electronics

MDPI

Article

Hybrid Graph Neural Network-Based Aspect-Level Sentiment Classification

Hongyan Zhao [1,*], Cheng Cui [1] and Changxing Wu [2]

[1] College of Engineering, Yang-En University, Quanzhou 362014, China
[2] School of Information and Software Engineering, East China Jiaotong University, Nanchang 330013, China
* Correspondence: hyzhao@yeu.edu.cn

Abstract: Aspect-level sentiment classification has received more and more attention from both academia and industry due to its ability to provide more fine-grained sentiment information. Recent studies have demonstrated that models incorporating dependency syntax information can more effectively capture the aspect-specific context, leading to improved performance. However, existing studies have two shortcomings: (1) they only utilize dependency relations between words, neglecting the types of these dependencies, and (2) they often predict the sentiment polarity of each aspect independently, disregarding the sentiment relationships between multiple aspects in a sentence. To address the above issues, we propose an aspect-level sentiment classification model based on a hybrid graph neural network. The core of our model involves constructing several hybrid graph neural network layers, designed to transfer information among words, between words and aspects, and among aspects. In the process of information transmission, our model takes into account not only dependency relations and their types between words but also sentiment relationships between aspects. Our experimental results based on three commonly used datasets demonstrate that the proposed model achieves a performance that is comparable to or better than recent benchmark methods.

Keywords: aspect-level sentiment classification; hybrid graph neural network; dependency syntax information; sentiment relationships

check for updates

Citation: Zhao, H.; Cui, C.; Wu, C. Hybrid Graph Neural Network-Based Aspect-Level Sentiment Classification. *Electronics* **2024**, *13*, 3263. https://doi.org/10.3390/electronics13163263

Academic Editor: Hyunjin Park

Received: 15 July 2024
Revised: 12 August 2024
Accepted: 15 August 2024
Published: 17 August 2024

1. Introduction

Sentiment analysis has a wide range of applications across various industries and fields—for example, consumer feedback analysis [1] and public opinion analysis [2,3]. As a subtask of sentiment analysis, aspect-level sentiment classification (ASC) aims to automatically distinguish the sentiment polarity of aspects in text [4], which is one of the research hotspots in the field of natural language processing. Based on the sentence shown in Figure 1, given the aspects 'food', 'service', and 'environment', aspect-level sentiment classification models should predict the corresponding polarities as positive, negative, and negative, respectively. An aspect typically refers to an entity or an attribute of an entity mentioned in text, which can be represented by a word or a text segment. Aspect-level sentiment classification offers finer granularity of sentiment information compared to sentence-level and document-level classifications. It has garnered increasing attention from academia and industry.

In recent years, aspect-level sentiment classification models based on deep learning have outperformed traditional models based on artificial features by automatically learning contextual information related to a given aspect in a sentence [5–9]. Among these models, the attention mechanism-based ones are noted for their simplicity and effectiveness, receiving considerable attention [10,11]. This type of model can more accurately capture contextual information related to a specific aspect in a sentence through attention mechanisms, improving sentiment polarity judgment. While attention-based models have historically achieved excellent performance, they can often be susceptible to noise. For example, assuming that the given aspect is 'service' in Figure 1, attention-based models may consider both 'good' and 'horrific' as its relevant contexts, making it challenging to accurately predict sentiment polarity. To address the above issue, researchers have endeavored to integrate dependency syntax information into aspect-level sentiment classification models. This integration aims to enhance the models' ability to capture context information related to specific aspects, thereby improving performance. For example, Zhang et al. [12], Sun et al. [13], and Wang et al. [14] leveraged the dependency syntax information of sentences through graph convolutional networks [15] and achieved notable performance improvements. From the example depicted in Figure 1, we can clearly observe the critical role of dependency syntax information. The aspect 'service' in the sentence depends on the word 'horrific', which conveys its negative polarity.

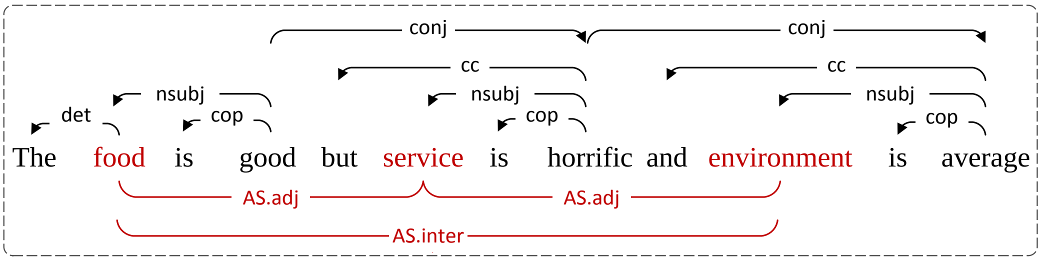
conj
conj
cc
cc
nsubj
nsubj
nsubj
det
cop
cop
cop
The food is good but service is horrific and environment is average
AS.adj
AS.adj
AS.inter

Figure 1. An example with dependency relations between words (above the sentence) and adjacency relationships between aspects (below the sentence).

Promising performance has been achieved by existing aspect-level sentiment classification models that incorporate dependency syntax information. However, these methods suffer from two shortcomings: (1) They only leverage the dependency relations between words in the dependency syntax tree, disregarding the categories of these relations. As shown in Figure 1, the words dependent on the aspect 'food' include 'The' and 'good'. Without considering the categories of dependency relations, classification models still struggle to determine which word is crucial for 'food'. When the related words are uncommon, knowing the categories of dependency relations becomes crucial for classification models to distinguish the importance of words. Therefore, a robust aspect-level sentiment classification model should simultaneously consider both the dependency relations between words and their categories. (2) Existing models typically predict the sentiment polarity of each aspect independently, overlooking the sentiment relationships between aspects within a sentence. Specifically, these models first decompose multiple aspects within a sentence. They then take the sentence and one aspect at a time as the input to predict the sentiment polarity for each aspect. Although the above strategy simplifies the design of ASC models, it fails to capture the sentiment relationships between multiple aspects within a sentence. For example, for the two aspects 'food' and 'service' shown in Figure 1, it is easy to infer that their sentiment polarities are opposite because there is a conjunction 'but' between them. When the sentiment polarity of an aspect is expressed implicitly—for example, 'food and service are horrific'—it becomes crucial for ASC models to understand the sentiment relationships between multiple aspects in a sentence in order to infer the sentiment polarities of these aspects.

To address the two aforementioned issues, we propose an aspect-level sentiment classification model based on hybrid graph neural networks. Specifically, we begin by constructing a hybrid graph based on both the dependency syntax tree and the adjacency relations between aspects. The constructed hybrid graph includes two types of nodes: words and aspects. It incorporates two types of edges: the dependency relations between words or between words and aspects (represented above the sentence in Figure 1) and the adjacency relations between aspects (represented below the sentence). Then, building upon the context coding layer (Section 3.1), we introduce multiple layers of hybrid graph neural networks to facilitate information transfer among words, between words and aspects, and among aspects (Section 3.2). In the process of information transmission, we consider not only the dependency relations between words and their categories but also the sentiment relationships between aspects. Finally, the classification layer takes the semantic vector representations of aspects learned by the hybrid graph neural networks as the input and jointly predicts the sentiment polarities of all aspects in a sentence (Section 3.3).

Existing models [5,16] use LSTM or graph neural networks to capture the sentiment relationships between multiple aspects within a sentence. However, a limitation of their models is its tendency to predict multiple aspects in a sentence as having the same sentiment polarity. By modeling both dependency types between words and adjacency relations between aspects, our model can place greater emphasis on connectives like *and* and *but*. This enables our model to effectively capture whether multiple aspects have the same or opposite sentiment polarities. Our experimental results based on three commonly used datasets show that our proposed model achieves performance comparable to or better than recent benchmark methods.

2. Related Work

This work is primarily related to research on aspect-level sentiment classification based on deep learning and graph neural networks, which is introduced below.

2.1. Aspect-Level Sentiment Classification

The key problem of aspect-level sentiment classification (ASC) is how to effectively identify the context information related to a specific aspect in the text. In recent years, models based on deep learning, particularly those utilizing attention mechanisms, have achieved promising performance in ASC. For example, Tang et al. [17] introduced two LSTM networks to capture the context information of an aspect from left to right and right to left, respectively. Wang et al. [10] first concatenated the representations of a given aspect and each word in the sentence as the input to an LSTM network to learn the aspect-specific word representations. They then used attention mechanisms to learn the final representation for classification. Ma et al. [11] adopted two attention mechanism modules to separately learn the representation of an aspect and its corresponding context, which were then combined for sentiment classification. Sun et al. [18] and Du and Liu [19] developed multi-head attention mechanisms and helical attention networks for aspect-level sentiment classification, respectively, achieving better performance on multiple datasets.

To better capture the aspect-specific context, researchers have integrated the dependency syntax information of sentences into ASC models, further improving the performance. For example, Huang and Carley [20] leveraged dependency syntax information via a graph attention network for ASC. Wang et al. [21] defined a variant of a dependency syntax tree for ASC to construct a graph neural network, achieving promising results. Tang et al. [22] combined the separately learned semantic vector representations via a transformer layer [23] and a graph neural network layer constructed on the dependency syntax tree in an interactive manner. To avoid possible errors caused by a single dependency syntax parser, Hou et al. [24] combined the dependency trees from multiple parsers to construct a graph neural network layer. Instead of incorporating syntactic structures of sentences, Ma et al. [25] made full use of semantic structures (abstract meaning representation, AMR). Wang et al. [14] developed a dual graph network to capture both the syntactic and semantic information. Additionally, both Zhu et al. [6] and Gu et al. [7] incorporated external knowledge via graph neural networks.

Most of the methods mentioned above predict the sentiment polarity of each aspect in a sentence independently, thereby overlooking the sentiment relationships between multiple aspects within the same sentence. To address this problem, Hazarika et al. [16] learned the contextualized representation of each aspect separately. These representations were then used as inputs to an LSTM network to capture the sentiment relationships between them, aiming for improved classification. Hu et al. [26] attempted to retrieve the different context of multiple aspects in a sentence using a constraint attention mechanism. Zhao et al. [5] modeled the sentiment dependency relations between aspects using a graph neural network. Han et al. [27] proposed a gate-based network to dynamically merge the sentiment of words and inter-aspect dependency.

Previous studies have achieved impressive performance improvements in aspect-level sentiment classification. The key differences between our method and existing approaches are summarized as follows: (1) Our model simultaneously considers the relationships among words, between words and aspects, and among aspects, whereas existing methods typically focus on only some of these relationships. (2) To capture these relationships, we design a hybrid graph neural network that includes both type-aware dependency edges and adjacency edges. Existing methods usually do not consider types of word dependencies. (3) Despite its simplicity, our method achieves comparable or superior performance compared to recent baseline models.

2.2. Graph Neural Networks

Graph neural networks (GNNs) [28] are deep learning models commonly used for processing graph-structured data. By facilitating information transmission between nodes in a graph, GNNs effectively model semantic relationships between nodes. In recent years, GNNs have been widely employed in the field of natural language processing. For example, Yao et al. [29] adapted a GNN for text classification and achieved state-of-the-art results at that time. They constructed a graph with nodes representing words and documents to capture relationships between documents and words, as well as the relationships between words. Yin et al. [30] fused information from text and images using a GNN for multi-modal machine translation. Wu et al. [31] employed a GNN to encode hierarchical label information for multi-level discourse relation recognition. Yan et al. [32] constructed several simpler local hyper-graphs to capture nested named entities instead of using a complex global hyper-graph. Mao et al. [33] developed a hierarchical graph fusion network for multi-party dialogue discourse parsing to integrate diverse contexts. Many existing studies have confirmed that GNNs can effectively model relationships between nodes in a graph, thereby improving the learning of semantic representations of nodes.

Our work is also related to the language and visual entity relationship graph model for agent navigation [34], which effectively captures both the inter-modal relationships between text and vision and the intra-modal relationships among visual entities. The main distinctions between these two methods are as follows: (1) Constructing an appropriate graph is critical for GNN-based models. These authors started by building a language attention graph and then a language-conditioned visual graph. In contrast, we construct a hybrid graph that integrates the dependency tree of a sentence with its aspects. (2) The two methods differ in how they propagate information between nodes. These authors' method relies on information fusion mechanisms for information transfer, while our method combines attention and gated mechanisms for this purpose.

3. Method

A sentence $S = x_1, \ldots, x_i, \ldots, x_n$ may contain m aspects, where each aspect can be a word or a text segment within the sentence, and n represents the total number of words in the sentence. Most previous studies take the sentence S and one aspect as the input at a time to predict the sentiment polarity of that aspect, overlooking the sentiment relationships between multiple aspects with the sentence. Differing from these methods, our proposed model is based on a hybrid graph neural network, which takes both the sentence and all aspects contained within it as the input to jointly infer the sentiment polarities of these aspects. The main contribution of our work is the development of a hybrid graph neural network that fully leverages both the dependency syntax information and the relationships between aspects.

As shown in Figure 2, the proposed aspect-level sentiment classification model comprises the input layer, the encoding layer, the hybrid graph neural network layers, and the classification layer. The input layer transforms words in a sentence into low-dimensional semantic vectors. The encoding layer then utilizes a BiLSTM network to further learn the contextualized representations of both words and aspects. The hybrid graph neural network layers first organize the aspects and words in a sentence into a hybrid graph. Subsequently, they capture relationships between words, between words and aspects, and among aspects. This approach enables the model to jointly learn improved semantic vector representations of multiple aspects within a sentence. Finally, the classification layer is responsible for generating the sentiment classification results. In the following subsections, we will elaborate on the input layer and the encoding layer (Section 3.1), the hybrid graph neural network layers (Section 3.2), and the classification layer, along with the loss function (Section 3.3).

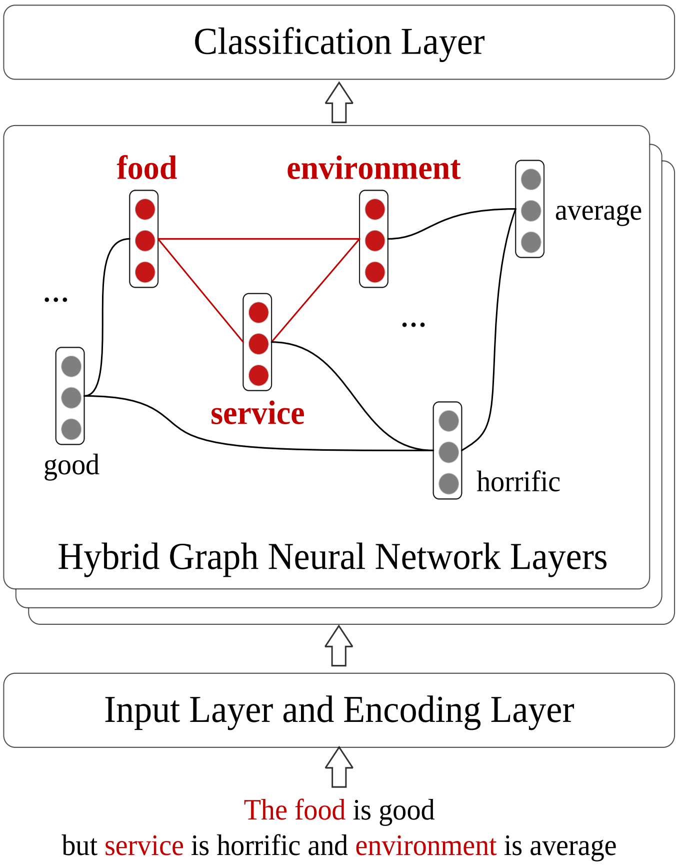
Classification Layer
food
environment
average
...
...
service
good
horrific
Hybrid Graph Neural Network Layers
Input Layer and Encoding Layer
The food is good
but service is horrific and environment is average

Figure 2. Hybrid graph neural network-based model for aspect-level sentiment classification.

3.1. Input Layer and Encoding Layer

Most natural language processing models based on deep learning typically represent words and other inputs as semantic vectors. Specifically, the input layer first converts the i-th word x_i in a sentence into the corresponding static word vector x_i^{glove}, according to the pretrained GloVe [35] embeddings. Then, based on the pretrained BERT [36], the i-th word x_i in the sentence is represented as the dynamic contextualized word embedding x_i^{bert}. Lastly, BIO tags, which are commonly used in sequence tagging, are introduced to mark the aspect words in a sentence. The corresponding tag vector of the i-th word x_i is denoted as x_i^{bio}. The tags B, I, and O represent the first word of an aspect, the subsequent words of an aspect, and words not in an aspect, respectively. Concatenating the above three types of information yields the complete semantic representation of a word, which serves as the input to the encoding layer:

$$w_i = [x_i^{glove}; x_i^{bert}; x_i^{bio}]. \tag{1}$$

The encoding layer further learns the representation of each word conditioned on its sentence-level context as follows:

$$h_1, \ldots, h_i, \ldots, h_n = \mathrm{BiLSTM}(w_1, \ldots, w_i, \ldots, w_n), \tag{2}$$

where BiLSTM refers to a bidirectional long short-term memory network.

An aspect in a sentence can be a single word or a text segment containing multiple words. For an aspect consisting of a single word, we directly use the word representation in Equation (2) as its semantic representation. For an aspect containing multiple words, we compute its semantic representation by averaging the word representations. For convenience, we denote the semantic representations of m aspects in a sentence as $\tilde{h}_1, \ldots, \tilde{h}_m$, respectively.

3.2. Hybrid Graph Neural Network Layers

We combine multiple aspects in a sentence, along with words not belonging to these aspects, into a hybrid graph to jointly learn better semantic representations of these aspects. By introducing the hybrid graph neural network layers, we consider not only the dependency relations of words and their categories but also the sentiment relationships between multiple aspects.

3.2.1. Construction of Hybrid Graph

For a sentence S, we construct a hybrid graph $G =< V, E >$, where G is an undirected graph, V is the set of nodes, and E is the set of edges in the graph. There are two types of nodes: aspect nodes and word nodes. Specifically, each aspect in the sentence corresponds to an aspect node $v_p, p \in [1, m]$ in the graph, and each word not in any aspect corresponds to a word node $v_p, p \in [m+1, m+c]$, where m is the number of aspects in the sentence and c is the number of words not in any aspect. As shown in Figure 3, the boxes denote aspect nodes, and the circles represent word nodes.

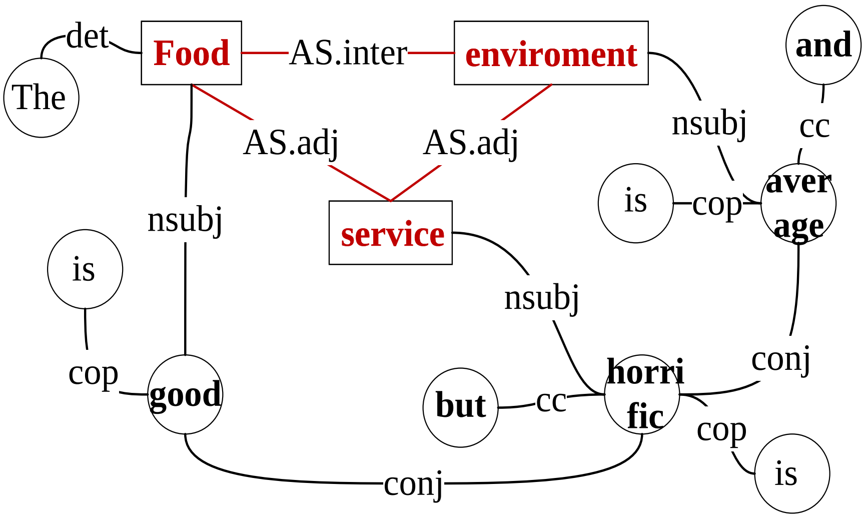

Figure 3. The corresponding hybrid graph for the sentence in Figure 1. The boxes denote aspect nodes and the circles represent word nodes.

A hybrid graph contains two kinds of edges: dependency edges and adjacency edges. Dependency edges exist between an aspect node and a word node, or between two word nodes. These dependency edges are derived from the dependency syntax tree of the sentence. Adjacency edges exist between aspect nodes and are based on the adjacency relationship between two aspects in a sentence.

The definition of a dependency edge is as follows: (1) A dependency edge exists between two word nodes if and only if there is a dependency relation between the two corresponding words in the dependency tree. (2) When there is a dependency relation between a word in an aspect and a word not in any aspect, a dependency edge exists between the corresponding aspect node and the word node. (3) The type of the dependency relation in the syntax tree is directly used as the category of the corresponding edge in the graph. There are two points that need to be noted. First, a hybrid graph does not contain the dependency edges between words within the same aspect because each aspect is treated as a cohesive unit. Second, there may be multiple dependency edges between an aspect node (which contains multiple words) and a word node. For convenience in processing, only the dependency edge with the smallest span (the shortest distance between two words involved) is retained.

The definition of an adjacency edge is as follows: (1) There is an adjacency edge between any two aspect nodes in the graph. (2) If two aspects are adjacent in the sentence, the category of the adjacency edge is 'adjacent' (AS.adj in Figure 1); otherwise, the category of the edge is 'interval' (AS.inter in Figure 1).

3.2.2. Information Transfer Mechanism

The hybrid graph neural network layers can effectively model the semantic relationships between nodes by transferring information among neighbor nodes, thereby enhancing the learning of semantic representations of nodes. To model the semantic relationships between nodes that are not directly connected, we stack L hybrid graph neural network layers. Specifically, we initialize the corresponding aspect nodes and word nodes in a graph with outputs from the encoding layer. The aspect nodes are initialized as the obtained semantic vectors $\tilde{h}_1, \ldots, \tilde{h}_m$, and the words' nodes are initialized as the learned contextualized vectors h_i as defined in Equation (2). For convenience, the initial representations of all nodes are uniformly referred to as $h_p^0, p \in [1, m+c]$. Each edge category is randomly initialized as a low-dimensional vector, which serves as the parameters of our model. Then, we calculate the semantic representations of nodes at the l-th ($l \in [1, L]$) hybrid graph neural network layer based on the representations at the $l-1$-th layer. Formally, we obtain the semantic representation of the node v_p at the l-th layer as follows:

$$\begin{aligned}
h_p^l &= \mathrm{Gate}_l(h_p^{l-1}, \sum_{q \in N(p)} \alpha_{p,q} \hat{h}_q^{l-1}), \\
\alpha_{p,q} &= \frac{(h_p^{l-1})^T \hat{h}_q^{l-1}}{\sum\limits_{q \in N(p)} \exp((h_p^{l-1})^T \hat{h}_q^{l-1})}, \\
\hat{h}_q^{l-1} &= \mathrm{FFN}_l([h_q^{l-1}; r_{p,q}]),
\end{aligned} \tag{3}$$

where $N(p)$ is the set of indices of nodes adjacent to the node v_p, $\alpha_{p,q}$ represents the weight obtained via the attention mechanism for the corresponding node, $r_{p,q}$ is the semantic vector of the edge between nodes v_p and v_q, FFN_l is a multi-layer feed-forward neural network used to calculate the edge category-enhanced node representation $\hat{h}_q^{l-1}$, and Gate_l is a gating mechanism used to fuse representations of the node itself and its neighbor nodes. Lastly, the node representations $h_p^L, p \in [1, m]$ at the L-th graph layer are used as the final semantic representations of m aspects in the sentence S, respectively.

In our hybrid graph neural network, the information transfer mechanism comprises three steps. First, edge category information is integrated with related nodes through a multi-layer feed-forward neural network (FFN_l), resulting in edge category-enhanced node representations. Next, we employ an attention mechanism to selectively aggregate information from neighboring nodes, ensuring that more significant information receives greater weight. Finally, a gated mechanism is utilized to combine a node's own information with that of its neighbors. This helps in filtering out noise and irrelevant information, leading to more accurate sentiment predictions. To some extent, the attention mechanism functions as a local information filter, while the gated mechanism serves as a global information filter, together improving the model's contextual understanding.

3.3. Classification Layer and Loss Function

Based on outputs of the last graph neural network layer, the classification layer calculates the final classification results as follows:

$$\bar{y}_p = \mathrm{softmax}(W_s h_p^L + b_s), p \in [1, m], \tag{4}$$

where $\bar{y}_p$ is the predicted result for the p-th aspect, and W_s and b_s are the parameters of the classification layer.

The commonly used cross-entropy loss function is adopted to guide the model training, defined as follows:

$$J_{ce} = -\sum_{p=1}^{m} E_{y_p}[\log \bar{y}_p], \tag{5}$$

where y_p is the ground-truth label (one-hot vector) of the p-th aspect and $E[]$ represents the expected value of the predicted result about the ground-truth label.

4. Experiments

4.1. Datasets and Settings

To verify the effectiveness of our proposed method, we compared it with benchmark methods based on three datasets: Laptop, Restaurant, and MAMS. The Laptop and Restaurant datasets are derived from review data provided by the SemEval 2014 competition [37]. The MAMS dataset [38] contains at least two aspects with different sentiment polarities in each sentence, making the dataset more challenging. The statistics of the three datasets are shown in Table 1. The officially provided Laptop and Restaurant datasets were split into only training and test sets, without validation sets. To determine the optimal hyperparameters, we randomly selected 10% instances from the official training set to create our validation set. The remaining 90% instances were used as our training data. For MAMS, we directly used the officially provided validation set.

In the Laptop dataset, more than 70% of sentences contain multiple aspects, and about 60% of sentences contain more than one aspect in the Restaurant dataset. For MAMS, each sentence contains multiple aspects. From the above statistical results, it is evident that modeling the sentiment relationships between multiple aspects in a sentence is necessary. The experimental results in the following subsections further verify this necessity.

Table 1. Statistics of experimental datasets.

Dataset	Split	Positive	Negative	Neutral
Laptop	Training	980	858	454
	Test	340	128	171
Restaurant	Training	2159	800	632
	Test	730	195	196
MAMS	Training	3380	5042	2764
	Validation	403	604	325
	Test	400	607	329

In our experiments, we used pretrained 300-dimensional GloVe word embeddings as static word vectors, which were not fine-tuned during model training. Additionally, we used the pretrained BERT-base model with 12 hidden layers to obtain dynamical word embeddings, which have 768 dimensions. The semantic vectors of BIO tags and the edge categories were randomly initialized from a uniform distribution between [−0.01, 0.01]. Other parameters in the model were set to their default initial values in PyTorch 1.8. To mitigate overfitting, we applied the dropout strategy [39] to the input layer. We used the Biaffine dependency parser [40] to obtain the dependency syntax tree of each sentence. The values of the hyper-parameters were determined based on achieving the best performance on the validation sets as depicted in Table 2.

Table 2. The values of hyper-parameters.

Hyper-Parameter	Value
Dimension of BIO tag vectors	50
Dimension of category vectors (d_1)	50
Sizes of BiLSTM hidden units (d_2)	300 (Forward), 300 (backward)
Dimension of node vectors	600
Hybrid graph layers (L)	2
Dropout rate (r)	0.3
Optimizer	Adam
Learning rate	0.00001
Batch size	32

As with most previous work for this task, we used accuracy (denoted as ACC) and macro-F_1 as the performance metrics. ACC is the ratio of correctly predicted instances (T) to the total number of instances (N). It is calculated as follows:

$$ACC = \frac{T}{N}. \tag{6}$$

The F_1 score is a harmonic mean of precision (P) and recall (R). For each class, it is calculated as follows:

$$F_1 = 2 \times \frac{P \times R}{P + R}, P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, \tag{7}$$

where TP, FP and FN are the true positives, false positives, and false negatives, respectively. In a multi-class setting, we first calculate the F_1 score for each class separately, treating that class as the positive class and all others as negatives. Then, the macro$-F_1$ is defined as:

$$\text{macro-}F_1 = \frac{1}{C}\sum_{i=1}^{C} F_1^i, \tag{8}$$

where C is the number of classes and F_1^i is the F_1 score for the i-th class.

We further utilized ROC (Receiver Operating Characteristic) and AUC (Area Under the Curve) metrics to evaluate the performance of our model [41]. Specifically, we calculated the macro-averaged ROC curve with the one vs. rest approach.

4.2. Results

We verified the effectiveness of our proposed hybrid graph neural network model based on the ASC task in two scenarios: HGNN (without the BERT word embeddings) and HGNN-BERT (with the BERT word embeddings). Specifically, we compared our method with the following two types of benchmark methods.

The first type of method only uses static word embeddings, such as GloVe, and includes the models that utilize the dependency syntax information as follows:

- IAN [11]: Two LSTM networks are employed to learn the semantic representations of an aspect and the corresponding sentence, respectively. The interaction between these representations is also taken into account.
- RAM [42]: The multi-layer attention mechanism and memory network are combined to learn the representation of context related to a specified aspect, which is then used for classification.
- MIAD [16]: Multiple aspects in a sentence are considered as a sequence, and an LSTM network is adopted to capture the sentiment relationships between them.

- ASGCN [12]: A graph convolution neural network based on a dependency syntax tree is proposed to better capture the long-distance dependencies between a specified aspect and other words in a sentence.

The second type of method uses dynamical word embeddings from pretrained language models, such as BERT, and includes the models that utilize the dependency syntax information and external knowledge as follows:

- AEN-BERT [43]: BERT word embeddings are used as inputs, and an attention mechanism learns the representation of the context related to a specified aspect. The learned representation is then used for sentiment classification.
- SDGCN-BERT [5]: A bidirectional attention mechanism serves as the encoder, followed by a graph neural network to capture sentiment dependencies between multiple aspects in a sentence.
- ASGCN-BERT: The BERT-enhanced ASGCN model.
- dotGCN [44]: To eliminate reliance on external dependency parsers, this model induces aspect-specific discrete opinion trees as an alternative structure to dependency trees.
- KDGN [45]: A knowledge-aware dependency graph network is proposed to incorporate domain knowledge, dependency labels, and constituency syntax path.
- IDGNN [14]: This model first develops a graph attention network and a graph convolutional network to encode the syntactic and semantic information, respectively. Then, two fusion strategies are introduced to merge these information.
- APARN [25]: Instead of incorporating syntactic structures of sentences, APARN is designed to make full use of the semantic structures of abstract meaning representation (AMR).

The experimental results based on the three datasets are shown in Table 3. It should be noted that the results of IAN and RAM based on the MAMS dataset were extracted from the previous work [46], while the results of T-GCN, dotGCN, and APARN were from [25]. We executed the authors' provided code to obtain the results of ASGCN on the MAMS dataset. We enhanced the publicly available ASGCN code with BERT word embeddings to obtain the results of ASGCN-BERT. The results for other benchmark methods were extracted from the corresponding literature.

Table 3. Performance comparison with baselines. * means that the sentiment relationships between multiple aspects within a sentence are considered. $^{\#}$ indicates that the structure information of the dependency syntax tree is used. The best results for each dataset under the two experimental settings are marked in bold, and the second-best results are underlined.

Model	**Laptop**		**Restaurant**		**MAMS**	
	ACC	**Macro-F_1**	*ACC*	**Macro-F_1**	*ACC*	**Macro-F_1**
IAN	72.0	67.4	79.3	70.1	70.0	68.8
RAM	72.1	68.4	78.5	68.5	75.5	74.4
MIAD *	72.5	-	79.0	-	-	-
ASGCN $^{\#}$	75.6	71.1	80.8	72.0	76.8	75.3
HGNN $^{*,\#}$	**77.3**	**73.2**	**82.7**	**73.9**	**78.3**	**77.0**
AEN-BERT	79.9	76.3	83.1	73.7	-	-
SDGCN-BERT *	81.3	78.3	83.5	76.4	-	-
ASGCN-BERT $^{\#}$	80.6	76.7	85.1	77.4	82.3	81.8
dotGCN	81.0	78.1	86.2	80.5	84.9	84.4
KDGN $^{\#}$	81.3	77.6	87.0	81.9	**86.1**	84.6
IDGNN $^{\#}$	81.1	77.7	87.3	81.2	84.6	83.4
APARN	**81.9**	79.1	**87.8**	**82.4**	85.6	85.1
HGNN-BERT $^{*,\#}$	81.6	**79.5**	87.0	82.2	84.8	**85.4**

From the experimental results in Table 3, we can conclude the following: (1) Integrating the dependency syntax information benefits the aspect-level sentiment classification (ASGCN-BERT, KDGN and IDGNN vs. AEN-BERT). (2) Modeling sentiment relationships between multiple aspects in a sentence is necessary and improves ASC performance (MIAD vs. RAM; SDGCN-BERT vs. AEN-BERT). (3) By leveraging both the dependency syntax information and the sentiment relationships between multiple aspects, our proposed model achieves a performance that is comparable to or better than baseline methods (HGNN-BERT vs. methods in Part 2; HGNN vs. methods in Part 1). (4) While our method may not always outperform KDGN, IDGNN, and APARN, this is because they leverage additional information such as constituent syntax trees in KDGN and semantic information in IDGNN and APARN.

Based on the above analyses, our proposed hybrid graph neural network ASC model is proven to be effective.

4.3. Ablation Study

To verify the effectiveness of key components of our proposed model, we conducted ablation experiments on the relatively larger Restaurant and MAMS datasets, and present the results in Table 4. Specifically, we compared the proposed model (HGNN-BERT) with the following three variants:

- W/o dependency edges: The hybrid graph does not include edges based on the dependency syntax tree, meaning our model does not utilize information from the dependency syntax tree.
- W/o categories of edges: The category of edges is not considered in the hybrid graph neural network layer. In other words, we replace $\hat{h}_q^{l-1}$ in Equation (3) with h_q^{l-1}.
- W/o adjacency edges: The hybrid graph does not consider the adjacency relationships between multiple aspects. That is to say, our model does not leverage the sentiment relationships between multiple aspects.
- W/o AS.adj edges: Only the edges between adjacent aspects are excluded.
- W/o AS.inter edges: Only the interval edges are excluded.
- W/o an attention mechanism: An average operator is used in place of the attention mechanism.
- W/o a gated mechanism: An addition operator is used as a replacement.

Table 4. Experimental results of the ablation study.

#	Model	Restaurant		MAMS	
		ACC	Macro-F_1	*ACC*	Macro-F_1
1	HGNN-BERT	87.0	82.2	84.8	85.4
2	w/o dependency edges	84.9	80.3	83.2	83.7
3	w/o categories of edges	85.7	81.7	84.1	84.9
4	w/o adjacency edges	86.1	81.6	83.7	84.4
5	w/o AS.adj edges	86.4	81.8	84.1	84.9
6	w/o AS.inter edges	86.7	82.0	84.3	85.0
7	w/o attention mechanism	85.8	81.2	83.6	84.9
8	w/o gated mechanism	86.7	82.1	84.3	85.1

From the results in Table 4, several observation can be made: (1) All three variants of HGNN-BERT show varying degrees of performance degradation. These results clearly demonstrate the necessity of introducing dependency edges and adjacency edges in the hybrid neural network layer, as well as capturing the category information of these edges. (2) When the category of edges is not considered, the variant model's performance substantially declines on both datasets, despite many previous models ignoring this type of information (Line 3 vs. Line 1). (3) Omitting the consideration of sentiment relationships between multiple aspects results in performance degradation, particularly evident for the MAMS dataset (Line 4, 5 and 6 vs. Line 1). As expected, the edges between adjacent aspects have more of an impact on performance than the edges between non-adjacent aspects (Line 5 vs. Line 6). (4) When the attention mechanism is not used, the performance based on both datasets degrades significantly (Line 7 vs. Line 1). Excluding the gated mechanism leads to a small but consistently decreased performance based on both datasets (Line 8 vs. Line 1).

On one hand, these findings highlight the advantage of jointly predicting sentiment polarities of multiple aspects in a sentence, compared to independently predicting them as in previous approaches. On the other hand, these results confirm that the attention and gated mechanisms are both effective and necessary for information transfer.

4.4. Performance on ROC and AUC

To further validate the effectiveness of our model, we conducted a comparative analysis against baseline models using the *ROC* and *AUC* metrics. Due to the lack of open-source code for KDGN and APARN, our evaluation was limited to comparing our model (HGNN-BERT) with the IDGNN model. The performance of these two models across all three datasets is depicted in Figure 4.

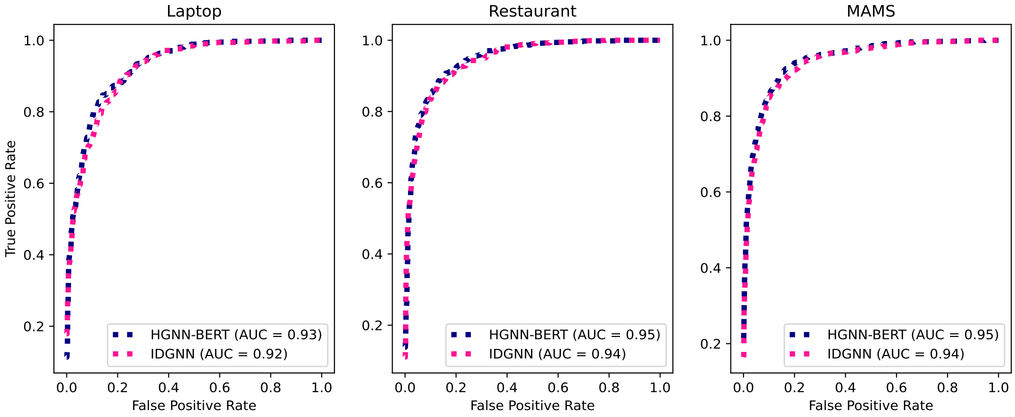

Figure 4. The macro-averaged *ROC* curves and *AUC* scores on three datasets.

Our model consistently achieved superior *AUC* values across all three datasets, highlighting its enhanced ability to discriminate between different classes across various threshold settings. Specifically, our model attained *AUC* scores of 0.93, 0.95, and 0.95, showcasing its high sensitivity and specificity in the ASC task. In comparison, the IDGNN model yielded scores 0.92, 0.94, and 0.94, falling short of matching our model's performance. This disparity indicates that our model is more adept at handling the complexities of the classification problem, particularly in situations where class distinctions are subtle or when the data are imbalanced.

4.5. Weights of Dependency Types

To further analyze which types of dependencies are important for our model, we averaged the weights assigned to each dependency type in the last graph neural network layer. The top 10 dependency types with their average weights based on the Restaurant dataset are listed in Figure 5. We can observe the following: (1) Our model assigns higher weights to *nsubj* and *acomp*, which are typically employed to express a subjective attitude toward a particular aspect (e.g., *the macbook looks very beautiful*). (2) The *amod* and *advmod* types also receive significant attention, as they frequently occur in phrases like *great battery* and *extremely useful*. (3) More importantly, our model assigns considerable weights to *conj* and *cc*. In the sentence *I have been impressed with the battery life and the performance for such a small amount of memory*, the aspects *battery life* and *performance* are connected by a *conj* edge, while *performance* and *and* are connected by a *cc* edge. With these higher weights, our model can easily deduce that the two aspects share the same sentiment polarity. Overall, these observations suggest that our proposed model is capable of allocating suitable weights to various dependency types, thereby better learning the context information related to the aspects.

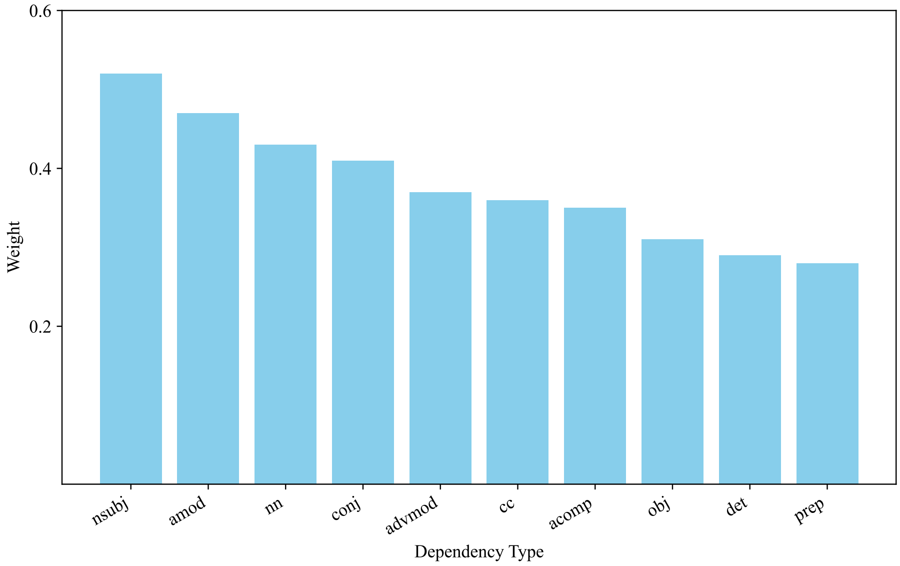

Figure 5. Weights of dependency types based on the Restaurant dataset.

4.6. *Effects of Hyper-Parameters*

To explore the influence of different values of hyper-parameters on the performance of our HGNN-BERT model, we varied these values and report the corresponding accuracies in Figure 6. Specifically, we varied the number of hybrid graph neural network layers $L = [1, 2, 3, 4, 5]$, the dimension of the category vectors of edges $d_1 = [25, 50, 75, 100, 125]$, the size of BiLSTM hidden units $d_2 = [100, 200, 300, 400, 500]$, and the dropout rate $r = [0.1, 0.2, 0.3, 0.4, 0.5]$, respectively. Please note that we changed one hyper-parameter at a time and kept the others fixed.

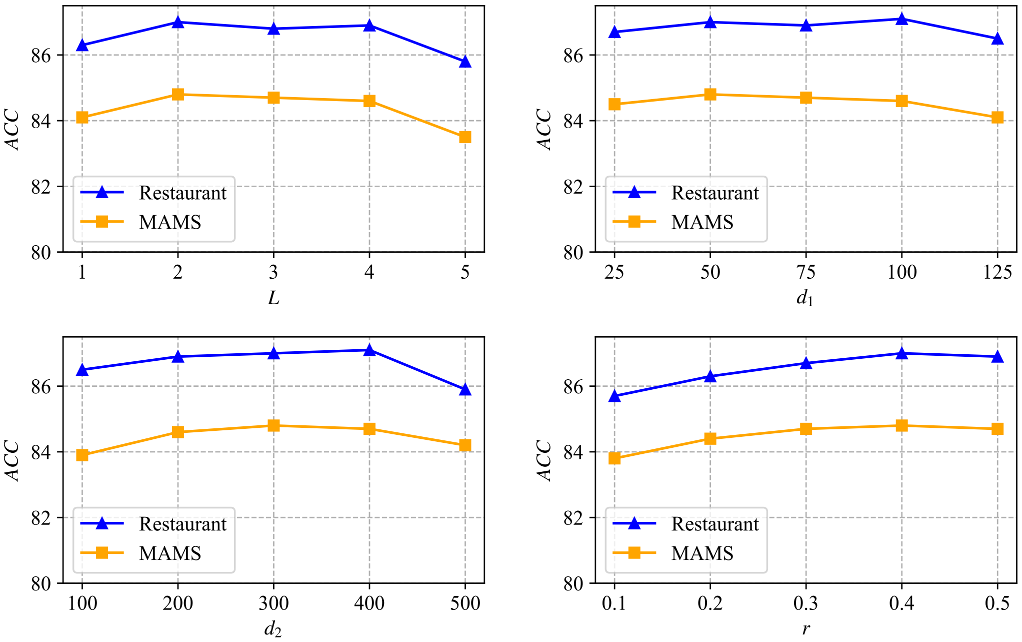

Figure 6. Effects of hyper-parameters.

From the results shown in Figure 6, we can observe that our model yielded relatively stable results when $L = [2, 3, 4]$, $d_1 = [25, 50, 75, 100]$, $d_2 = [200, 300, 400]$ and $r = [0.3, 0.4, 0.5]$. This indicates that our proposed model is not highly sensitive to these hyper-parameters.

4.7. Time Analysis

Time consumption is crucial for practical applications, as it directly impacts the usability and scalability of models. We compared the proposed model with baseline models in terms of time efficiency. Specifically, we ran SDGNCN-BERT, ASGCN-BERT, IDGNN, and our HGNN-BERT using an RTX 2080Ti GPU (MSI, Shanghai, China) with a batch size of 32. We did not include the time required for dependency parsing, as this can be performed in advance during training. During testing, the dependency parsing time was considered. From Table 5, we can observe the following: (1) Compared to traditional graph-based models (SDGCN-BERT and ASGCN-BERT), our model consumes more training and test time. The main reason for this is that the hybrid graph we used is more complex than the aspect graph or dependency graph used in these model. (2) IDGNN designs a dependency graph and a global-word graph, based on which it transfers two types of information via GAT and GCN, respectively. Our model requires less time than IDGNN while achieving better performance.

Table 5. Time comparison with baseline models based on the Restaurant dataset. GCN and GAT represent the graph convolutional network and the graph attention network, respectively. TR is the training time of a single epoch and IN represents the inference time on the test set.

Model	**Input Layer**	**Encoding Layer**	**External Tools**	**TR**	**IN**
SDGCN-BERT	BERT+BiLSTM	GCN on aspect graph	-	242 s	32 s
ASGCN-BERT	BERT+BiLSTM	GCN on dependency graph	Dependency Parser	286 s	36 s
IDGNN	BERT+BiLSTM	GAT on dependency graph, GCN on global-word graph	Dependency Parser	386 s	45 s
KDGN	BERT+BiLSTM	GAT on knowledge-aware dependency graph	Dependency Parser, Constituency parser, Knowledge retriever	-	-
APARN	BERT	Path aggregation layer	AMR Parser	-	-
HGNN-BERT	BERT+BiLSTM	GAT on hybrid graph	Dependency Parser	347 s	41 s

There is no open source code available for the baseline models KDGN and APARN, and it is difficult to reproduce these two models. Therefore, we do not list their training and test time. Theoretically, our model outperforms KDGN and APARN in terms of time efficiency. The reasons are for this as follows: (1) KDGN requires more external tools: a dependency parser, a constituency parser, and a knowledge retriever. The constructed knowledge-aware dependency graph is also more complex than our hybrid graph. (2) As described in the limitations of APARN [25], the path aggregation layer is more time-consuming than the existing methods. Overall, our proposed model performs well in terms of time consumption.

4.8. Case Study

To further explore the effectiveness of our proposed HGNN-BERT, we present several examples in Table 6, along with the predictions of three models: SDGCN-BERT, IDGNN, and HGNN-BERT. In the first example, SDGCN-BERT makes a mistake on both aspects because it focuses more on *like*. With the help of the dependency information (*not*, *version*) and (*version*, *feature*), both IDGNN and HGNN-BERT make correct predictions. In the second example, both SDGCN-BERT and HGNN-BERT effectively capture the sentiment relationships between adjacent aspects, resulting in accurate predictions. For example 3, the sentiment polarity for the aspect *vegetables* is implicitly expressed. Our model assigns higher weights to the relationships (*but*, *needed*, *cc*) and (*vegetables*, *needed*, *nsubj*), while also considering the relationship between the aspects *meat* and *vegetables*, thereby accurately predicting the different polarities of them. The above-mentioned examples clearly demonstrate that it is essential to use both types of dependency relation and the sentiment relationships between aspects.

Table 6. Case studies. Aspects in reviews are underlined. P, N, and O represent positive, negative, and neutral sentiment polarities, respectively. Underline means the aspects in a sentence.

#	**Review**	**SDGCN-BERT**	**IDGNN**	**HGNN-BERT**
1	Did not like the new Android version and the new navigation bar feature.	P×, P×	N✓, N✓	N✓, N✓
2	The pizza is very good, so is the atmosphere.	P✓, P✓	P✓, O×	P✓, P✓
3	The meat was great, but vegetables needed flavor and etc.	P✓, P×	P✓, O×	P✓, N✓
4	The mobile phone can be just put in my pocket.	N×	O×	O×
5	It is really thick around the battery.	N×	N×	N×

All three models failed to classify examples 4 and 5 correctly. Understanding the meanings of text like *just put in my pocket* and *around* is very challenging for current ASC models. We leave the study of these issues for future work.

5. Conclusions

In this paper, we propose a novel aspect-level sentiment classification model based on a hybrid graph neural network, achieving a performance comparable to or better than that of recent baseline models based on three commonly used datasets. Specifically, we design several hybrid graph neural network layers on the encoding layer to leverage both dependency syntax information (relations and their types) and sentiment relationships between multiple aspects. The primary distinction between our model and existing methods lies in our comprehensive consideration of relationships: among words, between words and aspects, and among aspects. In contrast, existing methods usually focus on only a subset of these relationships. Comprehensive experimental results confirmed both the effectiveness and efficiency of our method. The main drawback of our method is its reliance on the results of dependency parsing. When the dependency parsing results are suboptimal or in scenarios where a suitable dependency parser is unavailable, our method is not applicable.

In our future work, we plan to explore the application of hybrid graph neural networks in other natural language processing tasks. For instance, in multi-modal tasks, we can utilize a hybrid graph neural network to model interactions between objects from different modalities.

Author Contributions: Conceptualization, H.Z. and C.C.; methodology, H.Z.; software, C.C.; validation, H.Z.; formal analysis, H.Z.; data curation, H.Z. and C.C.; writing—original draft preparation, H.Z. and C.W.; funding acquisition, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been jointly supported by the National Natural Science Foundation of China under Grant No. 62266017 and the Natural Science Foundation of Jiangxi Province under Grant No. 20232BAB202050.

Data Availability Statement: The Laptop, Restaurant and MAMS datasets are all publicly available. Please refer to Section 4.1 for more details. We have released the source code and the processed datasets (https://github.com/NlperYeu/HGNN (accessed on 12 August 2024)).

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

References

1. Wu, C.; Cao, L.; Chen, J.; Wang, Y.; Su, J. Modeling Different Effects of User and Product Attributes on Review Sentiment Classification. *Appl. Intell.* **2024**, *54*, 835–850. [CrossRef]
2. Rasool, A.; Jiang, Q.; Qu, Q.; Kamyab, M.; Huang, M. HSMC: Hybrid Sentiment Method for Correlation to Analyze COVID-19 Tweets. In Proceedings of the Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery, Fuzhou, China, 30 July 2022; pp. 991–999.
3. Thapar, K.; Singh, R.; Fadlullah, Z.M.; Fouda, M.M.; Nasser, N.; Ali, A. A Hybrid AI Model for Improving COVID-19 Sentiment Analysis in Social Networks. In Proceedings of the ICC 2022—IEEE International Conference on Communications, Seoul, Republic of Korea, 16 May 2022; pp. 1752–1757.
4. Pang, B.; Lee, L. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135. [CrossRef]

5. Zhao, P.; Hou, L.; Wu, O. Modeling Sentiment Dependencies with Graph Convolutional Networks for Aspect-level Sentiment Classification. *Knowl.-Based Syst.* **2020**, *193*, 105443. [CrossRef]
6. Zhu, Z.; Zhang, D.; Li, L.; Li, K.; Qi, J.; Wang, W.; Zhang, G.; Liu, P. Knowledge-guided Multi-granularity GCN for ABSA. *Inf. Process. Manag.* **2023**, *60*, 103223. [CrossRef]
7. Gu, T.; Zhao, H.; He, Z.; Li, M.; Ying, D. Integrating External Knowledge into Aspect-based Sentiment Analysis using Graph Neural Network. *Knowl.-Based Syst.* **2023**, *259*, 110025. [CrossRef]
8. Zhou, T.; Shen, Y.; Li, Y. GCNet: Global-and-Context Collaborative Learning for Aspect-Based Sentiment Analysis. In Proceedings of the COLING, Torino, Italy, 20 May 2024; pp. 7570–7580.
9. Tian, Y.; Liu, C.; Song, Y.; Xia, F.; Zhang, Y. Aspect-based Sentiment Analysis with Context Denoising. In Proceedings of the Findings of NAACL, Mexico City, Mexico, 16 June 2024; pp. 3083–3095.
10. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the EMNLP, Austin, TX, USA, 1 November 2016; pp. 606–615.
11. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive Attention Networks for Aspect-level Sentiment Classification. In Proceedings of the AAAI, Melbourne, VIC, Australia, 19 August 2017; pp. 4068–4074.
12. Zhang, C.; Li, Q.; Song, D. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. In Proceedings of the EMNLP, Hong Kong, China, 3 November 2019; pp. 4568–4578.
13. Sun, K.; Zhang, R.; Mensah, S.; Mao, Y.; Liu, X. Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree. In Proceedings of the EMNLP, Hong Kong, China, 3 November 2019; pp. 5678–5687.
14. Wang, P.; Tao, L.; Tang, M.; Wang, L.; Xu, Y.; Zhao, M. Incorporating Syntax and Semantics with Dual Graph Neural Networks for Aspect-Level Sentiment Analysis. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108101. [CrossRef]
15. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the ICLR, Toulon, France, 24 April 2017.
16. Hazarika, D.; Poria, S.; Vij, P.; Krishnamurthy, G.; Cambria, E.; Zimmermann, R. Modeling Inter-Aspect Dependencies for Aspect-Based Sentiment Analysis. In Proceedings of the NAACL, New Orleans, LA, USA, 1 June 2020; pp. 266–270.
17. Tang, D.; Qin, B.; Feng, X.; Liu, T. Effective LSTMs for Target-Dependent Sentiment Classification. In Proceedings of the COLING, Osaka, Japan, 11 December 2016; pp. 3298–3307.
18. Sun, X.; Wang, Y.; Wang, X.; Sun, Y. Aspect-Based Sentiment Analysis Model Based on Dual-Attention Networks. *J. Comput. Res. Dev.* **2019**, *56*, 2384–2395.
19. Du, C.; Liu, P. Helical Attention Networks for Aspect-level Sentiment Classification. *J. Chin. Inf. Process.* **2020**, *34*, 70–77.
20. Huang, B.; Carley, K. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. In Proceedings of the EMNLP, Hong Kong, China, 3 November 2019; pp. 5472–5480.
21. Wang, K.; Shen, W.; Yang, Y.; Quan, X.; Wang, R. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In Proceedings of the ACL, Online, 5 July 2020; pp. 3229–3238.
22. Tang, H.; Ji, D.; Li, C.; Zhou, Q. Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification. In Proceedings of the ACL, Online, 5 July 2020; pp. 6578–6588.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the Proceeding of NIPS, Long Beach, CA, USA, 4 December 2017; pp. 6000–6010.
24. Hou, X.; Qi, P.; Wang, G.; Ying, R.; Huang, J.; He, X.; Zhou, B. Graph Ensemble Learning over Multiple Dependency Trees for Aspect-level Sentiment Classification. In Proceedings of the NAACL, Online, 6 June 2021; pp. 2884–2894.
25. Ma, F.; Hu, X.; Liu, A.; Yang, Y.; Li, S.; Yu, P.S.; Wen, L. AMR-based Network for Aspect-based Sentiment Analysis. In Proceedings of the ACL, Toronto, ON, Canada, 9 July 2023; pp. 322–337.
26. Hu, M.; Zhao, S.; Zhang, L.; Cai, K.; Su, Z.; Cheng, R.; Shen, X. CAN: Constrained Attention Networks for Multi-Aspect Sentiment Analysis. In Proceedings of the EMNLP, Hong Kong, China, 3 November 2019; pp. 4593–4602.
27. Han, Y.; Zhou, X.; Wang, G.; Feng, Y.; Zhao, H.; Wang, J. Fusing Sentiment Knowledge and Inter-Aspect Dependency Based on Gated Mechanism for Aspect-level Sentiment Classification. *Neurocomputing* **2023**, *551*, 126462. [CrossRef]
28. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral Networks and Locally Connected Networks on Graphs. In Proceedings of the ICLR, Banff, AB, Canada, 14 April 2014.
29. Yao, L.; Mao, C.; Luo, Y. Graph Convolutional Networks for Text Classification. In Proceedings of the AAAI, Honolulu, HI, USA, 27 January 2019; pp. 7370–7377.
30. Yin, Y.; Meng, F.; Su, J.; Zhou, C.; Yang, Z.; Zhou, J.; Luo, J. A Novel Graph-based Multi-modal Fusion Encoder for Neural Machine Translation. In Proceedings of the ACL, Online, 5 July 2020; pp. 3025–3035.
31. Wu, C.; Cao, L.; Ge, Y.; Liu, Y.; Zhang, M.; Su, J. A Label Dependence-Aware Sequence Generation Model for Multi-Level Implicit Discourse Relation Recognition. In Proceedings of the AAAI, Vancouver, BC, Canada, 23 February 2022; pp. 11486–11494.
32. Yan, Y.; Cai, B.; Song, S. Nested Named Entity Recognition as Building Local Hypergraphs. In Proceedings of the AAAI, Washington, DC, USA, 7 February 2023; pp. 13878–13886.
33. Mao, T.; Hao, T.; Fu, J.; Yoshie, O. Hierarchical Graph Fusion Network and a New Argumentative Dataset for Multiparty Dialogue Discourse Parsing. *Inf. Process. Manag.* **2024**, *61*, 103613. [CrossRef]
34. Hong, Y.; Rodriguez, C.; Qi, Y.; Wu, Q.; Gould, S. Language and Visual Entity Relationship Graph for Agent Navigation. In Proceedings of the NIPS, Online, 12 December 2020; pp. 7685–7696.

35. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the EMNLP, Doha, Qatar, 25 October 2014; pp. 1532–1543.
36. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL, Minneapolis, MN, USA, 2 June 2019; pp. 4171–4186.
37. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the the 8th International Workshop on Semantic Evaluation (SemEval), Dublin, Ireland, 23 August 2014; pp. 27–35.
38. Jiang, Q.; Chen, L.; Xu, R.; Ao, X.; Yang, M. A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis. In Proceedings of the EMNLP, Hong Kong, China, 3 November 2019; pp. 6280–6285.
39. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
40. Dozat, T.; Manning, C.D. Deep Biaffine Attention for Neural Dependency Parsing. In Proceedings of the ICLR, Toulon, France, 24 April 2017.
41. Tom, F. An Introduction to ROC Analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874.
42. Chen, P.; Sun, Z.; Bing, L.; Yang, W. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In Proceedings of the EMNLP, Copenhagen, Denmark, 7 September 2017; pp. 452–461.
43. Song, Y.; Wang, J.; Jiang, T.; Liu, Z.; Rao, Y. Attentional Encoder Network for Targeted Sentiment Classification. *arXiv* **2019**, arXiv:1902.09314.
44. Chen, C.; Teng, Z.; Wang, Z.; Zhang, Y. Discrete Opinion Tree Induction for Aspect-based Sentiment Analysis. In Proceedings of the ACL, Dublin, Ireland, 22 May 2022; pp. 2051–2064.
45. Wu, H.; Huang, C.; Deng, S. Improving Aspect-based Sentiment Analysis with Knowledge-aware Dependency Graph Network. *Inf. Fusion* **2023**, *92*, 289–299. [CrossRef]
46. Wu, Z.; Ying, C.; Dai, X.; Huang, S.; Chen, J. Transformer-Based Multi-aspect Modeling for Multi-aspect Multi-sentiment Analysis. In Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing, Zhengzhou, China, 14 October 2020; pp. 546–557.

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.