# Aspect-Based Sentiment Analysis with Dependency Relation Weighted Graph Attention

**Tingyao Jiang, Zilong Wang \*, Ming Yang and Cheng Li**

College of Computer and Information, China Three Gorges University, Yichang 443002, China; jiangty@ctgu.edu.cn (T.J.)
\* Correspondence: wzl@ctgu.edu.cn

**Abstract:** Aspect-based sentiment analysis is a fine-grained sentiment analysis that focuses on the sentiment polarity of different aspects of text, and most current research methods use a combination of dependent syntactic analysis and graphical neural networks. In this paper, a graph attention network aspect-based sentiment analysis model based on the weighting of dependencies (WGAT) is designed to address the problem in that traditional models do not sufficiently analyse the types of syntactic dependencies; in the proposed model, graph attention networks can be weighted and averaged according to the importance of different nodes when aggregating information. The model first transforms the input text into a low-dimensional word vector through pretraining, while generating a dependency syntax graph by analysing the dependency syntax of the input text and constructing a dependency weighted adjacency matrix according to the importance of different dependencies in the graph. The word vector and the dependency weighted adjacency matrix are then fed into a graph attention network for feature extraction, and sentiment polarity is predicted through the classification layer. The model can focus on syntactic dependencies that are more important for sentiment classification during training, and the results of the comparison experiments on the Semeval-2014 laptop and restaurant datasets and the ACL-14 Twitter social comment dataset show that the WGAT model has significantly improved accuracy and F1 values compared to other baseline models, validating its effectiveness in aspect-level sentiment analysis tasks.

**Keywords:** aspect-based sentiment analysis; dependent syntactic analysis; dependency weighting; graph attention network; pretrained model

## 1. Introduction

Sentiment analysis is one of the current research hotspots in natural language processing [1]. With the development of information technology, increasingly more text with emotional tendencies is appearing on the internet [2], and the analysis of this information to understand people's views is crucial for solving practical problems, as well as for informative decision making.

The main types of text sentiment analysis are chapter-level sentiment analysis, sentence-level sentiment analysis, and aspect-level sentiment analysis [3]. Both chapter-level and sentence-level sentiment analyses are coarse-grained sentiment analyses that analyse an entire document or an entire sentence as a basic unit, respectively, and can only analyse a single sentiment of the text, not multiple aspects. Aspect-level sentiment analysis is a more fine-grained sentiment analysis that addresses specific aspects of a sentence [3]. For example, the emotional polarity of "The food at this restaurant is good, but the service is terrible" is positive and negative. In this case, it is not possible to accurately analyse the sentiment of the whole sentence, and specific aspects need to be analysed. With increasingly complex semantic environments and increased analysis requirements, coarse-grained sentiment analysis techniques can no longer meet today's needs, and more fine-grained sentiment analysis has become a key research problem in natural language processing [4].

Early approaches for aspect-based sentiment analysis used machine learning to manually engineer features, such as building sentiment dictionaries and then classifying sentiments through machine learning classifiers such as decision trees and support vector machines [5]. Particularly complex feature engineering is often needed, which is costly and poorly generalised.

With the development of neural network technology, deep learning methods have become the main approach for current aspect-based sentiment analysis research [6]. Deep learning overcomes the traditional machine learning problem of relying on manual extraction of text features, and it is able to automatically learn the emotional information embedded in text through neural networks and map words in text to low-latitude word embedding vectors. Some researchers further combined attentional mechanisms with deep learning [7–9], obtaining desirable results. To be specific, the attention mechanism enables the model to concentrate more on words with high similarity during sentiment classification by calculating the semantic similarity of the aspect word embedding vector to other word embedding vectors in the text. However, merely taking into account semantic similarity without considering the syntactic structure between words may cause the model to focus excessively on sentiment words that are not syntactically correlated with aspectual words and, thus, make incorrect judgments [10,11]. Nonetheless, traditional neural networks can only process data in Euclidean space and cannot make effective use of syntactic information.

To cope well with these problems, some researchers have conducted systematic and comprehensive dependency syntactic analysis on texts, generated dependency syntactic graphs to express the syntactic dependencies between words, and extracted the syntactic information from them through graph neural networks, which has materialised a considerably noticeable improvement in the model effect [11]. Nevertheless, the majority of current studies simply take into consideration the syntactic dependencies between words, ignoring the important feature of the type of dependency. Aside from that, in the process of classifying aspectual words for sentiment, the model needs to fix attention on the dependencies that are more important for the sentiment judgement of aspectual words. As a consequence, we incorporated dependency types into the graph neural network. Under such circumstances, the model can focus more on the dependencies that are important for sentiment classification. To achieve this, we set the weights of the dependencies in accordance with their importance and weight them as trainable parameters in the graph neural network for feature extraction, so that the model can fix its attention on the important dependencies. In the word embedding layer, we also use a sentiment pretraining model for word embedding. This enables the word embedding weights to be better adapted to the sentiment analysis task.

This paper presents the following main work:

1.  We use a BERT-pretrained sentiment corpus for word embedding of input text. In such case, BERT can better capture the sentiment information in the text during pretraining, and demonstrate the superiority of the sentiment pretraining model compared to the normal pretraining model in sentiment analysis tasks through experimental results.
2.  We probe deep into the text for dependency syntax, set initial weights for dissimilar dependencies in line with their relative importance for sentiment classification, and further optimise them through model training.
3.  We come up with a relation-weighted graph attention network model (WGAT). This model ameliorates the existing attention mechanism and weights the attention score. In this way, the model can assign weights according to the importance of different types of dependencies, so as to concentrate more on important dependencies and heighten classification accuracy.
4.  Experiments on different datasets prove the effectiveness of our proposed method, whereby the effect of our model is better than other baseline models.

## 2. Related Work

The main aim of aspect-based sentiment analysis is to examine the sentiment categories of the different aspects of a sentence. One of the main approaches used for aspect-based sentiment analysis based on machine learning involves using supervised classification algorithms such as plain Bayes algorithm [12] and support vector machines [13]. Although such an approach has achieved good performance, it relies heavily on manual feature engineering, which is costly. The performance of the model is also directly affected by how well the features are manually extracted; the manually extracted features are only valid in the corresponding domain and have poor generalisation capabilities.

In recent years, the rise of neural networks has led to the widespread use of deep learning methods in natural language processing tasks. Compared to traditional machine learning methods, deep learning can automatically extract text features without the need for particularly complex feature engineering and has greater expressive power. However, the large number of parameters used in various neural network models and the often small corpus for downstream tasks in natural language processing (sentiment analysis, machine translation, etc.) prevent the neural network parameters from being better trained. For this reason, pretraining on a large-scale corpus and then applying the resulting word embeddings to downstream natural language processing tasks is currently a proven approach [14].

Early word embedding techniques represented each word as a vector, where words that were closer in semantics were represented as vectors that were also closer in space. The word2vec model proposed by Mikolov et al. [15] uses a CBOW (predicting central words with surrounding words) and skip-gram (predicting surrounding words with central words). The Glove model proposed by Pennington et al. [16] constructs a co-occurrence matrix of words based on the corpus and then vectorises the word list on the basis of the co-occurrence matrix and the Glove model. The above two are static word vectors, which have been widely used in sentiment analysis tasks.

Recurrent neural networks (RNNs), with their temporal nature, exhibit great advantages in natural language processing tasks, but RNNs experience gradient disappearance for longer text. The long short-term memory (LSTM) network can solve this problem; hence, various aspect-based sentiment analysis models based on LSTM networks have been proposed. Tang et al. [7] proposed a target-dependent LSTM (TD-LSTM) model to address the inadequate attention given to target words in LSTM networks by using two LSTM networks, modelled separately on the left and right of the target words. They further proposed a target-connected LSTM (TC-LSTM) model, which further strengthens the relationship between target words and context by splicing target word vectors with contextual word vectors in the input part of the LSTM model based on TD-LSTM. The attention mechanism can focus on important parts of the text and has been successfully applied to a variety of natural language processing tasks. Wang et al. [8] proposed an attention-based LSTM (ATAE-LSTM) model using target embeddings, where the context and aspectual word splicing is fed into the LSTM for encoding, and the encoded vector is spliced with the aspectual word vector to obtain the sentiment category by computing through the attention mechanism. Tang et al. [17] applied the idea of memory networks to the aspect-based sentiment analysis task by storing the contextual information of a given aspect into a memory network and capturing the degree of aspect word–context association through an attention mechanism, which has a higher computational speed than LSTM. However, all these models use static word vectors and cannot dynamically adjust the meaning of words according to the context during model training to address the multiple meanings of words.

BERT is pretrained by working on a large corpus and fine-tuned in downstream tasks. It has better word embedding representation and stronger feature extraction than traditional word vectors. Song et al. [18] proposed the BERT_AEN model to address the excessive training data required for LSTM by using a multiheaded attentional encoder network for modelling between contextual words and aspect words and learning the interaction

information between them. Zeng et al. [19] suggested that the sentiment polarity of aspectual words is related to nearby contextual words, and that distant words can interfere with words that may interfere with the final classification; hence, they proposed an aspect-based sentiment classification mechanism that focuses on local context. However, the emotional polarity of aspectual words is not always related to words that are close' for example, in the sentence "So delicious were the noodles but terrible were the vegetables", the aspectual word "noodles" is closer to "terrible", but its affective polarity is negative. In addition, calculating weights between words through a self-attentive mechanism fails to consider syntactic knowledge and may assign greater weights to words unrelated to aspectual word sentiment classification, leading to misclassification [10,11].

In recent years, some scholars have applied knowledge of dependent syntactic analysis to aspect-based sentiment analysis. Dependent syntactic analysis focuses on analysing the dependencies between words in a sentence [20], explaining the dependencies between word nodes through a kind of graph structure. For example, in the above example, the adjective "delicious" directly modifies the aspectual word "noodles" through a dependency relation, and they are at a distance of 1 in the syntactic map, while the aspectual word "noodles" has no path to reach the adjective "terrible". Syntactic distance is, therefore, preferable over spatial distance. Graph convolutional networks (GCNs) [21] can better handle graph structured data by aggregating and messaging neighbouring nodes. Because traditional neural networks cannot address the syntactic constraints and long-term dependencies of text, Zhang et al. [11] built graph convolutional networks in syntactic dependency trees to facilitate the use of syntactic information and dependencies of text. Huang et al. [22] proposed a graph attention network based on target dependency [23], which represents sentences as dependency graphs and directly connects aspect words to related words to extract information through graph attention networks. Wang [24] et al. proposed the relational graph attention (R-GAT) network model to reconstruct syntactic graphs using dependencies with aspectual words as root nodes and to encode dependencies for feature extraction through relational graph attention networks. Current research has enabled models to learn rich syntactic knowledge by combining syntactic graphs with graph neural networks, but less consideration has been given to the types of dependencies. Although R-GAT encodes dependencies, it does not consider that different types of dependencies have different importance and may assign larger weights to unimportant nodes when calculating the attention weights of two nodes.

## 3. Methodology

### 3.1. Task Definition

The data for an aspect-based sentiment analysis task can usually be defined as a binary group $(W, A)$ consisting of a sentence and the aspect words therein, where $W = \{w_1, w_2, \ldots w_n\}$ is an emotionally charged sentence consisting of $m$ words and $A = \{w_i, w_{i+1}, \ldots w_{i+n-1}\}$ represents the $n$ aspect words in the sentence. Its affective polarity includes positive, negative, and neutral. For example the text "The staff should be more friendly.", $W$ is a sequence of words within the text, i.e., $W$ = {"the", "staff", "should", "be ", "more", "friendly", "."}, and $A$ is the sequence of aspectual words in it, i.e., $A$ = {"staff"}. The goal of the task is to input $W$ and $A$ into Model $M$ to predict the sentiment polarity of aspectual word $A$, as shown in Equation (1). $y\_pred$ is the model prediction result.

$$y\_pred = M(W, A). \tag{1}$$

This paper proposes a graph attention network aspect-based sentiment analysis (WGAT) model based on sentiment pretraining and the weighting of dependencies. The model mainly consists of a word embedding layer, a dependency syntactic analysis layer, a graph attention network layer, and a classification layer. The original text is encoded at the word embedding layer as an input feature to the graph attention network. Dependency syntax analysis is performed on the text at the syntactic analysis layer to generate a dependency syntax graph and construct a dependency-weighted adjacency matrix based on

the dependency relationships between nodes. Lastly, the weighted adjacency matrix and input features are fed into the graph attention network for feature extraction and sentiment prediction through the classification layer. The specific model structure is shown in Figure 1.
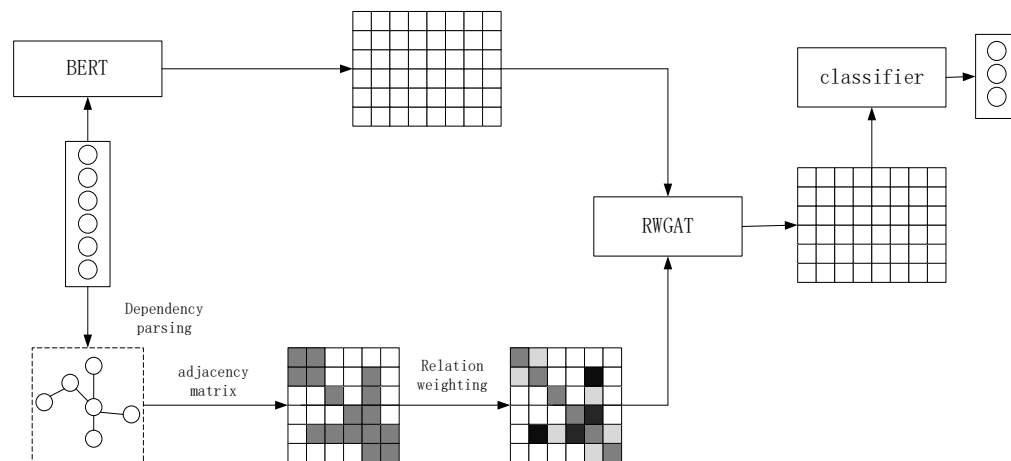
**Figure 1.** Model structural diagram.

### 3.2. Word Embedding Layer

The BERT model uses a bidirectional transformer [25] encoder structure, which, at its core, uses a self-attentive mechanism to encode text. The self-attention mechanism allows the model to focus on the important parts of the text during training by calculating the attention weight of each word in the text in relation to all other words. Although the pretrained model has performed well in a variety of NLP tasks, it is pretrained on a generic corpus based on its inability to better capture sentiment relations in text [26], limiting its performance in sentiment analysis tasks. This paper, therefore, uses a BERT model pretrained on a large-scale sentiment corpus for word embedding, making its pretrained weights better suited for sentiment analysis.

In this paper, the input text sequence is $W = \{w_1, w_2, \ldots w_m\}$, which is mapped into a low-dimensional semantic vector $H^{m \times bert\_\dim}$ by the BERT model, where m is the input text length and bert_dim is the BERT word embedding dimension. The features that are graph nodes are then fed into the graph neural network for training. This is shown in Equation (2).

$$H = \{h_1, h_2, \ldots h_m\} = BERT(W). \tag{2}$$

### 3.3. Syntactic Analysis Layer

Dependency syntactic analysis focuses on the dependency relationships between words in a sentence [24]. Although the attention mechanism can focus on important parts of text, it cannot adequately capture the syntactic dependencies in sentences [21], which affects the classification effect. This paper, therefore, requires syntactic analysis of the input text. The StanfordNLP syntactic parser was used to convert the sentences into a syntactic dependency graph. Taking "Great food but the service was dreadful!" as an example, the syntactic dependency graph is shown in Figure 2.
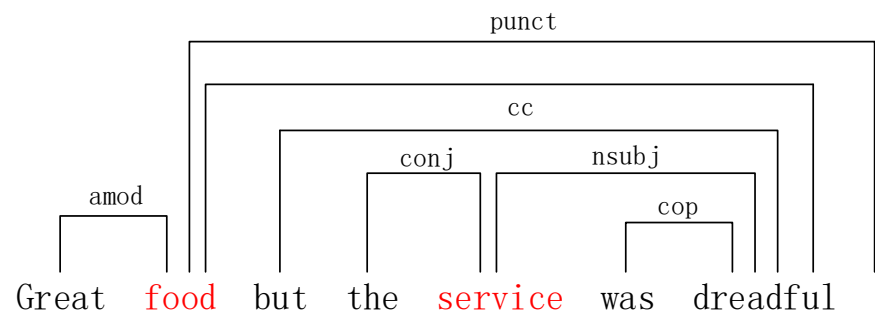
**Figure 2.** Dependency syntax example 1.

In Figure 2, the results of the syntactic analysis of dependencies include both the syntactic dependencies that exist between words and the types of dependencies. Different dependencies contribute differently to the affective classification of aspectual words, e.g., the aspectual word 'food' is related to the words "great", "dreadful", and "," in a modifying way. Different dependencies contribute differently to the affective classification of aspectual words; for example, the aspectual word "food" has a modifying relationship with the words "great", "dreadful", and ",", whose dependencies are "amod", "conj", and "punct", respectively. "Great" is an emotion word for the aspect word "food", indicating that its emotional overtone is positive and that it plays a positive role in the classification of emotions. "Punct" is usually connected with punctuation and can potentially interfere with the final classification. "Food" is linked to the word "dreadful" via "conj" with negative emotional overtones, which can lead to misclassification. For the aspectual word "service", the dependency "nsubj" can be directly connected to its affective word "dreadful", but "det" has no effect on affective classification. Different types of dependencies have different effects on the affective prediction of aspectual words. Therefore, this paper sets different weights for different types of dependencies, with larger weights used for dependencies that have an affective effect on aspectual words and smaller weights for the rest of the dependencies so that the model can focus on the important parts in the process of extracting the syntactic dependency graph. In addition to aspectual words, adverbial relations modifying adjectives or verbs play a crucial role in classification, as shown in Figure 3.
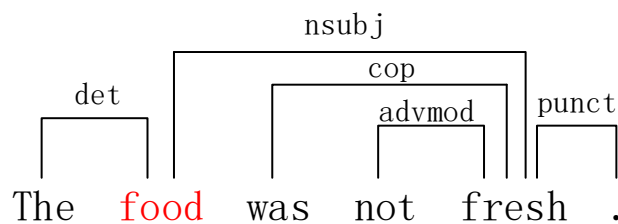
**Figure 3.** Dependency syntax example 2.

In Figure 3, the aspectual word "food" is modified by the emotion word "fresh". The emotion polarity could be positive; however, since "fresh" is not modified by the adverb "not", the emotion polarity of food should be negative.

The StanfordNLP syntactic parser used in this paper has more than 50 dependencies [27]. On the basis of a priori knowledge, we set larger weights for dependencies that are more important for aspectual word sentiment classification, such as amod, nsubj, and advmod, and smaller weights for other dependencies, such as det, punct, and cop. Here, the initialisation weight of important dependencies is set to 3, the weight of other dependencies is set to 0.5, and they are stored in the dependency dictionary wei.

After setting the dependency weights, we construct the dependency weighted adjacency matrix on the basis of the dependency weights. The construction method is shown in Equation (3).

$$A_{ij}^{weight} = \begin{cases} wei(r(i,j)) & i \neq j \ and \ \mathrm{j} \in \mathrm{N}(i) \\ 1 & i = j \\ 0 & other \end{cases}, \tag{3}$$

where $r(i,j)$ denotes the type of dependency relationship corresponding to node $i$ and node $j$. The $wei(r(i,j))$ function denotes the weight of the relationship between node $i$ and node $j$, and $N(i)$ denotes all the neighbouring nodes of node $i$. If node $j$ is a neighbour node of node $i$ (i.e., node $i$ and node $j$ are at a distance of 1), then the value of $A_{ij}^{weight}$ is the weight of the relationship between node $i$ and node $j$; otherwise, the value of $A_{ij}^{weight}$ is 0. In addition to preserving the information of the node itself, we also add self-connected edges for the node, i.e., the value of node $i$ with respect to itself is 1.

*3.4. Graph Attention Network Layer*

Ordinary GCNs are aggregated by averaging the characteristics of each of the node's neighbours. However, neighbouring nodes with different dependencies are of different importance to the aspect word nodes, and some may even lead to incorrect classification results. Therefore, this paper uses the GAT model for feature extraction of syntactic graphs.

The GAT network is essentially a variant of the upper GCN, where the attention weights of the current node and each neighbouring node are calculated according to the different importance of the neighbouring information and the weighted average when aggregating the neighbouring information. However, the use of the attention mechanism alone to calculate the neighbour node weights may cause the aspect nodes to focus too much on error messages. In this paper, we propose a graph attention network based on dependency weighting based on the GAT model. By adding dependency weights to the calculation of attention weights between neighbouring nodes, the model is trained to focus more on the important parts of the sentence by giving greater weights to important dependencies, as shown in Equation (4).

$$\partial = W^r A^{weight} + b. \tag{4}$$

Equation (4) shows the dependency weight matrix, where $W^r$ and $b$ are the weight and bias, respectively. The dependency weight matrix is set as a trainable parameter by setting the weights and biases, and the model dynamically adjusts the relationship weights during training according to the initial values we set.

The attention weights of different nodes are then calculated by weighting the dependencies, as shown in Figure 4. The calculation is shown in Equations (5) and (6).

$$e_{ij} = \partial_{ij} \big[ Wh_i || Wh_j \big]. \tag{5}$$

$$a_{ij} = \frac{exp\big(LeakyRelu\big(\partial_{ij}\big[Wh_i || Wh_j\big]\big)\big)}{\sum\limits_{u \in N(i)} exp\big(LeakyRelu\big(\partial_{iu}\big[Wh_i || Wh_u\big]\big)\big)}. \tag{6}$$
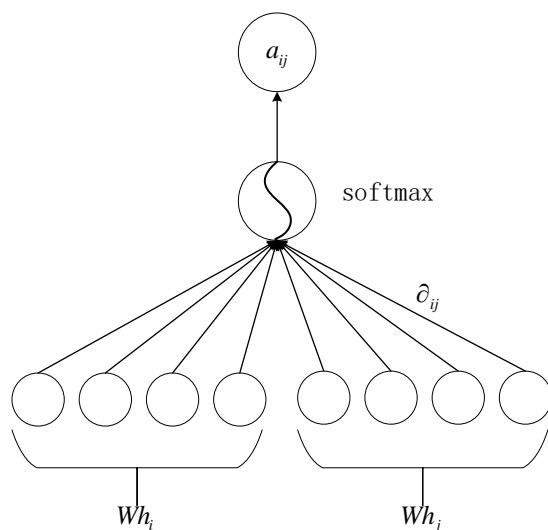
**Figure 4.** Attention calculation diagram.

Equation (5) calculates the attention weights of two nodes. The two nodes are stitched together by a linear transformation and multiplied by the corresponding relationship weights of the two nodes. $\partial$ is the weight of trainable dependencies in Equation (4), and || is the splicing operation. Equation (6) performs a softmax operation on the computed attention mechanism so that it maps to a probability between 0 and 1, where $N(i)$ denotes the neighbouring nodes of node $i$, and *LeakyRelu*() is the activation function.

Lastly, all neighbouring nodes of node $i$ are multiplied by their corresponding attention weights for weighted summation, and the final attention characteristics of node i are obtained through the nonlinear layer, as shown in Equation (7).

$$h'_i = \sigma \left( \sum_{j \in N(i)} a_{ij} W h_j \right). \tag{7}$$

To improve the model fitting ability, this paper adopts a multiheaded attention mechanism for information extraction. The calculation method is shown in Equation (8).

$$h'_i = \overset{K}{\underset{k=1}{||}} \sigma \left( \frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N(i)} a_{ij}^k W^k h_j \right). \tag{8}$$

In Equation (8), $\sigma$ is the activation function, where $K$ denotes the number of attention heads. We obtain the feature $h'_i$ of node $i$ after aggregating information about its neighbours by multiplying the weights calculated from the $K$ attention heads by the weighted average of the corresponding nodes. Lastly, we obtain the output vector $H' = \{h'_1, h'_2, \ldots h'_m\}$ of the dependency weighted graph attention network.

### 3.5. Emotional Classification Layer

As this paper implements a multiclassification task, a softmax function is used for the final sentiment classification. In this paper, the features extracted from the graph attention network layer are passed through a fully connected layer and a softmax layer to obtain the probability values of various sentiments of the corresponding aspect words. The feature with the highest probability is the final prediction result, as in Equation (9), where $W$ and $b$ are the weight and bias, respectively.

$$y\_pred = softmax(WH' + b). \tag{9}$$

The loss function of the model uses a cross-entropy loss function, as shown in Equation (10), where $y$ is the true label. To prevent overfitting of the model, L2 regularisation is used in this paper to train parameters and weights, and $\lambda$ is the regularisation parameter.

$$L(\theta) = -\sum y log(y\_pred) + \lambda||\theta||_2. \tag{10}$$

## 4. Experiment

### 4.1. Dataset

This paper conducts experiments on publicly available aspect-based sentiment analysis datasets, namely, the Semeval2014 Task4 dataset [28] and the ACL-14 [29] dataset. Among them, Semeval2014 includes two review datasets, laptop and restaurant, and ACL-14 is a Twitter social review dataset. Each dataset is divided into a training set and a test set and has three types of labels: positive, negative, and neutral. The specific information of the three datasets is shown in Table 1.

**Table 1.** Experimental data statistics.

| Dataset | Positive | | Negative | | Neutral | |
|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test |
| Laptop | 994 | 341 | 870 | 128 | 464 | 169 |
| Restaurant | 2164 | 728 | 807 | 196 | 637 | 196 |
| Twitter | 1561 | 173 | 1560 | 173 | 3127 | 346 |

### 4.2. Hyperparameter Setting

In this paper, experiments are conducted on 12 models for each of the three datasets. Seven of the models are based on the Glove word vector, and five are based on the BERT pretrained model. The word embedding dimension is 300 for Glove and 768 for the BERT-pretrained model. As the BERT-pretrained model requires a relatively low learning rate while fine-tuning the shared layer [10], the learning rate is set to $2 \times 10^{-5}$ for the BERT-based model and $1 \times 10^{-3}$ for the Glove-based model. The BERT model used above is bert-base-uncased, and the sentiment pretrained model used in this paper is bert-base-uncased-emotion. Other hyperparameters are shown in Table 2.

**Table 2.** Training parameters.

| Parameter | Value |
|---|---|
| Batch size | 16 |
| Dropout rate | 0.1 |
| Epoch | 6 |
| L2 regularisation coefficient | 0.000001 |
| Optimiser | Adam |
| Maximum sentence length | 80 |

### 4.3. Evaluation Metrics

The main model evaluation metrics commonly used for sentiment analysis are accuracy, precision, recall, and F1 value. Accuracy indicates the proportion of correctly predicted samples with respect to all samples, as shown in Equation (11). Precision is the proportion of positive predictions in the sample that are actually positive, as shown in Equation (12). Recall is the proportion of positive cases in the original sample that are correctly predicted, as shown in Equation (13). The F1 value combines precision and recall and is the summed average of the two, as shown in Equation (14). Some of the datasets in this experiment have a large difference in the numbers of categories, while macro-F1 (macro-F1) can consider the number of each category and is more suitable for unevenly distributed samples, as shown in Equation (15). To evaluate the model results, accuracy

and macro-F1 are used as evaluation metrics, and the model is tested five times on each of the three datasets. The best result is taken as the final result.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{11}$$

$$Precision = \frac{TP}{TP + FP}, \tag{12}$$

$$Recall = \frac{TP}{TP + FN}, \tag{13}$$

$$F1 = \frac{2 \times TP}{N + TP \times TN}, \tag{14}$$

$$Macro - F1 = \frac{1}{n}\sum_{i=1}^{n} F1_i, \tag{15}$$

where $N$ is the total number of samples, $TP$ is the number of samples with positive labels predicted by the model to be positive, $TN$ is the number of samples with negative labels predicted to be negative, FP is the number of samples with negative labels predicted to be positive, and $FN$ is the number of samples with positive labels predicted to be negative.

### 4.4. Comparison Models

To verify the validity and reasonableness of the model proposed in this paper, 11 baseline models used in aspect-based sentiment analysis were selected for comparative analysis.

TD-LSTM [7]: LSTM is extended by using two LSTM networks to model a left context with a target and a right context with a target. The left and right target-dependent representations are connected and used to predict the affective polarity of the target.

ATAE-LSTM [8]: ATAE-LSTM is an improved AE-LSTM model. The aspect word vector is spliced with the context vector in the input part of the model and fed into the LSTM network, enabling the context hidden states to carry information from the aspects.

MEMNET [17]: MEMNET stores the context of a particular aspectual word into a memory network and captures the importance of the aspectual word in relation to the contextual word through an attention mechanism.

IAN [9]: This model proposes an interactive attention network that extracts contextual and aspectual features through two LSTM models and interactively learns the representation of context and aspectual words through an attention mechanism.

Bi-GCN [30]: Bi-GCN uses global lexical graphs to encode word co-occurrence information and designs interactive graph convolutional networks to learn syntactic and lexical graphs.

ASGCN [11]: A multilayer GCN is constructed to extract syntactic information based on the ability of LSTM to capture word order information, and a masking mechanism is used to remove non-aspect word information and retain only aspectual high-level features.

TD-GAT [22]: Syntactic graphs using aspectual targets directly linked to related targets are applied to GAT networks instead of word sequences, and node features are updated by LSTM.

BERT-SPC [18]: This model uses a pretrained BERT classification model to generate word vectors and extracts the vectors corresponding to the sign bit [CLS] for final classification based on each vector output from the BERT model containing full text information.

AEN-BERT [18]: This model uses BERT for pretraining to generate context and target word vectors and an attention encoder to model the context and target.

LCF-BERT [19]: This model uses a local context-focusing mechanism that uses a dynamic masking layer for contextual features and a dynamic weighting layer for contextual features to focus on local contextual words.

Mem + BERT [31]: The syntactic structure information is extracted using GCN, and the attention mechanism is used to fuse syntactic structure information, semantic information, lexical information, location information, and aspects.

SEDC-GCN [32]: A two-channel graph convolutional network model based on GCN structural enhancement is proposed.

R-GAT-BERT [24]: R-GAT-BERT involves the pruning of dependent syntactic graphs, reconstruction of syntactic graphs with aspectual words as root nodes, and feature extraction via relational graph attention networks.

SenticGCN [33]: This model incorporates affective common-sense knowledge into graph networks to facilitate model extraction of affective dependencies between contextual words and specific aspects.

### 4.5. Experiment Results

Table 3 shows the results of the different models on the three datasets, where data marked with "*" refer to the model results in the published paper, while "-" refers to the results not given in the original paper, and the bolded parts are the models in this paper. The model results show that, among the word vector-based models, the ATAE_LSTM, IAN, and AOA models based on the attention mechanism have better results than the TD_LSTM model based on LSTM alone on both the notebook and the restaurant datasets due to the inclusion of the attention mechanism. However, the static word vector limits the performance of the model, with accuracy rates on the Twitter, restaurant, and laptop datasets topping out at approximately 70%, 77%, and 72%, respectively. Moreover, as the attention mechanism cannot use the syntactic information of the text, it is possible to assign larger attention weights to irrelevant or even wrong word nodes, leading to classification errors and limiting the performance of the model. ASGCN and Bi-GCN use GCNs for text syntactic feature extraction and show a large improvement over the attention-based mechanism model on all three datasets. The GAT model improves the effect of the TD-GAT model compared to the above two GCN models due to the consideration of the different importance between different nodes. The BERT_SPC model combines aspect words with context as input only by constructing auxiliary sentence pairs, and then uses the first output vector of BERT for classification. The model is relatively simple but also achieves better results, with accuracy improvements of 1.77%, 1.91%, and 2.6% compared to the best static word vector-based model. The R-GAT model discards all dependencies between non-aspect words by pruning the dependency tree, which reduces redundant information but may also ignore information that is important for sentiment classification. The model in this paper sets the corresponding weights for the dependencies of different importance and optimises them through model training. The best results are improved by 1.39%, 0.89%, and 1.36% compared to those of the best model above, proving the effectiveness of the model.

**Table 3.** Experimental results contrasting.

| Dataset | Twitter | | Restaurant | | Laptop | |
|---|---|---|---|---|---|---|
| | **Acc** | **F1** | **Acc** | **F1** | **Acc** | **F1** |
| TD-LSTM | - | - | 75.63 * | - | 68.13 * | - |
| ATAE-LSTM | - | - | 77.20 * | - | 68.70 * | - |
| MEMNET | 68.50 | 67.48 | 77.36 | 67.36 | 71.56 | 65.34 |
| IAN | 67.92 | 65.61 | 76.47 | 67.49 | 71.88 | 66.16 |
| ASGCN | 70.95 | 69.46 | 77.92 | 69.98 | 73.41 | 69.05 |
| Bi-GCN | 70.12 | 68.94 | 78.25 | 70.21 | 74.15 | 69.12 |
| TD-GAT | 71.35 | 69.17 | 79.11 | 71.64 | 75.21 | 72.09 |
| BERT-SPC | 73.12 | 71.32 | 81.02 | 72.81 | 77.81 | 73.41 |
| BERT-AEN | 73.70 | 72.10 | 81.69 | 73.64 | 77.19 | 72.57 |
| BERT-LCF | 73.84 | 72.87 | 82.13 | 74.52 | 78.12 | 73.43 |
| Mem-BERT | 73.13 | 71.94 | 84.76 | 78.46 | 78.70 | 74.06 |
| SEDC-GCN | 73.56 | 71.34 | 84.33 | 79.12 | 77.74 | 74.68 |
| R-GAT-BERT | 74.86 | 72.52 | 84.55 | 78.34 | 78.73 | 75.12 |
| SenticGCN | 74.69 | 72.28 | 84.82 | 79.37 | 79.13 | 76.06 |
| WGAT-SBERT | **76.25** | **74.56** | **85.71** | **80.23** | **80.49** | **77.21** |

* in the table refers to the model results in the published paper.

*4.6. Ablation Studies*

To further verify the effect of the different modules of the model on the final experimental results, several sets of ablation experiments were performed on the three datasets above, as described below.

GAT-Glove: GAT-Glove performs word embedding using the Glove word vector and LSTM for feature extraction, trained by a general graph attention network.

WGAT-Glove: WGAT-Glove performs word embeddings using Glove word vectors and LSTM for feature extraction, trained with a relational weighted graph attention network.

GAT-SBERT: To verify the effect of dependency weighting on the experimental effect, the dependency weights are removed, and only the weights between two nodes are calculated by the attention mechanism.

WGAT-BERT: To verify the effect of sentiment pretrained models on experimental effects, word embeddings are performed on text using a pretrained model on a large-scale general corpus.

As shown in Table 4, changing either part of the model resulted in a lower final classification. First, the use of different word embedding methods had a large impact on the model effects. Both Glove word embedding-based models showed a significant decrease in effectiveness compared to the BERT word embedding-based models, demonstrating the powerful feature extraction capability of the BERT model. The BERT model pretrained on the sentiment corpus was better than the BERT model pretrained on the general corpus, which shows the positive effect of domain data on model training. The dependency weights also had a greater impact on the models for the same word embedding cases. The accuracy of the models increased by 2.13%, 3.2%, and 3.6% with the addition of dependency weights in the Glove word embedding case and 2.27%, 2.12%, and 2.37% with the addition of dependency weights in the SBERT word embedding case. This is because graph attention networks may assign larger attention weights to unimportant nodes in the process of calculating the importance between two nodes. With the addition of dependency weights, the model can assign greater attention weights to nodes that have a strong influence on the classification, reducing the interference of non-disjoint points, with a significant improvement in accuracy.

**Table 4.** Ablation experiment.

| Dataset | Twitter | | Restaurant | | Laptop | |
|---|---|---|---|---|---|---|
| | Acc | F1 | Acc | F1 | Acc | F1 |
| GAT-Glove | 69.92 | 67.68 | 76.64 | 67.91 | 72.35 | 67.37 |
| WGAT-Glove | 72.05 | 69.23 | 79.84 | 70.11 | 75.95 | 72.24 |
| GAT-SBERT | 73.12 | 71.39 | 83.16 | 76.69 | 77.12 | 73.50 |
| WGAT-BERT | 75.13 | 72.76 | 84.75 | 78.53 | 79.32 | 76.03 |
| WGAT-SBERT | **76.25** | **74.56** | **85.71** | **80.23** | **80.49** | **77.21** |

*4.7. Effect of Initial Values of Weights*

To study the effect of the initialisation weights of the dependencies on the effectiveness of the model, we set different initialisation weights for the dependencies and conducted comparative experiments to observe the results and explore the best initialisation weights. Here, we denote the important dependency weights mentioned in Section 3.2 (*a*) and denote the other dependency weights (*b*). The range of *a* values is [1,5], and the range of *b* values is [0.1, 1]. The control variable method was used to find the optimal solution, and the experimental results are shown in Figure 5.

(**a**) Important dependency weights *a*    (**b**) Other dependency weights *b*
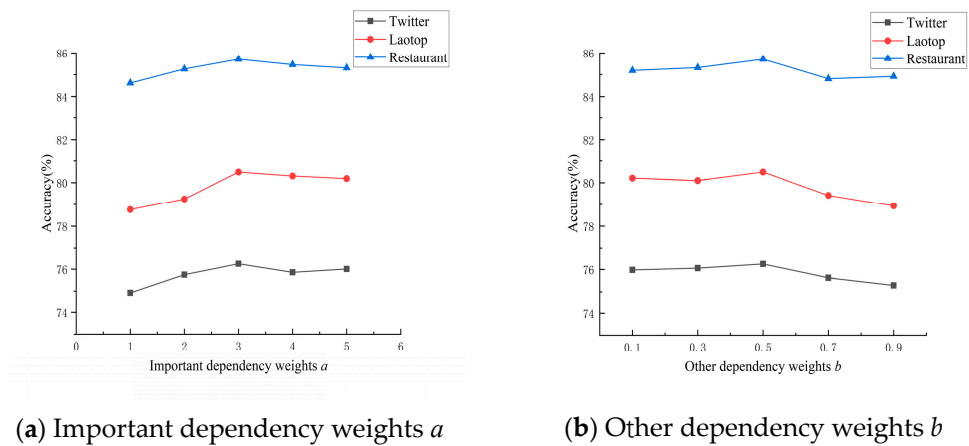
**Figure 5.** Performance of WGAT model on each dataset under different *a* and *b*.

In Figure 5a, $b = 3$, and different values of *a* are set. The accuracy of the model increases as the value of *a* decreases, and the model performance is optimal when $a = 0.5$, after which the model performance stabilises and decreases as the value of *a* decreases. In Figure 5b, $a = 0.5$, and different values of *b* are set. As the value of *b* increases, the accuracy of the model gradually increases and reaches an optimum performance at $b = 3$. Figure 5a,b show that setting larger *a* values and smaller *b* values allows the model to assign greater weight to important dependencies when calculating attention scores, which helps to improve classification accuracy. However, too large *a* values or too small *b* values may also lead to a degradation in model performance. This is due to the model focusing too much on important dependencies, which may lead to classification errors in some specific sentence types. For example, in the sentence "The staff should be more friendly.", the adjective "friendly" modifies the aspect word "staff" through the dependency nsubj, and the adverb "more" modifies the emotive word "friendly" through the dependency advmod. When $a < 0.5$ and $b > 3$, the aspectual word "staff" is predicted to be positive. However, since this sentence is a dummy, the aspectual word is negative.

### 4.8. Visualisation of Attention Mechanisms

WGAT makes the model focus more on the words that contribute more to the aspect words in the calculation of attention through dependencies. Here, we visualise the attention weights for the sentence "Great food but the service is dreadful!". The attention weights for this sentence are shown in Figure 6.

(**a**) Unweighted attention weight matrix    (**b**) Weighted attention weight matrix
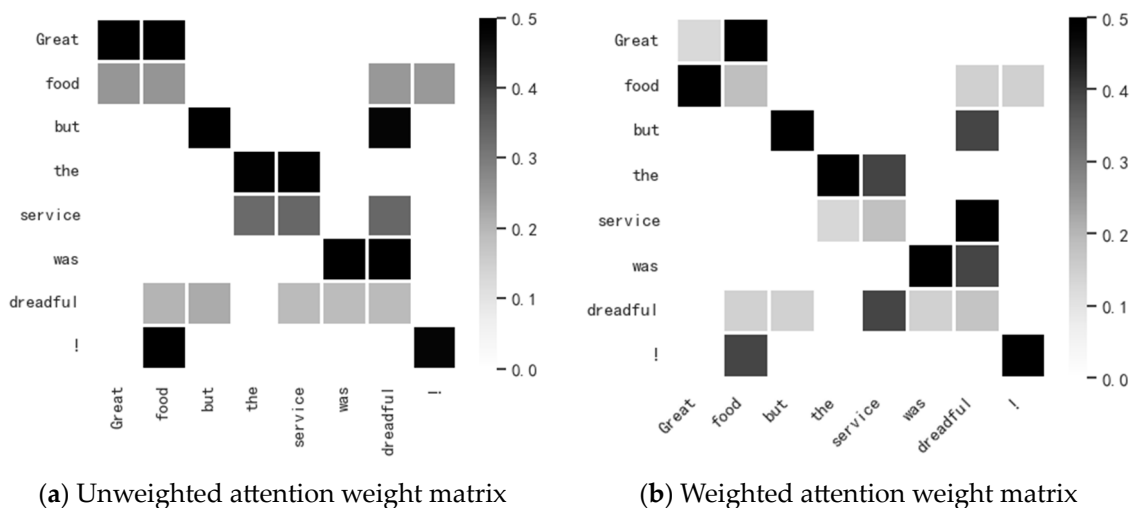
**Figure 6.** Attention visualisation.

As shown in the figure, it is possible for GAT to assign larger weights to unimportant words when calculating the attention of two nodes. For example, the attention weights calculated without relational weights in Figure 6a, where the aspect word "food" is assigned a larger weight for "dreadful", may cause the model to predict "food" as being negative, affecting model accuracy. The addition of dependency weights in Figure 6b allows the aspectual word "food" to focus more on "great" and the aspectual word "service" to focus more on "dreadful", reducing the interference of irrelevant words in the sentiment classification of aspectual words. The example in the figure above shows that the weights we set are better able to help the model find sentiment words corresponding to aspect words during the training process.

## 5. Conclusions

In this paper, we proposed a graph attention network aspect-level sentiment analysis model (WGAT) based on the weighting of dependencies. On the one hand, the influence of different dependencies on sentiment classification is analysed, corresponding weights are set for them, and the optimal weights are found through experiments. On the other hand, the incorporation of relational weights into the graph attention network allows the model to pay more attention to words that are more important to the aspectual sentiment and to ignore words that have a negative impact on the outcome during the training process. In addition, this paper achieved better results by using a BERT model pretrained on a sentiment corpus for word embedding of text. According to comparative experiments on three datasets, the WGAT model achieved the best results in terms of accuracy and F1 values compared to other baseline models, demonstrating the effectiveness of our approach. We also analysed specific cases of this to show more visually how relational weighting contributes to our task.

Although our model achieves better results, it is significantly less accurate than the normal text when faced with abnormal text (e.g., Twitter dataset). In addition, the model sometimes fails to correctly identify the sentiment in complex sentences such as virtual speech, leading to errors in judgment. In the future, more in-depth syntactic analysis will be conducted on more complex sentences to improve the prediction accuracy.

**Author Contributions:** Conceptualisation, T.J., Z.W., M.Y. and C.L.; data curation, T.J., Z.W., M.Y. and C.L.; methodology, T.J. and Z.W.; writing—original draft, T.J. and Z.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China, grant number 61871258.

**Data Availability Statement:** https://alt.qcri.org/semeval2014/task4/ (accessed on 7 March 2023).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhang, L.; Wang, S.; Liu, B. Deep Learning for Sentiment Analysis: A Survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
2. Wang, T.; Yang, W. Review of Text Sentiment Analysis Methods. *Comput. Eng. Appl.* **2021**, *57*, 11–24.
3. Zhang, Y.; Li, T. Review of Comment-oriented Aspect-based Sentiment Analysis. *Comput. Sci.* **2020**, *47*, 7. [CrossRef]
4. Tang, X.; Liu, G. Research Review on Fine-grained Sentiment Analysis. *Libr. Inf. Serv.* **2017**, *61*, 132–140.
5. Wang, S.Y.; Yuan, K. Sentiment analysis model of college student forum based on RoBERTa-WWM. *Comput. Eng.* **2022**, *48*, 292–298, 305.
6. Shen, Y.; Zhao, X. A Review of Research on Aspect-based Sentiment Analysis based on Deep Learing. *Inf. Techol. Stand.* **2020**, *1*, 50–53.
7. Tang, D.; Qin, B.; Feng, X.; Liu, T. Effective LSTMs for target-dependent sentiment classification. In Proceedings of the 26th International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 3298–3307.
8. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for aspect-based sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 606–615.
9. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-based sentiment classification. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 4068–4074.

10. Meng, F.; Feng, J.; Yin, D.; Chen, S.; Hu, M. Sentiment Analysis with Weighted Graph Convolutional Networks. In *EMNLP (Findings)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 586–595.

11. Zhang, C.; Li, Q.; Song, D. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional 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; ACL: Stroudsburg, PA, USA, 2019; pp. 4560–4570.

12. Mubarok, M.S.; Adiwijaya; Aldhi, M.D. Aspect-based sentiment analysis to review products using Naïve Bayes. *AIP Conf. Proc.* **2017**, *1867*, 020060.

13. Kiritchenko, S.; Zhu, X.; Cherry, C.; Mohammad, S. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In Proceedings of the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, 23–24 August 2014; pp. 437–442.

14. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional Transformer for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186.

15. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.

16. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

17. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 214–224.

18. Song, Y.; Wang, J.; Jiang, T.; Liu, Z.; Rao, Y. Targeted sentiment classification with attentional encoder network. In *International Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2019; pp. 93–103.

19. Zeng, B.; Yang, H.; Xu, R.; Zhou, W.; Han, X. LCF: A Local Context Focus Mechanism for Aspect-Based Sentiment Classification. *Appl. Sci.* **2019**, *9*, 3389. [CrossRef]

20. Tang, X.; Xiao, L. Research on Micro-Blog Topics Mining Model on Dependency Parsing. *Inf. Sci.* **2021**, *57*, 11–24.

21. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.

22. Huang, X.; Carley, K. Syntax-aware aspect level sentiment classification with graph attention networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing(EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5469–5477.

23. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.

24. Wang, K.; Shen, W.; Yang, Y.; Quan, X.; Wang, R. Relational Graph Attention Network for Aspect-based Sentiment Analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 6–8 July 2020; pp. 3229–3238.

25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; p. 30.

26. Zhang, D.; Huang, L.; Zhang, R.; Xue, H.; Lin, J.; Yao, L. Fake Review Detection Based on Joint Topic and Sentiment Pre-Training Model. *J. Compt. Res. Dev.* **2021**, *58*, 1385–1394.

27. de Marneffe, M.C.; Manning, C.D. *Stanford Typed Dependencies Manual*; Technical report; Stanford University: Stanford, CA, USA, 2008.

28. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 task 4: Aspect based sentiment analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation, Dublin, Ireland, 23–24 August 2014; 24 August 2014; pp. 27–35.

29. Dong, L.; Wei, F.; Tan, C.; Tang, D.; Zhou, M.; Xu, K. Adaptive recursive neural network for target-dependent twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, 22–27 June 2014; Volume 2, pp. 49–54, Short papers.

30. Zhang, M.; Qian, T. Convolution over Hierarchical Syntactic and Lexical Graphs for Aspect Level Sentiment Analysis. In *Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 3540–3549.

31. Wang, G.; Li, H.; Qiu, Y.; Yu, B.; Liu, T. Aspect-based Sentiment Classification via Memory Graph Convolutional Network. *J. Chin. Inf. Process.* **2021**, *35*, 98–106.

32. Zhu, L.; Zhu, X.; Guo, J.; Dietze, S. Exploring rich structure information for aspect-based sentiment classification. *J. Intell. Inf. Syst.* **2022**, *60*, 97–117. [CrossRef]

33. Liang, B.; Su, H.; Gui, L.; Cambria, E.; Xu, R. Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks. *Knowl.-Based Syst.* **2022**, *235*, 107643. [CrossRef]