DualGCN: Exploring Syntactic and Semantic Information for Aspect-Based Sentiment Analysis

Ruifan Li, *Member, IEEE*, Hao Chen, Fangxiang Feng, Zhanyu Ma, *Senior Member, IEEE*, Xiaojie Wang, and Eduard Hovy

***Abstract*— The task of aspect-based sentiment analysis aims to identify sentiment polarities of given aspects in a sentence. Recent advances have demonstrated the advantage of incorporating the syntactic dependency structure with graph convolutional networks (GCNs). However, their performance of these GCN-based methods largely depends on the dependency parsers, which would produce diverse parsing results for a sentence. In this article, we propose a dual GCN (DualGCN) that jointly considers the syntax structures and semantic correlations. Our DualGCN model mainly comprises four modules: 1) *SynGCN*: instead of explicitly encoding syntactic structure, the SynGCN module uses the dependency probability matrix as a graph structure to implicitly integrate the syntactic information; 2) *SemGCN*: we design the SemGCN module with multihead attention to enhance the performance of the syntactic structure with the semantic information; 3) *Regularizers*: we propose orthogonal and differential regularizers to precisely capture semantic correlations between words by constraining attention scores in the SemGCN module; and 4) *Mutual BiAffine*: we use the BiAffine module to bridge relevant information between the SynGCN and SemGCN modules. Extensive experiments are conducted compared with up-to-date pretrained language encoders on two groups of datasets, one including Restaurant14, Laptop14, and Twitter and the other including Restaurant15 and Restaurant16. The experimental results demonstrate that the parsing results of various dependency parsers affect their performance of the GCN-based models. Our DualGCN model achieves superior performance compared with the state-of-the-art approaches. The source code and preprocessed datasets are provided and publicly available on GitHub (see https://github.com/CCChenhao997/DualGCN-ABSA).**

***Index Terms*— Aspect-based sentiment analysis, graph convolutional network (GCN), semantic correlation, syntactic dependency.**

I. Introduction

SENTIMENT analysis is a long-standing research domain. Broadly speaking, this domain includes textual [1], [2], [3], [4], audio [5], [6], [7], visual [8], [9], [10], and multimodal [11], [12] sentiment analysis. Among them, textual sentiment analysis is a popular yet challenging research topic. The aim of textual sentiment analysis is to analyze people's opinions, sentiments, evaluations, and attitudes within text. This article focuses on the study of textual sentiment analysis.

Except for the traditional sentence-level or document-level sentiment classification tasks, the aspect-based sentiment analysis (ABSA) task has recently been proposed. This task is supposed as an entity-level oriented fine-grained sentiment analysis. Generally, the ABSA aims to determine the sentiment polarities of given aspects in a sentence. For example, in the foodservice industry, diners are extraordinarily concerned about certain aspects of comments, such as food, price, and service. Fig. 1 shows a sentence about a restaurant review. The sentiment polarities of those two aspects "food" and "service" are positive and neutral, respectively. Thus, the ABSA can precisely identify consumer's attitudes toward a certain aspect, rather than simply assigning a rough sentiment polarity for that sentence. Such fine-grained sentiment analysis task can not only bring detailed analysis to users but also pave the foot stones for other downstream tasks, such as recommendation [13], [14], [15] and advertisement computation [16].

The key to addressing the ABSA task is to model the dependencies between an aspect and its expression of the corresponding opinion. However, there probably exist multiple aspects and different expressions of opinions within a sentence. To identify the sentiment polarity of a particular aspect, previous studies [17], [18], [19], [20], [21], [22], [23], [24], [25] have proposed a variety of attention mechanisms based on recurrent neural networks (RNNs) to extract aspect-specific sentence representations, achieving impressive performance. Nevertheless, the lack of linguistic knowledge makes the attention mechanism susceptible to noise within sentences. Take Fig. 1 as an example. As for the aspect "service," the opinion word "good" may receive more attention than the opinion word "ok," while the "good" refers to another aspect "food."

Manuscript received 26 July 2021; revised 7 April 2022 and 14 June 2022; accepted 2 November 2022. Date of publication 14 November 2022; date of current version 4 June 2024. This work was supported in part by the Beijing Natural Science Foundation under Project Z200002 and in part by the National Natural Science Foundation of China under Grant 61922015, Grant 61906018, Grant U19B2036, Grant 62076032, and Grant 62225601. *(Corresponding author: Zhanyu Ma.)*

Ruifan Li, Fangxiang Feng, and Xiaojie Wang are with the School of Artificial Intelligence, and the Engineering Research Center of Information Networks, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: rfli@bupt.edu.cn; fxfeng@bupt.edu.cn; xjwang@bupt.edu.cn).

Hao Chen is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: ccchenhao997@bupt.edu.cn).

Zhanyu Ma is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Beijing Academy of Artificial Intelligence, Beijing 100084, China (e-mail: mazhanyu@bupt.edu.cn).

Eduard Hovy is with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: hovy@cmu.edu).

Digital Object Identifier 10.1109/TNNLS.2022.3219615

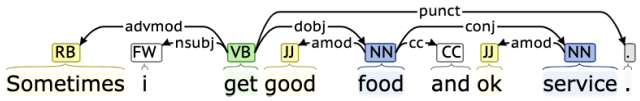


Fig. 1. Exemplar dependency parsing tree of a sentence produced by the Stanford CoreNLP parser. The sentence "Sometimes I get good food and OK service" contains two aspects "food" and "service" but having different sentiment polarities, positive and neutral.

Very recently, graph neural networks (GNNs) [26] including graph convolutional networks (GCNs) and graph attention networks (GATs) are overnight used over dependency trees for capturing the syntactic structure of a sentence [27], [28], [29], [30], [31], [32], [33], [34]. Consider the dependency tree in Fig. 1; the syntactic dependency can establish the connections between words in a sentence. For example, a dependency relationship exists between the aspect "food" and the opinion word "good." However, two challenges arise when applying the syntactic dependency knowledge onto the ABSA task: 1) since the training corpus of the dependency parser and the datasets performed on the ABSA task are often in significantly different domains, the inaccuracy of the dependency parsing results would arise and 2) GCNs over dependency trees do not work well as expected on datasets that are not sensitive to the syntactic dependency due to the informal expression and complexity of online reviews.

To solve the aforementioned challenges, in this article, we propose a novel architecture, the dual graph convolution network (DualGCN). For the first challenge, we use the probability matrix of all the dependency arcs from a dependency parser to build a syntax-based GCN (SynGCN). The idea behind is that the probability matrix representing dependencies between words contains richer syntactic information compared with the final discrete output of a dependency parser. For the second, we construct a semantic correlation-based GCN (SemGCN) using a self-attention mechanism. The idea behind is that the attention matrix shaped by self-attending, also viewed as an edge-weighted directed graph, can represent semantic correlations between words. Moreover, motivated by the work of DGEDT [34], we use a BiAffine module to bridge relevant information between the SynGCN and SemGCN modules.

To further enhance our DualGCN model, we design two regularizers. We observe that the semantically related terms of each word should not overlap. Therefore, we encourage the attention probability distributions over words to be orthogonal. To this end, we incorporate an orthogonal regularizer on the attention probability matrix for the SemGCN module. Moreover, these two representations learned from the SynGCN and SemGCN modules should contain significantly different information captured by the syntactic dependency and the semantic correlation. Therefore, we expect that the SemGCN module could learn semantic representations different from syntactic representations. Thus, we propose a differential regularizer between the SynGCN and SemGCN modules.

Our main contributions are highlighted as follows.

1) We propose a DualGCN model for the ABSA task. Our DualGCN considers both the syntactic structure and the semantic correlation within a given sentence. Specifically, our DualGCN integrates the SynGCN and SemGCN networks through a mutual BiAffine module.
2) We propose orthogonal and differential regularizers. The orthogonal regularizer encourages the SemGCN network to learn an orthogonal semantic attention matrix, whereas the differential regularizer encourages the SemGCN network to learn semantic features distinct from the syntactic ones built from the SynGCN network.
3) We conduct extensive experiments on the Restaurant14, Laptop14, and Twitter datasets. The experimental results demonstrate the effectiveness of our DualGCN model.

Note that this work has been presented in our previously published conference paper [35]. In this article, we make the following significant extensions to our previous work.

1) We extend our proposed DualGCN with a posttraining (PT) BERT for domain adaptation. The experimental results show that the PT BERT-based DualGCN significantly outperforms all the compared approaches.
2) We extend our proposed DualGCN with various pretrained language models (PLMs). The experimental results indicate the appealing performance of DualGCN equipped with PLMs. In addition, we adopt two additional datasets for evaluation with robustness performance.
3) We conduct extensive experiments to evaluate the impact of different dependency parsers on the our model. The experimental results demonstrate that the performance of the parser is not robust enough and the semantic information can compensate for the lack of syntactic structure.

The remainder of this article is organized as follows. In Section II, we describe two preliminary techniques, including GCN and BERT, used in our article. In Section III, we provide our DualGCN model in detail. Subsequently, the experimental settings and experimental results, including ablation studies, are reported in Sections IV and V, respectively. Section VI gives a brief review on related works. Finally, Section VII gives our conclusions and suggests future directions.

II. Preliminary

A. Graph Convolutional Network

Recently, GCN has achieved outstanding performance in a wide range of applications [36], [37], [38]. Here, we briefly introduce the main idea of GCN. GCN [39] is a type of variant of convolutional neural networks (CNNs) which is motivated mainly by the conventional CNNs and graph embeddings. A GCN can efficiently capture nodes' information by operating directly on graphs. In other words, for graph-structured data, a GCN can apply the convolution operation directly on connected nodes to encode local information. Then, through the message passing of multilayer GCNs, each node in a graph can learn more global information. In the NLP domain, most recent works such as [27], [28] extend the GCN models by encoding dependency trees and incorporating dependency paths between words. Specifically, given a graph

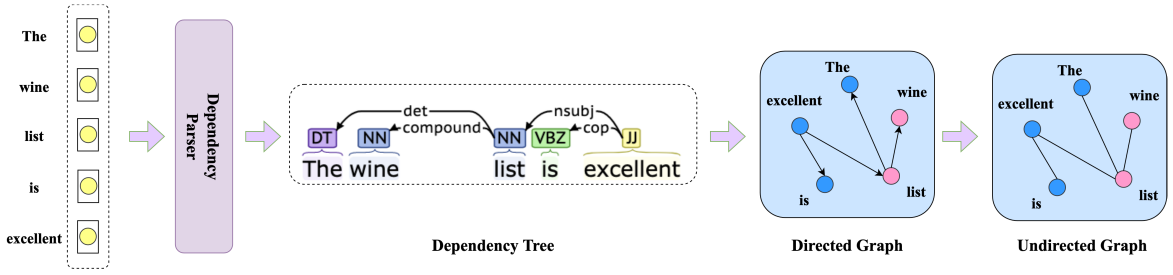


Fig. 2. Procedure of representing a sentence using graph. The sentence is first parsed into a dependency tree with a parser. Then the tree is converted into a directed graph and further into an undirected graph.

with n nodes, the graph can be represented as an adjacency matrix $A \in \mathbb{R}^{n\times n}$. For a sentence, an adjacency matrix A over its syntactical dependency tree is then built. An element A_{ij} in the adjacency matrix A indicates whether the ith node is connected to the jth node. Specifically, the element A_{ij} equals one, if the ith node is connected to the jth node, and A_{ij} equals zero, otherwise. More formally, the graph adjacency matrix A can be given as follows:

$$A_{ij} = \begin{cases} 1, & \text{if node } i \text{ is connected to node } j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Furthermore, the hidden state representation h_i^l of the ith node at the lth layer is updated according to the equation

$$h_i^l = \phi\left(\sum_{j=1}^{n} A_{ij} W^l h_j^{l-1} + b^l\right) \tag{2}$$

where the symbol W^l denotes a weight matrix, the symbol b^l denotes a bias term, and the symbol ϕ denotes an activation function, such as ReLU, i.e., $\phi(x) = \max(0, x)$.

In addition, the adjacency matrix A which is composed of ones and zeros can be taken as the final discrete output of a dependency parser. Fig. 2 shows an example how a sentence is transformed to an undirected graph. The sentence "The wine list is excellent" is parsed to a dependency tree. The tree is then transformed into a directed graph and an undirected graph.

B. Bidirectional Encoder Representation From Transformers

Recently, BERT has demonstrated its effectiveness in various NLP tasks, such as information extraction [40], question answering [41], and semantic matching [42]. Specifically, BERT [43] is a bidirectional language model based on the Transformer encoder [44]. Hence, BERT can be formulated as follows:

$$\hat{h}^l = \text{LN}\left(h^{l-1} + \text{MHAtt}\left(h^{l-1}\right)\right) \tag{3}$$

$$h^l = \text{LN}\left(\hat{h}^l + \text{FFN}\left(\hat{h}^l\right)\right) \tag{4}$$

where the symbol l denotes the depth of transformer layers. The symbol h^0 denotes the BERT input representation, which is the summation of token embeddings, position embeddings, and segment embeddings. The symbol LN is the layer normalization, and the symbol MHAtt is the multihead self-attention. The FFN contains successively three layers: the first is a linear projection layer, the second an activation layer, and the third linear projection layer. The vanilla version of BERT includes 12 Transformer layers and its MHAtt includes 12 attention heads.

During the pretraining stage, BERT is pretrained with large-scale corpus on two tasks, i.e., masked language modeling (MLM) and next sentence prediction (NSP). In the MLM task, 15% of the tokens in a sentence are manipulated in three manners. Specifically, 10%, 10%, and 80% of them are replaced by a random token, itself, or an "[MASK]" token, respectively. In the NSP task, two sentences, such as S_a and S_b, are concatenated before being fed into the BERT. Given 50% of the time when the sentence S_b is the next utterance of the sentence S_a, BERT needs to use the vector representation of "[CLS]" to figure out whether the input is continuous.

Usually, the pretraining corpus of the vanilla BERT is built from BooksCorpus [45] and Wikipedia. Consequently, the vanilla BERT suffers from the domain challenge, since the original LM pretraining domain is inconsistent with that of the target task. Some of the latest studies [46], [47] have demonstrated that the PT approach before fine-tuning is effective. Generally, the data scale of the downstream fine-tuning task is small, and the vanilla BERT cannot bridge the gap between the universal domain and the specific domain. PT is an adaptive method for pretraining language models. Specifically, in the ABSA task, we need to reduce the bias introduced by nonreview knowledge (e.g., from Wikipedia corpora) and integrate domain knowledge such as foodservice and laptop. Hence, PT is an important approach that leads to the performance gain.

III. Proposed DualGCN

Fig. 3 provides an overview of our DualGCN model. In the ABSA task, a sentence–aspect pair $(\mathcal{S}, \mathcal{A})$ is given as input. The entire sentence has n words, i.e., $\mathcal{S} = \{w_1, w_2, \ldots, w_n\}$ and an aspect term $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ has m words. The DualGCN model figures out what type of sentimental polarity is for each aspect. To this end, we first use BiLSTM, BERT, and other PLMs' encoder to obtain contextual representation. Then, the hidden representations of a sentence are input into the SynGCN and SemGCN modules, simultaneously. A BiAffine module is then adopted for effective information flow between these two modules. Finally, we aggregate all the aspect nodes' representations from the SynGCN and SemGCN modules via the pooling and concatenation operations, forming the final aspect representation. A softmax classifier is used to output the sentimental polarity for the sentence–aspect pair.

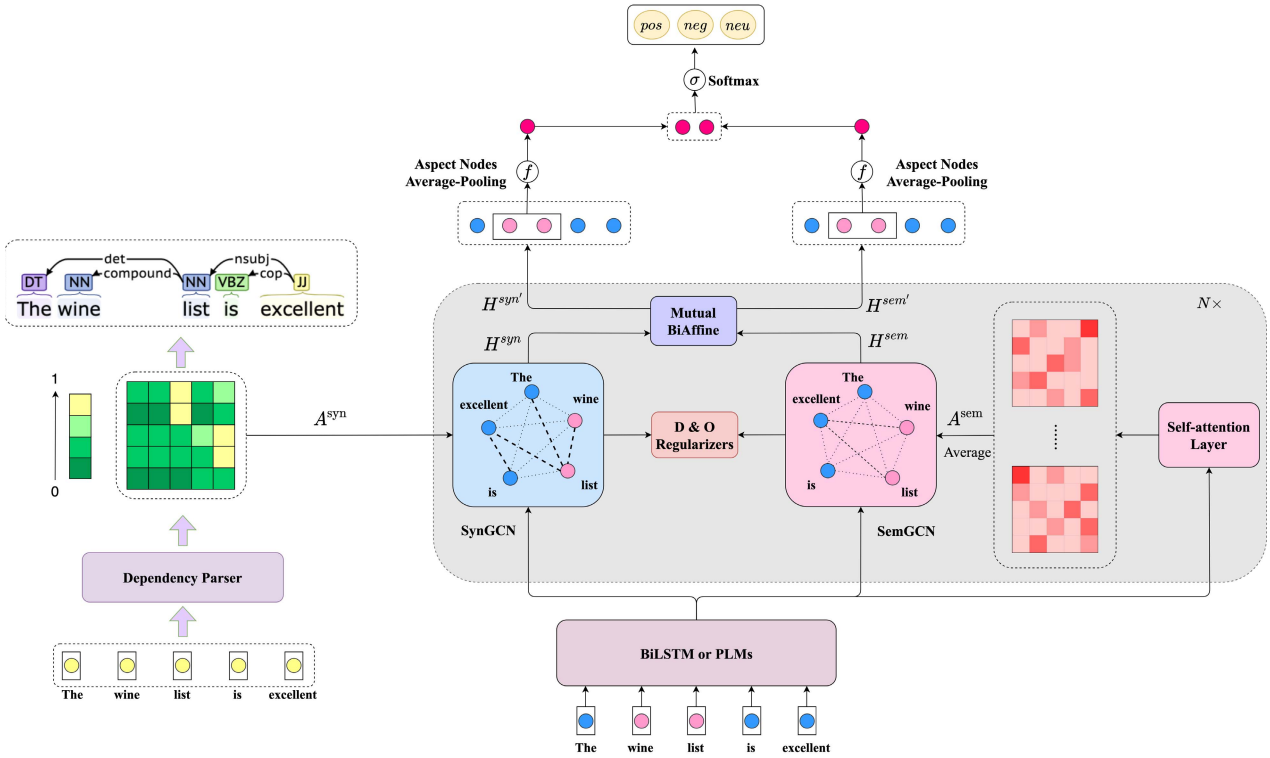


Fig. 3. Overall architecture of DualGCN. The DualGCN model is primarily composed of the SynGCN and SemGCN modules. SynGCN uses the probability matrix generated by the dependency parser, while SemGCN uses the attention score matrix generated by the self-attention layer. The module of D & O regularizers denotes the differential and orthogonal regularizers, which are designed to further improve the model's capability of capturing semantic correlations. Details of these modules are described in the main text.

In the following sections, we elaborate on the details of our proposed DualGCN model.

A. Contextual Representation

In our model, we use BiLSTM, BERT, and other PLMs as the sentence encoder to extract hidden contextual representations. For the BiLSTM encoder, the inputs have three types of representation: word embeddings, part-of-speech (POS) tag embeddings, and position embeddings. We provide details of these three embeddings one after another. We obtain the word embeddings $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ of the sentence $\mathcal{S} = \{w_1, w_2, \ldots, w_n\}$ from an embedding lookup table $E \in \mathbb{R}^{|V| \times d_e}$, where $|V|$ is the size of vocabulary and d_e denotes the dimensionality of word embeddings.

For POS tag embeddings, the main idea is to map each POS tag type into a real-valued vector. For each word w_i in the sentence under consideration, we create a POS tag embedding $t_i \in \mathbb{R}^{d_t}$ based on its POS tag, where d_t denotes the dimensionality of POS tag embeddings. The POS tag embedding matrix of the sentence is denoted as $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$. The POS tag embedding could be trained during the training phase to learn a tailor-made representation.

We create a position embedding $p_i \in \mathbb{R}^{d_p}$ for each word w_i in the sentence under consideration, in which the symbol d_p denotes the dimensionality of position embeddings. The position embedding p_i is calculated based on the relevant distance v_i between the ith word w_i and aspect terms. The position embedding matrix of the sentence is denoted as $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$. The position embedding could be trained during the training phase to learn a tailor-made representation. Specifically, we calculate the relevant distance v_i as follows:

$$v_i = \begin{cases} |i - j_s|, & \text{if } i < j_s \\ 0, & \text{if } j_s \le i \le j_e \\ |i - j_e|, & \text{if } i > j_e. \end{cases} \tag{5}$$

where the symbols j_s and j_e denote the starting and ending indices of the aspect term, respectively. The obtained distance v_i can be viewed as the relative distance of the ith word w_i in the sentence to the aspect term.

Next, we concatenate word embeddings $\mathcal{E}$, POS tag embeddings $\mathcal{T}$, and position embeddings $\mathcal{P}$ of all the words in the sentence to form the final word representations $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$. And then we feed them into a BiLSTM to produce hidden state vectors $H = \{h_1, h_2, \ldots, h_n\}$, where $h_i \in \mathbb{R}^{2d_u}$ is the hidden state vector at time i from BiLSTM. The symbol d_u denotes the dimensionality of the hidden state vector output by a unidirectional LSTM. Given an input word embedding vector x at each time t, the forward LSTM model is updated as follows:

$$f_t = \sigma\left(W_f\left[h_{t-1}; x_t\right] + b_f\right) \tag{6}$$
$$i_t = \sigma\left(W_i\left[h_{t-1}; x_t\right] + b_i\right) \tag{7}$$
$$o_t = \sigma\left(W_o\left[h_{t-1}; x_t\right] + b_o\right) \tag{8}$$
$$\tilde{c_t} = \tanh\left(x_t W_c + h_{t-1} W_c + b_c\right) \tag{9}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{10}$$
$$\overrightarrow{h_t} = o_t \odot \tanh\left(c_t\right) \tag{11}$$

where the five symbols f_t, i_t, o_t, $\tilde{c}_t$, and c_t denote the forget gate, input gate, output gate, candidate memory cell, and memory cell, respectively. The symbol σ represents a sigmoid activation function. The symbol tanh denotes the hyperbolic tangent function, i.e., $\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z})$. The symbols W_f, W_i, and W_o denote the weight matrices of the corresponding forget gate, input gate, and output gate. Accordingly, the symbols b_f, b_i, and b_o denote biases for the three gates. The symbol [;] denotes the concatenation operation for two vectors, and the symbol $\odot$ denotes the elementwise product. Similarly, we can obtain the backward hidden states $\overleftarrow{h_t}$. Then, we concatenate two hidden states $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ to form the final hidden state h_t, i.e., $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$ for the word x_t at the tth position of the input sentence.

For the BERT encoder, we construct a sentence–aspect pair, i.e., "[CLS] sentence [SEP] aspect [SEP]" as input to obtain aspect-aware hidden representations of the sentence. Moreover, to match the word-piece-based representations of BERT with the result of syntactic dependency based on words, we expand dependencies of a word into all its subwords. For example, the word "extendable" can be tokenized as "extend" and "##able" by BERT tokenizer.

B. Syntax-Based GCN

The SynGCN module takes the syntactic encoding as input. To encode syntactic information, we use the probability matrix of all the dependency arcs from a dependency parser. Compared with the final discrete output of a dependency parser, the dependency probability matrix could capture richer structural information by providing all the latent syntactic structures. Therefore, the dependency probability matrix is used to alleviate dependency parsing errors. Here, we use the dependency parsing model LAL-Parser [48]. The purpose of using the dependency parser is to obtain the dependency probability matrix of a sentence. We note that some parsers, such as CoreNLP, AllenNLP, and Stanza, can only output the hard structure of the syntactic tree. In other words, they can only produce discrete results. The LAL-Parser meets our needs and it achieves state-of-the-art performance on the syntactic dependency parsing task.

With the syntactic encoding of an adjacency matrix $A^{\text{syn}} \in \mathbb{R}^{n\times n}$, the SynGCN module takes the hidden state vectors H from a BiLSTM as initial node representations in the syntactic graph. The syntactic graph representation $H^{\text{syn}} = \{h_1^{\text{syn}}, h_2^{\text{syn}}, \ldots, h_n^{\text{syn}}\}$ is then obtained from the SynGCN module. Here, the symbol $h_i^{\text{syn}} \in \mathbb{R}^{d_u}$ denotes a hidden representation of the ith node. The representation of the ith node at the lth layer is updated as follows:

$$h_i^{\text{syn}l} = \phi\left(\sum_{j=1}^{n} A^{\text{syn}}{}_{ij} W^l h_j^{\text{syn}l-1} + b^l\right) \tag{12}$$

where the symbol W^l denotes a weight matrix, and the symbol b^l denotes a bias term. Note that for aspect nodes, we use symbols $\{h_{a_1}^{\text{syn}}, h_{a_2}^{\text{syn}}, \ldots, h_{a_m}^{\text{syn}}\}$ to denote their hidden representations.

C. Semantics-Based GCN

Instead of using additional syntactic knowledge as in SynGCN, our SemGCN obtains an attention matrix as an adjacency matrix via a self-attention mechanism. Self-attention can capture the semantically related terms of each word in a sentence, which is more flexible than the syntactic structure. Thus, SemGCN can adapt to online reviews that are not sensitive to syntactic information. We describe SemGCN starting from self-attention.

Self-attention [44] computes the attention score of each pair of elements in parallel. In our DualGCN, we compute the attention score matrix $A^{\text{sem}} \in \mathbb{R}^{n\times n}$ using a self-attention layer. We then take the attention score matrix A^{sem} as the adjacency matrix of our SemGCN module. The attention score matrix A_i^{sem} computed by the ith attention head in self-attention mechanism can be formulated with a softmax as

$$A_i^{\text{sem}} = \text{softmax}\left(\frac{QW_i^Q \times \left(KW_i^K\right)^{\text{T}}}{\sqrt{d_k}}\right) \tag{13}$$

$$d_k = \frac{d_{\text{in}}}{k} \tag{14}$$

where two vectors Q and K are equal to the hidden state vector H produced by the sentence encoder, while W_i^Q and W_i^K are the learnable weight matrices. The symbol T denotes the transpose operation. In addition, the symbol d_k denotes a scaling factor, while the symbol d_{in} is the dimensionality of the input node feature and the constant k is the number of attention heads. As shown in the right part of Fig. 3, each attention head can be calculated to obtain an adjacency matrix.

To enhance semantic graph representation, we apply multiple attention heads to generate more robust adjacency matrix, i.e., graph representation. Therefore, the final semantic graph averages the attention score matrices of all the attention heads as follows:

$$A^{\text{sem}} = \frac{1}{k}\sum_{i=1}^{k} A_i^{\text{sem}}. \tag{15}$$

Similar to the SynGCN module, the SemGCN module obtains the graph representation H^{sem}. In addition, we use the symbols $\{h_{a_1}^{\text{sem}}, h_{a_2}^{\text{sem}}, \ldots, h_{a_m}^{\text{sem}}\}$ to denote the hidden representations of all the aspect nodes.

To effectively exchange relevant features between these SynGCN and SemGCN modules, we adopt a mutual BiAffine transformation as a bridge. We formulate the process as follows:

$$H^{\text{syn}\prime} = \text{softmax}\left(H^{\text{syn}} W_1 \left(H^{\text{sem}}\right)^{\text{T}}\right) H^{\text{sem}} \tag{16}$$

$$H^{\text{sem}\prime} = \text{softmax}\left(H^{\text{sem}} W_2 \left(H^{\text{syn}}\right)^{\text{T}}\right) H^{\text{syn}} \tag{17}$$

where symbols W_1 and W_2 are trainable parameters. To summarize, two vector outputs $H^{\text{syn}\prime}$ and $H^{\text{sem}\prime}$ are produced with two vector inputs H^{syn} and H^{sem} in the BiAffine function.

D. Classifier

Finally, we apply the average pooling and concatenation operations on the aspect nodes of the SynGCN and SemGCN

modules. Thus, we obtain the final feature representation r, i.e., the concatenation of syntactic and semantic representations h_a^{syn} and h_a^{sem} of all aspects for the ABSA task, that is,

$$h_a^{\text{syn}} = \text{ap}\left(h_{a_1}^{\text{syn}} h_{a_2}^{\text{syn}}, \ldots, h_{a_m}^{\text{syn}}\right) \tag{18}$$

$$h_a^{\text{sem}} = \text{ap}\left(h_{a_1}^{\text{sem}} h_{a_2}^{\text{sem}}, \ldots, h_{a_m}^{\text{sem}}\right) \tag{19}$$

$$r = \left[h_a^{\text{syn}}; h_a^{\text{sem}}\right] \tag{20}$$

where an average pooling function $\text{ap}(\cdot)$ is applied over the aspect node representations. Then, the final representation r is fed into a linear layer, followed by a softmax function to produce a sentiment probability distribution p for a given aspect a, that is,

$$p(a) = \text{softmax}\left(W_p r + b_p\right) \tag{21}$$

where the symbols W_p and b_p are the learnable weight and bias, respectively.

E. Regularizer

To further improve the semantic representation, we propose two regularizers for the SemGCN module, i.e., orthogonal and differential regularizers. For the orthogonal regularizer, we have the following observation. Since the related items of each word should be in different regions in a sentence, the distributions of attention scores rarely overlap. Therefore, we expect a regularizer to encourage orthogonality among the attention score vectors of all the words. Given an attention score matrix $A^{\text{sem}} \in \mathbb{R}^{n \times n}$, the orthogonal regularizer R_O is formulated as follows:

$$R_O = \left\| A^{\text{sem}} (A^{\text{sem}})^{\text{T}} - I \right\|_F \tag{22}$$

where the symbol I denotes an identity matrix. The subscript F denotes the Frobenius norm of a matrix. As a result, each nondiagonal element of $A^{\text{sem}}(A^{\text{sem}})^{\text{T}}$ is minimized to maintain the matrix A^{sem} orthogonal.

For the differential regularizer, we have the following expectation. Specifically, we expect that two types of feature representations learned from the SynGCN and SemGCN modules represent distinct information contained within the syntactic dependency trees and semantic correlations. Therefore, we adopt a differential regularizer between the two adjacency matrices built from the SynGCN and SemGCN modules. Note that the differential regularizer R_D is only restrictive to A^{sem} and is given as follows:

$$R_D = \left\| A^{\text{sem}} - A^{\text{syn}} \right\|_F^{-1}. \tag{23}$$

F. Loss and Training

Our training goal is to minimize the following objective:

$$\ell_T = \ell_C + \lambda_1 R_O + \lambda_2 R_D + \lambda_3 \|\Theta\|_2 \tag{24}$$

where these three symbols λ_1, λ_2, and λ_3 denote regularization coefficients, and the symbol $\|\Theta\|_2$ represents L_2 norm for all the trainable model parameters Θ. The loss ℓ_C is a standard cross-entropy loss. For the ABSA task, the loss ℓ_C is defined as follows:

$$\ell_C = -\sum_{(s,a)\in\mathcal{D}} \sum_{c\in\mathcal{C}} \log p(s, a, c) \tag{25}$$

where the set $\mathcal{D}$ contains all the sentence–aspect pairs, and the set $\mathcal{C}$ denotes the collection of distinct sentiment polarities.

Thus, with the total loss ℓ_T, we can train the DualGCN model using the back-propagation algorithm. Once the training is finished, we use the model to inference the sentimental polarities for given sentences and their aspects.

IV. Experimental Settings

In this section, we introduce two groups of datasets used for evaluation and briefly review these baselines to be compared. Then, we concisely describe evaluation metrics and present all the implementation details.

A. Datasets

To evaluate our proposed model, we conduct experiments on two groups of benchmark datasets. The first group comprises three datasets, i.e., Restaurant14, Laptop14, and Twitter. The Restaurant14 and Laptop14 datasets are released publicly from the SemEval ABSA challenge 2014 [49]. For a fair comparison, we remove the instances with the "conflict" label following previous work [20]. In addition, the Twitter dataset is a collection of tweets [50]. The second group comprises two datasets, i.e., Restaurant15 and Restaurant16, which are from SemEval ABSA challenge 2015 [51] and SemEval ABSA challenge 2016 [52], respectively. We preprocess the two datasets as same as Restaurant14 and Laptop14. All the five datasets have three sentimental polarities: positive, neutral, and negative. Each sentence in these datasets is annotated with marked aspects and their corresponding sentimental polarities. The statistics for the five datasets are summarized in Table I. Note that the first group of datasets is more commonly adopted in previous works. In contrast, only a few works adopt the second group of datasets for evaluation. Therefore, we adopt the first group for main evaluation and the second group for subsidiary evaluation.

B. Baseline Methods

To evaluate our proposed DualGCN model, we compare DualGCN with the state-of-the-art baselines. All these models are grouped and briefly reviewed as follows.

1) The first seven methods are the attention-based models.
   a) ATAE-LSTM [17] combines the aspect embedding with all the input word embeddings in a sentence, and then input them into LSTM. Furthermore, this method designs an aspect-to-sentence attention mechanism that can concentrate on the key part of a sentence given the aspect.
   b) MemNet [18] proposes a deep memory network that considers contexts as external memories and benefits from a multihop attention layer structure.
   c) IAN [19] proposes an interactive attention mechanism to interactively learn attentions within the

TABLE I
STATISTICS OF FIVE DATASETS USED FOR EVALUATION

Dataset	Division	# Positive	# Negative	# Neutral
Restaurant14	Train	2164	807	637
	Test	727	196	196
Laptop14	Train	976	851	455
	Test	337	128	167
Twitter	Train	1507	1528	3016
	Test	172	169	336
Restaurant15	Train	1178	382	50
	Test	439	328	35
Restaurant16	Train	1620	709	88
	Test	597	190	38

contexts and aspects and generate the representations for aspects and contexts, respectively.

d) RAM [20] designs a tailor-made memory for different opinion targets of a sentence. Then, it uses multiple attention and a GRU cell to learn the sentence representation for final classification.

e) Inter-aspect [53] proposes a way to incorporate inter-aspect dependencies in the task of ABSA.

f) AOA [22] leverages an attention-over-attention module to learn the important parts in the aspect and sentence, which generates the final representation of the sentence for aspect-level sentiment classification.

g) MGAN [21] leverages both fine-grained and coarse-grained attention mechanisms to build its framework. The fine-grained attention mechanism can capture the word-level interaction between aspects and their context.

2) The eighth and ninth baselines are the CNN-based models.

a) GCAE [54] proposes two convolutional layers with gating mechanisms to capture the aspect and sentiment information, respectively, for the ABSA task.

b) TNet [55] transforms the BiLSTM embeddings into target-specific embeddings and then uses a CNN to obtain final representation for aspect sentiment classification.

3) The successive seven baselines are the GNN-based models.

a) ASGCN [27] first proposes using GCN over dependency tree to exploit syntactical information and word dependencies and to learn the aspect-specific representations for aspect-based sentiment classification.

b) CDT [28] presents a convolution over a dependency tree model. This model exploits a BiLSTM to learn representations for a sentence, and further enhances the sentence embeddings with a GCN.

c) BiGCN [30] uses a global lexical graph to encode the corpus-level word co-occurrence information. Then, the BiGCN builds a concept hierarchy on both the syntactic and lexical graphs to differentiate various types of dependency relationships or lexical word pairs.

d) kumaGCN [31] proposes a gating mechanisms to dynamically combine information from word dependency graphs and latent graphs.

e) InterGCN [32] explores a novel solution of constructing a heterogeneous graph for each instance. The graph is built by leveraging aspect-focused and inter-aspect contextual dependencies for specific aspects.

f) R-GAT [33] defines an aspect-oriented dependency tree structure and then encodes the new dependency tree with a relational graph attention network.

g) DGEDT [34] proposes a dependency graph-enhanced dual-transformer network. The DGEDT jointly considers flat representations learned from transformer and graph-based representations learned from the corresponding dependency graph in an iterative interaction manner.

4) The last five methods are the BERT-based models.

a) BERT [43] uses the vanilla BERT model by feeding the sentence–aspect pairs and then uses the obtained "[CLS]" representation for sentimental predictions.

b) BERT-PT [46] proposes a PT approach on the BERT to improve the performance of the fine-tuning stage. We reimplement the model in our experiments.

c) R-GAT + BERT [33] is a variant of R-the GAT model. This variant uses a pretrained BERT to replace the initial BiLSTM encoder.

d) DGEDT + BERT [34] is a variant of the DGEDT model. Similar to R-GAT + BERT, this variant also uses a pretrained BERT to replace the initial BiLSTM encoder.

e) BERT-ADA [56] first fine-tunes a pretrained BERT model on a domain specific corpus with subsequent training on the down-stream classification task, which is similar to BERT-PT.

Note that major aforementioned baselines are evaluated only on the first group of datasets and a few of them are evaluated on two groups of datasets.

C. Evaluation Metrics

To quantitatively evaluate the model performance, we use the accuracy and macro-averaged F1-score as the main metrics. The accuracy is the fraction of the number of correct predictions in the total number of predictions, that is,

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{Total predictions}}. \tag{26}$$

The macro-averaged F1-score is defined as the mean of classwise F1-scores, that is,

$$\text{Macro F1-score} = \frac{1}{N}\sum_{m=1}^{N}\left(2 \times \frac{P_m \times R_m}{P_m + R_m}\right) \tag{27}$$

in which P_m and R_m denote the precision and recall values for the mth class, respectively, and N is the total number of

TABLE II
COMPARISON RESULTS ON THE FIRST GROUP OF DATASETS. ALL BASELINE RESULTS ARE COPIED FROM THE ORIGINAL ARTICLES AND UNREPORTED RESULTS ARE REPRESENTED WITH A HYPHEN MARK "-"

Models	Restaurant14		Laptop14		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
ATAE-LSTM [17]	77.20	-	68.70	-	-	-
MemNet [18]	79.61	69.64	70.64	65.17	71.48	69.90
IAN [19]	78.60	-	72.10	-	-	-
RAM [20]	80.23	70.80	74.49	71.35	69.36	67.30
Inter-aspect [53]	79.00	-	72.50	-	-	-
AOA [22]	79.97	70.42	72.62	67.52	72.30	70.20
MGAN [21]	81.25	71.94	75.39	72.47	72.54	70.81
GCAE [54]	77.28	-	69.14	-	-	-
TNet [55]	80.69	71.27	76.54	71.75	74.90	73.60
ASGCN [27]	80.77	72.02	75.55	71.05	72.15	70.40
CDT [28]	82.30	74.02	77.19	72.99	74.66	73.66
BiGCN [30]	81.97	73.48	74.59	71.84	74.16	73.35
kumaGCN [31]	81.43	73.64	76.12	72.42	72.45	70.77
InterGCN [32]	82.23	74.01	77.86	74.32	-	-
R-GAT [33]	83.30	76.08	77.42	73.76	75.57	73.82
DGEDT [34]	83.90	75.10	76.80	72.30	74.80	73.40
Our DualGCN	**84.27**	**78.08**	**78.48**	**74.74**	**75.92**	**74.29**
BERT [43]	86.15	80.29	81.01	76.69	75.18	74.01
BERT-PT [46]	87.76	81.48	81.80	78.08	75.04	73.92
R-GAT+BERT [33]	86.60	81.35	78.21	74.07	76.15	74.88
DGEDT+BERT [34]	86.30	80.00	79.80	75.60	**77.90**	75.40
BERT-ADA [56]	87.14	80.05	79.19	74.18	-	-
Our DualGCN+BERT	87.13	81.16	81.80	78.10	77.40	**76.02**
Our DualGCN+BERT-PT	**88.47**	**82.92**	**82.59**	**79.34**	76.37	75.44

classes. Note that macro F1-score is more suitable for micro F1-score in our settings.

D. Implementation Details

We adopt LAL-Parser [48] for the dependency parsing in all our experiments. LAL-Parser provides an off-the-shelf parser with the state-of-the-art performance.[1] To initialize word embeddings, we use pretrained 300-D Glove[2] vectors [57] in all the experiments. In addition, the dimensionality of position (i.e., the relative position of each word in a sentence with respect to the aspect) embeddings and that of POS embeddings is set to 30. Thus, we concatenate the three embeddings of words, POS, and position, and then input them into a BiLSTM model, whose hidden size is set to 50. To summarize, four hyperparameters d_u, d_e, d_t, and d_p are set to 50, 300, 30, and 30, respectively.

Furthermore, to alleviate the problem of overfitting, we apply dropout [58]. Dropout is a technique where randomly selected neurons are ignored during training. Neurons are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass. Specifically, the dropout rate of 0.7 is set to the input word embeddings of BiLSTM. The dropout rate of the SynGCN and SemGCN modules is set to 0.1, and the number of SynGCN and SemGCN layers is set to 2. All the model weights are initialized using a uniform distribution, i.e., $U(-0.3, 0.3)$. We use the Adam optimizer with a learning rate of 0.002. The DualGCN model is trained in 50 epochs with a batch size of 16. The two regularization coefficients, λ_1 and λ_2, are set to (0.2, 0.3), (0.2, 0.2), (0.3, 0.2), (0.8, 0.1), and (0.9, 0.1) for the Restaurant14, Laptop14, Twitter, Restaurant15, and Restaurant16 datasets, respectively. The coefficient λ_3 is set to 10^{-4}. For the DualGCN + BERT model, we use the English version of BERT-base-uncased.[3] DualGCN + BETR-PT replaces the vanilla BERT with BERT-PT[4] which is a PT BERT model for cross-domain reviews from Amazon and Yelp. Our source code gives more details about BERT's experiments. In addition, following Marcheggiani and Titov [59], we add a self-loop for each node in the SynGCN and SemGCN modules.

V. RESULTS AND ANALYSIS

In this section, we report the quantitatively comparison results on two groups of datasets. Then, based on the first group of datasets, we report the experimental results of ablation studies and show qualitatively experimental results.

A. Major Comparison Results

The main experimental results compared with the state-of-the-art models are reported in Table II. A few baselines reported their performance only on Restaurant14 and Laptop14 without Twitter with the accuracy metric. Hence, we use a

[1]https://github.com/KhalilMrini/LAL-Parser
[2]https://nlp.stanford.edu/projects/glove/
[3]https://github.com/huggingface/transformers
[4]https://github.com/howardhsu/BERT-for-RRC-ABSA/blob/master/transformers/amazon_yelp.md

hyphen mark "-" to denote those unreported results. Our DualGCN model consistently outperforms all the attention-based and syntax-based methods on the Restaurant14, Laptop14, and Twitter datasets. These results demonstrate that our DualGCN effectively integrates syntactic knowledge and semantic information. In addition, the DualGCN can accurately fit datasets that contain formal, informal, or complicated reviews. Compared with the attention-based methods such as ATAE-LSTM, IAN, and RAM, our DualGCN model uses the syntactic knowledge to establish dependencies between words. Thus, DualGCN could to some extent avoid noises introduced by the attention mechanism. Moreover, the syntax-based methods such as ASGCN, CDT, and R-GAT achieve better performance than those attention-based methods. Note that these syntax-based methods ignore the semantic correlation between words. However, when addressing informal or complicated sentences, these methods based only on syntactic knowledge usually result in poor performance.

In addition, we note that the results of the basic BERT outperform most of the models based on static word embedding, which is shown in the last group in Table II. A BERT-enhanced version, our DualGCN + BERT achieves better performance compared with the basic DualGCN model. Moreover, our DualGCN + BERT achieves better performance in most metrics compared with the BERT-based methods, including BERT, R-GAT + BERT, and DGEDT + BERT. Furthermore, we observe that DualGCN + BERT-PT (i.e., with PT) significantly surpasses other methods on the two datasets of Restaurant14 and Laptop14. However, DualGCN + BERT-PT does not perform well in the Twitter dataset. We suppose that this is because BERT-PT is posttrained on Amazon and Yelp domain reviews, which are more similar to the Restaurant14 and Laptop14 datasets compared with the Twitter dataset.

B. Minor Comparison Results

To verify the robustness of our DualGCN model, we conduct further experiments on two datasets Restaurant15 and Restaurant16. As shown in Table III, we compare DualGCN with the attention-based methods (i.e., MemNet and IAN), CNN-based method (i.e, TNet), and GNN-based methods (i.e., ASGCN, CDT and BiGCN). Under the accuracy metric, the experimental results show that our DualGCN model outperforms all other baselines. DualGCN performs very well even on more datasets, indicating its strong consistency and robustness.

C. On Pretrained Models

To demonstrate the role of extending various pretrained language models (PLMs), we compare our DualGCN equipped with ALBERT [60], DistilBERT [61], BERT [43], and RoBERTa [62]. The experiments are conducted on the Restaurant14 and Laptop14 datasets, and the results are shown in Table IV. Specifically, we exploit different PLMs as sentence encoders for DualGCN. Note that ALBERT and DistilBERT are lightweight PLMs. ALBERT reduces the parameters of PLM by factorized embedding parameterization and cross-layer parameter sharing, while DistilBERT leverages knowledge distillation during the pretraining phase. The experimental results show that although they have fewer model parameters, their performances are competitive. Compared with BERT, RoBERTa set up larger batch size and used more training data with dynamic masking technique during the pretraining phase. In consequence, DualGCN with RoBERTa achieve better results than BERT.

TABLE III
EXPERIMENTAL RESULTS ON RESTAURANT15 AND RESTAURANT16. THE RESULTS WITH THE MARK "†" ARE COPIED FROM [27]. UNREPORTED RESULTS ARE REPRESENTED WITH A HYPHEN MARK "-"

Models	Restaurant15		Restaurant16	
	Accuracy	Macro-F1	Accuracy	Macro-F1
MemNet† [18]	77.31	58.28	85.44	65.99
IAN† [19]	78.54	52.65	84.74	55.21
TNet† [55]	78.47	59.47	89.07	70.43
ASGCN† [27]	79.89	61.89	88.99	67.48
CDT [28]	-	-	85.58	69.93
BiGCN [30]	81.16	64.79	88.96	**70.84**
DualGCN	**81.73**	**65.05**	**89.29**	68.08

D. Ablation Study

To further investigate the effectiveness of modules in the DualGCN model, we conduct thorough ablation studies. All the experimental results are reported in Table V. The SynGCN-head model uses the discrete outputs of a dependency parser to construct the adjacency matrix of GCNs. In contrast, the SynGCN model directly leverages the probability matrix generated in a dependency parser as the adjacency matrix. The SynGCN model outperforms SynGCN-head on the Restaurant14 and Laptop14 datasets, which demonstrates that rich syntactic knowledge can alleviate dependency parsing errors. The SemGCN model uses a self-attention layer to construct the adjacency matrix of the semantic graph. This SemGCN model outperforms SynGCN on the Twitter dataset. This is because reviews from Twitter, compared with those from the Restaurant14 and Laptop14 datasets, are largely informal and insensitive to syntactic information. DualGCN w/o BiAffine means that we remove the BiAffine module so that the SynGCN and SemGCN modules cannot interact with each other. Therefore, the performance degrades substantially on the Restaurant14 and Laptop14 datasets. DualGCN w/o $R_O \& R_D$ indicates that we remove both the orthogonal and differential regularizers. Similarly, DualGCN w/o R_O or R_D denotes that we remove only one regularizer. The experimental results show that our proposed two regularizers encourage DualGCN to accurately capture semantic correlations. Overall, our DualGCN with full modules achieves the best performance.

E. Case Study

Table VI shows a few sample cases analyzed using different models. In this table, notations P, N, and O represent positive, negative, and neutral sentiment categories, respectively. We underline and italicize the aspect words in these cases. For the aspect "food" in the first sample, the attention-based methods, i.e., ATAE-LSTM and IAN, are prone to attend to the noisy word "dreadful." Although syntactic dependency

TABLE IV
COMPARISON OF DUALGCN EQUIPPED WITH VARIOUS PRETRAINED LANGUAGE MODELS ON RESTAURANT14 AND LAPTOP14

PLMs	Parameter Scale	Restaurant14		Laptop14	
		Accuracy	Macro-F1	Accuracy	Macro-F1
ALBERT-base [60]	12M	84.99	77.05	78.48	74.65
ALBERT-large [60]	18M	86.15	80.31	79.59	76.43
DistilBERT-base [61]	66M	85.25	78.59	79.27	75.18
BERT-base [43]	110M	87.13	81.16	**81.80**	78.10
RoBERTa-base [62]	125M	**87.40**	**82.30**	**81.80**	**78.57**

TABLE V
EXPERIMENTAL RESULTS OF ABLATION STUDY

Models	Restaurant14		Laptop14		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
SynGCN-head	82.93	75.29	76.27	72.39	75.04	73.85
SynGCN	83.74	76.97	76.58	73.17	74.59	72.86
SemGCN	83.29	76.30	76.90	73.72	75.18	73.86
DualGCN w/o BiAffine	82.84	75.31	76.90	73.23	75.33	73.92
DualGCN w/o R_O & R_D	82.93	75.79	76.58	72.03	74.59	73.20
DualGCN w/o R_O	83.56	77.43	76.58	72.78	75.18	73.55
DualGCN w/o R_D	83.65	76.34	77.53	73.72	74.45	72.82
DualGCN	**84.27**	**78.08**	**78.48**	**74.74**	**75.92**	**74.29**

TABLE VI
CASE STUDIES OF OUR DUALGCN MODEL COMPARED WITH STATE-OF-THE-ART BASELINES

#	Review	ATAE-LSTM	IAN	SynGCN	SemGCN	DualGCN	DualGCN+BERT-PT
1	Great [food] but the [service] was dreadful!	(N✗, N✓)	(N✗, N✓)	(P✓, N✓)	(P✓, N✓)	(P✓, N✓)	(P✓, N✓)
2	[Works] well, and I am extremely happy to be back to an [apple OS].	(P✓, P✓)	(P✓, P✓)	(P✓, O✗)	(P✓, P✓)	(P✓, P✓)	(P✓, P✓)
3	Did not enjoy the new [Windows 8] and [touchscreen functions].	(O✗, P✗)	(O✗, N✓)	(P✗, O✗)	(N✓, N✓)	(N✓, N✓)	(N✓, N✓)
4	I never tried any [external mics] with that iMac.	O✓	N✗	N✗	N✗	O✓	O✓
5	In mi burrito, here was nothing but dark [chicken] that had that cooked last week and just warmed up in a microwave [taste].	(N✓, P✗)	(N✓, N✓)	(N✓, O✗)	(N✓, O✗)	(N✓, N✓)	(N✓, N✓)
6	The [staff] should be a bit more friendly.	P✗	P✗	P✗	P✗	P✗	N✓

can establish direct connections between an aspect and some words, no association exists between the aspect and the opinion words for complicated sentences. Take the second sample as an example; the aspect "apple os" is far away from the opinion word "happy" in terms of syntactic distance. Thus, the SynGCN model fails. In addition, in the third sample, feature representations of the key words "did not" are not captured by the SynGCN model. In contrast, the SemGCN model can attend to the semantic correlation between words. The fourth and fifth cases demonstrate that our DualGCN, which fully considers the complementarity of syntactic knowledge and semantic information, can address complicated and informal sentences with the help of orthogonal and differential regularizers. However, for the last review (i.e., #6) using subjunctive style, except DualGCN + BERT-PT, the other models fail to predict the sentimental polarities. In this case, the difficulty of prediction is to infer the implicit semantics, which requires a strong ability of natural language understanding. Our DualGCN + BERT-PT model can correctly predict the sentiment polarity by means of the PT language model BERT-PT.

F. Effect of Different Parsers

The dependency parsing plays an important role in those GCN-based models. To quantitatively evaluate the impact of different parsers, we conduct an investigation based on the SynGCN and DualGCN models. We use four well-known dependency parsers, including CoreNLP Parser,[5] AllenNLP Parser,[6] Stanza Parser[7], and LAL-Parser.[8] Note that here we only use the discrete syntactic dependency structure output by these parsers rather than the dependency probability matrix to construct the adjacency matrix of graph. Table VII shows the performance of SynGCN and DualGCN using different parsers. As shown at the top group in Table VII, SynGCN-LALParser performs better on Restaurant14, while SynGCN-Stanza performs better on Laptop14. CoreNLP Parser seems to be more suitable for the Twitter dataset. At the bottom group in Table VII, DualGCN with different parsers in general achieves an overall improvement against the corresponding SynGCN. That is because SemGCN effectively enhances semantic information. Therefore, we can conclude that the method combining semantic and syntactic information can achieve better performance than the syntactic structure information alone on the ABSA task.

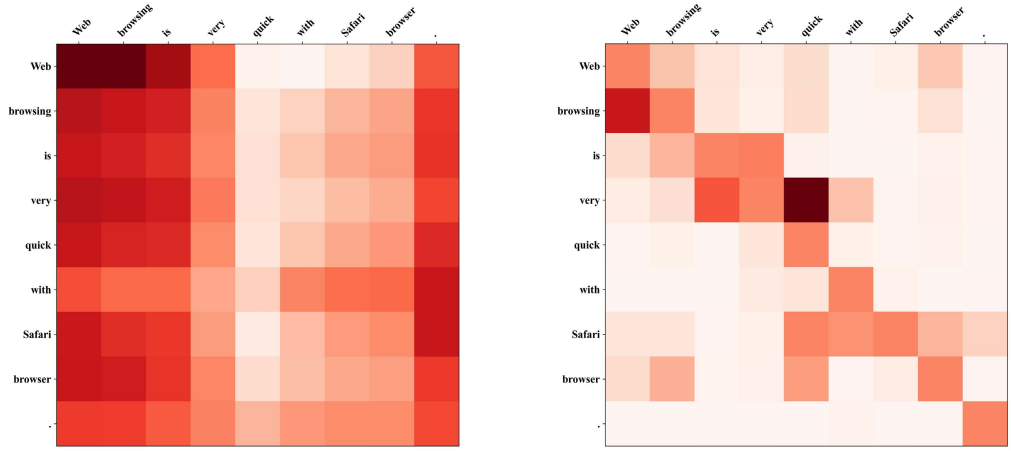
G. Attention Visualization

To investigate the effectiveness of the two regularizers in capturing the semantic correlations between words, we visualize the attention score matrix of DualGCN w/o R_O&R_D

[5]https://stanfordnlp.github.io/CoreNLP/
[6]https://demo.allennlp.org/dependency-parsing
[7]https://stanfordnlp.github.io/stanza/
[8]https://github.com/KhalilMrini/LAL-Parser

TABLE VII
EXPERIMENTAL RESULTS WITH VARIOUS DEPENDENCY PARSERS

Models	Restaurant14		Laptop14		Twitter	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
SynGCN-CoreNLP	82.13	72.46	76.42	72.75	**75.33**	**73.89**
SynGCN-AllenNLP	82.13	74.27	74.53	70.55	73.71	72.22
SynGCN-Stanza	82.22	73.98	**76.74**	**72.78**	72.97	70.47
SynGCN-LALParser	**82.93**	**75.29**	76.27	72.39	75.04	73.85
DualGCN-CoreNLP	82.66	75.23	77.53	**73.88**	74.45	72.69
DualGCN-AllenNLP	**82.84**	**76.03**	76.58	72.13	74.30	73.31
DualGCN-Stanza	82.22	73.96	**77.85**	73.48	74.30	72.11
DualGCN-LALParser	**82.84**	75.80	76.90	73.24	**75.48**	**74.12**



(a) (b)

Fig. 4. Illustration on how orthogonal and differential regularizers contribute to the self-attention layer. (a) The attention score matrix of DualGCN w/o R_O & R_D. (b) The attention score matrix of DualGCN.

and intact DualGCN. Consider the sample sentence, i.e., "Web browsing is very quick with Safari browser." with "Safari browser" as an aspect. As shown in Fig. 4(a), the attention score matrix is dense, and the related terms of each word overlap in the DualGCN w/o R_O&R_D model. This result is attributed to the lack of semantic constraints in the self-attention layers. The overlap of semantic correlations will lead to redundancy and noise during information propagation. The seventh and eighth rows of the attention score matrix are the attention probability distributions of "safari" and "browser," respectively. The information to which "safari browser" pays attention is redundant and it does not pay more attention to the key opinion word "quick." Thus, DualGCN w/o R_O&R_D fails. In comparison, in Fig. 4(b), the attention score matrix produced by our DualGCN is relatively sparse. Both "safari" and "browser" are semantically related to "quick," and their other attended items are also semantically reasonable. In addition, the attention scores of the related terms of each word tend to be distinct and precise due to the semantic constraints of these two regularizers. Therefore, our DualGCN model can readily predict the correct sentiment polarity of the aspect "safari browser."

H. Effect of the DualGCN Layer Number

To investigate the impact of the DualGCN layer number, we evaluate our DualGCN model from one to eight layers on the three public datasets. As shown in Fig. 5, our model with two DualGCN layers performs the best. On one hand, node representations cannot propagate far when the number of layers is small. On the other hand, if the number of layers is excessive, the model will become unstable due to the vanishing gradient and information redundancy. From another perspective, if there are many iterations of graph convolution, the representation of each node in the same connected domain will tend to converge to the same value, that is, the same position on the feature space, also known as over-smoothing phenomenon. Then, it will be difficult to distinguish the information of each node.

VI. RELATED WORKS

Textual sentiment analysis tasks are traditionally sentence-level or document-level oriented. In contrast, ABSA is an entity-level oriented and a more fine-grained task for textual sentiment analysis. Earlier methods [63], [64], [65], [66] are usually based on handcrafted features but fail to model the dependency between the given aspect and its context.

Recently, various attention-based neural networks [17], [18], [18], [19], [20], [21], [22], [23], [67], [68], [69] have been proposed to implicitly model the semantic relationship of an aspect and its context. For instance, Wang et al. [17] proposed attention-based LSTMs for aspect-level sentiment classification. Chen et al. [20] and Tang et al. [18] introduced a hierarchical attention network to identify important sentiment information related to the given aspect. Fan et al. [21]

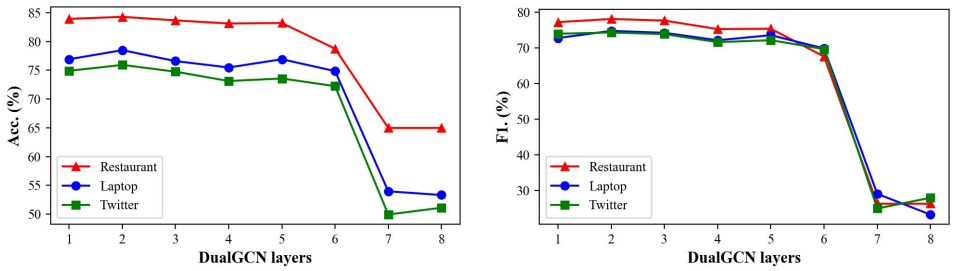


Fig. 5. Effect of the number of DualGCN layers on the Restaurant14, Laptop14, and Twitter datasets.

exploited a multigrained attention mechanism to capture word-level interaction between aspects and their context. Tan et al. [68] designed a dual attention network to recognize conflicting opinions. In addition, the pretrained language model BERT [43] has achieved remarkable performance in various NLP tasks, including ABSA. Sun et al. [70] transformed the ABSA task into a sentence pair classification task by constructing an auxiliary sentence. Xu et al. [46] proposed a PT approach on the BERT to enhance the performance of fine-tuning stage for the ABSA task.

Another trend explicitly leverages syntactic knowledge. This type of knowledge helps establish connections between the aspects and the other words in a sentence to learn syntax-aware feature representations of aspects. Dong et al. [50] proposed a recursive neural network to adaptively propagate the sentiment of words to the aspect along the dependency tree for handling the negation or intensification phenomena in aspect-based sentiment analysis. Che et al. [71] presented a framework for using a sentiment sentence compression model for ABSA which can compress a complicated sentiment sentence into one that is shorter and easier to parse. Accordingly, the most important features for ABSA, i.e., syntactic features, can be more correctly acquired to enhance the performance of this task. He et al. [72] introduced an attention model that incorporated syntactic information to compute attention weights. Phan and Ogunbona [73] used the syntactic relative distance to reduce the impact of irrelevant words.

Following this line, a few works extend the GCN and GAT models by means of a syntactical dependency tree and develop several outstanding models [27], [28], [29], [33], [34], [74]. For instance, Zhang et al. [27] proposed using GCN over dependency tree to exploit syntactical information and word dependencies and to learn the aspect-specific representations for aspect-based sentiment classification. Wang et al. [33] defined a unified aspect-oriented dependency tree structure rooted at a target aspect by reshaping and pruning an ordinary dependency parse tree, and then used a relational graph attention network to encode the novel tree structure for sentiment prediction. Zhang et al. [74] proposed a capsule attention network with the GCN-based syntactic layer to integrate the syntactic knowledge into the attention layer. As mentioned above, these works explicitly exploit the syntactic structure information to learn node representations from adjacent nodes. Thus, the dependency tree shortens the distance between the aspects and opinion words of a sentence and alleviates the problem of long-range dependency.

Most recently, several works explore the idea of combining different types of graph for ABSA task. For instance, Zhang and Qian [30] observed the characteristics of word co-occurrence in linguistics and designed hierarchical syntactic and lexical graphs. Chen et al. [31] combined a dependency graph and a latent graph to generate aspect representation. Liang et al. [32] constructed aspect-focused and inter-aspect graphs to learn dependency feature of the key aspect words and sentiment relationships between different aspects.

In this article, we propose a GCN-based method combining syntactic and semantic features. We use a dependency probability matrix with richer syntactic information and elaborately design orthogonal and differential regularizers to enhance the ability to further precisely capture the semantic associations.

VII. Conclusion

In this article, we propose a DualGCN architecture to address the disadvantages of the attention-based and dependency-based methods for the ABSA task. Our DualGCN model integrates syntactic knowledge and semantic information by means of the SynGCN and SemGCN modules. Moreover, to effectively capture the semantic correlation between words, we propose orthogonal and differential regularizers in the SemGCN module. These regularizers can attend to the semantically related items with less overlap of each word and capture feature representations that differ from the syntactic structure. Extensive experiments on benchmark datasets show that our DualGCN model outperforms baselines.

In the future, two interesting research issues could be extended from this article. First, the dependency parser is off-the-shelf, which could be improved by replacing it with a trainable module; in addition, BERT can implicitly embed dependency information. Thus, it is worthwhile to explore whether the two parts can be merged in a multitask learning method with the shared BERT encoder. Second, we would like to consider exploring syntactic and semantic information with graph neural networks for the aspect–opinion–sentiment triplet extraction task. We hope that this interesting work could obtain better interpretation for the clues of aspect-based sentiment classification.

Acknowledgment

The authors thank the editors and anonymous reviewers for their constructive feedback.

References

[1] 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. [Online]. Available: https://www.morganclaypool.com/doi/abs/10.2200/S00416ED1V01Y201204HLT016, doi: 10.2200/S00416ED1V01Y201204HLT016.

[2] J. Li and E. Hovy, "Reflections on sentiment/opinion analysis," in *A Practical Guide to Sentiment Analysis*, E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, Eds. Cham, Switzerland: Springer, 2017, pp. 41–59.

[3] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," in *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4. U.K.: Wiley, 2018, p. e1253. [Online]. Available: https://wires.onlinelibrary.wiley.com/journal/19424795, doi: 10.1002/widm.1253.

[4] G. Brauwers and F. Frasincar, "A survey on aspect-based sentiment classification," *ACM Comput. Surv.*, Dec. 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3503044, doi: 10.1145/3503044.

[5] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, and B. Schuller, "Deep neural networks for acoustic emotion recognition: Raising the benchmarks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 5688–5691.

[6] P. Tzirakis, J. Zhang, and B. W. Schuller, "End-to-end speech emotion recognition using deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5089–5093.

[7] M. S. Fahad, A. Ranjan, J. Yadav, and A. Deepak, "A survey of speech emotion recognition in natural environment," *Digit. Signal Process.*, vol. 110, Mar. 2021, Art. no. 102951.

[8] L. Vadicamo et al., "Cross-media learning for image sentiment analysis in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 308–317.

[9] A. Ortis, G. M. Farinella, and S. Battiato, "Survey on visual sentiment analysis," *IET Image Process.*, vol. 14, no. 8, pp. 1440–1456, Jun. 2020.

[10] S. Zhao et al., "Emotional semantics-preserved and feature-aligned CycleGAN for visual emotion adaptation," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 1–14, Mar. 2021.

[11] M. Soleymani, D. Garcia, B. Jou, B. Schuller, S.-F. Chang, and M. Pantic, "A survey of multimodal sentiment analysis," *Image Vis. Comput.*, vol. 65, no. 1, pp. 3–14, Sep. 2017.

[12] L. Stappen, B. Schuller, I. Lefter, E. Cambria, and I. Kompatsiaris, *Summary of MuSe: Multimodal Sentiment Analysis, Emotion-Target Engagement and Trustworthiness Detection in Real-Life Media*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 4769–4770.

[13] C. Musto, P. Lops, M. de Gemmis, and G. Semeraro, "Justifying recommendations through aspect-based sentiment analysis of users reviews," in *Proc. 27th ACM Conf. User Model., Adaptation Personalization*, New York, NY, USA, Jun. 2019, pp. 4–12.

[14] C. Huang, W. Jiang, J. Wu, and G. Wang, "Personalized review recommendation based on Users' aspect sentiment," *ACM Trans. Internet Technol.*, vol. 20, no. 4, pp. 1–26, Oct. 2020.

[15] P. Liu, L. Zhang, and J. A. Gulla, "Multilingual review-aware deep recommender system via aspect-based sentiment analysis," *ACM Trans. Inf. Syst.*, vol. 39, no. 2, pp. 1–33, Jan. 2021.

[16] M. Dragoni, "A three-phase approach for exploiting opinion mining in computational advertising," *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 21–27, May 2017.

[17] 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 for Computational Linguistics, Nov. 2016, pp. 606–615.

[18] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Austin, TX, USA: Association for Computational Linguistics, Nov. 2016, pp. 214–224.

[19] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Palo Alto, CA, USA: AAAI Press, Aug. 2017, pp. 4068–4074.

[20] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 452–461.

[21] F. Fan, Y. Feng, and D. Zhao, "Multi-grained attention network for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 3433–3442.

[22] B. Huang, Y. Ou, and K. M. Carley, "Aspect level sentiment classification with attention-over-attention neural networks," in *Proc. Social, Cultural, Behav. Model. 11th Int. Conf. (SBP-BRiMS)*, in Lecture Notes in Computer Science, vol. 10899, C. L. Dancy, A. Hyder, and H. Bisgin, Eds. Washington, DC, USA: Springer, Jul. 2018, pp. 197–206.

[23] S. Gu, L. Zhang, Y. Hou, and Y. Song, "A position-aware bidirectional attention network for aspect-level sentiment analysis," in *Proc. 27th Int. Conf. Comput. Linguistics*. Santa Fe, NM, USA: Association for Computational Linguistics, Aug. 2018, pp. 774–784.

[24] N. Jiang, F. Tian, J. Li, X. Yuan, and J. Zheng, "MAN: Mutual attention neural networks model for aspect-level sentiment classification in SIoT," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2901–2913, Apr. 2020.

[25] P. Lin, M. Yang, and J. Lai, "Deep selective memory network with selective attention and inter-aspect modeling for aspect level sentiment classification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1093–1106, 2021.

[26] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Jan. 2020.

[27] C. Zhang, Q. Li, and D. Song, "Aspect-based sentiment classification with aspect-specific graph convolutional networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 4568–4578.

[28] K. Sun, R. Zhang, S. Mensah, Y. Mao, and X. Liu, "Aspect-level sentiment analysis via convolution over dependency tree," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 5679–5688.

[29] B. Huang and K. Carley, "Syntax-aware aspect level sentiment classification with graph attention networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 5469–5477.

[30] M. Zhang and T. Qian, "Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 3540–3549.

[31] C. Chen, Z. Teng, and Y. Zhang, "Inducing target-specific latent structures for aspect sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2020, pp. 5596–5607.

[32] B. Liang, R. Yin, L. Gui, J. Du, and R. Xu, "Jointly learning aspect-focused and inter-aspect relations with graph convolutional networks for aspect sentiment analysis," in *Proc. 28th Int. Conf. Comput. Linguistics*. Barcelona, Spain: International Committee on Computational Linguistics, Dec. 2020, pp. 150–161.

[33] K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, "Relational graph attention network for aspect-based sentiment analysis," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 3229–3238.

[34] H. Tang, D. Ji, C. Li, and Q. Zhou, "Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 6578–6588.

[35] R. Li, H. Chen, F. Feng, Z. Ma, X. Wang, and E. Hovy, "Dual graph convolutional networks for aspect-based sentiment analysis," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2021, pp. 6319–6329.

[36] X. Zhang, C. Xu, X. Tian, and D. Tao, "Graph edge convolutional neural networks for skeleton-based action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3047–3060, Aug. 2020.

[37] Y. Xu, C. Han, J. Qin, X. Xu, G. Han, and S. He, "Transductive zero-shot action recognition via visually connected graph convolutional networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 1–9, Aug. 2020.

[38] W. Liu et al., "Item relationship graph neural networks for E-commerce," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 1–15, Mar. 2021.

[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–14.

[40] C. Lin, T. Miller, D. Dligach, S. Bethard, and G. Savova, "A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction," in *Proc. 2nd Clin. Natural Lang. Process. Workshop*. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 65–71.

[41] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang, "Multi-passage BERT: A globally normalized BERT model for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 5878–5882.

[42] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.* Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992.
[43] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.
[44] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 5998–6008.
[45] Y. Zhu et al., "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Washington, DC, USA: IEEE Computer Society, Dec. 2015, pp. 19–27.
[46] H. Xu, B. Liu, L. Shu, and P. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 2324–2335.
[47] S. Gururangan et al., "Don't stop pretraining: Adapt language models to domains and tasks," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2020, pp. 8342–8360.
[48] K. Mrini, F. Dernoncourt, Q. Tran, T. Bui, W. Chang, and N. Nakashole, "Rethinking self-attention: Towards interpretability in neural parsing," 2019, *arXiv:1911.03875*.
[49] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 27–35.
[50] L. Dong, F. Wei, C. Tan, and D. Tang, "Adaptive recursive neural network for target-dependent Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*. Baltimore, MD, USA: Association for Computational Linguistics, Jun. 2014, pp. 49–54.
[51] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 Task 12: Aspect based sentiment analysis," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*. Denver, CO, USA: Association for Computational Linguistics, Jun. 2015, pp. 486–495.
[52] M. Pontiki et al., "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*. San Diego, CA, USA: Association for Computational Linguistics, Jun. 2016, pp. 19–30.
[53] D. Hazarika, S. Poria, P. Vij, G. Krishnamurthy, E. Cambria, and R. Zimmermann, "Modeling inter-aspect dependencies for aspect-based sentiment analysis," in *Proc. Conf. North Amer. Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2. New Orleans, LA, USA: Association for Computational Linguistics, Jun. 2018, pp. 266–270.
[54] W. Xue and T. Li, "Aspect based sentiment analysis with gated convolutional networks," 2018, *arXiv:1805.07043*.
[55] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*. Melbourne, VIC, Australia, Jul. 2018, pp. 946–956.
[56] A. Rietzler, S. Stabinger, P. Opitz, and S. Engl, "Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification," in *Proc. 12th Lang. Resour. Eval. Conf.* Marseille, France: European Language Resources Association: Association for Computational Linguistics, May 2020, pp. 4933–4941.
[57] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 1532–1543.
[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[59] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in *Proc. EMNLP*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 1506–1515.
[60] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.
[61] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.
[62] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[63] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proc. 17th Int. Conf. World Wide Web (WWW)*, New York, NY, USA: Association for Computing Machinery, 2008, pp. 111–120.
[64] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter sentiment classification," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics*. Portland, OR, USA: Association for Computational Linguistics, Jun. 2011, pp. 151–160.
[65] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "NRC-Canada-2014: Detecting aspects and sentiment in customer reviews," in *Proc. 8th Int. Workshop Semantic Eval.* Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 437–442.
[66] D.-T. Vo and Y. Zhang, "Deep learning for event-driven stock prediction," in *Proc. IJCAI*. BueNos Aires, Argentina, Aug. 2015, pp. 1–7.
[67] L. Li, Y. Liu, and A. Zhou, "Hierarchical attention based position-aware network for aspect-level sentiment analysis," in *Proc. 22nd Conf. Comput. Natural Lang. Learn.* Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 181–189.
[68] X. Tan, Y. Cai, and C. Zhu, "Recognizing conflict opinions in aspect-level sentiment classification with dual attention networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 3426–3431.
[69] Y. Zhang, B. Xu, and T. Zhao, "Convolutional multi-head self-attention on memory for aspect sentiment classification," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 4, pp. 1038–1044, Jul. 2020.
[70] C. Sun, L. Huang, and X. Qiu, "Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol. (NAACL-HLT)*. vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 380–385.
[71] W. Che, Y. Zhao, H. Guo, Z. Su, and T. Liu, "Sentence compression for aspect-based sentiment analysis," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 12, pp. 2111–2124, Dec. 2015.
[72] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "Effective attention modeling for aspect-level sentiment classification," in *Proc. 27th Int. Conf. Comput. Linguistics*. Santa Fe, NM, USA: Association for Computational Linguistics, Aug. 2018, pp. 1121–1131.
[73] M. H. Phan and P. O. Ogunbona, "Modelling context and syntactical features for aspect-based sentiment analysis," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 3211–3220.
[74] B. Zhang, X. Li, X. Xu, K.-C. Leung, Z. Chen, and Y. Ye, "Knowledge guided capsule attention network for aspect-based sentiment analysis," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2538–2551, 2020.

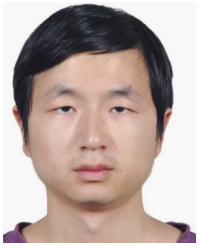
Ruifan Li (Member, IEEE) received the B.S. degree in control systems and the M.S. degree in circuits and systems from the Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001, respectively, and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2006.

Since 2006, he has been with the School of Computer Science, BUPT. Since February 2011, he has been a Visiting Scholar with the Information Sciences Institute, University of Southern California, Los Angeles, CA, USA. He is currently an Associate Professor with the School of Artificial Intelligence, BUPT. His current research activities include multimedia information processing, natural language processing, and statistical machine learning.

Dr. Li is a member of the IEEE Signal Processing Society and the IEEE Computer Society.

Hao Chen received the B.E. degree in network engineering from Hefei University, Hefei, China, in 2019, and the master's degree in computer technology from the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China, in June 2022.

His research interests include natural language processing and deep learning.



Fangxiang Feng received the B.S. and Ph.D. degrees from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2010 and 2015, respectively.

He is currently an Assistant Professor with the School of Artificial Intelligence, BUPT. His research interests include multimedia information retrieval, multimodal deep learning, and computer vision.

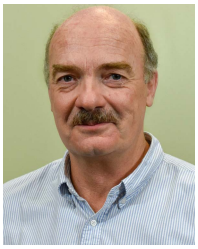
Zhanyu Ma (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2011.

From 2012 to 2013, he was a Post-Doctoral Research Fellow with the School of Electrical Engineering, KTH Royal Institute of Technology. He has been an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China, from 2014 to 2019, where he is currently a Professor. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in computer vision and multimedia signal processing.

Xiaojie Wang received the Ph.D. degree from Beihang University, Beijing, China, in 1996.

He is currently a Full Professor and the Director of the Centre for Intelligence Science and Technology, Beijing University of Posts and Telecommunications, Beijing. His research interests include natural language processing and multimodal cognitive computing.

Dr. Wang is an Executive Member of the Council of Chinese Association of Artificial Intelligence and the Director of the Natural Language Processing Committee. He is a member of the Council of Chinese Information Processing Society and the Chinese Processing Committee of China Computer Federation.



Eduard Hovy received the Ph.D. degree in computer science from Yale University, New Haven, CT, USA, in May 1987.

He received the honorary doctorates from the National University of Distance Education (UNED), Madrid, Spain, in 2013, and the University of Antwerp, Antwerp, Belgium, in 2015. He is currently a Research Professor with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA. He is one of the original 17 fellows of the Association for Computational Linguistics (ACL). He has published more than 500 research articles. His researches focus on various topics, including aspects of the computational semantics of human language.

Dr. Hovy is a fellow of the Association for the Advancement of Artificial Intelligence (AAAI). He serves or has served on the editorial boards of several journals, such as the *ACM Transactions on Asian Language Information Processing (TALIP)* and *Language Resources and Evaluation (LRE)*.