

Article

Text Triplet Extraction Algorithm with Fused Graph Neural Networks and Improved Biaffine Attention Mechanism

Yinghao Piao¹ and Jin-Xi Zhang^{2,*} ¹ College of Sciences, Northeastern University, Shenyang 110819, China; 2170016@stu.neu.edu.cn² State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China

* Correspondence: zhangjx@mail.neu.edu.cn

Abstract: In the realm of aspect-based sentiment analysis (ABSA), a paramount task is the extraction of triplets, which define aspect terms, opinion terms, and their respective sentiment orientations within text. This study introduces a novel extraction model, BiLSTM-BGAT-GCN, which seamlessly integrates graph neural networks with an enhanced biaffine attention mechanism. This model amalgamates the sophisticated capabilities of both graph attention and convolutional networks to process graph-structured data, substantially enhancing the interpretation and extraction of textual features. By optimizing the biaffine attention mechanism, the model adeptly uncovers the subtle interplay between aspect terms and emotional expressions, offering enhanced flexibility and superior contextual analysis through dynamic weight distribution. A series of comparative experiments confirm the model's significant performance improvements across various metrics, underscoring its efficacy and refined effectiveness in ABSA tasks.

Keywords: aspect-based sentiment analysis; aspect sentiment triplet extraction; biaffine attention; graph attention network; graph convolutional network



Citation: Piao, Y. H.; Zhang, J.-X. Text Triplet Extraction Algorithm with Fused Graph Neural Networks and Improved Biaffine Attention Mechanism. *Appl. Sci.* **2024**, *14*, 3524. <https://doi.org/10.3390/app14083524>

Academic Editor: Rui Araújo

Received: 1 April 2024

Revised: 13 April 2024

Accepted: 19 April 2024

Published: 22 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Neural networks, by simulating the way neurons in the human brain connect to process information and learn, are capable of automatically identifying valuable features and patterns from vast amounts of data, thereby automating the execution of complex tasks. This technology has attracted widespread attention due to its broad application potential and its ability to effectively integrate concepts from multiple fields, e.g., [1,2]. In recent years, the field of image processing has also witnessed innovative integrations with fractional-order processing methods [3–6], demonstrating the powerful synergy of interdisciplinary technologies. As a cornerstone of deep learning, neural networks offer substantial potential and value in various domains such as image recognition, natural language processing, and complex decision support systems, thereby playing a crucial role in advancing artificial intelligence.

In the context of the current data-centric era, the progress of neural networks in the realm of deep learning has significantly fostered a deeper comprehension of textual content, placing a fine-grained analysis of text sentiment at the vanguard of research and practical application. Of particular note is the application of graph neural networks in aspect-based sentiment analysis, providing robust support for accurately identifying the sentiment orientations towards specific entities or aspects within texts. Within the multitude of subtasks in the aspect-based sentiment analysis domain, employing graph neural networks for aspect sentiment triplet extraction task poses unique challenges, not merely due to the increased complexity involved, but also due to the urgent requirement for the in-depth mining of textual details. Therefore, the thorough investigation and exploration of the aspect sentiment triplet extraction task hold substantial academic value and practical significance for

overcoming these challenges and further broadening the application spectrum of graph neural networks in complex natural language processing tasks.

Currently, the research methods are mainly divided into two-stage pipeline architecture, machine reading comprehension (MRC), sequence-to-sequence (seq2seq), and end-to-end approach, which includes annotation strategies and integrated solutions. A summary is provided in Table 1. Peng et al. [7] pioneered the ASTE task and implemented triplet extraction using a two-stage pipeline architecture, which integrates aspect extraction, aspect sentiment classification, and opinion extraction. However, there may be propagated errors in subsequent stages that are amplified. In addition, other model frameworks such as MRC [8,9] and seq2seq [10–12] have also been applied to this task. For example, Mao et al. [8] designed specific query questions to decompose the original task into two MRC tasks: one for extracting aspect terms and the other for predicting corresponding opinion terms and sentiment polarities. Chen et al. [9] adopted a bidirectional MRC framework and used a similar approach: first predicting aspects and opinions, and then predicting opinions and aspects in reverse. Zhang et al. [10] converted the original task into a text generation problem and proposed two sentiment triplet prediction models, including annotation style and extraction style. Yan et al. [11] and Hsu et al. [12] used sentences as input and pointer indices as targets to predict the start and end indices of aspect terms (or opinion terms). Mukherjee et al. [13] designed an unlabeled decoding framework using pointer networks for aspect sentiment triplet extraction, overcoming the limitations of complex label schemes in traditional methods. Fei et al. [14] proposed a non-autoregressive decoding method that models the ASTE task as an unordered triplet prediction problem. However, these methods may have limitations in capturing long-distance dependencies and understanding complex sentence structures, which may lead to poor performance when dealing with complex or ambiguous aspect and opinion expressions. Moreover, these models usually require a large amount of labeled data for training to achieve satisfactory performance. To deeply investigate the interactions between multiple sentiment factors, other researchers have proposed numerous integrated solutions. For example, Zhang et al. [15] proposed a multi-task learning framework that covers aspect term extraction, opinion term extraction, and sentiment polarity analysis, and further synthesizes sentiment triplets from the prediction results of these subtasks through heuristic rules. Another research direction focuses on developing a unified annotation strategy to complete triplet extraction in one go. For instance, Xu and Wu et al. [16,17] achieved the goal of extracting triplets in one shot. Specifically, Xu et al. [16] proposed a position-aware annotation strategy aimed at overcoming the limitations of the existing works by enhancing the expressiveness of labels. Wu et al. [17] proposed a scheme that achieved end-to-end task processing by tagging relationships between all word pairs. Despite the advantages of end-to-end methods and grid tagging schemes, it should be noted that end-to-end methods may be limited by their high demand for a large amount of labeled data and excessive reliance on data distribution, while grid tagging schemes may fail to handle highly complex or ambiguous text relationships.

Due to the complexity of language structure, accurate sentiment analysis often requires a comprehensive consideration of contextual information, syntactic features, and other relevant information. Graph neural network (GNN), neural networks tailored for graph-structured data, have garnered significant interest for their distinct benefits. Subsequently, graph convolutional neural network (GCN) and graph attention network (GAT) based on GNN have been introduced into sentiment analysis tasks to enhance the understanding of model and utilization of syntactic structure. For example, Bastings et al. [18] constructed directed graphs to integrate syntactic relationships, and GCNs were used to improve the accuracy of machine translation models, demonstrating their powerful ability to capture syntactic features in text. A common approach to more finely pair aspects with their corresponding opinion representations is to use syntactic dependency trees to construct adjacency matrices for sentence graph structures. However, there are still shortcomings in the analysis that solely relies on the adjacency matrix generated by the dependency

tree. To highlight highly relevant information in the dependency tree, Guo et al. [19], Chen et al. [20], and Li et al. [21] introduced attention mechanisms to construct new edges in the graph structure, thereby enhancing the ability of the model to grasp dependency relationships. Huang et al. [22] and Wang et al. [23] used GAT on graph-structured data, where words in sentences are nodes in the graph and grammatical relationships are edge labels. The dependency graphs used in the model are obtained from ordinary dependency tree parsers or modified using heuristic rules. Meanwhile, the widespread use of GCN in processing dependency graphs and the effectiveness of GAT in handling aspect triplet extraction tasks demonstrate the potential of these two graph neural networks in enhancing feature representation. Specifically, Huang et al. [22] used GAT to focus on sentiment words that are more closely related to aspect terms through target-dependent methods, which helps to address syntactic ambiguity and clearly establish dependencies between words. However, the above studies only used a single graph neural network and did not take full advantage of graph neural networks. Therefore, considering the respective advantages of GCN and GAT in processing graph-structured data and enhancing syntactic and semantic relationships, this study aims to fuse GCN and GAT to further improve the performance and accuracy of aspect sentiment triplet extraction tasks by integrating the strengths of these two graph networks, overcoming the limitations of existing methods, and effectively capturing and representing complex text relationships.

Table 1. Previous work.

	Method Type	Feature
Pipeline	Peng-two-stage [7]	Build a triplet by breaking down the task into two stages.
MRC	Dual-MRC [8] BMRC [9]	Build two machine reading comprehension tasks to jointly solve subtasks. Transforming the ASTE task into a multi round reading comprehension task.
seq2seq	GAS [10] Unified generative framework [11] Semantics-preserved data augmentation [12]	Develop the task as a text generation problem. Transforming multiple subproblems of sentiment analysis into a unified generative problem. Improving model performance by expanding the dataset and increasing data diversity.
end-to-end	PASTE [13] onautoregressive encoder–decoder [14] OTE-MTL [15] JET-BERT [16] GTS [17]	Propose a location-based approach to unify the representation of opinion triplets. Propose a high-order aggregation mechanism to fully interact with overlapping triplets. Propose a multi-task learning framework to jointly extract aspect words and viewpoint words. Using position aware tagging scheme to jointly extract triples. Propose a grid tagging scheme to solve the ASTE task through only a unified grid tagging task.

In this paper, a new end-to-end model is proposed, which integrates graph convolutional networks and graph attention networks with an improved biaffine attention mechanism for the task of aspect sentiment triplet extraction. The main contributions of this study are summarized as follows:

- An innovative end-to-end solution that enhances the flexibility and portability of a model through modular design, capable of effectively handling multiple subtasks within a single end-to-end process, significantly improving the processing efficiency and accuracy.
- By integrating the attention mechanism of GAT with the deep processing capabilities of GCN, there is a significant improvement in processing efficiency for graph-structured data and a deeper understanding of textual content, effectively capturing the complex relationships and rich contextual information within texts.

- By integrating BiLSTM to improve the biaffine attention mechanism, it effectively extracts high-dimensional aspect features, significantly enhancing the capability to process both local and global textual information, and increases the accuracy of the model in capturing long-distance dependencies and recognizing complex emotional expressions.

The following is the order in which each consecutive section is presented in the paper: In Section 2, the preliminaries of this paper are introduced to the reader in detail. In Section 3.1, a thorough discussion of the specific structure of the model we propose is given. Section 4 details an in-depth experimental study and analyzes the performance. Section 5 concludes this study and discusses future directions.

2. Preliminaries

2.1. Graph Neural Network

GNN represents a novel class of neural network models specifically used to process graph-structured data, distinguishing them from traditional neural network models such as convolutional neural network and recurrent neural network that handle regular data structures. Graph data, composed of nodes and edges, depict complex networks of entities and their interrelations, thus necessitating a unique approach to understanding and representing these entities and their connections. The core mechanism of GNN lies in aggregating the feature information of each node and its neighbors, constructing new node representations through this process. This is typically achieved through a message-passing framework, where nodes collect information from their neighbors and integrate this information to form updated node representations. Through multiple iterations, this captures the overall structural information of the graph.

2.1.1. GCN

Inspired by convolutional neural networks, GCN is an efficient variant of CNN that operates directly on graphs. Figure 1 depicts the architectural framework of GCN. The core formula lies in the propagation between layers, calculated by:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \tag{1}$$

where $\tilde{A} = A + I$ represents the adjacency matrix of the graph plus self-connections, $A \in \mathbb{R}^{n \times n}$ and I the adjacency matrix and identity matrix; \tilde{D} represents the degree matrix, with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$; H represents the features at each layer; $W^{(l)}$ is the weight matrix for the l th layer; σ is a nonlinear activation function, such as *Sigmoid*, *ReLU*.

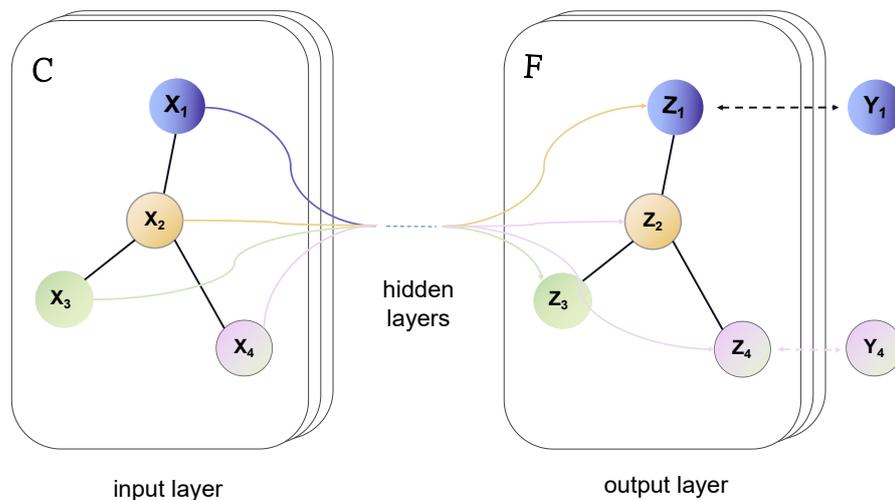


Figure 1. The GCN architecture.

Given the feature representation of node i at layer $l - 1$, h_i^{l-1} , the output, which is the feature representation of node i at layer l , h_i^l , is obtained by:

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

where $W^{(l)}$ is the weight matrix; $b^{(l)}$ is the bias vector; $h_i^{(0)}$ represents the initial input of feature x_i , with $x_i \in R^d$, and d the dimension of the input features.

2.1.2. GAT

GAT can be seen as a variant of GCN. While GCN aggregates nodes using the Laplacian matrix, GAT assigns different weights to each node through an attention mechanism, aggregating nodes according to the magnitude of these weights when updating the hidden layers of nodes. Therefore, by adding attention layers in the network, GAT makes the computation more efficient and allows for the assignment of different levels of importance to different nodes without relying on previous spectral-based methods, thereby enhancing the interpretability of model. Figures 2 and 3 depict the architectural framework of GAT with self and multi-head attention.

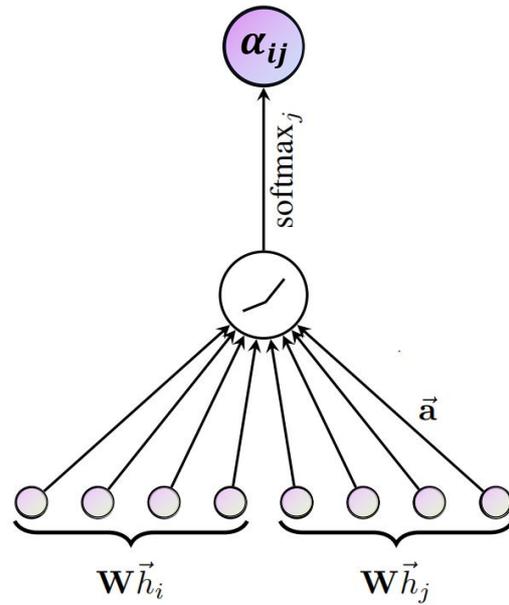


Figure 2. GAT with self attention.

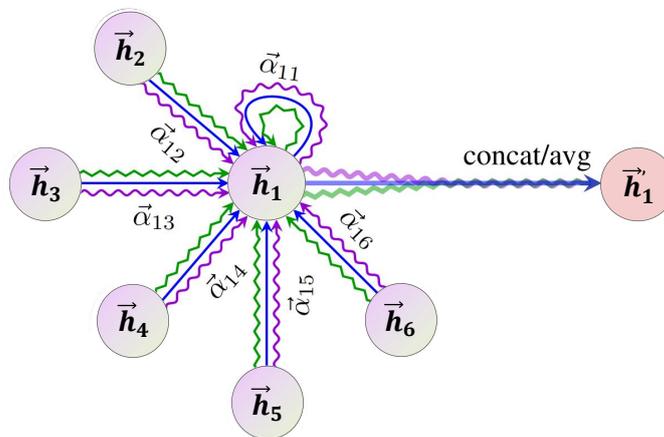


Figure 3. GAT with multi-head attention.

Firstly, an attention mechanism is applied to each node to train its own weight matrix, calculated as follows:

$$e_{ij} = a(W\vec{h}_i, W\vec{h}_j) \tag{3}$$

where the input is $h = [\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N]$, with $\vec{h}_i \in \mathbb{R}^F$; N is the number of nodes; F is the number of features per node; W is the weight matrix, and a represents the attention mechanism's function; e_{ij} is obtained through (3), which indicates the importance of node j to node i .

To facilitate the comparison of coefficients across different nodes, the softmax function is used to normalize them, as follows:

$$a_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \tag{4}$$

In the output layer of the aforementioned feedforward neural network, add a LeakyReLU function:

$$a_{ij} = \frac{\exp(\text{LeakReLU}(\vec{a}^T [W\vec{h}_i || W\vec{h}_j]))}{\sum_{k \in N_i} \exp(\text{LeakReLU}(\vec{a}^T [W\vec{h}_i || W\vec{h}_k]))} \tag{5}$$

where LeakyReLU is the activation function; T represents transpose; $||$ denotes the concatenation of the vector representations of nodes i and j after transformation. Subsequently, the final output result is obtained after passing through an activation function:

$$\vec{h}'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W\vec{h}_j\right) \tag{6}$$

Given that a single layer of self-attention mechanism has a limited capacity to learn from surrounding nodes, to further enhance the representational ability of each output features of node, GAT introduces the application of multi-head attention mechanisms within the network. By utilizing multiple attention mechanisms to calculate the attention coefficients of surrounding nodes, the learning effectiveness of the model becomes more stable. Executing k attention mechanisms and then concatenating the features they generate, the specific output feature representation is obtained through:

$$h'_i(K) = ||_{k=1}^K \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k W^k h_j\right) \tag{7}$$

where α_{ij}^k is the normalized coefficient computed by the k th attention mechanism α^k and W^k is the weight matrix corresponding to the input linear transformation.

2.2. Biaffine Attention

The biaffine attention mechanism is used as an alternative to the traditional MLP-based attention mechanism and affine label classifier, not using the LSTM recurrent state in the dual affine transformation, but instead first reducing dimensionality through MLP operations. Based on the aforementioned performance characteristics, the biaffine attention mechanism is employed to capture the probability distribution of the relationship between every pair of words in a sentence, as described by:

$$h_i^a = \text{MLP}_a(h_i) \tag{8}$$

$$h_j^o = \text{MLP}_o(h_j) \tag{9}$$

The hidden states h_i and h_j obtained from the word embeddings W_i and W_j are processed through a multilayer perceptron to obtain the aspect feature representation h_i^a

and the opinion feature representation h_j^o . The adjacency matrix $R \in \mathbb{R}^{n \times n \times m}$ is obtained through Equations (10)–(12), which represents the relationships between words.

$$g_{i,j} = h_i^{aT} U_1 h_j^o + U_2 (h_i^a \oplus h_j^o) + b \tag{10}$$

$$r_{i,j,k} = \frac{\exp(g_{i,j,k})}{\sum_{l=1}^m \exp(g_{i,j,l})} \tag{11}$$

$$R = \text{Biaffine}(\text{MLP}_a(H), \text{MLP}_o(H)) \tag{12}$$

where, U_1 , U_2 , and b are trainable weights and bias; \oplus denotes concatenation; $g_{i,j}$ represents the relationship between w_i and w_j ; m is the number of relationship types; $r_{i,j,k}$ represents the score for the k th relationship type of the word pair (w_i, w_j) . Biaffine is the integration of the aforementioned steps. Figure 4 shows a schematic diagram structure of model.

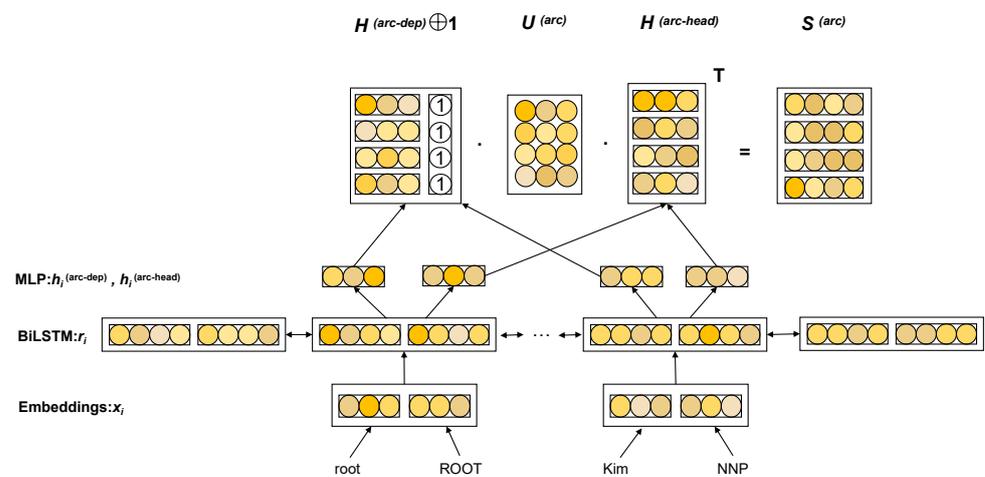


Figure 4. The Biaffine attention architecture.

2.3. BiLSTM

The bidirectional long short-term memory (BiLSTM) incorporates a refined gate mechanism and memory cells, optimizing the feature extraction process by selectively retaining significant features and omitting the less relevant ones. Specifically, i represents the input gate, which is responsible for handling the input at the current sequence position; f is the forgetting gate, indicating the degree of forgetting the hidden cell state of the previous layer, which is used to forget unimportant information; o indicates the output gate, which is used to determine the output value of the information; memory cells are used to record additional information. Figure 5 depicts the architectural framework of BiLSTM.

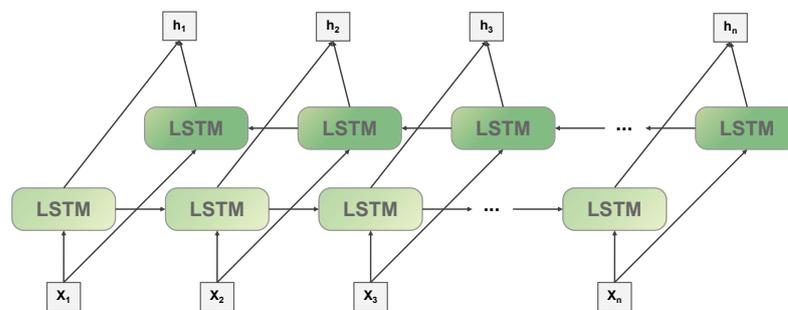


Figure 5. The BiLSTM architecture.

At time t , given the current input x_t , the calculation procedure of the bidirectional LSTM is as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{13}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{14}$$

$$c_t = i_t g_t + f_t c_{t-1} \tag{15}$$

$$g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + W_{cc}c_{t-1} + b_c) \tag{16}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{17}$$

$$h_t = o_t \tanh(c_t) \tag{18}$$

where t is the current moment; σ represents the sigmoid function; h_{T-1} and c_{t-1} represent the hidden vector of the previous moment, respectively; i_t representative enter door; W_{Ix} , W_{Ih} , W_{Ic} , b_I are, respectively, corresponding to the weighting matrix and deviation; the same, f_t forgotten door, W_{xf} , W_{hf} , W_{fc} , b_f , respectively, correspond to the weighting matrix and deviation; c_t is the cell state at time t ; o_t represents the output gate, W_{xo} , W_{ho} , W_{oc} , b_o , respectively, correspond to the weighting matrix and deviation. Finally, the results obtained from the forward LSTM and the backward LSTM are spliced together through (13) to (19) to obtain h_i :

$$h_i = \overrightarrow{h}_i \oplus \overleftarrow{h}_i \tag{19}$$

3. Proposed Framework

This paper describes a novel hybrid model known as BiLSTM-BGAT-GCN. It initially utilizes the hidden representation sequence obtained from the BERT pre-trained language model. Then, it processes through the four given types of language features, biaffine attention based on bidirectional long short-term memory, and graph attention neural networks. Subsequently, it employs a multi-branch GCN method to integrate multiple language feature representations obtained after multiple processing steps. Finally, the final aspect sentiment triplets are generated through surface-level interactions, output layer processing, and training steps. In addition, the architecture of BiLSTM-BGAT-GCN is described in Figure 6.

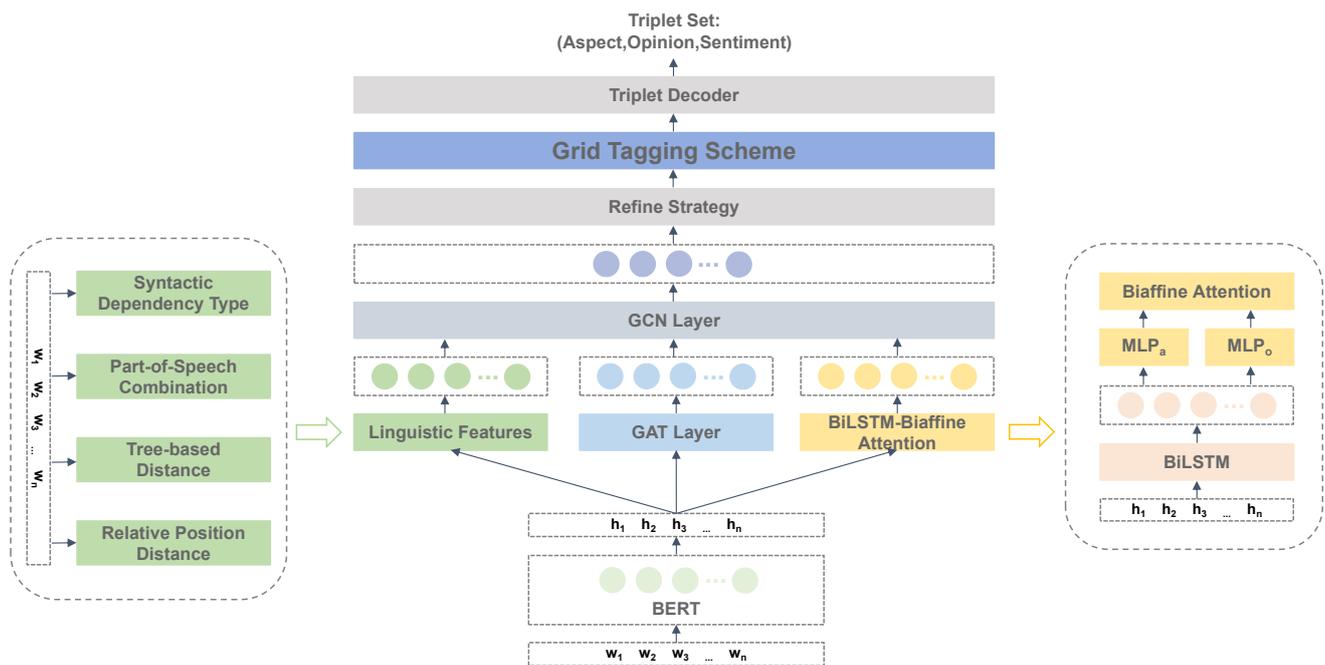


Figure 6. The architecture of the proposed BiLSTM-BGAT-GCN model.

3.1. Task Formulation

Given a sentence containing n words, the goal is to output a set of triplets $T = \{(a, o, s)_m\}_{m=1}^{|T|}$, where a and o represent aspect item and sentiment item. The affective polarity s belongs to the affective label set $S = \{POS, NEU, NEG\}$.

3.2. Input and Encoding Layer

The effectiveness of BERT pre-trained language models has been demonstrated on multiple tasks. For an input sentence denoted by $W = \{w_1, w_2, \dots, w_n\}$ with n signs, hidden by both the word embedded model based on BERT said sequence $H = \{h_1, h_2, \dots, h_n\}$.

3.3. Relation Definition and Table Filling

To provide more accurate information to the model and thus define the task objectives more precisely. Based on the four basic relations in the existing grid labeling scheme, the relationship between words is defined in greater detail, and the relationship definitions between ten kinds of words are given. Table 2 provides the corresponding interpretations, where NI represents relational inclusion that is not defined. According to the above rules, this paper constructs a relational table for each sentence using grid notation. In Figure 7, we visually represent the definitions provided by the table through an example sentence.

Table 2. The definitions of defined ten relations.

Sequence	Relationship	Meaning
1	$B - A$	Beginning of aspect term
2	$I - A$	Inside of aspect term
3	A	Aspect term
4	$B - O$	Beginning of opinion term
5	$I - O$	Inside of opinion term
6	O	Opinion term
7	POS	Sentiment polarity is positive
8	NEU	Sentiment polarity is neutral
9	NEG	Sentiment polarity is negative
10	NI	Not included in the above relationships

	The	gourmet	food	is	delicious	but	the	service	is	poor
The	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI
gourmet	NI	B-A	A	NI	POS	NI	NI	NI	NI	NI
food	NI	A	B-A	NI	POS	NI	NI	NI	NI	NI
is	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI
delicious	NI	POS	POS	NI	B-O	NI	NI	NI	NI	NI
but	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI
the	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI
service	NI	NI	NI	NI	NI	NI	NI	B-A	NI	NEG
is	NI	NI	NI	NI	NI	NI	NI	NI	NI	NI
poor	NI	NI	NI	NI	NI	NI	NI	NEG	NI	B-O

Figure 7. A grid marking of a sentence.

3.4. Linguistic Features

According to the structural features of the graph data, the relationship of linguistic features are established within the graph $G = (V, E)$, where V represents the vertices (i.e., nodes or words) and E represents the connection between two nodes (i.e., dependencies or linguistic relationships). Usually, this relationship is represented in matrix form. In this paper, inspired by Chen et al. [24], four linguistic feature matrices are introduced. Before being input into the graph convolutional neural network, these four language features need to be encoded as adjacency matrices and initialized as R^{sdt} , R^{psc} , R^{tbd} , and R^{rpd} , as shown in Figure 8.

In Figure 8, taking syntactic dependency types as an example, if there is some dependency between w_i and w_j , assuming the dependency is the noun subject (nsubj), then $R_{i,j}^{dep}$ is initialized as an embedding of nsubj; conversely, if there are no dependencies, $R_{i,j}^{dep}$ is initialized to an m-dimensional zero vector. Similarly, if w_i and w_j are a part-of-speech combination, such as a combination of a common noun (NN) and a determiner (DT), then $R_{i,j}^{dep}$ is initialized as an embedding of (NN-DT); or a combination of an adjective (JJ) and a common noun (NN) can initialize $R_{i,j}^{dep}$ as an embedding of (JJ-NN). Tree-based distance and relative position distance are similarly defined.

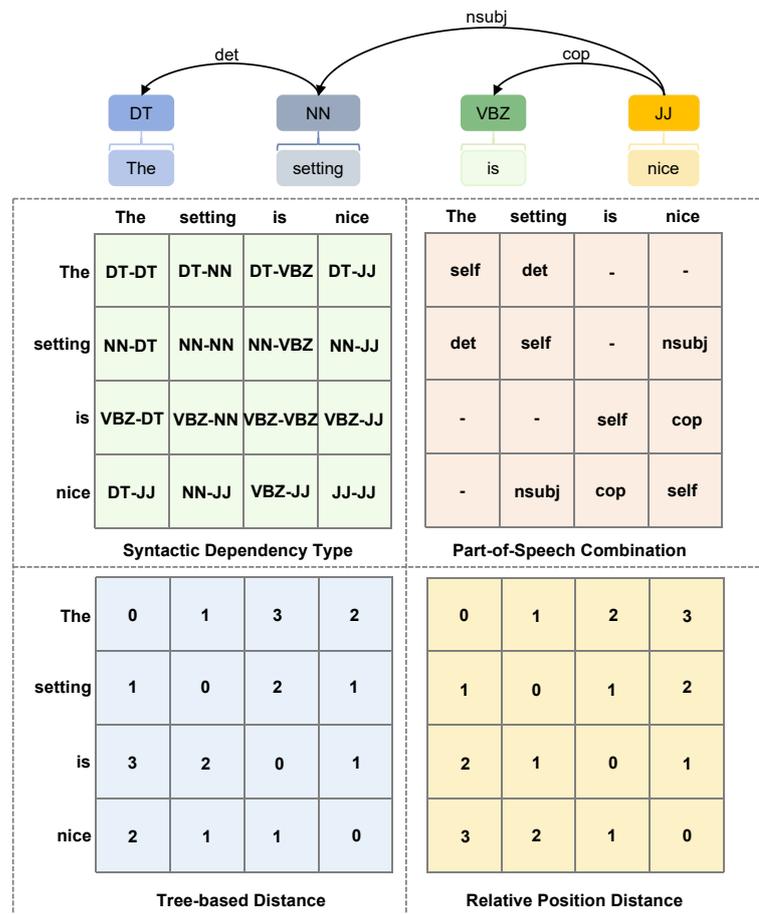


Figure 8. Four types of features for a sentence.

3.5. BiLSTM-Biaffine Attention

This method addresses the shortcomings of conventional feature extraction and relational understanding techniques, which often fail to effectively leverage long-distance contextual data and overlook complex interactions between aspect and opinion terms. By integrating the biaffine attention mechanism with BiLSTM, the model enhances aspect-

level sentiment analysis. BiLSTM excels in capturing long-term textual dependencies and contextual nuances, whereas the biaffine attention mechanism further refines how attention weights are distributed across text elements, particularly between aspect and opinion terms.

This fusion strategy not only enhances the ability of model to understand context, improving the precision of feature extraction and the capture of complex sentiment relationships, but also significantly boosts the generalization capabilities of model, enabling it to more accurately analyze and predict aspect-level emotions in diverse texts. By adopting this approach, we aim to overcome the limitations of traditional sentiment analysis methods and provide a more nuanced and efficient solution for complex textual sentiment analysis.

The sequence of hidden textual representations $H = \{h_1, h_2, \dots, h_n\}$ obtained by BiLSTM is first reduced by MLP and then input into the biaffine attention mechanism, which is calculated through:

$$h_i^a = \text{MLP}_a(h_i) \quad (20)$$

$$h_j^o = \text{MLP}_o(h_j) \quad (21)$$

Then, the adjacency matrix R is obtained by the biaffine attention mechanism as follows:

$$g_{i,j} = h_i^{aT} U_1 h_j^o + U_2 (h_i^a \oplus h_j^o) + b \quad (22)$$

$$r_{i,j,k} = \frac{\exp(g_{i,j,k})}{\sum_{l=1}^m \exp(g_{i,j,l})} \quad (23)$$

$$R = \text{Biaffine}(\text{MLP}_a(H), \text{MLP}_o(H)) \quad (24)$$

where U_1 , U_2 and b are trainable weights and bias; \oplus denotes connection; $g_{i,j}$ is the relationship between w_i and w_j ; m is the count of relationship types; and $r_{i,j,k}$ represents the rating of the word for the K -th relation type of (w_i, w_j) ; the adjacency matrix $R \in R^{n \times n \times m}$ represents the relationship between words. Biaffine is the integration of the aforementioned steps.

3.6. BiLSTM-BGAT-GCN Model

Prior to integration into the graph convolutional neural network, the data traverse six distinct pathways. The initial two pathways remain as they were, one facilitating the generation of a new adjacency matrix R^{ba} via the biaffine attention mechanism. Another pathway utilizes the graph attention network to derive a new node feature table \hat{H}^{gat} . Building on this, four additional pathways focus on linguistic features: syntactic dependency type, parts of speech combinations, syntactic tree-based distance, and positional relationships. Each of these pathways, respectively, results in four new adjacency matrices: R^{sdt} , R^{psc} , R^{tbd} , and R^{rpd} .

The adjacency matrix R^{sdt} , R^{psc} , R^{tbd} , and R^{rpd} obtained according to the four language features introduced above are, respectively, input into GCN to carry out repeated graph convolution operations. The new nodes represent H^{sdt} , H^{psc} , H^{tbd} , and H^{rpd} . Specific calculations are given in Equations (25)–(28):

(1) Syntactic dependency types

$$\tilde{H}_k^{sdt} = \sigma(R_{:, :, k}^{sdt} H W_k + b_k) \quad (25)$$

(2) Part-of-speech combination

$$\tilde{H}_k^{psc} = \sigma(R_{:, :, k}^{psc} H W_k + b_k) \quad (26)$$

(3) Tree-based distance

$$\tilde{H}_k^{tbd} = \sigma(R_{:, :, k}^{tbd} H W_k + b_k) \quad (27)$$

(4) Relative position distance

$$\tilde{H}_k^{rpd} = \sigma(R_{::,k}^{rpd}HW_k + b_k) \tag{28}$$

where $R_{::,k}^{sdt} \in R^{n \times n}$, $R_{::,k}^{psc} \in R^{n \times n}$, $R_{::,k}^{tbd} \in R^{n \times n}$ and $R_{::,k}^{rpd} \in R^{n \times n}$ said, respectively, according to the relationship between the first k type modeling of adjacency matrix; W_k and b_k are to learn the weights and bias; σ is the activation function. Then, the adjacency matrix modeled from all relationship types is aggregated to obtained through:

$$\hat{H}^{sdt} = f(\tilde{H}_1^{sdt}, \tilde{H}_2^{sdt}, \dots, \tilde{H}_m^{sdt}) \tag{29}$$

$$\hat{H}^{psc} = f(\tilde{H}_1^{psc}, \tilde{H}_2^{psc}, \dots, \tilde{H}_m^{psc}) \tag{30}$$

$$\hat{H}^{tbd} = f(\tilde{H}_1^{tbd}, \tilde{H}_2^{tbd}, \dots, \tilde{H}_m^{tbd}) \tag{31}$$

$$\hat{H}^{rpd} = f(\tilde{H}_1^{rpd}, \tilde{H}_2^{rpd}, \dots, \tilde{H}_m^{rpd}) \tag{32}$$

where f represents the average pooling function.

Similarly, input adjacency matrix R^{ba} from the BiLSTM-biaffine attention model into the graph convolutional neural network to obtain new node representation H^{ba} through:

$$\tilde{H}_k^{ba} = \sigma(R_{::,k}^{ba}HW_k + b_k) \tag{33}$$

where $R_{::,k}^{ba} \in R^{n \times n}$ are said to be according to the relationship between the first k type modeling by adjacency matrix; W_k and b_k are to learn the weights and bias; and σ is the activation function. Then, the adjacency matrix modeled by all relationship types is aggregated to obtain:

$$\hat{H}^{ba} = f(\tilde{H}_1^{ba}, \tilde{H}_2^{ba}, \dots, \tilde{H}_m^{ba}) \tag{34}$$

At the same time, input adjacency matrix R^{gat} from BiLSTM-biaffine attention model into the graph convolutional neural network to obtain new node representation H^{gat} through:

$$\tilde{H}_k^{gat} = \sigma(R_{::,k}^{gat}HW_k + b_k) \tag{35}$$

$$\hat{H}^{gat} = f(\tilde{H}_1^{gat}, \tilde{H}_2^{gat}, \dots, \tilde{H}_m^{gat}) \tag{36}$$

Finally, the final result is obtained through:

$$H = f(\hat{H}^{gat}, \hat{H}^{ba}, \hat{H}^{sdt}, \hat{H}^{psc}, \hat{H}^{tbd}, \hat{H}^{rpd}) \tag{37}$$

$$R = R^{ba} \oplus R^{sdt} \oplus R^{psc} \oplus R^{tbd} \oplus R^{rpd} \tag{38}$$

where f represents the average pooling function.

3.7. Refining Strategy and Predict

Firstly, assuming that w_i denotes the aspect terms and w_j denotes the opinion terms, the word (w_i, w_j) is generally predicted for affective polarity, i.e., POS, NEU, or NEG. Then, the introduction of r_{ii} and r_{jj} to refine the word (w_i, w_j) is calculated by:

$$s_{ij} = h_i \oplus h_j \oplus r_{ij} \oplus r_{ii} \oplus r_{jj} \tag{39}$$

Passing through Equation (40) to produce the probability distribution of the label p_{ij} :

$$p_{ij} = \text{softmax}(W_p s_{ij} + b_p) \tag{40}$$

where softmax is an activation function, W_p and b_p are learnable weights and biases, respectively.

3.8. Loss Function

Each of the four language features introduced in this chapter sets constraints L_{sdt} , L_{psc} , L_{tbd} , and L_{rpd} , which are obtained through:

$$L_{sdt} = - \sum_i^n \sum_j^n \sum_{c \in C} \mathbb{I}(y_{ij} = c) \log(r_{ij|c}) \tag{41}$$

$$L_{psc} = - \sum_i^n \sum_j^n \sum_{c \in C} \mathbb{I}(y_{ij} = c) \log(r_{ij|c}) \tag{42}$$

$$L_{tbd} = - \sum_i^n \sum_j^n \sum_{c \in C} \mathbb{I}(y_{ij} = c) \log(r_{ij|c}) \tag{43}$$

$$L_{rpd} = - \sum_i^n \sum_j^n \sum_{c \in C} \mathbb{I}(y_{ij} = c) \log(r_{ij|c}) \tag{44}$$

Set a constraint on the adjacency matrix obtained by the biaffine attention mechanism L_{ba} :

$$L_{ba} = - \sum_i^n \sum_j^n \sum_{c \in C} \mathbb{I}(y_{ij} = c) \log(r_{ij|c}) \tag{45}$$

Cross-entropy loss is used for prediction results L_p , which is obtained through:

$$L_p = - \sum_i^n \sum_j^n \sum_{c \in C} \mathbb{I}(y_{ij} = c) \log(p_{ij|c}) \tag{46}$$

where \mathbb{I} represents the indicator function; y_{ij} is the true value of the word (w_i, w_j); c represents a set of the relationship.

The final objective function L is obtained through:

$$L = L_p + \alpha L_{ba} + \beta (L_{sdt} + L_{psc} + L_{tbd} + L_{rpd}) \tag{47}$$

where the coefficients α and β are used to adjust the impact of the loss resulting from the corresponding constraint on the total loss.

4. Experiments and Analysis

4.1. Datasets

In part of datasets, we select datasets that are widely used in aspect level sentiment analysis tasks, which are derived from SemEval ABSA challenges (Pontiki et al., 2014, 2015, 2016) [25–27]. In addition, the first dataset was modified by Wu et al. [17] according to the ASTE task, and the second dataset was annotated by Xu et al. [16], whilst Peng et al. [7] further proposed a corrected version of the dataset. The above two datasets are labeled D_1 and D_2 , respectively. The options include Laptop 14, Restaurant 14, Restaurant 15, and Restaurant 16. Statistics for these two groups of datasets are shown in Table 3.

Table 3. Data information of datasets.

Dataset		14res		14lap		15res		16res	
		#S	#T	#S	#T	#S	#T	#S	#T
D_1	train	1259	2356	899	1452	603	1038	863	1421
	dev	315	580	225	383	151	239	216	348
	test	493	1008	332	547	325	493	328	525
D_2	train	1266	2338	906	1460	605	1013	857	1394
	dev	310	577	219	346	148	249	210	339
	test	492	994	328	543	322	485	326	514

4.2. Experimental Parameter Setting

The aspect-level sentiment analysis model based on fusion graph neural network proposed in this paper is designed and written using Python programming language. The batch size is set to 8 and the number of model iterations is 100. The dimensional settings for GAT and GCN are established at 500, with a learning rate of 10^{-5} .

4.3. Baselines

To verify the effectiveness of the proposed BiLSTM-BGAT-GCN model, we conducted comparisons with several leading baseline models, with the results presented as follows:

Peng-two-stage [7] and its related variants are a series of models in the field of sentiment analysis, which extract emotional pairs, opinion terms, and construct triplets through a two-stage process. Among them, Peng-two-stage+IOG [17] combines IOG technology to enhance performance; GTS-CNN [17] and GTS-BiLSTM [17] incorporate grid tagging schemes with CNN and BiLSTM techniques for feature extraction; Dual-MRC [8] constructs two machine reading comprehension problems to extract relevant information; IMN + IOG [17] combines the advantages of interactive multi-task learning networks and IOG; CMLA [7] uses the attention mechanism for collaborative extraction. RINANTE [7] is based on word dependency in sentences and extraction rules; Li-unified-R [7] is an improved version of its original OE component; OTE-MTL [15] achieved triplet extraction through multi-task learning; JET-BERT [16] simultaneously extracts triplets using a position-aware tagging approach; BMRC [8] converts the task into a machine reading comprehension task; and EMC-GCN [24] processes words and edges in sentences through multichannel graphs. These models possess unique characteristics and advantages, providing sophisticated and efficient solutions for complex text sentiment analysis.

4.4. Main Results

This section presents the main experimental findings, as illustrated in Table 4 and Figure 9. Drawing on the data displayed in the table, the following conclusions can be made:

Under the $F1$ metric, our proposed BiLSTM-BGAT-GCN model achieves scores of 73.36%, 58.05%, 58.61%, and 69.14% on the D_1 datasets, respectively. For the D_2 datasets, the performance results are 71.55%, 60.83%, 58.81%, and 66.18%, respectively. A careful review of Table 3 clearly demonstrates that our BiLSTM-BGAT-GCN model outperforms all existing pipeline models, including Peng-two-stage + IOG, IMN + IOG, CMLA, RINANTE, Li-unified-R, and Peng-two-stage. Moreover, it surpasses all MRC-based methods, such as Dual-MRC and BMRC.

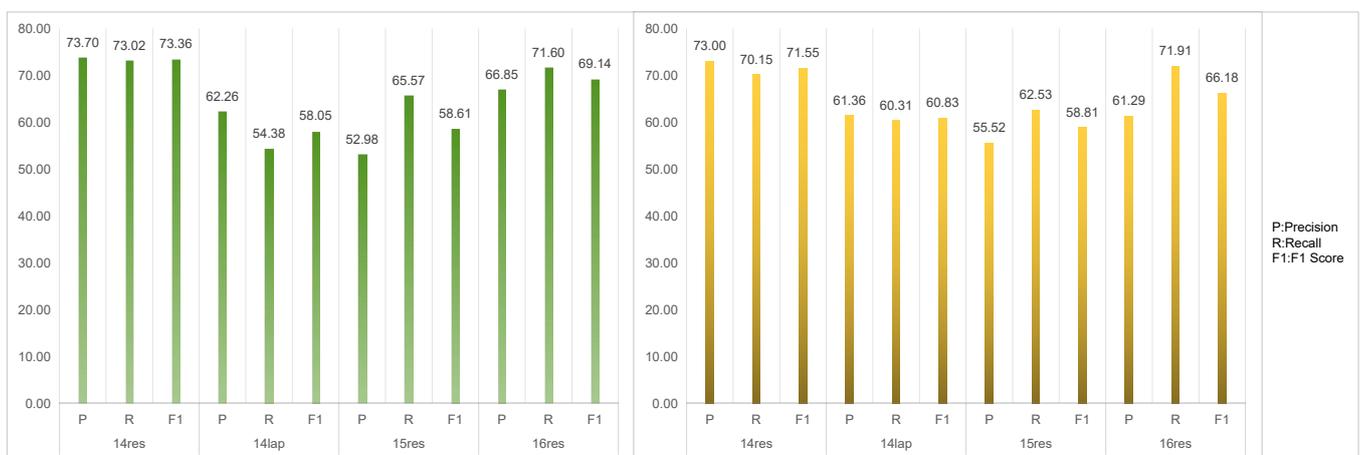


Figure 9. The results on D_1 and D_2 datasets.

Further comparison with end-to-end methods reveals that our model also exhibits a superior performance, outdoing GTS-CNN, GTS-BiLSTM, OTE-MTL, and JET-BERT. Not

only that, but in comparison with the best baseline model EMC-GCN, our BiLSTM-BGAT-GCN model also shows significant progress. On the D_1 dataset, it secures improvements of 1.65% (73.36–71.71%), 1.25% (58.05–56.80%), (58.61–58.61%), and 0.45% (69.14–68.69%) in $F1$ scores across the four sub-datasets. On the D_2 dataset, the performances of the BiLSTM-BGAT-GCN model are only slightly below the EMC-GCN on $F1$ in Restaurant 15. Furthermore, the remaining performances of the BiLSTM-BGAT-GCN model in Restaurant14, Laptop14, and Restaurant16 also achieves 1.78% (71.55–69.77%), 5.23% (60.83–55.60%), 0.08% (66.18–66.10%) increases, respectively.

Table 4. Experimental results of each model (%).

	Model	14res			14lap			15res			16res		
		<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
D_1	Peng-two-stage + IOG	58.89	60.41	59.64	48.62	45.52	47.02	51.70	46.04	48.71	59.25	58.09	58.67
	GTS-CNN	70.79	61.71	65.94	55.93	47.52	51.38	60.09	53.57	56.64	62.63	66.98	64.73
	GTS-BiLSTM	67.28	61.91	64.49	59.42	45.13	51.30	63.26	50.71	56.29	66.07	65.05	65.56
	Dual-MRC	71.55	69.14	70.32	57.39	53.88	55.58	63.78	51.87	57.21	68.60	66.24	67.40
	IMN + IOG	59.57	63.88	61.65	49.21	46.23	47.68	55.24	52.33	53.75	-	-	-
	EMC-GCN	71.15	72.29	71.71	56.55	57.06	56.80	59.21	58.01	58.61	67.98	69.41	68.69
	BiLSTM-BGAT-GCN	73.70	73.02	73.36	62.26	54.38	58.05	52.98	65.57	58.61	66.85	71.60	69.14
D_2	CMLA	39.18	47.13	42.79	30.09	36.92	33.16	34.56	39.84	37.01	41.34	42.10	41.72
	RINANTE	31.42	39.38	34.95	21.71	18.66	20.07	29.88	30.06	29.97	25.68	22.30	23.87
	Li-unified-R	41.04	67.35	51.00	40.56	44.28	42.34	44.72	51.39	47.82	37.33	54.51	44.31
	Peng-two-stage	43.24	63.66	51.46	37.38	50.38	42.87	48.07	57.51	52.32	46.96	64.24	54.21
	OTE-MTL	62.00	55.97	58.71	49.53	39.22	43.42	56.37	40.94	47.13	62.88	52.10	56.96
	JET-BERT	70.56	55.94	62.40	55.39	47.33	51.04	64.45	51.96	57.53	70.42	58.37	63.83
	BMRC	75.61	61.77	67.99	70.55	48.98	57.82	68.51	53.40	60.02	71.20	61.08	65.75
	EMC-GCN	67.40	72.33	69.77	57.00	54.90	55.60	64.01	61.24	62.59	63.93	68.42	66.10
BiLSTM-BGAT-GCN	73.00	70.15	71.55	61.36	60.31	60.83	55.52	62.53	58.81	61.29	71.91	66.18	

P, *R*, and *F1* represent precision, recall, and *F1* score, respectively. Precision refers to the proportion of samples that are actually classified as positive categories among all samples classified as positive categories. Recall refers to the proportion of samples that have been successfully predicted by the model as positive categories among all actual positive category samples. *F1* score is the harmonic mean of precision and recall, which combines the information of precision and recall, helping us find a balance between precision and recall.

In conclusion, as demonstrated by the results, our proposed BiLSTM-BGAT-GCN model has proven to effectively perform the ASTE task.

To provide a more intuitive demonstration of the effectiveness of our improvements, we compared the two following models, as illustrated in Table 5 and Figure 10, where BGAT-GCN represents the model without the improved biaffine attention mechanism. The experimental results show that, by introducing bidirectional long short-term memory to improve the biaffine attention mechanism, the model achieved significant improvements in three key indicators: precision (*P*), recall (*R*), and *F1* score. This improvement measure not only effectively enhances the overall performance of the model, but also further proves the effectiveness and practicality of this improvement measure in model optimization.

Table 5. Experimental results of each model (%).

	Model	14res			14lap			15res			16res		
		<i>P</i>	<i>R</i>	<i>F1</i>									
D_1	BGAT-GCN	72.11	72.70	72.40	56.96	57.80	57.38	55.74	61.06	58.28	65.05	71.89	68.30
	BiLSTM-BGAT-GCN	73.70	73.02	73.36	62.26	54.38	58.05	52.98	65.57	58.61	66.85	71.60	69.14
D_2	BGAT-GCN	66.64	73.96	70.11	61.67	58.60	60.10	51.76	66.60	58.25	63.71	71.54	67.40
	BiLSTM-BGAT-GCN	73.00	70.15	71.55	61.36	60.31	60.83	55.52	62.53	58.81	61.29	71.91	66.18

P, *R* and *F1* represent precision, recall, and *F1* score, respectively.

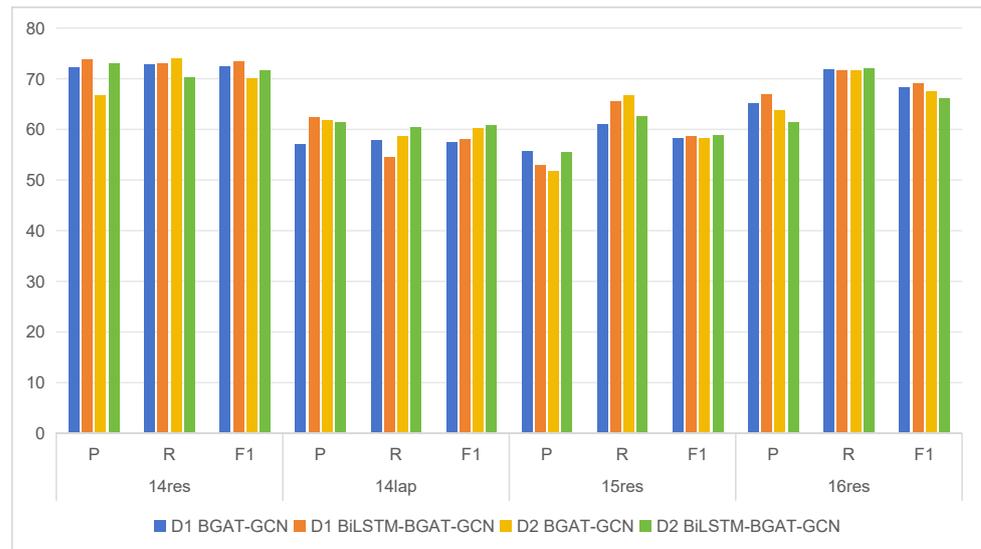


Figure 10. Comparison results on D1 and D2 datasets.

To delve deeper into the influence of each linguistic feature on our task, we carefully selected a sentence sample and depicted the distribution of attention weights across each word, as exemplified in Figure 11. This figure intricately displays each row as the visualization derived from the GCN for every distinct linguistic feature, while every column unravels the intricacies of how a word is interpreted across the four linguistic features. Initially, it is clear that GCN’s focus is variably distributed among the sentence’s words. Take, for instance, Figure 11, where the pivotal word for the branch $F_{sdt+GCN}^G$ is “is”, contrasting starkly with its positioning as the final word in the attention sequence ordered by $F_{psc+GCN}^G$, which, in turn, allocates a substantial attention weight to “this”. Similar patterns are observable in $F_{tbd+GCN}^G$ and $F_{rpd+GCN}^G$. It is further intriguing that should a branch overlook a specific word, say “is” in both $F_{psc+GCN}^G$ and $F_{rpd+GCN}^G$, an alternate branch of the GCN compensates by amplifying its attention on the word, as seen with $F_{sdt+GCN}^G$ and $F_{tbd+GCN}^G$. This interplay not only underscores the essence of feature integration but also highlights the adeptness of our model at synthesizing diverse branch features to enrich text representation, thereby significantly bolstering the performance of ASTE task.

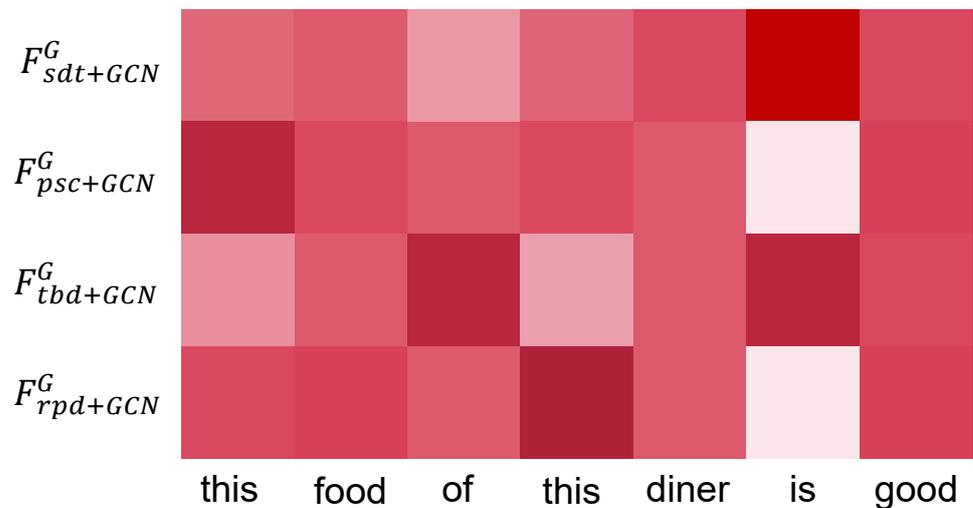


Figure 11. Attention distribution on word.

5. Conclusions and Future Work

This paper introduces an innovative end-to-end model, BiLSTM-BGAT-GCN, aimed at addressing the task of ASTE. By integrating GCN and GAT, and introducing an improved biaffine attention mechanism, this model significantly optimizes the capability to process the relationships between aspect and opinion terms within texts. It has enabled a deeper understanding of the complex relationships between aspect and opinion terms in texts, while also increasing the accuracy and efficiency of sentiment triplet extraction. Moreover, by incorporating multiple linguistic features, the model has further enhanced its comprehensive understanding of texts, making sentiment triplet extraction even more precise. This innovation not only provides an effective technical approach to solving the ASTE task but also offers a new perspective for sentiment analysis research in the field of natural language processing. Experimental results demonstrate that the proposed BiLSTM-BGAT-GCN model achieves outstanding performance across multiple standard datasets, showing significant improvements, especially in capturing complex textual relationships and enhancing sentiment analysis accuracy, compared to existing methods. Future research will explore the further optimization of model parameters and structures to adapt to a more diverse range of text types and sentiment analysis tasks. Additionally, considering the richness of multimodal data, combining textual information with other modalities such as images and videos to more comprehensively understand and analyze sentiments is also a direction worth further investigation.

Author Contributions: Conceptualization, methodology, validation, Y.P. and J.-X.Z.; writing—original draft preparation, Y.P.; writing—review and editing, J.-X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62103093, the National Key Research and Development Program of China under Grant 2022YFB3305905, the Xingliao Talent Program of Liaoning Province of China under Grant XLYC2203130, the Natural Science Foundation of Liaoning Province of China under Grant 2023-MS-087, and the Fundamental Research Funds for the Central Universities of China under Grants N2108003.

Data Availability Statement: Data are contained within the article.

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

References

1. Zhang, X.F.; Chen, S.N.; Zhang, J.X. Adaptive sliding mode consensus control based on neural network for singular fractional order multi-agent systems. *Appl. Math. Comput.* **2022**, *434*, 127442. [\[CrossRef\]](#)
2. Zhang, J.-X.; Yang, T.; Chai, T. Neural network control of underactuated surface vehicles with prescribed trajectory tracking performance. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–14. [\[CrossRef\]](#)
3. Zhang, X.F.; Driss, D.; Liu, D.Y. Applications of fractional operator in image processing and stability of control systems. *Fractal Fract.* **2023**, *7*, 359. [\[CrossRef\]](#)
4. Yan, H.; Zhang, J.X.; Zhang, X.F. Injected infrared and visible image fusion via L1 decomposition model and guided filtering. *IEEE Trans. Comput. Imaging* **2022**, *8*, 162–173. [\[CrossRef\]](#)
5. Zhang, X.F.; Dai, L.W. Image enhancement based on rough set and fractional order differentiator. *Fractal Fract.* **2020**, *6*, 214. [\[CrossRef\]](#)
6. Zhang, X.F.; Liu, R.; Wang, Z.; Ren, J.X.; Gui, L. Adaptive fractional image enhancement algorithm based on rough set and particle swarm optimization. *Fractal Fract.* **2022**, *6*, 100. [\[CrossRef\]](#)
7. Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; Si, L. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8600–8607. [\[CrossRef\]](#)
8. Mao, Y.; Shen, Y.; Yu, C.; Cai, L. A joint training dual-mrc framework for aspect based sentiment analysis. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 13543–13551. [\[CrossRef\]](#)
9. Chen, S.; Wang, Y.; Liu, J.; Wang, Y. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 12666–12674. [\[CrossRef\]](#)
10. Zhang, W.; Li, X.; Deng, Y.; Bing, L.; Lam, W. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*; Online; Association for Computational Linguistics: New York, NY, USA, 2021; pp. 504–510.

11. Yan, H.; Dai, J.; Qiu, X.; Zhang, Z. A unified generative framework for aspect-based sentiment analysis. *arXiv* **2021**, arXiv:2106.04300.
12. Hsu, T.-W.; Chen, C.-C.; Huang, H.-H.; Chen, H.-H. Semantics-preserved data augmentation for aspect-based sentiment analysis. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 4417–4422.
13. Mukherjee, R.; Nayak, T.; Butala, Y.; Bhattacharya, S.; Goyal, P. PASTE: A tagging-free decoding framework using pointer networks for aspect sentiment triplet extraction. *arXiv* **2021**, arXiv:2110.04794.
14. Fei, H.; Ren, Y.; Zhang, Y.; Ji, D. Nonautoregressive encoder–decoder neural framework for end-to-end aspect-based sentiment triplet extraction. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 5544–5556. [[CrossRef](#)] [[PubMed](#)]
15. Zhang, C.; Li, Q.; Song, D.; Wang, B. A multi-task learning framework for opinion triplet extraction. *arXiv* **2020**, arXiv:2010.01512.
16. Xu, L.; Li, H.; Lu, W.; Bing, L. Position-aware tagging for aspect sentiment triplet extraction. *arXiv* **2020**, arXiv:2010.02609.
17. Wu, Z.; Ying, C.; Zhao, F.; Fan, Z.; Dai, X.; Xia, R. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. *arXiv* **2020**, arXiv:2010.04640.
18. Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; Sima'an, K. Graph convolutional encoders for syntax-aware neural machine translation. *arXiv* **2017**, arXiv:1704.04675.
19. Guo, Z.; Zhang, Y.; Lu, W. Attention guided graph convolutional networks for relation extraction. *arXiv* **2019**, arXiv:1906.07510.
20. Chen, C.; Teng, Z.; Zhang, Y. Inducing target-specific latent structures for aspect sentiment classification. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 5596–5607.
21. Li, R.; Chen, H.; Feng, F.; Ma, Z.; Wang, X.; Hovy, E. Dual graph convolutional networks for aspect-based sentiment analysis. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 6319–6329.
22. Binxuan, H.; Carley, K. Syntax-aware aspect level sentiment classification with graph attention networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 5469–5477.
23. Wang, K.; Shen, W.; Yang, Y.; Quan, X.; Wang, R. Relational graph attention network for aspect-based sentiment analysis. *arXiv* **2020**, arXiv:2004.12362.
24. Chen, H.; Zhai, Z.; Feng, F.; Li, R.; Wang, X. Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 2974–2985.
25. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Manandhar, S. SemEval-2014 task 4: Aspect based sentiment analysis. In Proceedings of the International Workshop on Semantic Evaluation, San Diego, CA, USA, 16–17 June 2014; pp. 27–35.
26. Papageorgiou, H.; Androutsopoulos, I.; Galanis, D.; Pontiki, M.; Manandhar, S. SemEval-2015 task 12: Aspect based sentiment analysis. In Proceedings of the 9th International Workshop on Semantic Evaluation, Denver, CO, USA, 4–5 June 2015; pp. 486–495.
27. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Al-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O. Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the ProWorkshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 19–30.

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