Fine-Grained Sentiment Analysis Based on Segment Awareness and Discrete Dependency Tree

Junpeng Rao, Lingli Tong

School of Artificial Intelligence Hubei University, Wuhan 430415, China

Abstract

To address the issues that mainstream Graph Neural Network methods rely on complex global structures to integrate syntactic and semantic information, which easily introduces irrelevant noise, and highly depends on external syntactic parsers to generate dependency trees, resulting in extremely poor applicability in low-resource languages and non-canonical text scenarios, this paper proposes a syntactic and semantic enhanced graph attention network model based on segment-aware and discrete opinion tree information. The model adopts a dual-branch architecture. The first branch constructs a local semantic graph based on constituent tree boundaries to split complex sentences, filter irrelevant noise, and focus on the core semantics within phrases. The second branch autonomously derives implicit global dependency structures via task-oriented attention and reinforcement learning to mine long-distance sentiment trigger words, thus getting rid of the dependence on external syntactic parsers. Equipped with a hierarchical attention and adaptive aggregation module, the model achieves complementary feature fusion. Experimental results demonstrate the effectiveness of the proposed method.

Keywords

Fine-grained Sentiment Analysis; Constituent Tree; Dependency Tree; Attention Mechanism.

1. Introduction

With the rapid development of Web 2.0 and mobile Internet, user-generated content such as social media comments, product reviews and forum discussions has witnessed explosive growth. Unlike structured data, such unstructured texts are featured by large scale, high timeliness and strong subjectivity, carrying rich information on user attitudes, preferences and emotions. Automatically identifying sentiment polarity from massive unstructured texts has thus become a core research topic in natural language processing.

Sentiment lexicon-based methods identify sentiment polarity by matching sentiment words via high-quality lexicons and calculating sentiment intensity combined with grammatical rules. The core limitation of these methods lies in lexicon quality and coverage. Current lexicon construction mainly follows two paradigms: expert-based manual annotation and computational model-driven automatic generation. Manual annotation ensures fine-grained accuracy but incurs prohibitive labor costs, while self-supervised techniques, including corpus statistics-based and lexicon expansion-based methods, offer higher practicability.

Machine learning-based sentiment analysis methods classify sentiment polarity via supervised learning, which requires manually labeled datasets for model training. Typical algorithms include Conditional Random Fields (CRF) and Support Vector Machines (SVM).

Deep learning-based Aspect-Based Sentiment Analysis (ABSA) methods realize end-to-end text representation learning, drastically reducing the reliance on manual feature engineering, and have become the mainstream research direction[1].mergence of pre-trained language models (e.g., BERT) has greatly improved contextual semantic modeling; further integration with Graph Neural Networks (GNNs) such as GCN and GAT enables explicit integration of dependency syntax, semantic graphs or external knowledge, enhancing structured modeling for aspect-related contexts.

Nevertheless, existing deep learning approaches still suffer from two critical drawbacks: syntax-semantic fusion modules easily introduce irrelevant structural noise in complex sentences, and the models rely heavily on high-quality external syntactic parsers.

2. Basic Model Architecture

Vaswani et al.[2] proposed the Transformer model in 2017, a deep learning architecture centered on the self-attention mechanism. Initially designed for sequence-to-sequence tasks in natural language processing such as machine translation, it has been successfully applied to various tasks including text summarization, sentiment classification, and speech recognition. Its innovation mainly lies in the encoder-decoder structure built with self-attention.

The input of the BERT model consists of three types of embedding vectors. First, the input text is preprocessed and tokenized, and each token is mapped to a pre-trained word embedding to obtain Token Embeddings. Second, Segment Embeddings are assigned to each token to distinguish different sentences. Finally, Positional Embeddings are used to retain the positional information of each token in the sequence. The final representation of each token is obtained by summing these three embeddings. A special token is inserted at the beginning of the sequence, whose hidden state is used for classification tasks, and a token is added at the end of each sentence. The vectors processed by the Transformer encoder layer are illustrated in Figure 1.

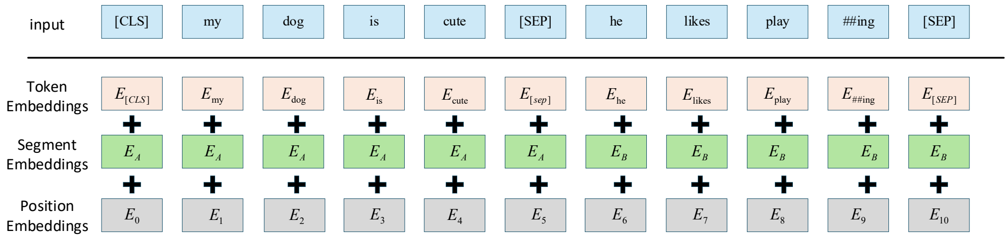

Fig.1 Input Representation of BERT

Graph Attention Network (GAT)[3] redefines the rules of information interaction between nodes by introducing an attention mechanism. The model abandons the static neighborhood weight allocation strategy of traditional Graph Convolutional Networks (GCN), and instead dynamically generates attention coefficients using feature similarity metrics (e.g., cosine similarity or dot-product operation). Let the set of neighbor nodes be N_i , and j be a neighbor node of nodes, where W denotes the shared parameter matrix. The formulation is as follows:

$$e_{ij} = attn([Wh_i \| Wh_j]), j \in N_i \tag{1}$$

The formula for calculating the attention score using the correlation coefficient e_{ij} is as follows:

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU(e_{ij})\right)}{\sum_{k \in N_i} \exp\left(LeakyReLU(e_{ik})\right)} \tag{2}$$

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU(e_{ij})\right)}{\sum_{k \in N_i} \exp\left(LeakyReLU(e_{ik})\right)} \tag{3}$$

3. Model Architecture Design

In this chapter, to address the core problems of existing sentiment analysis methods—easily introducing structural noise in multi-aspect complex sentences, misalignment between aspect terms and sentiment triggers, and strong reliance on external syntactic parsing—we propose a syntax- and semantics-enhanced graph attention network model based on segment awareness and discrete opinion trees(SeDOT-SSEGAT). The model architecture is illustrated in Figure 2.The model is designed following the philosophy of "divide and conquer, collaborative fusion". First, the pre-trained language model (BERT) is used to obtain the contextual representation of the input sentence. On the one hand, a local semantic graph is constructed under the boundary constraint of constituent trees, focusing on capturing the close correlations within phrases. On the other hand, an implicit global dependency structure is automatically derived via task-oriented attention scoring, aiming to mine long-distance sentiment trigger words. Finally, a hierarchical attention mechanism is introduced within both branches, and an adaptive aggregation module is adopted to achieve complementary fusion of the outputs from the two branches. The fused features are then fed into the sentiment classification layer to obtain the final sentence representation for prediction.

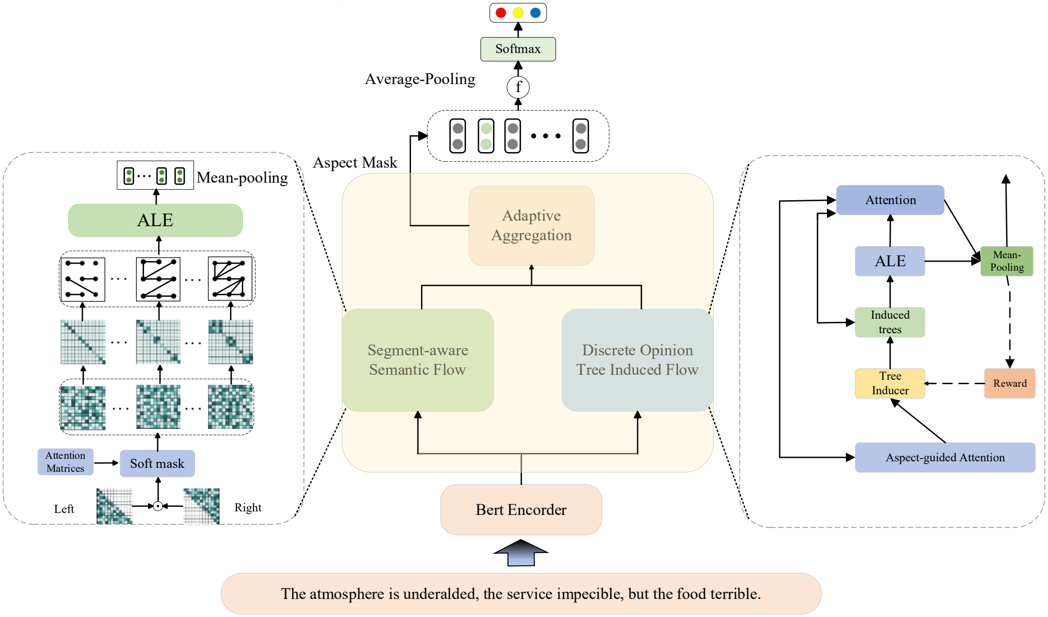

Fig.2 Overall Architecture of SeDOT-SSEGAT

3.1. Input Representation of BERT

Given a sentence $s=\left\{w_1,w_2,\ldots,w_{\gamma+1}\ldots,w_{\gamma+m}\ldots,w_n\right\}$ with $a=\left\{w_{\gamma+1},\ldots,w_{\gamma+m}\right\}$ words containing an aspect n, we employ the pre-trained language model BERT as the sentence encoder to extract contextual representations. Following the BERT-spc approach, we concatenate the sentence and the aspect term into a standard input sequence for the BERT encoder:

$$X=[\mathrm{CLS}]s[\mathrm{SEP}]a[\mathrm{SEP}] \tag{4}$$

The sequence X is fed into the BERT model, and the hidden states of the last layer are extracted as the initial contextual representations:

$$H=\mathrm{BERT}(X)=\{h_1^c,h_2^c,\ldots,h_n^c\}\in \square^{n\times \mathrm{d}} \tag{5}$$

Were d denotes the dimension of the hidden layer and c denotes the context. We take H as the input node features for the subsequent dual-branch graph network.

3.2. Segmented Semantic Perception Module

To address the issues of mutual interference among different aspects in multi-aspect complex sentences and the misalignment between aspect terms and sentiment terms, this section designs a Segment-Aware Semantic Stream based on the boundary constraints of constituent trees.

The core idea of this module is to divide complex sentences into several local consecutive segments according to the semantic segment boundaries provided by constituent trees. In this way, the model is forced to prioritize information aggregation within the segment where the target aspect is located, thereby reducing the interference caused by irrelevant clauses and information from other aspects.

Firstly, we obtain the hierarchical structure of the sentence via a pre-trained constituent parser, extract phrase boundary information, and convert it into two boundary probability matrices: the left boundary soft mask matrix $\bar{\phi}_l \in \mathrm{R}^{n\times n}$ and the right boundary soft mask matrix $\bar{\phi}_r \in \mathrm{R}^{n\times n}$, to generate token-level attention segments for each sentence. The calculation formulas are given as follows:

$$\bar{\phi}_l = \mathrm{Softmax}\left(\frac{QW_L^Q(KW_L^K)^T}{\sqrt{d}} \square \hat{M}\right) \tag{6}$$

$$\bar{\phi}_r = \mathrm{Softmax}\left(\frac{QW_R^Q(KW_R^K)^T}{\sqrt{d}} \square \hat{M}^T\right) \tag{7}$$

$$\hat{M}_{ij} = \begin{cases} 1, & i \geq j \\ -\infty, & i < j \end{cases} \tag{8}$$

Were □ denotes the element-wise product, and $W_L^Q, W_L^K, W_R^Q, W_R^K \in R^{d*d}$ is the trainable parameter. Notably, a mask matrix is introduced to ensure that the left boundary position l_p and the right boundary position r_p generated at position i satisfy the constraint $0 \leq l_p \leq i \leq r_p \leq N$.

Notably, the segment-aware mask matrix M_{seg} can be synthesized from the left and right boundary mask matrices:

$$M_{seg} = \left(\bar{\phi}_l \cdot L_N\right) \square \left(\bar{\phi}_r \cdot L_N^T\right) \tag{9}$$

Were $L_N \in \{0,1\}^{n\times n}$ is an all-one upper triangular matrix (i.e., $(L_N)_{ij} = 1$ if and only if $i \leq j$), and □ denotes the Hadamard product (element-wise product).

Based on the semantic segment mask matrix, we further construct a locally constrained graph attention layer, which strictly restricts the receptive field of graph convolution within semantic segments and computes the locally constrained attention matrix A^{Se}. For the l -th graph convolution layer, the computation process is as follows:

$$E^{Se} = \frac{(H^{(l-1)}W_Q)(H^{(l-1)}W_K)^T}{\sqrt{d_k}} \tag{10}$$

$$\mathrm{A}_{ij}^{Se} = \mathrm{Softmax}\left(E_{ij}^{Se} + \left(1 - M_{seg,ij}\right) \cdot (-\infty)\right) \tag{11}$$

Where $W_Q, W_K \in \mathrm{R}^{d_{model}\times d_k}$ denotes the learnable projection matrix. Here, M_{seg} acts as a hard attention mask, forcing the model to ignore noisy words outside the segments and aggregate information only within semantically continuous fragments.

3.3. Discrete Opinion Tree Induction

Syntactic dependency has been proven effective in capturing the interactions between aspects and opinion contexts. However, generating dependency trees with the aid of external parsers suffers from two core drawbacks: poor performance in low-resource scenarios and poor task adaptability. Furthermore, the dependency trees produced by external parsers are not optimized specifically for aspect-based sentiment classification.

To break the reliance on external dependency parsers and capture global long-range dependencies, this branch designs a discrete opinion tree induction flow. It automatically generates an implicit tree structure tailored to the current task via task-oriented structural scoring, which is able to capture long-range sentiment trigger information and enhance the robustness of structural modeling.

We first define a scoring function $F\left(w_i,a\right)$ to measure the importance of a context word w_i for judging the sentiment polarity of the corresponding aspect term a. Specifically, a multi-layer perceptron (MLP) is employed to compute the attention scores:

$$v_i = \mathrm{u}_p^T \tanh\left(W_p h_i + W_a h_a + \mathrm{b}_p\right) \tag{12}$$

where v_i denotes the structural score of the context word i , h_i and h_a represent the representations of the context word and the aspect term respectively, and, u_p W_p , W_a are learnable parameters. A higher scoring value indicates that the corresponding word plays a more critical role in sentiment expression.

Based on the set of word-level structural scores $V=\left\{v_1,\ldots,v_n\right\}$, we adopt an aspect-centric recursive generation strategy to construct the binary tree T_{latent} . We define the function BuildTree(span) as follows: for the current text span, if it contains the aspect term a , the aspect term is set as the root node of the current subtree. If the current span $\left[i,j\right]$ does not contain the aspect term, the index k with the highest structural score v_k within the span is selected as the splitting point (i.e., the root node of the subtree). We recursively call the BuildTree function on the left and right sub-spans of the root node, and establish parent-child connections between nodes accordingly.

To meet the computational requirements of the graph attention network, we map the generated discrete tree T_{latent} into a symmetric adjacency matrix $A^{DOT} \in \mathrm{R}^{n\times n}$. To promote effective information propagation, we define four types of connection edges to construct the adjacency matrix. Firstly, Self-loop: connect the node itself to preserve the original information of the node. Secondly, Top-down: if j is the parent node of i, add an edge $j \to i$; Thirdly, Bottom-up: if j is the parent node of i, add an edge $i \to j$; Finally, Full-connection (Optional):

$$A_{ij}^{DOT} = \begin{cases} 1, & \text{if} i = j \\ 1, & \text{if} w_i, w_j \in a \\ 1, & \text{if} (w_i, w_j) \in \mathrm{Edges}(T_{latent}) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Finally, normalize the adjacency matrix to ensure the training stability of the graph network:

$$\hat{A}^{DOT} = D^{-\frac{1}{2}} A^{DOT} D^{-\frac{1}{2}} \tag{14}$$

3.4. Feature Extraction Based on Multi-Layer Attention Integration

Traditional methods usually only adopt the output of the last layer for classification, ignoring the complementarity between shallow lexical details and deep semantic dependencies. To this end, this section introduces the multi-layer attention integration module to perform dynamic weighted fusion on representations of different layers, so as to make fuller use of multi-order neighborhood information and alleviate the over-smoothing problem caused by deep graph networks.

The multi-layer attention integration module is embedded inside the two branches, dynamically fusing the output features of all layers and adaptively balancing shallow lexical details and deep semantic logic. Taking the DOT branch as an example, the node feature update formula of the $l \in [1, L]$-th layer is as follows:

$$H_{DOT}^{(l)} = \mathrm{ReLU}\left(\hat{A}^{DOT} H_{DOT}^{(l-1)} W^{(l)} + b^{(l)}\right) \tag{15}$$

Where $W^{(l)} \in \mathrm{R}^{\mathrm{d}\times\mathrm{d}}$ denotes the layer weight.

Taking the DOT branch as an example, the ALE module dynamically fuses features of all layers. Firstly, perform aspect-based pooling on the output feature $H^{(l)}$ of each layer l to obtain the hierarchical representation vector $z^{(l)}$:

$$z^{(l)} = \mathrm{Pooling}_{w_k \in a}\left(H_{DOT}^{(l)}[k,:]\right) \tag{16}$$

Then, calculate the importance weight $\alpha^{(l)}$ of each layer via the attention mechanism:

$$e_l = \mathrm{q}_{att}^{T} \tanh\left(W_{att} \mathrm{z}^{(l)} + b_{att}\right) \tag{17}$$

$$\alpha^{(l)} = \frac{\exp\left(e_l\right)}{\sum_{k=1}^{L} \exp\left(e_k\right)} \tag{18}$$

The final branch feature representation O_{DOT} is the weighted sum of all layers:

$$O_{DOT} = \sum_{l=1}^{L} \alpha^{(l)} \cdot z^{(l)} \tag{19}$$

Similarly, we can obtain the final representation of the Se branch in the same way

3.5. Adaptive Aggregation Module

To fully exploit the complementarity between the explicit local semantic feature O_{Se} and the implicit global structural feature O_{DOT}, and address the noise problem that may be introduced by a single structure, this chapter designs an adaptive aggregation module based on cross-attention. Different from simple concatenation or gated fusion, this mechanism enables the feature interaction between the two branches: on the one hand, semantic information is used to guide feature screening of the discrete tree structure; on the other hand, the global tree structure is adopted to enhance the local semantic representation.

Specifically, the module contains two cross-attention flows. Firstly, to obtain the semantics-enhanced structural representation $H_{DOT}^{'}$, we take the output O_{DOT} of the discrete opinion tree branch as the query vector, and the output O_{Se} of the segment-aware semantic branch as the key and value, which are fed into the multi-head attention mechanism:

$$\tilde{H}_{DOT} = \mathrm{MH}(Q = O_{DOT}, K = O_{Se}, V = O_{Se}) \tag{20}$$

$$H_{DOT}^{'} = \mathrm{LayerNorm}(\mathrm{FFN}(\tilde{H}_{DOT}) + O_{DOT}) \tag{21}$$

Similarly, to obtain the structure-enhanced semantic representation $H_{Se}^{'}$, we take O_{Se} as the query vector and interact with O_{DOT} which serves as the key and value:

$$\tilde{H}_{Se} = \mathrm{MH}(Q = O_{Se}, K = O_{DOT}, V = O_{DOT}) \tag{22}$$

$$H_{Se}^{'} = \text{LayerNorm}(\text{FFN}(\tilde{H}_{Se}) + O_{Se}) \tag{23}$$

Were LayerNorm denotes layer normalization and FFN represents the feed-forward neural network. Through such cross-interaction, the model can dynamically seek the most relevant features in the alternative view for the current view.

To balance the roles of different views, we further introduce adaptive gating weights. Concatenate the two enhanced features, and calculate the fusion weight α via a linear layer:

$$\alpha = \sigma(W_{agg}[H_{DOT}^{'} \oplus H_{Se}^{'}] + b_{agg}) \tag{24}$$

Where W_{agg} and b_{agg} denote the learnable linear transformation weights and biases, respectively.

The final fused feature vector H^{F} is the weighted sum of the two:

$$H^{F} = \alpha H_{DOT}^{'} + (1-\alpha) H_{Se}^{'} \tag{25}$$

Finally, input H^{F} into the SoftMax classifier for sentiment polarity prediction:

$$p(y \mid s, a) = \text{Softmax}(W_{out} H^{F} + b_{out}) \tag{26}$$

Where W_{out} and b_{out} denote the learnable weights and biases of the classifier, respectively.

4. Experimental Analysis

4.1. Experimental Data and Experimental Settings

This paper evaluates the proposed method on four public English datasets, including the Laptop and Restaurant datasets from SemEval2014 Task4, the MAMS dataset, and the Twitter dataset . The aforementioned datasets cover both single-aspect and multi-aspect scenarios, which can comprehensively test the model performance in complex sentence modeling and cross-scenario generalization.

In the experiments of this chapter, the pre-trained BERT model is adopted as the text encoder, and the constituent trees are obtained via the SuPar2 parser. The AdamW optimizer is utilized with a learning rate of 2×10^{-5} and a weight decay coefficient of 1e-5. The batch size is set to 16 for the Restaurant and Laptop datasets, and 32 for the Twitter and MAMS datasets. The number of training epochs is set to 20, and the early stopping strategy is adopted on the validation set. To reduce the impact of randomness, the random seed is fixed to 42 and all experiments are run three times repeatedly. The model evaluation metrics adopt classification Accuracy (Acc.%) and Macro-F1 score (F1.%), to comprehensively reflect the model's ability to recognize various types of samples.

Table 1Experimental Datasets

Dataset		Sentence-Type		Aspect-Type		
		Mul.-	Sin.-Asp.	#Pos.	#Neg.	#Neu.
Res.	Train	971	1009	2164	807	637
	Test	315	284	727	196	196
Laptops.	Train	538	916	937	861	455
	Test	150	259	337	128	167
MAMS.	Train	4297	0	3380	2764	5042
	Test	500	0	400	329	607
Twitter.	Train	0	6057	1507	1528	3016

4.2. Baseline Models

We compare our proposed model with several mainstream and state-of-the-art models in aspect-based sentiment analysis (ABSA), covering three major categories of typical methods as follows:Attention-based Methods: DualGCN (2022) [4], AG-VSR (2022) [5], ASHGAT (2024) [6];Deep Learning-based Methods: SSEGCN (2022) [7], WGAT-SBERT (2023) [8], DMAN (2024) [9], MASGCN (2025) [10];Multi-aspect Oriented Methods: RMN (2022) [11], IA-HiNET (2023) [12], APSCL (2023) [13], AGCL (2025) [14].

Table 2 Comparison with Baseline Models

Model		Dataset							
		Laptops		Restaurants		Twitter		MAMS	
		Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)
Att.	DualGCN	78.58	74.76	84.27	78.08	75.92	74.29		
	AG-VSR	79.43	75.45	84.43	77.52	74.23	72.56		
	ASHGAT	80.22	77.80	85.61	80.70	74.84	74.85		
Syn.	SSEGCN	78.89	75.04	83.55	79.93	74.51	74.32		
	WGATS	80.49	77.21	85.71	80.23	76.25	74.56		
	DMAN	80.49	77.21	85.71	80.23	74.25	74.56		
	MASGCN	81.48	77.38	86.22	80.82				
Multi	RMN	79.98	76.58	85.49	79.23			79.78	78.97
	IA-HiNET	80.36	76.76	85.61	72.85			83.83	83.44
	APSCL	79.85	76.57	84.79	80.64			84.06	83.52
	AGCL	81.85	77.90	85.80	80.28	74.95	73.37		
Our.	SeDOT-SSEGAT	80.83	77.02	86.71	80.66	76.66	75.82	84.20	83.83

As shown in Table 2, the proposed SeDOT-SSEGAT achieves superior performance on all four datasets, and its overall effectiveness outperforms most of the comparative models.

Specifically, graph-based deep learning methods generally perform better than attention-based methods in this task. This is because graph-based methods usually construct the graph structure between words in a sentence to represent semantic information, thereby modeling the overall sentence semantics for more accurate sentiment analysis. In contrast, attention-based mechanisms mainly focus on the local relationships between individual words, which may fail to fully capture the global semantic information of the entire sentence.

In addition, compared with multi-aspect oriented methods, our method reduces cross-segment interference in multi-aspect sentences via the segment-aware semantic stream, and enhances the modeling capability for long-distance sentiment trigger words through the discrete opinion tree-induced stream. Therefore, it exhibits more stable classification performance in complex sentences and multi-aspect scenarios.

4.3. Ablation Study

To evaluate the specific contribution of each module in the SeDOT-SSEGAT model to the overall performance, this paper designs and conducts ablation experiments, with relevant results listed in the corresponding table. During the experiments, different components of the model are removed separately, and the performance evaluation is carried out on the four datasets: Restaurant, Laptop, Twitter and MAMS.

In Table 3, the notation "w/o" denotes the removal of the specified module from SeDOT-SSEGAT. Specifically, the segment-aware module is denoted as Se, the discrete opinion tree module is denoted as DOT, the attention integration module is denoted as ALE, and the adaptive aggregation module is denoted as AAM.

Table 2 Comparison with Baseline Models

Ablation	Laptops		Restaurants		Tweets		MAMS	
	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)
Ours	80.83	77.02	86.71	80.66	76.66	75.82	84.20	83.83
w/o se	78.78	73.84	84.83	77.32	73.79	71.44	81.63	81.33
w/o dot	78.85	75.01	84.71	79.20	74.15	73.72	81.57	81.01
w/o ale	79.32	75.13	85.16	77.58	74.88	73.34	82.70	82.11
w/o aam	79.23	74.92	85.22	77.62	74.67	73.23	82.71	82.10

5. Result Analysis

To validate the effectiveness of each core component, we conduct ablation studies by removing individual modules sequentially.

Removing either the Se branch or the DOT branch leads to a 2%–3% drop in both Accuracy and F1-score across all four datasets, with the most pronounced performance degradation observed on the MAMS dataset, which contains the highest proportion of multi-aspect sentences.

Removing the Se branch eliminates the model's ability to perform local semantic filtering guided by constituent tree boundary constraints, rendering it incapable of effectively isolating interference from conflicting aspects within the same sentence.

Removing the DOT branch strips the model of its independence from external syntactic parsers and impairs its capacity to capture sentiment correlations between long-distance contexts and target aspect terms, thereby confirming the complementary nature of the dual-branch representation.

After removing the ALE module, model performance declines by 1%–2%, which validates the limitations of traditional Graph Convolutional Networks (GCNs) that solely rely on the final layer's output. The ALE module dynamically fuses lexical detail features from shallow GCN layers and long-range semantic features from deep layers via hierarchical attention, effectively mitigating the over-smoothing issue of deep GCNs and enabling the model to more fully exploit syntactic and semantic information from multi-order neighborhoods.

Removing the AAM module also results in a 1%–2% performance drop. In contrast to naive feature concatenation, the cross-attention-based adaptive aggregation module facilitates deep interaction between the two branch features: it uses local semantic information to guide the selection of global structural features, while leveraging global structural cues to enhance local semantic representation. By dynamically balancing the contributions of the two branches through adaptive gating weights, this module further elevates the efficacy of cross-branch feature fusion.

5.1. The influence of segmented perception on semantic flow

To further verify the effectiveness of segmented-aware semantic flow, this paper conducts a visual analysis of the attention distributions of different model variants. As shown in Figure 3 and Figure 4, in the absence of segmentation constraints, the model tends to allocate attention to sentiment words corresponding to other aspects, resulting in misalignment. In contrast, after introducing segmented-aware semantic flow, the model's attention can be more intensively focused on semantic segments related to the target aspect, thereby effectively mitigating cross-aspect interference in multi-aspect sentences.

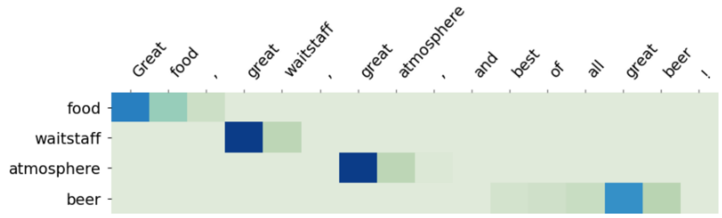

Fig.3 SeDOT-SSEGAT

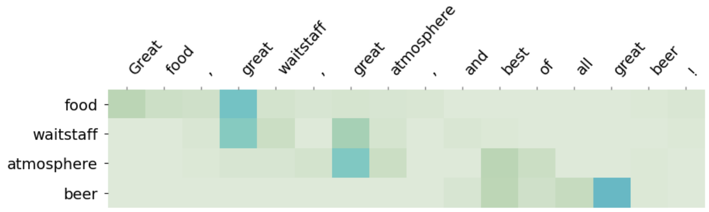

Fig.4 SeDOT-SSEGAT w/o Se

5.2. Comparison between Discrete Opinion Tree and Dependency Tree

To verify the rationality of the induced discrete opinion tree, this paper further compares it with dependency trees generated by humans or parsers, and analyzes the distance distribution between aspect terms and opinion terms. As shown in Figure 5, the discrete opinion tree and the dependency tree exhibit high consistency in the overall distribution, indicating that the model can effectively capture the key structural relationships between aspects and opinions. Furthermore, in scenarios involving low-frequency or unseen aspects, the proposed method shows stronger robustness compared with models relying on external parsers. This demonstrates that the discrete opinion tree can avoid the propagation of parsing errors while adapting more flexibly to task objectives, thereby improving the structural modeling capability in complex scenarios.

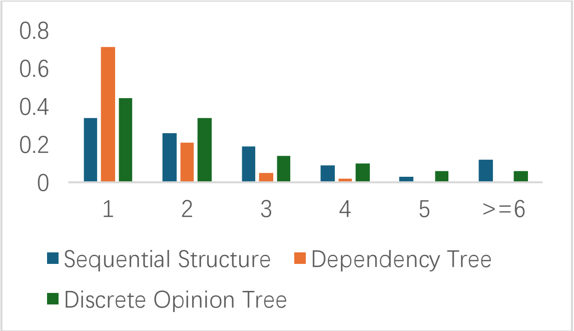

Fig.5 The distance between aspect terms and opinion terms

6. Conclusion

To address the limitations of existing graph-based ABSA methods that are prone to introducing irrelevant structural noise in complex sentences, rely heavily on external syntactic parsers, and frequently suffer from aspect misalignment, this paper proposes a syntax and semantic enhanced graph attention network with segmented perception and discrete opinion tree induction, namely SeDOT-SSEGAT. Specifically, the model first employs BERT to obtain contextual token representations. Subsequently, it leverages the segmented-aware semantic flow to model local semantic fragments, and utilizes the discrete opinion tree induction branch to capture global structural dependencies. Furthermore, the multi-layer attention integration module and adaptive aggregation module are adopted to achieve collaborative fusion of the dual-branch features. Experimental results on multiple public benchmark datasets demonstrate that the proposed model achieves superior performance over most baseline models, verifying that this method can effectively improve aspect alignment accuracy and sentiment classification precision in complex sentences.

References

[1] L. Zhu, M. Xu, Y. Bao, et al.: Deep Learning for Aspect-Based Sentiment Analysis: A Review, PeerJ Computer Science, Vol. 8 (2022), p. e1044.

[2] A. Vaswani, N. Shazeer, N. Parmar, et al.: Attention Is All You Need, Advances in Neural Information Processing Systems 30 (Long Beach, USA, 2017), Vol. 30, p. 5998–6008.

[3] S. Chaudhari, G. Polatkan, R. Shen, et al.: An Attentive Survey of Attention Models, ACM Transactions on Intelligent Systems and Technology, Vol. 12 (2021) No.5, p. 1–32.

[4] R. Li, H. Chen, F. Feng, et al.: Dual Graph Convolutional Networks for Aspect-Based Sentiment Analysis, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Online, 2021), p. 6319–6329.

[5] S. Feng, B. Wang, Z. Yang, et al.: Aspect-Based Sentiment Analysis with Attention-Assisted Graph and Variational Sentence Representation, Knowledge-Based Systems, Vol. 258 (2022), p. 109975.

[6] J. Ouyang, C. Xuan, B. Wang, et al.: Aspect-Based Sentiment Classification with Aspect-Specific Hypergraph Attention Networks, Expert Systems with Applications, Vol. 248 (2024), p. 123412.

[7] Z. Zhang, Z. Zhou, Y. Wang: SSEGCN: Syntactic and Semantic Enhanced Graph Convolutional Network for Aspect-Based Sentiment Analysis, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Seattle, USA, 2022), p. 4916–4925.

[8] T. Jiang, Z. Wang, M. Yang, et al.: Aspect-Based Sentiment Analysis with Dependency Relation Weighted Graph Attention, Information, Vol. 14 (2023) No.3, p. 185.

[9] Y. Chen, K. Zhang, F. Hu, et al.: Dynamic Multi-Granularity Attribution Network for Aspect-Based Sentiment Analysis, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (Miami, USA, 2024), p. 10920–10931.

[10] X. Huang, H. Peng, S. Sun, et al.: Multi-View Attention Syntactic Enhanced Graph Convolutional Network for Aspect-Based Sentiment Analysis, Database Systems for Advanced Applications: DASFAA 2025 (Singapore, 2026), Vol. 15987, p. 289–304.

[11] J. Zeng, T. Liu, W. Jia, et al.: Relation Construction for Aspect-Level Sentiment Classification, Information Sciences, Vol. 586 (2022), p. 209–223.

[12] T. Gu, H. Zhao, M. Li: Effective Inter-Aspect Words Modeling for Aspect-Based Sentiment Analysis, Applied Intelligence, Vol. 53 (2023) No.4, p. 4366–4379.

[13] P. Li, P. Li, X. Xiao: Aspect-Pair Supervised Contrastive Learning for Aspect-Based Sentiment Analysis, Knowledge-Based Systems, Vol. 274 (2023), p. 110648.

[14] Z. Jian, D. Wu, S. Wang, et al.: AGCL: Aspect Graph Construction and Learning for Aspect-Level Sentiment Classification, Proceedings of the 31st International Conference on Computational Linguistics (Abu Dhabi, UAE, 2025), p. 841–854.