IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 January 2024, accepted 6 February 2024, date of publication 8 February 2024, date of current version 15 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3364353

RESEARCH ARTICLE

Syntactic and Semantic Aware Graph Convolutional Network for Aspect-Based Sentiment Analysis

JUNJIE CHEN, HAO FAN, AND WENCONG WANG
College of Computer Science and Information Engineering, Inner Mongolia Agricultural University, Hohhot 010018, China
Inner Mongolia Autonomous Region Key Laboratory of Big Data Research and Application for Agriculture and Animal Husbandry, Hohhot 010018, China

Corresponding author: Junjie Chen (chenjj@imau.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62066037, in part by the Inner Mongolia Natural Science Foundation of China under Grant 2020MS06013, and in part by the Inner Mongolia Agricultural University High Level Talent Introduction and Research Initiation under Project NDYB2020-32.

ABSTRACT In recent years, there has been a growing interest in utilizing dependency parsing with graph convolutional networks for aspect-based sentiment analysis. Dependency relations between words are used to construct graphs that integrate syntactic information into deep learning frameworks. However, most existing methods fail to consider the impact of different relation types between content words, which makes it difficult to distinguish important related words. Moreover, the semantic relationship between words can enhance the text understanding ability, which has been largely neglected in previous works. To address these limitations, in this paper, we propose a novel model named as SS-GCN. Our model automatically learns syntactic weighted matrix and leverages semantic information to obtain the text semantic representation, and an attention module is introduced to obtain the specific aspect-context hidden vectors. The model enhances the text representation ability from syntactic and semantic graph convolutional networks. We conducted comprehensive experiments on publicly available datasets to demonstrate its validity and effectiveness. The experimental results demonstrate that our model outperforms strong baseline models.

INDEX TERMS Aspect-based sentiment analysis, automated syntactic dependency weighting, graph convolutional network, semantic relation graph.

I. INTRODUCTION

Sentiment analysis refers to extract and understand public opinion from the sentence or document level [1]. However, the user-generated content usually contains many topics or aspects, toward different target has different sentiment. Aspect-based sentiment analysis (ABSA) is a fine-grained task within sentiment analysis that aims to distinguish sentiment polarities towards specific entities in a given text [2]. For example, given the sentence "i liked the atmosphere very much but the food was not worth the price.". There are two aspects "atmosphere" and "food", the user holds positive polarity to "atmosphere", whereas negative towards "food".

In recent years, ABSA has garnered significant attention from both industry and research communities [3], [4]. Deep learning methods have emerged as popular models for ABSA task due to their ability to automatically extract relevant features. Most early works exploited recurrent neural network (RNN) and its variants, such as long short-term memory (LSTM) and bidirectional long short-term memory (Bi-LSTM), to obtain the aspect category from the context and aspect hidden representation [5], [6], [7], [8]. Zhang et al. [2] pointed out that it is important for modelling connection

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna D'Ulizia.

between context words and aspects. Attention mechanism [9] is introduced into the models, usually assuming that words closer to the target words are more likely to be related to its polarity [10]. However, these methods may attend the irrelevant sequence words to the specific aspect [11]. To address this issue, graph-based models [12], [13], [14] were employed to utilize the relationship between words through syntactic dependency trees. These models construct a graph for the entire text and leverage graph convolutional networks (GCN) [15] to incorporate non-sequential information. Moreover, some studies [16], [17] focused on aspect, constructing text graph according to relation between context words and aspect words. Graph based methods enrich the representation vectors of context and aspect terms, leading to improved the performance of ABSA. Expression ability of the GCN is closely related to the graph relation weight score [18]. Liang et al. [19] introduced external affective commonsense knowledge between words for relation weight score. However, most of these methods regard the dependency relations as same weight score, without considering their dependency types. This limitation restricts the expressive capabilities of the dependency relations. For example, considering the sentence "great food but the service was dreadful !", the dependency graph is shown as Fig.1.

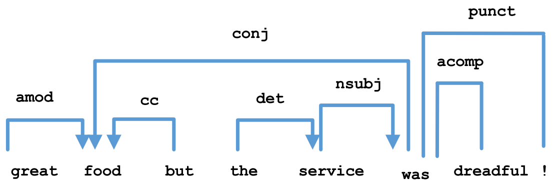


FIGURE 1. **Example of sentence dependency relationship.**

From the Fig.1, we can see "great" and "food" have a dependency relation of "amod", while "food" and "was" have a relation of "conj". Obviously, the "amod" type should be more important than "conj" and "det". Therefore, the weight score of the edge between "great" and "food" should be higher than the score between "food" and "was" in the text graph. However, most studies assign the same weight to all dependency relations, neglecting the distinction between different relation types. Hence, it is necessary to develop an appropriate approach that can differentiate between different relation types and compute weight scores accordingly.

Furthermore, graph-based models often overlook the semantic relationships between words. To illustrate, consider the following text:

"i had a terrific meal, and our server guided us toward a wine in our price range."

In this example, the aspect "wine" is perceived as positive, which can be inferred from the phrase "terrific meal". There exists a semantic relationship between the words "meal" and "wine". The word "terrific" influences the word "meal", which in turn impacts the aspect "wine". Consequently, it is crucial to establish semantic relationships between words within the text. This enables a comprehensive understanding of the overall meaning conveyed by the text.

In this paper, we propose an approach to effectively utilize syntax information by employing type embeddings to represent different dependency relation types. We introduce a syntactic weighted module to automatically learn the weight scores for these different types. Additionally, we introduce a text semantic graph to enhance the representation ability between words. Graph convolutional network is employed to get the text and aspect representation from semantic and syntactic graphs.

The main contributions of our work can be summarized as follows:

- We explore a type embedding layer to obtain representations of dependency relation types, and introduce a syntactic module to automatically learn the weight scores for these different types. This module overcomes the limitation in graph models where all types of relationships have the same weight, thereby enhancing the expressive power of syntactic information in the text.
- We introduce a text semantic graph to capture the semantic relationships between words. This module incorporates semantic information to the text, enabling a more comprehensive understanding of the text and thereby enhancing the text representation ability.
- We propose a novel model, called SS-GCN, which automatically learns the text syntactic and semantic features from graph convolutional networks. Furthermore, we incorporate an attention module to obtain specific context-aspect representations for the ABSA task. We evaluate our method on four publicly available datasets. The experimental results indicate that our proposed model achieves higher accuracy than most of the baseline models and outperforms the strong baseline models.

II. RELATED WORK

Aspect-based sentiment analysis is an important fine-grained sentiment analysis task that has received increasing attention in recent years. Traditional methods [20], [21] for this task primarily focus on extracting a set of handcrafted features, such as sentiment lexicons, to train a sentiment statistic-based classifier. However, these methods heavily rely on handcrafted features, which are labor-intensive and costly. Deep learning methods have shown strong text representation capabilities through multiple hidden layers. They are more scalable than manual-based features. Many works based on deep learning approaches are developed for ABSA.

Sequential models, such as recurrent neural network, long short-term memory and gated recurrent neural networks (GRU), have demonstrated strong represent ability in natural language processing (NLP). TD-LSTM [5] splits a sentence into two parts from the target, and utilizes two LSTMs to obtain the left and right representation of the target. Further more, Zhang et al. [22] proposed bi-direction GRU perform even better than LSTM, compared to TD-LSTM.

Not all context words are important for the aspect words. Sequential models combined with attention mechanisms have been employed to address the relationship between context and aspect words [23], [24], [25], [26]. Feng et al. [27] introduced an attention mask mechanism to ignore irrelevant words in the context. The integration of complex attention mechanisms with sequential models has greatly improved ABSA performance and become a prevalent framework. In addition, some previous works [28], [29], [30], [31] utilized memory networks or Capsule Networks to extract aspect-relevant information from the context. For example, TransCap [31] developed semantic capsules for sentence representation, employing aspect routing to compute the weight between aspects and the context words. CPA-SA [32] introduced two asymmetric positional weighting functions to model aspect-specific context position information.

Despite the effectiveness of these methods, they implicitly get the related information from attention score, making it challenging to capture long-range dependencies between words.

Graph convolutional network has the ability to process data on the generalized graph, which can capture the dependencies of the graph structure [33]. It performs excellent for processing rich relational data by updating nodes feature from adjacent nodes. Therefore, it can capture the long-range sentiment relation through syntactic or semantic information between words. Graph convolutional networks combined with dependency trees have been explored to predict aspect sentiment polarity [12], [13], [14]. For instance, Chen et al. [13] utilized graph convolutional networks to enhance text representation capabilities by considering syntactic graphs.

However, these methods consider all relations between words, which do not focus on the connection between aspect and opinion words. R-GAT [16] and dotGCN [34] reconstructed the dependency parser tree specifically for aspect words, generating a graph structure with the aspect words as root nodes. AGGCN [34] used gated LSTM to encode sentences, which selected specific aspect representation information from the context. Nevertheless, all these methods ignore the dependency relation type between words, and they regard them as the same weight in the adjacent matrix. Phan et al. [35] pointed out that the performance of graph-based methods strongly depends on the graph structure and edge weights. DREGCN [36] and T-GCN [37] exploited dependency relation type by embedding them as type vectors and updated node features by adjacent node features and type vectors. These works are most similar to our approach. However, in these works, the edge weights are dependent on the relation type and word representation. Obviously, it is inconsistent with our intuition, as the same type of relation should have the same weight. Researchers are increasingly recognizing the importance of word relations and incorporating additional knowledge in analyzing these relationships. PD-RGAT [38] constructed graph structure by incorporating rich information from both dependency trees and phrase trees. Liang et al. [11] used constituent tree to build adjacent matrix for each phrase layer. Hete_GNNs [39] and Sentic GCN [19] introduced sentiment dictionary into words relationship. In addition, BERT-based context representation models [40], [41], [42] can improve the performance whatever it is a graph-based model or none graph-based model [43].

In addition, another trend is to explicitly utilize knowledge, such as syntax, semantics, and affective dictionaries, to construct text graphs. Graph convolutional networks are applied to perform operations on these graph structures. Constructing text graph based on dependency relationships is a prevalent approach in graph-based models. However, all of these works did not take into account the semantic information between words. The semantic relationship between words is very important for text comprehension, as it can enhance the representation ability of words, and subsequently improve the performance of the ABSA task.

III. OUR PROPOSED MODEL

In this section, we explain the proposed model in detail. Given the sentence $s = w_1, w_2, \ldots, w_t, w_{t+1}, \ldots, w_{t+1-k}, \ldots, w_n$, the task is defined as predicting the sentiment polarity category corresponding to the specific aspect term $a = w_t, w_{t+1}, \ldots, w_{t+1-k}$. Where w_i is the ith word in sentence s, and a is the subsequence of s.

For graph-based models, constructing text graphs is an important step. First, we introduce the process of constructing the syntactic and semantic text graph. Subsequently, we present the architecture of our proposed model, as depicted in Fig. 2 (All matrix scores in Fig. 2 are provided as examples.).

Our model can be mainly divided into three parts, that is:

- Syntactic representation: Learn the syntactic weighting matrix from the text syntactic graph and obtain the text syntactic representation through the syntactic GCN layer.
- Semantic representation: leverage a semantic weighted matrix to get text semantic representation from the text semantic GCN layers.
- Attention module: the syntactic representation and semantic representation are combined to obtain the text representation. Afterwards, an aspect mask is applied to the text representation to extract the aspect representation. The attention module then computes the attention weight for the specific aspect and outputs the final representation for sentiment classification.

A. GRAPH CONSTRUCTION

For graph based methods, establishing an appropriate graph is a critical step as it determines the structure of the graph. The differences in graph structures can have an impact on the relationships between words, which are closely related to the text representation. In our model, the syntactic graph and semantic graph are constructed separately. We enhance

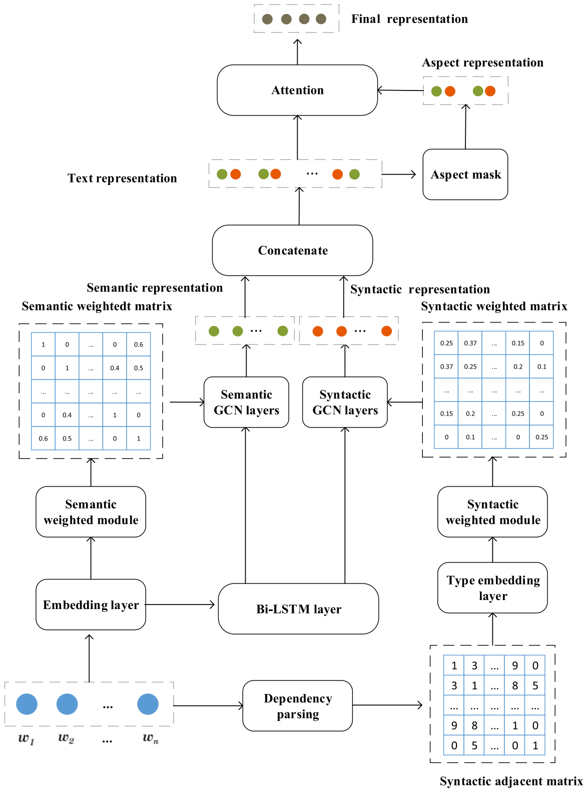


FIGURE 2. Overall architecture of the proposed SS-GCN model.

the text representation ability by leveraging syntactic and semantic information.

1) SYNTACTIC GRAPH

The syntactic undirected graph is constructed by dependency relation between words. Each word in sentence corresponds to the node in graph, and the edges between words are dependency type. We add self loop for each node, and the type is defined as ‘‘self’’. The dependency type index is built, starting from 1. The syntactic adjacency matrix A^{syn}:

$$A_{ij}^{syn} = \begin{cases} T, & \text{if } w_i \text{ and } w_j \text{ have dependency relation} \\ 0, & \text{others} \end{cases} \tag{1}$$

where T is dependency type index.

2) SEMANTIC GRAPH

Word embedding models such as Glove [44] and Word2vec [45] represent words in the form of vector, and previous studies have demonstrated that they can be used for measuring semantic similarity between words [46]. Semantic similarity plays an important role in the field of linguistics. We build edges for any two nodes in the text, where the edge weight is semantic similarity. The semantic similarity is defined as:

$$distance_{ij} = \left|cosine(E_i, E_j)\right| \tag{2}$$

E_i and E_j are embeddings corresponding to w_i and w_j, respectively. The distance between w_i and w_j is the absolute value of cosine similarity between their embedding vectors. The semantic weighted matrix A^{sem} is:

$$A_{ij}^{sem} = \begin{cases} distance_{ij}, & \text{if } distance_{ij} > \varepsilon \\ 0, & \text{if } distance_{ij} \leq \varepsilon \end{cases} \tag{3}$$

ε is the threshold of the distance.

B. SS-GCN MODEL

In the embedding layer, each word in the sentence is embedded into a dw-dimensional vector by looking up table $E \in \mathbb{R}^{dw \times N}$, where N denotes the size of the vocabulary. The semantic weighting matrix A^{sem} is obtained by embedding words in the semantic weighting module as shown in (3). As shown in the weighted score examples in Fig.2, the syntactic adjacency matrix is symmetric. After passing through the type embedding layer, the syntactic type embedding $H_{type} \in \mathbb{R}^{dt \times n \times n}$ is generated, where dt is the dimension of type embedding. The syntactic weighted matrix is produced by the syntactic weighted module, and the formula is:

$$A^{syn} = \sigma(W_s H_{type} + b_s) \tag{4}$$

where $W_s \in \mathbb{R}^{dt \times 1}$ and b_s are the trainable parameters, and σ is the sigmoid activation function. The $A^{syn} \in \mathbb{R}^{n \times n}$ is a self-learning weighted matrix, which indicates that different types of dependencies correspond to different the dependency weight.

With the word embeddings of the sentence X, a bidirectional LSTM is used to generate the context hidden state vectors H_c. Then semantic weighted matrix and syntactic weighted matrix are fed into semantic GCN layers and syntactic GCN layers, respectively. In the GCN layers, the node representation is updated by aggregating neighbor nodes information from adjacent matrix, where the weight denotes the importance of the adjacency. The syntactic GCN layer integrates syntactic related words into the word representation, and the hidden state of the word in GCN layers is formulated as:

$$H_{syn}^{(l+1)} = ReLU(L^{syn} H_{syn}^{(l)} W_{syn}^{(l)}) \tag{5}$$

where $W_{syn}^{(l)}$ is the learned parameter in the $l-th$ layer of GCN, and L^{syn} is the Laplacian matrix that is defined as:

$$L^{syn} = \tilde{D}^{-\frac{1}{2}} \tilde{A}^{syn} \tilde{D}^{\frac{1}{2}} \tag{6}$$

where $\tilde{A}^{syn}$ is denoted as $\tilde{A}^{syn} = A^{syn} + I$, and I is the identity matrix. $\tilde{D}$ is a diagonal matrix, and the $\tilde{D}_{ii} = \sum_j A_{ij}^{syn}$. The initial hidden state $H_{syn}^{(0)}$ is set to H_c. The l layers GCN is considered to capture information from $l-$hops neighbors. Similar to the syntactic GCN layers, the formula for the semantic GCN layer is given by:

$$H_{sem}^{(l+1)} = ReLU(L^{sem} H_{sem}^{(l)} W_{sem}^{(l)}) \tag{7}$$

where L^{sem} is the Laplacian matrix of the A^{sem}. The last layer of syntactic GCN and semantic GCN are concatenated to obtain the text representation H_{text}. The aspect representation H_{aspect} is produced by aspect mask module from H_{text}, where the word vectors of non-aspect words are set to 0. This can be expressed as follows:

$$H^i_{aspect} = \begin{cases} H^i_{text}, & \text{if } t \leq i \leq t+k-1 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The attention module derives significate features form the contextual representation, and the final representation H_{final} and attention weights are calculated as follows:

$$H_{final} = \sum_{m=1}^{n} \alpha_m H^m_c \tag{9}$$

$$\alpha_m = \frac{exp(\beta_m)}{\sum_{i=1}^{n} exp(\beta_i)} \tag{10}$$

$$\beta_m = (H^m_{text})^T H_{aspect} = \sum_{i=t}^{t+k-1} (H^m_{text})^T H^i_{aspect} \tag{11}$$

Here α_m is attention weight of the m-th word with respect to the aspect. The final text vectors H_{final} are forwarded to softmax layer for aspect sentiment classification.

$$y = softmax(W_f H_{final} + b_f) \tag{12}$$

where W_f and b_f are trainable parameters, and $softmax(\cdot)$ represents the softmax function, which is employed to learn the output distribution for the sentiment classifier.

C. MODEL TRAINING

The model is trained to optimize all the parameters to minimize the objective function. In our model, the cross-entropy with $L2$-regularization term is used as the loss function, and it is formulated as:

$$l(\theta) = -\sum_{i}^{N} \sum_{j}^{C} y^j_i log(\hat{y}^j_i) + \lambda \, ||\theta||^2 \tag{13}$$

where N is the number of samples, and C is the number of classes. The $\hat{y}_i$ is the estimated probability, and λ is the $L2$-regularization factors.

IV. EXPERIMENTS

A. DATASETS AND EXPERIMENTAL SETTINGS

To evaluate our model, extensive experiments are performed on four publicly available datasets, abbreviated as Rest14, Lap14, Rest15, and Rest16 respectively. Rest14 and Lap14 respectively come from the restaurant and laptop domains of SemEval-2014 Task 4 [47]. Rest15 and Rest16 are restaurant reviews collected from SemEval-2015 Task 12 [48] and SemEval-2016 Task 5 [49], respectively. To ensure a fair comparison with the baseline models, we utilized the same data partitioning scheme as employed by the baseline models [12], [13], [19], [30], [39]. The statistics of datasets are shown in Table 1.

TABLE 1. Statistics of the experimental datasets.

Dataset	Train			Test		
	Positive	Neutral	Negative	Positive	Neutral	Negative
Lap14	994	464	870	341	169	128
Rest14	2164	637	807	728	196	196
Rest15	912	36	256	326	34	182
Rest16	1240	69	439	469	30	117

In this paper, we construct syntactic graph using dependency relations and types obtained from Spacy parse tree.[1] 300-dimensional pretrained word embeddings of GloVe are used to represent the word vectors, and semantic graph is constructed by computing the semantic distance through word embeddings with the threshold of 0.1. The embedding size of the dependency type is set to 300 and randomly initialized. Following pilot studies, the depth of GCN layers is set to 2. The dimensionality of hidden state vectors in the model is set to 250 and initialized randomly with uniform distribution. The coefficient of L2 regularization is set to 0.00001. The model is trained using Adam optimizer with 0.001 learning rate. Accuracy and Macro-Averaged F1 are utilized as the evaluation metrics.

B. BASELINE MODELS

To evaluate the effectiveness of the proposed model, we compare its performance with the following methods on four public datasets:

- TD-LSTM [5]: utilizes two LSTM networks to obtain the left and right context-aspect hidden vectors separately. These hidden vectors are then combined to generate the specific aspect representation.
- IAN [24]: employs interactive attention networks dynamically learns the relationship between the context and aspect.
- Transcap [31]: transfers document-level knowledge to aspect-level sentiment classification by a transfer capsule network framework.
- IACapsNet [30]: learns vector-based feature representation through a capsule network and obtains clustered features from an EM routing algorithm.
- GCNSA [13]: adopts graph convolutional networks over syntactic dependency graph for text representation and an extended structural attention model for aspect representation.
- ASGCN [12]: leverages syntactical information and word dependency relations to construct text graph, and aspect-specific features are generated using retrieval-based attention mechanism.
- CPA-SA [32]: introduces two asymmetric positional weighting functions to adjust the weights of contextual words at different positions. It also utilizes a multiple-sentence-level Bi-GRU to capture contextual features.

[1] Spacy tools:https://spacy.io/

TABLE 2. Experimental results on four datasets.

Model	Rest14		Lap14		Rest15		Rest16	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
TD-LSTM	75.17 ‡	65.25 ‡	70.53 ‡	64.27 ‡	74.76 ‡	53.22 ‡	81.06 ‡	53.79 ‡
IAN	77.25‡	69.87‡	71.88 ‡	66.23 ‡	75.54 ‡	51.25 ‡	83.69‡	54.38 ‡
TransCap	79.29	70.85	73.87	70.10	-	-	-	-
IACapsNet	81.79	73.40	**76.80**	**73.29**	-	-	-	-
GCNSA	80.41 ♮	73.23♮	73.35 ♮	68.4♮	77.68♮	62.26♮	87.34♮	68.22♮
ASGCN	80.77	72.02	75.55	71.05	79.89	61.89	88.99	67.48
CPA-SA	82.64	73.38	75.18	71.50	79.61	60.15	89.02	72.43
Sentic GCN	81.87 ♮	73.25 ♮	75.71♮	71.18 ♮	80.44 ♮	63.09♮	89.61♮	69.88♮
Hete_GNNs	81.91	73.74	74.08	69.45	80.37	63.21	87.92	70.18
SS-GCN(Ours)	**82.96**	**74.26**	75.86	71.78	**81.18**	**66.16**	**89.77**	**74.28**

- Hete_GNNs [39]: incorporates syntax dependency relations, speech part-of-speech (POS) relations, and sentiment relations into the adjacency matrix. Graph attention networks are employed to capture heterogeneous graph relations information.
- Sentic GCN [19]: constructs a graph over the dependency tree, incorporating affective commonsense knowledge. Graph convolutional networks are then used to capture affective dependencies to specific aspects.

C. RESULTS

We conducted a comparison of our proposed method with baselines on the four datasets, and the results are reported in Table 2. The performance of the model is influenced by the choice of optimizer and batch size. To accurately assess the performance of each model, we re-implemented the models and trained them using the same settings as the others. The results marked with '♮' indicate that they have been re-implemented, whereas those denoted with '‡' are obtained from open source codes. The remaining results are retrieved from the original papers. The best results are highlighted in bold, the second best results are underlined.

As shown in Table 2, our proposed method SS-GCN achieves the highest performance on the Rest14, Rest15 and Rest16 datasets. Among the compared methods, TD-LSTM performs the worst as it disregards the representation of context. The IAN is slightly better than TD-LSTM for introducing attention mechanism into sequential model. The performance of the capsule network-based methods TransCap and IACapsNet is superior to the sequential models TD-LSTM and IAN. In particular, IACapsNet utilizes EM routing for specific aspect representations, achieving the highest accuracy and F1 score on Lap14. CPA-SA achieves the second highest accuracy on the Rest14 dataset, with F1 scores only second to our proposed method on the Rest16 dataset. CPA-SA differs from ASGCN by assigning asymmetric weights to contextual words. However, the performance improvement is limited because aspect-related words are not only influenced by their position in the sentence but also by syntax and semantics.

Graph based models GCNSA, ASGCN, Sentic-GCN, Hete_GNNs and our model SS-GCN generally outperform sequential models because these graph-based models are able to incorporate more information, such as syntactic, affective features into models. As a result, they can better capture the relationship between context and specific aspect. As show on Rest15 and Rest16 in Table 2, we can observe that the F1 scores of our method are 66.16% and 74.28%, respectively. Compared with the second result, it increased by 2.95% and 4.1%, respectively.

Our method, SS-GCN, incorporates a syntactic weighted module that automatically learns the weights of syntactic dependency relations, enabling it to assign weight information based on the importance of dependency types. In comparison to SAGCN, ASGCN, and Sentic GCN, SS-GCN overcomes the drawback of assigning equal weights to all dependency types. The semantic weighted module introduced in our approach obtains semantic relation weights between words, which aggregates information from words with similar semantics. This integration of semantic information into word features enhances the representation ability of the text. However, existing methods have overlooked this aspect of information. The incorporation of these two modules fully leverages the semantic and syntactic information and further improves performance.

D. ABLATION STUDY

To evaluate the impact of each component of the proposed SS-GCN approach on the overall performance, we compare different variants of SS-GCN on four datasets as presented below:

- Semantic-GCN: the syntactic graph convolution network is ablated, and other parts are similar to SS-GCN.
- Syntactic-Embedding-GCN:The semantic graph convolution network is excluded and syntactic dependency relation is embedded as SS-GCN.
- Syntactic-GCN:It is similar to the proposed variant Syntactic-embedded-GCN, without inclusion of the semantic graph convolution component. The weight of all the dependency relation type is set to a fixed value 1, rather than being learned from the embedding.
- C-SS-GCN: The attention module is removed. The context representation vector and the aspect hidden vector are passed through mean pooling to obtain

TABLE 3. Ablation study.

Model	Rest14		Lap14		Rest15		Rest16	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Semantic-GCN	79.46	70.43	74.14	70.10	79.52	61.81	87.18	70.19
Syntactic-Embedding-GCN	81.79	74.04	74.76	70.71	80.81	65.03	88.8	70.51
Syntactic-GCN	81.16	73.15	74.29	70.07	79.34	62.07	87.99	68.55
C-SS-GCN	82.32	74.10	74.92	70.91	80.07	64.33	89.02	72.77
SS-GCN	**82.96**	**74.26**	**75.86**	**71.78**	**81.18**	**66.16**	**89.77**	**74.28**

the context hidden vector and aspect hidden vector, which are then concatenated to form the final representation.

The results of the ablation study are demonstrated in Table 3. We can observe that Syntactic-Embedding-GCN outperforms Syntactic-GCN on all datasets, which demonstrates the effectiveness of our proposed syntactic weighted module for learning dependency weights. Furthermore, Syntactic-Embedding-GCN is superior to Semantic-GCN, indicating that the syntactic graph convolutional network in our module can capture more important features than the semantic graph convolutional network. However, Syntactic-GCN does not outperform Semantic-GCN on all datasets, which further demonstrates the close relationship between dependency relation weight and the performance of graph-based models. The C-SS-GCN model generally performs better than Syntactic-Embedding-GCN and Semantic-GCN due to its incorporation of both syntactic and semantic information. The results also indicate that the semantic graph relation is a necessary complement to the syntactic graph. However, when considered separately, the syntactic relation is more important than the semantic relation. C-SS-GCN is inferior to SS-GCN, suggesting that attention-based representation of the context-aspect relationship is more effective than using concatenated approaches.

E. IMPACT OF THE SEMANTIC GRAPH THRESHOLD

To investigate the impact of threshold value of semantic graph to the performance, we conduct experiments on the Lap14, Rest14, Rest15, and Rest16 datasets. The threshold values are varied from 0 to 0.9 with an interval of 0.1, resulting in a total of 10 values. The accuracy results are presented in Fig.3, while the F1 scores are displayed in Fig.4.

The variation of the threshold value has a significant impact on the accuracy. When the threshold value is 0, the accuracy is not optimal across all datasets, which indicates that including all semantic relations introduces a lot of noisy information. When the threshold is set to 0.5, all datasets demonstrate the lowest accuracy. As the threshold value increases beyond 0.5, the accuracy tends to decrease initially and then stabilize.

The F1 score is also affected by the variation of threshold values. Generally, as the threshold value increases to greater than 0.4, the F1 score significantly decrease, similar to the changes observed in the accuracy. These results indicate that setting a threshold too high can filter out many useful semantic relationships.

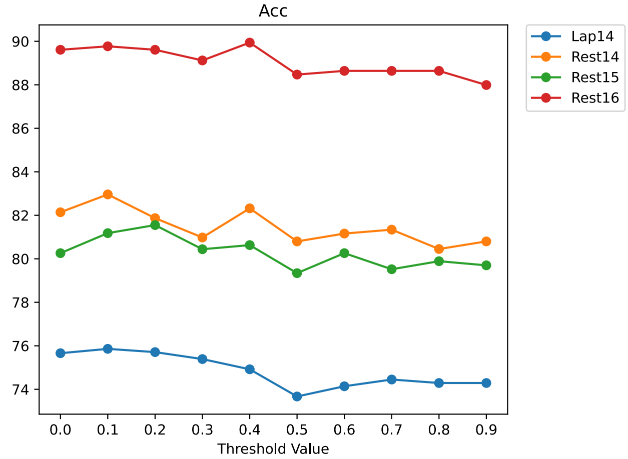


FIGURE 3. The impact of threshold values on accuracy.

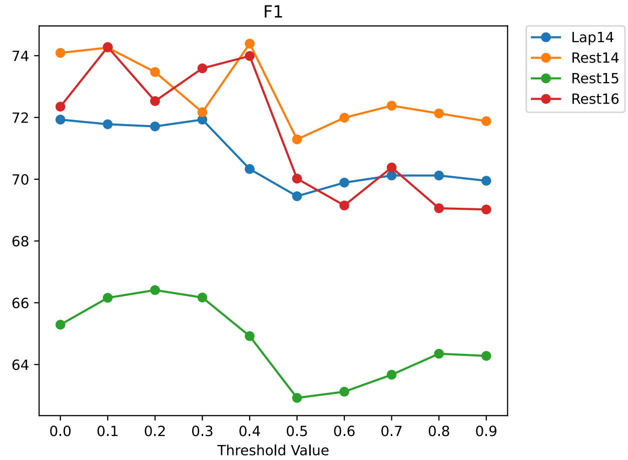


FIGURE 4. The impact of threshold values on the F1 score.

F. IMPACT OF DEPENDENCY EMBEDDING SIZE

In this section, we investigate the impact of dependency embedding size by changing the dimension from 50 to 400. Our experiment focused on the Rest 15 and Rest 16 datasets. The accuracy results of the experiment are shown in Fig.5.

The size of dependency embeddings has a notable impact on the performance of the model. Generally, increasing the size of dependency embeddings leads to higher accuracy.

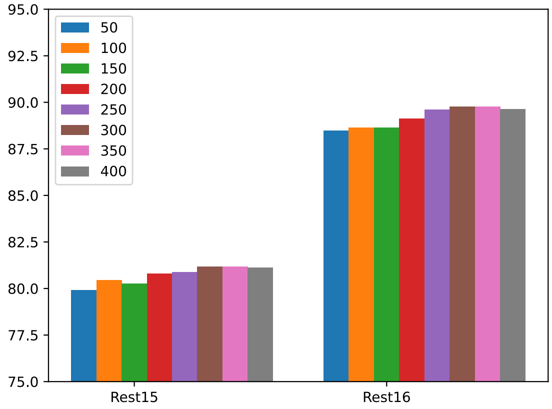


FIGURE 5. The impact of dimension of dependency embedding.

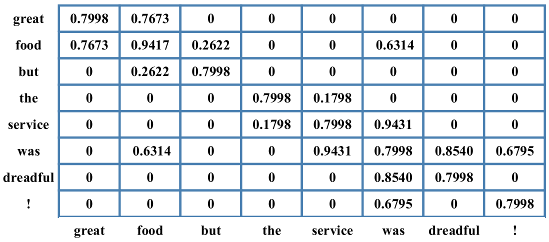
In particular, when the size is set to 300, the model achieves the highest accuracy on Rest15 and Rest16 datasets. When the size exceeds 300, the performance changes are not significant. This observation supports the notion that when the dependency embedding size is small, the model may face challenges in capturing fine-grained syntactic relationships between words. On the contrary, increasing the dependency embedding size enables a more expressive representation of the dependency relationships.

G. CASE STUDY

To perform a comprehensive analysis of the importance of semantic and syntactic graph relationships, we conducted a case study in which we reported the results of several representative testing examples.

For the sentence "great food but the service was dreadful!", the syntactic dependency relations are depicted in Fig.1. We proposed to automatically learn the weights of different dependency relation types. The results of the learned dependency weights are shown in Fig.6.

From Fig.1, we can observe that "great" and "food" have the relation type "amod", whereas "food" and "but" have the relation type "cc". After learning, we found that in our model, the weight of "amod" is 0.7673 and the weight of "cc" is 0.2622, as shown in Fig.6. This implies that, according to our learned weights, the "amod" relation is considered more important than the "cc" relation. This weighting approach is more suitable than simply setting all relation type weights to 1, as done in other methods.



great	0.7998	0.7673	0	0	0	0	0	0
food	0.7673	0.9417	0.2622	0	0	0.6314	0	0
but	0	0.2622	0.7998	0	0	0	0	0
the	0	0	0	0.7998	0.1798	0	0	0
service	0	0	0	0.1798	0.7998	0.9431	0	0
was	0	0.6314	0	0	0.9431	0.7998	0.8540	0.6795
dreadful	0	0	0	0	0	0.8540	0.7998	0
!	0	0	0	0	0	0.6795	0	0.7998
	great	food	but	the	service	was	dreadful	!

FIGURE 6. Weights of the dependency relations for the example sentence.

The sentence "The food was very good and I was pleasantly surprised to see so many vegan options." contains "food" and "vegan" two aspects. It is relatively easy to judge the sentiment for the aspect "food" as it is closely related to the adjective "good". However, the aspect "vegan" is distantly related to the adjective words. Without semantic information, it becomes difficult to determine the correct sentiment for this aspect. By leveraging the semantic relation, we can infer the sentiment by understanding that vegan is a type of food and food is described as good, thus allowing us to further judge that "vegan" is also good. In our proposed method, we construct a semantic graph for the sentence, and the weights of the semantic graph relations are visualized in Fig.7.

From Fig.7, we can observe that the word "vegan" has the highest relation weights with "food" compared to other words. In our experiments, this weight helped us correctly determine the sentiment polarity for the aspect "vegan". Although, it is challenging to determine the sentiment of "vegan" solely based on syntactic and positional information, our approach relies on semantic relations to accurately capture the sentiment. This highlights the significance of the semantic graph in the aspect-based sentiment analysis task.

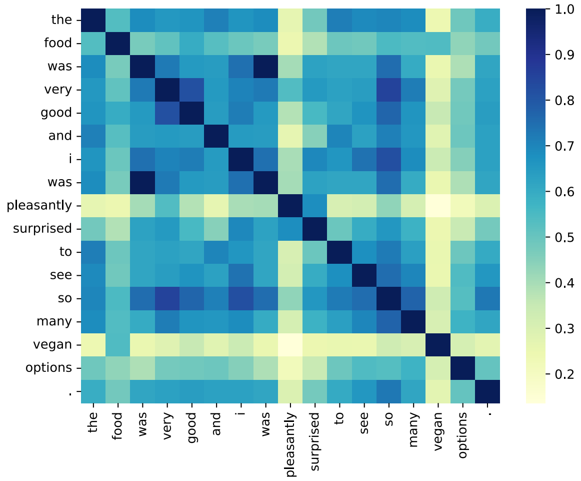


FIGURE 7. Visualization of semantic relation weight for the example sentence.

V. CONCLUSION

For aspect-level sentiment analysis tasks, it is crucial to have a comprehensive understanding of the overall meaning of the entire text and the relationships between aspects and the entire text. The comprehension of text heavily relies on the syntactic dependency relationships and semantic relationships between words. However, the importance of different syntactic and semantic relationships varies, and existing methods struggle to effectively capture the weights of these two types of relationships, which consequently impacts the performance of graph-based methods.

In this paper, we propose a graph-based model called SS-GCN, which incorporates both semantic and syntactic information. Our model consists of a syntactic weighted module that automatically learns the weights of syntactic dependency relations, and a semantic weighted module that captures the semantic relations between words. Experimental results demonstrate the importance of semantic relations in text representation. Experimental results and case study demonstrate that the learned weights can effectively represent the importance of different relationships. The incorporation of these two modules overcomes the limitation of assigning equal weights to all relationships, thereby enhancing the text and aspect representation capability. Specifically, in longer sentences, different words may be used to express the same concept, making semantic information even more crucial. Furthermore, our experiments reveal that the sentiment polarity of an aspect is related to the entire sentence, indicating the importance of comprehensively understanding the entire sentence. Simply considering syntactic and semantic information alone is not sufficient.

In the future, it is essential to consider additional information such as common sense knowledge and affective knowledge. An appropriate fusion method for all these types of information will be crucial for advancing this task. In addition, in aspect-level sentiment tasks, the text can be considered as a whole, while aspects are specific parts within the text. Understanding the relationship between the whole and its parts is crucial for aspect-level tasks. Irrelevant content related to aspects in the text may introduce noise and affect the accuracy of the task. Therefore, in the future, it is crucial to segment the relevant aspects from the unrelated portions.

REFERENCES

[1] B. Liu, *Sentiment Analysis and Opinion Mining*. Cham, Switzerland: Springer, 2022.

[2] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 11019–11038, Nov. 2022.

[3] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis: A comprehensive survey," *IEEE Trans. Affect. Comput.*, vol. 13, no. 2, pp. 845–863, Apr. 2022.

[4] M. M. Trușcă and F. Frasincar, "Survey on aspect detection for aspect-based sentiment analysis," *Artif. Intell. Rev.*, vol. 56, no. 5, pp. 3797–3846, May 2023.

[5] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 3298–3307.

[6] S. Ruder, P. Ghaffari, and J. G. Breslin, "A hierarchical model of reviews for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 999–1005.

[7] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 5876–5883.

[8] L. Bao, P. Lambert, and T. Badia, "Attention and lexicon regularized LSTM for aspect-based sentiment analysis," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Student Res. Workshop*, 2019, pp. 253–259.

[9] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[10] X. Wang, M. Tang, T. Yang, and Z. Wang, "A novel network with multiple attention mechanisms for aspect-level sentiment analysis," *Knowl.-Based Syst.*, vol. 227, Sep. 2021, Art. no. 107196.

[11] S. Liang, W. Wei, X.-L. Mao, F. Wang, and Z. He, "BiSyn-GAT+: Bi-syntax aware graph attention network for aspect-based sentiment analysis," 2022, *arXiv:2204.03117*.

[12] C. Zhang, Q. Li, and D. Song, "Aspect-based sentiment classification with aspect-specific graph convolutional networks," 2019, *arXiv:1909.03477*.

[13] J. Chen, H. Hou, Y. Ji, and J. Gao, "Graph convolutional networks with structural attention model for aspect based sentiment analysis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–7.

[14] J. Chen, H. Hou, J. Gao, Y. Ji, T. Bai, and Y. Jing, "GCNDA: Graph convolutional networks with dual attention mechanisms for aspect based sentiment analysis," in *Proc. Int. Conf. Neural Inf. Process.* Springer, 2019, pp. 189–197.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–14.

[16] 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*, 2020, pp. 3229–3238.

[17] 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*, 2020, pp. 150–161.

[18] L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei, and B. Long, "Graph neural networks for natural language processing: A survey," *Found. Trends Mach. Learn.*, vol. 16, no. 2, pp. 119–328, 2023.

[19] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowl.-Based Syst.*, vol. 235, Jan. 2022, Art. no. 107643.

[20] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter sentiment classification," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2011, pp. 151–160.

[21] T. Nakagawa, K. Inui, and S. Kurohashi, "Dependency tree-based sentiment classification using CRFs with hidden variables," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 786–794.

[22] M. Zhang, Y. Zhang, and D.-T. Vo, "Gated neural networks for targeted sentiment analysis," in *Proc. AAAI Conf. Artif. Intell.*, 2016, vol. 30, no. 1, pp. 3087–3093.

[23] 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.*, 2016, pp. 606–615.

[24] 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.*, Aug. 2017, pp. 4068–4074.

[25] 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.*, 2018, pp. 181–189.

[26] E. F. Ayetiran, "Attention-based aspect sentiment classification using enhanced learning through CNN-BiLSTM networks," *Knowl.-Based Syst.*, vol. 252, Sep. 2022, Art. no. 109409.

[27] A. Feng, X. Zhang, and X. Song, "Unrestricted attention may not be all you need–masked attention mechanism focuses better on relevant parts in aspect-based sentiment analysis," *IEEE Access*, vol. 10, pp. 8518–8528, 2022.

[28] 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.*, 2017, pp. 452–461.

[29] Q. Liu, H. Zhang, Y. Zeng, Z. Huang, and Z. Wu, "Content attention model for aspect based sentiment analysis," in *Proc. World Wide Web Conf.*, 2018, pp. 1023–1032.

[30] C. Du, H. Sun, J. Wang, Q. Qi, J. Liao, T. Xu, and M. Liu, "Capsule network with interactive attention for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 5489–5498.

[31] Z. Chen and T. Qian, "Transfer capsule network for aspect level sentiment classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 547–556.

[32] B. Huang, R. Guo, Y. Zhu, Z. Fang, G. Zeng, J. Liu, Y. Wang, H. Fujita, and Z. Shi, "Aspect-level sentiment analysis with aspect-specific context position information," *Knowl.-Based Syst.*, vol. 243, May 2022, Art. no. 108473.

[33] S. Abadal, A. Jain, R. Guirado, J. López-Alonso, and E. Alarcón, "Computing graph neural networks: A survey from algorithms to accelerators," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–38, Dec. 2022.

[34] C. Chen, Z. Teng, Z. Wang, and Y. Zhang, "Discrete opinion tree induction for aspect-based sentiment analysis," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 2051–2064.

[35] H. T. Phan, N. T. Nguyen, and D. Hwang, "Aspect-level sentiment analysis: A survey of graph convolutional network methods," *Inf. Fusion*, vol. 91, pp. 149–172, Mar. 2023.

[36] Y. Liang, F. Meng, J. Zhang, Y. Chen, J. Xu, and J. Zhou, "A dependency syntactic knowledge augmented interactive architecture for end-to-end aspect-based sentiment analysis," *Neurocomputing*, vol. 454, pp. 291–302, Sep. 2021.

[37] Y. Tian, G. Chen, and Y. Song, "Aspect-based sentiment analysis with type-aware graph convolutional networks and layer ensemble," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2021, pp. 2910–2922.

[38] H. Wu, Z. Zhang, S. Shi, Q. Wu, and H. Song, "Phrase dependency relational graph attention network for aspect-based sentiment analysis," *Knowl.-Based Syst.*, vol. 236, Jan. 2022, Art. no. 107736.

[39] G. Lu, J. Li, and J. Wei, "Aspect sentiment analysis with heterogeneous graph neural networks," *Inf. Process. Manage.*, vol. 59, no. 4, Jul. 2022, Art. no. 102953.

[40] H. T. Phan, N. T. Nguyen, and D. Hwang, "Aspect-level sentiment analysis using CNN over BERT-GCN," *IEEE Access*, vol. 10, pp. 110402–110409, 2022.

[41] H. Xu, B. Liu, L. Shu, and S. Y. Philip, "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, 2019, pp. 2324–2335.

[42] P. Yang, P. Zhang, B. Li, S. Ji, and M. Yi, "Aspect-based sentiment analysis using adversarial BERT with capsule networks," *Neural Process. Lett.*, vol. 55, no. 6, pp. 8041–8058, Dec. 2023.

[43] G. Brauwers and F. Frasincar, "A survey on aspect-based sentiment classification," *ACM Comput. Surv.*, vol. 55, no. 4, pp. 1–37, Apr. 2023.

[44] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

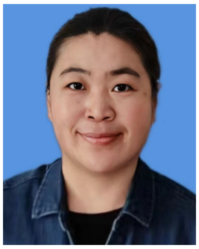
[45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–9.

[46] S. M. Mohammed, K. Jacksi, and S. R. M. Zeebaree, "A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 22, pp. 552–562, Apr. 2021.

[47] 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.*, 2014, pp. 27–35.

[48] 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.*, Denver, CO, USA, 2015, pp. 486–495.

[49] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, and V. Hoste, "SemEval-2016 Task 5: Aspect based sentiment analysis," in *Proc. Workshop Semantic Eval. (SemEval)*, 2016, pp. 19–30.



JUNJIE CHEN received the M.S. and Ph.D. degrees in computer application technology from Inner Mongolia University, in 2004 and 2020, respectively.

She is currently a Master's Tutor with the College of Computer and Information Engineering, Inner Mongolia Agricultural University. Her research interests include natural language processing, ontology, knowledge engineering, information retrieval, and computer vision.

Dr. Chen is a Reviewer of the International Conference on Neural Information Processing (ICOIP), from 2019 to 2023, a top international conference on the CCF Conference List, and the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), from 2020 to 2022. She is invited as a Keynote Speaker at the Third International Conference on Computer Information Science and Artificial Intelligence (CISAI 2020) International Conference.

HAO FAN was born in Hohhot, Inner Mongolia, China, in 1998. He received the bachelor's degree in computer science and technology from Inner Mongolia Agricultural University, in 2021, where he is currently pursuing the master's degree in computer science and technology with the College of Computer Science and Information Engineering.

His research interests include natural language processing, named entity recognition, and multimodal sentiment analysis.

WENCONG WANG was born in Liaocheng, Shandong, China, in 2000. She received the bachelor's degree in computer science and technology from Shandong Normal University, in 2022. She is currently pursuing the master's degree in computer science and technology with the College of Computer and Information Engineering, Inner Mongolia Agricultural University.

Her research interests include natural language processing, computer vision, and multimodal sentiment analysis.