Rest15. The results again demonstrate that the SEGCN module is really useful for ALSC task.

E. IMPACT OF THE NUMBER OF MULTI-HEADS AND GCN LAYERS
In this section, we investigate how the number of multi-heads and GCN layers affect the performance of our proposed model as shown in Figure 3. From Figure 3, we can infer that as the number increases, the performance decreases accordingly. The model can achieve the best performance when the number of heads and the GCN layers are set to 3 and 2, respectively. We conceive that as the number of heads increases, the semantic information of the contextual words will generate too much noise. Meanwhile, when the number
TABLE 4. Aspect-level sentiment classification results on various datasets. The bold indicates the best results and the underline represents the second best results.

| Model       | Twitter Acc | Twitter $F_1$ | Lap14 Acc | Lap14 $F_1$ | Rest14 Acc | Rest14 $F_1$ | Rest15 Acc | Rest15 $F_1$ | Rest16 Acc | Rest16 $F_1$ |
|-------------|-------------|---------------|-----------|------------|-----------|------------|-----------|------------|-----------|------------|
| LSTM        | 69.56       | 67.70         | 69.28     | 63.09      | 78.13     | 67.47      | 77.37     | 55.17      | 86.80     | 63.88      |
| MemNet      | 71.48       | 69.90         | 70.64     | 65.17      | 79.61     | 69.64      | 77.31     | 58.28      | 85.44     | 65.99      |
| ATAE-LSTM   | N/A         | N/A           | 68.70     | N/A        | 77.20     | N/A        | N/A       | N/A        | N/A       | N/A        |
| AOA         | 72.30       | 70.20         | 72.62     | 67.52      | 79.97     | 70.42      | 78.17     | 57.02      | 87.50     | 66.21      |
| RAM         | 69.36       | 70.85         | 74.49     | 71.35      | 80.23     | 70.80      | N/A       | N/A        | N/A       | N/A        |
| IAN         | 72.50       | 70.81         | 72.05     | 67.38      | 79.26     | 70.09      | 78.54     | 52.65      | 84.74     | 55.21      |
| TNet-LF     | 72.98       | 71.43         | 74.61     | 70.14      | 80.42     | 71.03      | 78.47     | 59.47      | 89.07     | 70.43      |
| GCAE        | N/A         | N/A           | 69.14     | N/A        | 77.28     | N/A        | N/A       | N/A        | N/A       | N/A        |
| PBAN        | N/A         | N/A           | 74.12     | N/A        | 81.16     | N/A        | N/A       | N/A        | N/A       | N/A        |
| MGAN        | 72.54       | 70.81         | 75.39     | 72.47      | 81.25     | 71.94      | N/A       | N/A        | N/A       | N/A        |
| AEN         | 72.83       | 69.81         | 73.51     | 69.04      | 80.98     | 72.14      | N/A       | N/A        | N/A       | N/A        |
| TD-GAT      | N/A         | N/A           | 74.00     | N/A        | 81.20     | N/A        | N/A       | N/A        | N/A       | N/A        |
| ASGCA-DG    | 72.15       | 70.40         | 75.55     | 71.05      | 80.77     | 72.02      | 79.89     | 61.89      | 88.99     | 67.48      |
| ASEGCA (ours)| 73.84     | 72.42         | 76.49     | 72.64      | 81.61     | 73.42      | 80.87     | 64.04      | 88.80     | 68.95      |

TABLE 5. Macro-Precision and Macro-Recall metrics for the results on various datasets.

| Model       | Twitter Prec. | Twitter Recall | Lap14 Prec. | Lap14 Recall | Rest14 Prec. | Rest14 Recall | Rest15 Prec. | Rest15 Recall | Rest16 Prec. | Rest16 Recall |
|-------------|----------------|----------------|-------------|--------------|--------------|---------------|--------------|---------------|--------------|---------------|
| ASEGCA      | 73.45          | 71.61          | 72.86       | 72.68        | 75.10        | 71.81         | 67.06        | 61.28         | 71.97        | 66.17         |
| ASEGCA-BERT | 76.55          | 71.68          | 76.98       | 80.02        | 80.53        | 73.01         | 74.90        | 65.84         | 74.92        | 69.98         |

FIGURE 3. Accuracy vs. the number of GCN layers and multi-heads, respectively.

of GCN layers continues to increase, it generates too many parameters, thus making it very difficult to train the model.

F. ABLATION STUDY

In order to investigate how each component contributes to the performance, we conduct an ablation study in Table 6. First, we removed the SEGCN module. The results show that the scores on all the Lap14, Rest14, and Rest15 datasets decreased to varying degrees, but there was no obvious change in the Twitter and Rest16 datasets, and even the $F_1$ score of the Twitter dataset increased. As SEGCN captures the underlying dependencies between the word pairs and the relationship of long-range words, it is found to be more helpful to the model. We speculate that as the text in the Twitter dataset is relatively colloquial and insensitive to syntactic information, a strong syntactic structure may interfere with the prediction of the emotional polarity of a specific target. Overall, SEGCN has significantly improved the prediction of the target emotional polarity of most of the datasets.
The multi-head interactive attention (MHA) module includes extracting semantic information, encoding syntactic information, and interactively learning the association between semantic information and syntactic information. After removing it, the results on the scores of the five datasets dropped significantly, which completely indicate the importance of interactive learning in semantic information and syntactic information. Therefore, it can be concluded that both the SEGCN model and the MHA model are of great help to the model.

**G. ERROR ANALYSIS**

In order to provide an intuitive understanding of our model, we select representative examples from the statistical mis-classified instances of Restaurant14 dataset shown in Table 7. By analyzing these mis-classified instances, we divide these errors into three categories.

The first is misleading due to neutral polarity, these opinion words towards the specific aspect usually have direct dependencies, resulting in the mis-classification issue.

The second is the difficulties due to the need of thorough comprehension, since many sentences don’t have the explicit clues indicating the sentiment polarity of the target aspects, and even contains some seemingly irrelevant opinion words.

The third is the trap due to double negation, which requires judgment based on meaning and context, this is challenging for the model.

Through the collation of the error analysis, we can see that although the current model has achieved some impressive results, for many more complicated sentences, we need to combine advanced natural language processing technologies and algorithms like word polarity disambiguation [72] and natural language inference [73] to solve them in the future.
In order to have a better understanding of our model, we adopt the attention scores of ASEGCN and ASEGCN (w/o SEGCN) in Table 8.

For the first example “I love the drinks and the lychee martini”, with two aspects “drinks” and “lychee martini”. We can see that ASEGCN (w/o SEGCN) primarily pays attention on the word “love” to predict the polarities for the two aspects. However, owing to modeling the dependency edge types (drinks $\rightarrow$ lychee martini, i.e. through the coordinating conjunction), ASEGCN can also concentrate on the conjunction “and” besides the word “love”, which correctly predicts the sentiment polarities of the two aspects simultaneously. This result indicates that the dependency edge types are indeed crucial to our model.

For the second example “The falafal was over cooked but the chicken was fine”, with two aspects “falafal” and “chicken”. We can see that ASEGCN (w/o SEGCN) actually predicts the polarity of the two aspects independently, which ignoring the relations between the two aspects, such as predicting the polarity of “chicken” by the word “fine”. While ASEGCN give the high score to “but” where the sentiments of the two aspects “falafal” and “chicken” are opposite connected.

The case study shows that the multi-head attention mechanism can focus attention on text words which are interdependent between different aspects and SEGCN model can effectively represent the sentiment dependencies between different aspects in a sentence. The combination of the two beneficially integrates the syntactic dependency information into the rich semantic representation, and fully interacts to make correct predictions.

V. CONCLUSION AND FUTURE WORK

In this paper, we present a bidirectional syntactic edge-enhanced graph convolutional network for aspect-level sentiment classification with the interactive attention mechanism. The specific idea is that the bidirectional LSTM and the multi-head attention mechanism are combined to generate semantic encoding, and SEGCN is applied based on the syntax dependency tree to completely use the directionality between the nodes and the dependency edge of the label to encode the syntactic information, which was completely missing from the previous research studies. Finally, the multi-head interactive attention mechanism and semantic information are used for full interaction. Our experimental results on the five benchmark datasets demonstrate that the proposed method outperforms various baseline methods on four datasets and even achieves the comparative results on the Rest16 dataset.

In future work, this research can be applied in the following three ways. First, we will discuss how to build a more accurate sentiment graph structure between various aspects. Second, different weights should be assigned to different nodes (e.g., aspect words) in a neighborhood, thus a natural avenue for the future work will explore the graph attention network (GAT) [74]–[76] to better model the relationships between the aspects and the opinion words so that the GCN module can be replaced. Third, the effective ensemble method should be explored to integrate the GNN-based models and sequence-based models into a unified framework in order to avoid potential noise.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] E. Cambria, S. Poria, A. Gelbukh, and M. Thelwall, “Sentiment analysis is a big suitcase,” IEEE Intell. Syst., vol. 32, no. 6, pp. 74–80, Nov. 2017, doi: 10.1109/MIS.2017.4531228.
[2] H. Peng, E. Cambria, and A. Hussain, “A review of sentiment analysis research in Chinese language,” Cognit. Comput., vol. 9, no. 4, pp. 423–435, Aug. 2017, doi: 10.1007/s12559-017-9470-x.
[3] O. Oueslati, E. Cambria, M. B. Hajhmida, and H. Ounelli, “A review of sentiment analysis research in Arabic language,” Future Gener. Comput. Syst., vol. 112, pp. 408–430, Nov. 2020.
[4] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Found. Trends Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008, doi: 10.1561/1500000011.
[5] B. Liu, “Sentiment analysis and opinion mining,” in Synthesis Lectures on Human Language Technologies, vol. 5, no. 1. San Rafael, CA, USA: Morgan & Claypool, 2012, pp. 1–167, doi: 10.2200/S00416ED1V01Y201204HLT016.
[6] D. Tang, B. Qin, X. Feng, and T. Liu, “Effective LSTMs for target-dependent sentiment classification,” in Proc. 25th Int. Conf. Comput. Linguistics, Tech. Papers. Osaka, Japan: ACL, Dec. 2016, pp. 3296–3307. [Online]. Available: https://www.aclweb.org/anthology/C16-1311/
[7] Y. Wang, M. Huang, X. Zhu, and L. Zhao, “Attention-based LSTM for aspect-level sentiment classification,” in Proc. Conf. Empirical Methods Natural Lang. Process. Austin, TX, USA: Association Computational Linguistics, 2016, pp. 606–615, doi: 10.18653/v1/11-1058.
[8] W. Xue and T. Li, “Aspect based sentiment analysis with gated convolutional networks,” in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics. Melbourne, VIC, Australia: Association Computational Linguistics, vol. 1, 2018, pp. 2514–2523. [Online]. Available: https://www.aclweb.org/anthology/P18-1234/
