


## Article

# Graph Convolutional Networks with Bidirectional Attention for Aspect-Based Sentiment Classification

Jie Liu <sup>1</sup>, Peiyu Liu <sup>1,\*</sup>, Zhenfang Zhu <sup>2</sup> , Xiaowen Li <sup>1</sup> and Guangtao Xu <sup>1</sup>

<sup>1</sup> School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China; 2019020851@stu.sdnu.edu.cn (J.L.); 2019309037@stu.sdnu.edu.cn (X.L.); 2019020878@stu.sdnu.edu.cn (G.X.)

<sup>2</sup> School of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan 250357, China; zhuzf@sdjtu.edu.cn

\* Correspondence: liupy@sdnu.edu.cn; Tel.: +86-131-8889-9297

**Abstract:** Aspect-based sentiment classification aims at determining the corresponding sentiment of a particular aspect. Many sophisticated approaches, such as attention mechanisms and Graph Convolutional Networks, have been widely used to address this challenge. However, most of the previous methods have not well analyzed the role of words and long-distance dependencies, and the interaction between context and aspect terms is not well realized, which greatly limits the effectiveness of the model. In this paper, we propose an effective and novel method using attention mechanism and graph convolutional network (ATGCN). Firstly, we make full use of multi-head attention and point-wise convolution transformation to obtain the hidden state. Secondly, we introduce position coding in the model, and use Graph Convolutional Networks to obtain syntactic information and long-distance dependencies. Finally, the interaction between context and aspect terms is further realized by bidirectional attention. Experiments on three benchmarking collections indicate the effectiveness of ATGCN.

**Keywords:** natural language processing; attention mechanism; aspect-based sentiment classification; graph convolutional network



**Citation:** Liu, J.; Liu, P.; Zhu, Z.; Li, X.; Xu, G. Graph Convolutional Networks with Bidirectional Attention for Aspect-Based Sentiment Classification. *Appl. Sci.* **2021**, *11*, 1528. <https://doi.org/10.3390/app11041528>

Academic Editor: Tobias Meisen

Received: 22 December 2020

Accepted: 3 February 2021

Published: 8 February 2021

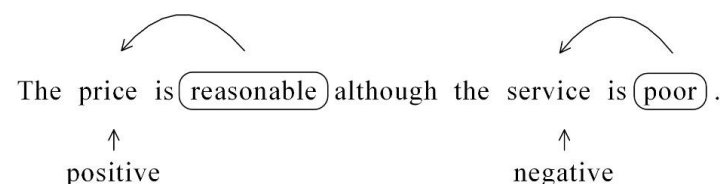
**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Aspect-based sentiment classification (ABSC) is a key subtask of sentiment classification [1], which is a fine-grained task in the field of text sentiment classification. Its objective is to determine the sentiment polarity (such as positive, negative, and neutral) of aspects that appears in a sentence. As a specific example, Figure 1, this sentence expresses positive sentiment about food prices, but it expresses negative sentiment about service.



**Figure 1.** Example of aspect-based sentiment classification.

Some scholars have successively proposed various models based on neural networks, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), etc., and they have achieved outstanding results on multiple datasets. For example: Xue et al. [2] proposed an accurate and effective model based on CNN and gate mechanism. Wang et al. [3] focused on the critical role of capturing different contextual information for a given aspect, and combined the attention mechanism with LSTM to model the semantics of sentences. Li et al. [4] introduced a model, which integrated aspect extraction and target sentiment classification,

this model used two LSTM networks to generate aspect-based sentiment classification results, but it cannot obtain important clues of the aspect. Ruder et al. [5] studied features within and between sentences of the input context to distinguish the sentiment polarity. This model is mainly aimed at the situation where a sentence contains one aspect and it is difficult to recognize the sentiment polarity of sentences with many aspects. However, most neural network models are the same as those mentioned above. They are still immature when dealing with the task of text sentiment classification.

Recently, in order to better portray different contextual information and reflect the influence of different words in sentences on sentiment polarity, some scholars have introduced the attention mechanism into ABSC tasks. The attention mechanism can well solve the problem of determining the sentiment polarity of each aspect in the sentence. It was originally used for image processing to enable neural networks to focus on certain information when processing data. Mnih et al. [6] used attention mechanism in image classification tasks, which verified the effectiveness of the attention mechanism in the image field. The neural network model combined with the attention network has become a major research focus. At present, the attention mechanism has been triumphantly applied in the fields of machine translation [7], reading comprehension [8] and question answering [9]. It is used to strengthen the model's attention to context words that are more closely related to the target semantics. In recent studies, it is common to create aspect precise sentence representations by using attention or transforming sentence representations based on aspect term. In addition, when people are asked to complete the task of ABSC, they will selectively focus on part of the content and the place where the information needs to be established to remember terms. Inspired by this, attention mechanism has been successfully applied, and introduced to the application of neural network structure of ABSC task. For example, Ma et al. [10] designed a model that uses two attention mechanisms to interactively learn aspects and contextual representations. Thence, attention mechanism is indispensable in ABSC task. However, complex recursive neural networks are the main dependence of these studies, acting as sequence encoders to calculate the hidden semantics of the context.

Although the research has made great progress, the results can still be improved. The neural network model for context modeling relies on recurrent neural networks, such as LSTM. LSTM has strong expressive power, but it is difficult to parallelize. The above research recognizes the importance of aspect goals and develops methods for precise modeling contexts by generating aspect-specific representations. However, we think that the truly improvement of the effectiveness of sentiment classification lies in the coordination of aspect and corresponding context. This paper proposed an attention-based graph convolutional network model. In order to make full use of syntactic information and remote dependencies of sentences, we used GCN in the model. In order to achieve and fuse context and aspect information, we used bidirectional attention to obtain useful interactive information.

We summarize main contributions as the following three points:

1. We propose to adopt the syntactic dependency structure in sentences and solve the problem of long-distance multi-word dependence based on aspect-based sentiment classification;
2. We design a bidirectional attention mechanism to enhance the interaction between aspect and context to obtain aspect specific context representation;
3. Experiments on three benchmarking collections indicated that the effectiveness of our model compared with other popular models.

## 2. Related Work

In this section, we primarily describe the related work of ABSC, attention mechanism and graph convolutional networks.

Aspect-based sentiment classification: Aspect-based sentiment classification is usually considered as a branch of sentiment classification and it is a popular research task recently. Some traditional works have designed rule-based models [11] and statistical-based

methods [12], which basically focus on extracting a series of features such as emotional vocabulary features and packet features to train sentiment classifiers [13]. The performance of these models are immensely decided by the effectiveness of feature engineering work. In this field, deep learning have demonstrated powerful context representation capabilities. Compared with manual-based features, they have better scalability. As a result, more and more deep learning methods are involved in ABSC tasks and have achieved significant success [14–16]. In order to further integrate the aspect words into a model, Tang et al. [17] proposed a target-dependent TD-LSTM, which used two Bi-LSTMs to model the left and right contexts of the aspect words respectively. Tang et al. [18] designed MemNet, which mainly included a multi-hop attention and external memory to gain the importance of each contextual word which was related to a given target. Ma et al. [19] integrated common-sense knowledge of emotion-related concepts into training of deep neural networks for aspect-based sentiment classification tasks. In addition, due to the lack of good application of syntactic information and long-distance dependence to aspect-based sentiment classification in previous research, we solved the aspect-based sentiment classification problem through syntactic GCN and attention mechanism.

**Attention mechanism:** Studies have shown that deep neural networks that introduce attention mechanisms [20] have achieved significant success, and they have also been extensively used in ABSC tasks. Wang et al. [21] proposed the hierarchical network of word-level and item-level attention of ABSC, which used attention mechanism to pay attention to the relationship between all clauses. He et al. [22] adopted the attention based LSTM network because they wanted to transfer knowledge through document level data, and imposed some syntactic restrictions on the weight of attention, but the influence of syntactic structure was not fully utilized. Qiang et al. [23] implemented a new multi-attention network that extracted sentiment features with self-and position-aware attention mechanism. Wang et al. [24] extracted global context information using an interactive attentional coding network. Inspired by this, we used multi-head attention to obtain the interaction information between aspect and context, and then we used bidirectional attention to obtain further interaction information. Attention-based models are promising, but they are not sufficient to capture syntactic dependencies between contextual words and various aspects of sentences.

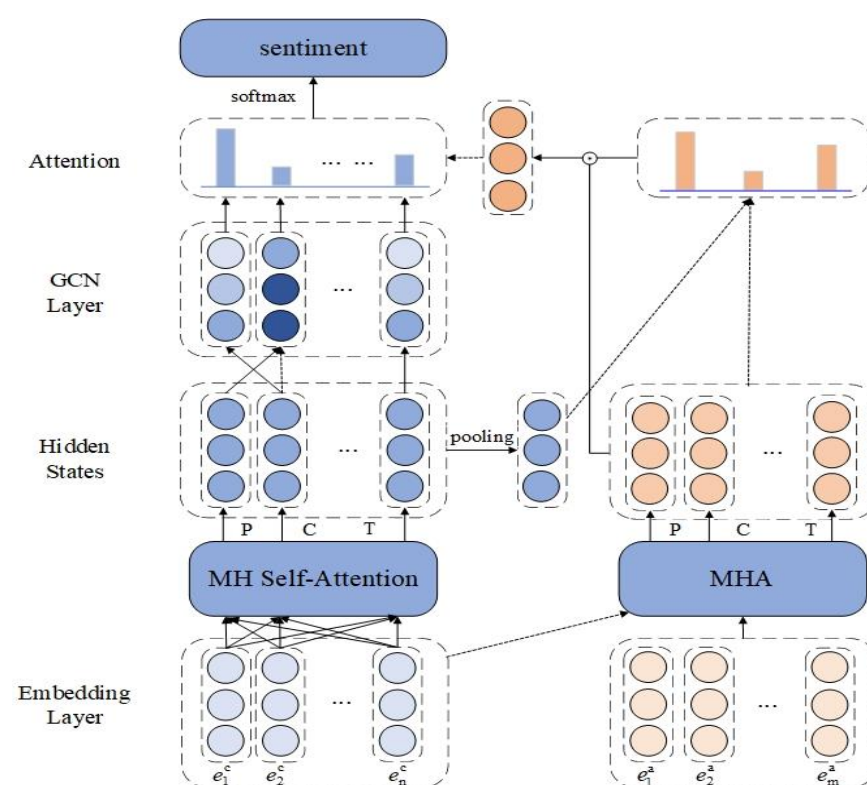
**Graph convolutional network:** The graph convolutional network can be regarded as an adaptation of traditional CNN [25,26], which is used to encode luxuriant aspects of sentence information. Graph convolutional networks are widely used in the field of vision. Recently, GCN has been immensely adopted in NLP, such as semantic role labeling [27], machine translation [28] and relation classification [29]. Since GCN was first introduced [30], GCN has recently achieved very good performance in graph structure representation in the field of natural language processing. For example, in order to use the syntactic information between aspect and context words, Zhang et al. [31] introduced a novel aspect-based GCN to deal with ABSC task using dependency graphs. Yao et al. [32] took the text-word and word-word relationship as a guide and introduced GCN into the context classification task to improve the performance of context classification. Peng et al. [33] adopted a deep network method based on GCN-CNN, which converts text into graphic words, and then uses the graph convolutional network to carry out convolution operation on word graphics. Zhang et al. [34] designed a graph convolution network for text relation extraction, which applied a pruning scheme to the input tree. Marcheggiani et al. [35] used syntactic GCN to integrate the syntactic structure information in the original sentence. As syntactic dependencies are graph-structured data, it is natural to use GCN for encoding. Inspired by this, we use the syntactic dependency structure of sentences to resolve the long-distance multi-word dependency problem in the ABSC.

Our ATGCN model adopted a different method, which utilized multi-head attention network to obtain contextual interaction information, we used point-wise convolution transformation to further analyze aspect terms and contextual information. Considering the influence of position information, we added position coding before graph convolution

network, then we used graph convolution network to gain long-distance dependency and syntax information. Then, we used bidirectional attention network to further enhance the interaction between aspect term and context. Finally, we used the softmax function for sentiment classification.

### 3. Methodology

Figure 2 shows the overall framework structure of the proposed ATGCN, which consists of an embedding layer, an attention encoder layer, a position encoding and the GCN layer, a bidirectional attention layer and an output layer. Next, the rest of the model is described in detail in this chapter.



**Figure 2.** The Overall Architecture of Our Model.

#### 3.1. Embedding Layer

The goal of ABSC is to predict the emotional polarity of aspects in sentences. Given context sequence  $W^c = \{w_1^c, w_2^c, \dots, w_n^c\}$  and aspect sequence  $W^a = \{w_1^a, w_2^a, \dots, w_m^a\}$  where the aspect sequence is a subsequence of the context sequence. In the next section, we concisely introduce our embedding layer, which includes GloVe embedding and BERT embedding.

**GloVe Embedding.** We use the pre-trained embedding matrix GloVe to get the fixed word embedding of each word. Then, each word is represented by an embedding vector  $e_i \in R^{d_e \times 1}$ , where  $d_e$  is the dimension of the word vector. After the embedding layer is completed, the context embedding is marked as matrix  $E^c \in R^{d_e \times n}$ , and the  $i$ -th aspect embedding is marked as matrix  $E^{ai} \in R^{d_e \times m_i}$ .

**BERT Embedding.** As one of the recent innovations in contextualized representation learning, BERT embedding uses the pre-trained BERT to create word vector sequences. To better fit the processing method of data BERT, we transform the given context and target to “[CLS] + context + [SEP]” and “[CLS] + aspect term + [SEP]”.

### 3.2. Attention Encoder Layer

At present, RNN or related conversion models are mostly used to gain the hidden state. Instead of using neural network methods to capture the hidden state, we apply the attention encoder layer to capture the hidden state.

#### 3.2.1. Multi-Head Attention

In order to calculate the selection of multiple input pieces of information in parallel, we used the multi-head attention (MHA) mechanism, which can reduce the complexity of the neural network. MHA is an attention mechanism that can execute multiple attentions simultaneously. We use MHSA to process context embedding. MHA is used to realize the interaction between context sequence and aspect. The goal is to model according to the given context. The substance of the attention mechanism is an addressing process. The attention function maps the query sequence  $q = \{q_1, q_2, \dots, q_m\}$  and our defined key sequence  $k = \{k_1, k_2, \dots, k_n\}$  to output sequence. The corresponding distribution of attention is:

$$Attention(k, q) = \text{softmax}(s(k, q))k \quad (1)$$

where  $s$  represents the score function of learning semantic association. The specific calculation method is as follows:

$$s = \tanh([k_i; q_j] \cdot W_{att}) \quad (2)$$

compared with the ordinary attention mechanism, the MHA can perform parallel calculation on the input information and learn different scores of  $n$  heads in parallel subspaces, which is effective for alignment.

Connect and project the output of  $n$  heads to the specified hidden dimension  $d_{hid}$ ,

$$MHA(k, q) = [O^1; O^2; \dots; O^h] \cdot W_O \quad (3)$$

$$o^h = Attention^h(k, q) \quad (4)$$

the “;” represents the connection of vectors.  $W_O \in R^{d_{hid} \times d_{hid}}$  is the corresponding weight, and  $d_{hid}$  represents the dimension of the hidden state. Among them,  $o^h$  is the output of the  $h$ -th head attention,  $h \in [1, 8]$ .

According to the difference of  $q$  value and  $k$  value, the attention mechanism can be divided into two types, namely MHSA and MHA. MHSA is a special case of the multi-head attention mechanism when  $q = k$ . Given a context embedding  $e^c$ , we can get the context representation by the following methods:

$$c^s = \text{MHSA}(e^c, e^c) \quad (5)$$

where  $c^s = \{c_1^s, c_2^s, \dots, c_n^s\}$ . To emphasize the interaction between context and aspect, we put context embedding and aspect embedding into MHA,

$$a^m = \text{MHA}(e^c, e^a) \quad (6)$$

After the interaction process, each given aspect word  $e_j^a$  will have a combined representation selected from the context embedding  $e^c$ , and we obtain the context-aware aspect representation  $a^m = \{a_1^m, a_2^m, \dots, a_i^m\}$ .

#### 3.2.2. Point-Wise Convolution Transformation

The role of PCT is similar to the conversion of multi-layer perception. The complete connection layer has two dense layers. We use Relu as the activation function of the first layer, then the linear function as the activation function of the second layer. In order

to analyze the information in more detail, we use PCT to convert context and aspect information. The equation is:

$$\text{PCT}(h) = \text{Relu}(h * W^1 + b^1) * W^2 + b^2 \quad (7)$$

where  $*$  is the convolution operator,  $W^1 \in R^{d_{hid} \times d_{hid}}$  and  $W^2 \in R^{d_{hid} \times d_{hid}}$  are the weights of the two convolutional kernels,  $b^1 \in R^{d_{hid}}$  and  $b^2 \in R^{d_{hid}}$  are biases. Finally, the results generated by the context and aspect terms of  $c^s$  and  $a^m$  will be further applied to obtain hidden representations. They are defined as:

$$h^c = \text{PCT}(c^s) \quad (8)$$

$$h^c = \{h_1^c, h_2^c, \dots, h_n^c\} \quad (9)$$

$$h^a = \text{PCT}(a^m) \quad (10)$$

$$h^a = \{h_1^a, h_2^a, \dots, h_m^a\} \quad (11)$$

### 3.3. Position Encoding

As we all know, words that have a significant impact on aspect terms are generally closer to aspect terms. If a word is too far away from the aspect, it has almost nothing to do with this aspect. Based on this intuition, we introduce position encoding. Formally, a given aspect  $W^a$  is one of  $K$  aspects, where  $i \in [1, K]$  is the index of the aspects, and the relative distance  $pos_i$  between the  $t$ -th word and the  $i$ -th aspect is defined as follows:

$$pos_i = \begin{cases} 1, & dis = 0 \\ 1 - \frac{dis}{N}, & 1 \leq dis \leq s \\ 0, & dis > s \end{cases} \quad (12)$$

$pos_i$  is the position weight to  $i$ -th token.  $dis$  is the distance between the context word and the aspect,  $N$  is the length of the context, and  $s$  is a predetermined constant. Finally, we can get a position-aware representation through position information:

$$H(h^c) = pos_i h^c \quad (13)$$

### 3.4. GCN Module

GCN can easily obtain the dependence of the graph through the message passing between the graph nodes, so it is extensively used to deal with data with abundant relationships and interdependence between objects. Specifically, we first obtain the adjacency matrix  $A \in R^{n \times n}$  according to the words in the sentence. It should be noted that GCN is executed in a multi-layered manner. Then, the representation of all nodes are updated with the graph convolution operation with a normalization factor. The specific graph convolution operation is updated as follows:

$$\tilde{h}_i^l = \sum_{j=1}^n A_{ij} W^l g_j^{l-1} \quad (14)$$

$g_j^{l-1}$  is the representation of the  $j$ -th token evolved from the previous GCN. The weights  $W^l$  is trainable parameters.

$$h_i^l = \sigma(\tilde{h}_i^l / (d_i + 1) + b^l) \quad (15)$$

where  $\sigma$  is a nonlinear function, e.g., ReLU.  $d_i = \sum_{j=1}^n A_{ij}$ ,  $g_i^l = H(h^c)$  where  $H$  is the same as the position weight function used by Li et al. [36].

### 3.5. Bidirectional Attention

We use a bidirectional attention in the model to gain interactive information between context and aspects. First, the attention mechanism is used to obtain a new aspect representation from context to aspect.

Context to aspect attention: By averaging the hidden output of the context, we get  $\bar{h}^c$ . Among them  $\bar{h}^c \in R^{d_{hid} \times 1}$ , for each aspect of the hidden word vector, the calculation formula of the attention weight is as follows:

$$f_1(\bar{h}^c, h_{a_i}^a) = \bar{h}^c{}^T \cdot W_a \cdot h_{a_i}^a \quad (16)$$

$$\beta = \frac{\exp(f_1(\bar{h}^c, h_{a_i}^a))}{\sum_{i=1}^M \exp(f_1(\bar{h}^c, h_{a_i}^a))} \quad (17)$$

where  $W_a \in R^{d_{hid} \times d_{hid}}$  is the attention weight matrix,  $\bar{h}^c \in R^{d_{hid} \times 1}$  is received by average pooling,  $h_{a_i}^a \in R^{d_{hid} \times 1}$  is the hidden word vector. By calculating the attention weight of the word, the hidden representation of the aspect is obtained:

$$r = \sum_{i=1}^M \beta \cdot h_{a_i}^a \quad (18)$$

Aspect to context attention: Through the obtained new aspect representation, the important features related to the aspect in semantics are retrieved from the hidden vector, and the attention weight based on retrieval is set for each context. The specific calculation is as follows:

$$f_2(r, h_i^L) = r^T \cdot W_c \cdot h_i^L \quad (19)$$

$$\alpha = \frac{\exp(f_2(r, h_i^L))}{\sum_{i=1}^N \exp(f_2(r, h_i^L))} \quad (20)$$

where  $W_c \in R^{d_{hid} \times d_{hid}}$  is the attention weight matrix,  $h_i^L$  is the output representation of GCN. The above formula is used to calculate the semantic relevance of the aspect and context words. The final representation of the forecast:

$$\gamma = \sum_{i=1}^n \alpha \cdot h^c \quad (21)$$

### 3.6. Output Layer

We input the obtained representation  $\gamma$  into the fully connected layer,

$$Z = \text{softmax}(W_z \gamma + b_z) \quad (22)$$

where  $W_z$  is the learned weight and  $b_z$  is the bias. The proposed ATGCN is optimized by the standard gradient descent algorithm with the cross-entropy loss and L2-regularization.

$$\text{Loss} = - \sum_{(d, y_Z) \in D} \log Z_{y_Z} + \lambda \|\theta\|_2 \quad (23)$$

where  $D$  means the collection of datasets, and  $Z_{y_Z}$  is the  $y_Z$ -th element of  $Z$ .  $\lambda$  is the coefficient for L2-regularization, and  $\theta$  denotes all trainable parameters.

## 4. Experiments

Firstly, we described the three datasets and experimental setup for the experiment. Secondly, in order to distinguish ATGCN from other aspects of sentiment classification



methods, we compared ATGCN with several accepted models. Finally, in order to prove the reliability of our ATGCN, we conducted experiments from multiple perspectives.

#### 4.1. Experimental Datasets

Our experiments are conducted on three public and authoritative standard datasets: TWITTER, LAP14, and REST14. TWITTER is a dataset created by Dong et al. [37] from the TWITTER review. LAP14 and REST14 are datasets in the SemEval task [38] in 2014. These datasets are labeled with three emotional polarities, namely positive, neutral and negative. It's easy to distinguish between positive and negative. But the classification of neutral emotion words is easy to make mistakes, because neutral means that the word is neither positive nor negative. The Table 1 shows the number of each sentiment category of train and test examples in each category.

**Table 1.** The details of the experimental datasets.

Dataset	Positive		Neural		Negative	
	Train	Test	Train	Test	Train	Test
<b>Twitter</b>	1561	173	3127	346	1560	173
<b>Restaurant</b>	2164	728	637	196	807	196
<b>Laptop</b>	994	341	464	169	870	128

#### 4.2. Experimental Settings

In this section, we will introduce some experimental details. We use pre-training matrix GloVe and pre-training BERT to initialize word embedding. For GloVe, the dimension of each word vector is 300. There are two sizes of BERT namely,  $BERT_{BASE}$  and  $BERT_{LARGE}$ . In this paper, we mainly use  $BERT_{BASE}$ , both word and aspect embedding dimensions are initialized to 768. The hidden state of dimension hidden layer after processing is set to 300. All weights are initialized according to the gloot unified policy, and the other weights are uniformly distributed. In training, our model is trained using the Adam [39] optimizer with a learning rate is 0.00001. We set dropout to 0.1 and the batch size is 32. In addition, the L2 regularization coefficient is  $10^5$ . Experimental results show that when the number of GCN layers is set to 2, the model performs best. We use Pytorch to implement our model, and the experimental results are obtained by randomly initializing an average of 3 runs. Accuracy (Acc) and Macro-F1 (F1) are used as the evaluation metrics. Compared with other methods, our model has achieved the latest results on two evaluation indicators.

#### 4.3. Model for Comparison

In order to comprehensively evaluate ATGCN, we compare ATGCN with some baselines and state-of-the-art models. The models we consider include models based on syntax and neural network:

##### 4.3.1. Syntactic-Based Model

AdaRNN adaptively transmits the sentiments of words based on context and syntactic relationships [37].

PhraseRNN is a recursive neural network that takes into account both sentence dependencies and composition trees [40].

LSTM + SynATT + TarRep integrates syntax information into the attention.

##### 4.3.2. Neural Network Models

TD-LSTM constructs aspect-specific representations by using the upper left and upper right context of an aspect, and then uses the two LSTMS to model the target for the upper left and upper right context, respectively. In order to predict the affective polarity of aspect, the related representations of the left and right targets are connected.



ATAE-LSTM proposes an attention-based LSTM model that takes the aspect representation and the word embedded as input, and calculates the attention weights using the hidden state of the LSTM. The attention is used to get sentence representation, and finally the aspect affective polarity is classified. Not only does it solve the problem of inconsistency between word vectors and aspect embeddings, but it also captures the most important information that corresponds to a given aspect.

MemNet: each word in the sentence is converted into a word vector for representation, and then the weighted sum of the word vectors is used to obtain the context vector. Since the attention mechanism does not have the ability to model the location information, the location embedded in the model is introduced to encode the location information.

IAN designs the interactive relation between modeling and context model, the model with the context and the target of LSTM network, respectively, based on attention generated terminology and context, said the average pool and pay attention to the mechanism analysis of the weight of different words to find out the important information of the context and specific goal, the final context and aspect representation are linked to predict the sentiment polarity of the aspect.

AEN models context and aspect separately. AEN uses a submodule similar to Transform to obtain hidden state. During the training process, the model adopts a unique label smoothing regularization method in the loss function. AEN is to model the context and aspects separately, and the submodule similar to transform is used to obtain the hidden state, and the model employs the label smoothing regularization method in the loss function.

AS-GCN proposes an aspect-oriented graph convolutional network, which uses Bi-LSTM to obtain the context information of words, and then multi-layered graph convolutional network to capture syntactic information and long distance dependencies, after which only aspect-oriented information characteristics are retained, and the generated representation is used to predict the emotional polarity based on aspects.

Bert-pair-QA-M transforms aspect-based sentiment classification into sentence pair classification task, and fine-tuned BERT's pretraining model [41].

#### 4.4. Results and Analysis

##### 4.4.1. Effectiveness of ATGCN

In this section we report our experimental results and analyze them extensively. In order to evaluate the performance of ATGCN, we use Accuracy and Macro-F1 metrics, as shown in Table 2, our experiments almost always outperform all the comparison models on the three datasets.

**Table 2.** Overall performance of accuracy and F1 on three datasets.

Model	Twitter		Lap14		Rest14	
	Acc	F1	Acc	F1	Acc	F1
AdaRNN	66.30	65.90	-	-	60.42	45.73
PhraseRNN	-	-	-	-	66.20	59.32
LSTM+SynATT+TarRep	70.46	67.55	70.87	66.53	79.33	69.25
TD-LSTM	70.80	69.00	68.13	-	75.63	-
ATAE-LSTM	-	-	68.70	-	77.20	-
MemNet	71.48	69.90	70.64	65.17	79.61	69.64
IAN	72.50	70.81	72.05	67.38	79.26	70.09
AEN	72.83	69.81	73.51	69.04	80.98	72.14
BERT-pair-QA-M	<b>74.28</b>	72.38	78.21	73.56	81.96	73.29
ASGCN	72.15	70.40	75.55	71.05	80.77	72.02
ATGCN(Glove)	73.02	70.98	75.92	71.70	81.85	73.31
ATGCN(BERT)	74.26	<b>73.81</b>	<b>80.06</b>	<b>75.61</b>	<b>82.79</b>	<b>75.10</b>

In order to evaluate the performance of ATGCN, we use Accuracy and Macro-F1 metrics, as shown in Table 2, our experiment consistently outperforms all comparison models on the three datasets.

We can know intuitively that Table 2 indicates the comparison of ATGCN with other methods. The proposed ATGCN model achieved a significant improvement over the GloVe based model in all datasets, such as TD-LSTM, MemNet and IAN. In addition, the performance of ATGCN (BERT) is better than the BERT-based models, such as Bert-pair-QA-M, indicating that ATGCN is effective even if the pre-trained BERT [42] model is used to aspect-based sentiment classification. Specifically, our model performs better than other model, and in most cases our model significantly outperformed the strongest baseline model.

According to the experimental results, in terms of accuracy and macro F1, ATGCN performs better than the previously proposed syntactic-based models (such as AdaRNN, LSTM+SynATT+TarRep, and PhraseRNN), which proves that our model can capture syntactic information more effectively.

In the model based on neural network, we can observe that the performance of TD-LSTM model is poor, because it considers the role of aspect information roughly, and it does not make full use of the correlation information between the aspect word and its corresponding context, so that the overall effect is poor.

The performance of ATAE-LSTM is superior to TD-LSTM. However, its performance is poor because LSTM is inefficient in capturing message. It does not make a comprehensive two-way analysis of the influence between aspect and context, but it only considers the use of attention mechanism to get important parts of sentences. Our approach not only enables multiple interactions between contexts and aspects, but also uses MHA to analyze the impact of words simultaneously.

One of the advantages of MemNet is the introduction of location embedding, which performs marginally better than the two models above, because the attention mechanism does not have the ability to model location information. In general, we believe that adjacent words have a significant influence on aspect terms. Therefore, the introduction of position coding can improve the classification effect. However, the hidden semantics of embedding were not modeled, and the outcome of the last focus was substantially a linear combination of word embedding, resulting in poor performance.

IAN achieved better performance on the three datasets because context and aspect interactions were accomplished. IAN uses two kinds of attention to perform ABSC tasks. However, compared to the approach proposed in this article, IAN is still not good enough, possibly because its context and target word interactions are coarse-grained, which may result in the loss of interaction information.

AEN performs better than IAN on the LAP14 and Rest14 datasets because it not only implements the interaction between context and aspect well, but also uses a submodule similar to Transform to capture hidden state. However, compared with our experimental results, there is still a certain gap, may not take into account the syntactic information and long-distance dependence.

ASGCN is the first to use the graph neural network for sentiment classification, showing good performance, but ASGCN is slightly worse than our proposed approach. As our method takes into account the interaction between context and aspect, this adequately explains the efficiency of our proposed model.

To further demonstrate the effectiveness of our ATGCN, we show the configuration and running time of the experiment in Table 3. We carry out ASGCN in our experimental environment. In the same environment, the running time of our ATGCN is less, which indicates that our model is more lightweight and efficient.

**Table 3.** Experimental configuration.

	ASGCN	ATGCN (GloVe)
<b>GPU</b>	2080ti	20180ti
<b>Programing language</b>	Python 3.6	Python 3.6
<b>Pytorch</b>	1.1.0	1.1.0
<b>Training Time</b>	5.5 h	4 h

#### 4.4.2. Ablation Experiment

In order to further research the impact of each part of ATGCN in performance, we also designed several ablation experiments. As shown in Table 4, in terms of Accuracy and Macro-F1 value, the performance of ATGCN ablation experiment is not as good as that of the ATGCN. The results show that these discarded components are critical and essential to model performance.

**Table 4.** Ablation study results.

Model	Twitter		Lap14		Rest14	
	Acc	F1	Acc	F1	Acc	F1
ATGCN w/o pos	<b>73.16</b>	70.63	72.39	68.15	<b>81.03</b>	<b>73.52</b>
ATGCN w/o Biatt	72.80	69.59	72.07	69.25	78.83	69.01
ATGCN-Siatt	72.91	69.73	<b>73.68</b>	<b>69.93</b>	79.37	70.95
ATCNN	72.65	<b>70.83</b>	73.64	68.91	79.24	69.57

After removing the position encoder module, the model becomes ATGCN w/o pos. We can observe the performance changes on the three datasets. The performance on the Twitter has slightly improved, but the performance on the Lap14 dataset is significantly reduced. According to the experimental results of Rest14, we found that if the grammar is not critical to the context, then the position weight is not helpful in reducing the noise of the context content.

Remove the bidirectional attention module, namely ATGCN w/o Bi-att. We have observed that removing the bidirectional attention module has a key impact on the performance of the model, indicating that the bidirectional attention module is very important to the model. Moreover, the performance of ATGCN w/o pos is better than ATGCN w/o Bi-att, indicating that the bidirectional attention module is more important.

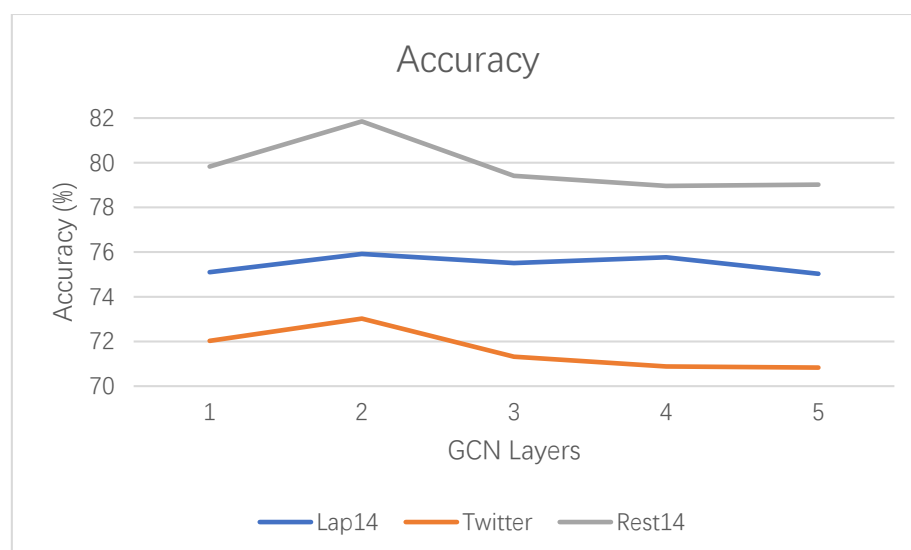
The bidirectional attention module is replaced with single-directional attention, namely ATGCN-Siatt. It is noteworthy that our model is superior to ATGCN-Siatt, which indicates that our model can enhance the interaction between aspects and context and significantly improve performance.

Remove the GCN module and use CNN instead, that is, ATCNN. It is not difficult to find that the performance of ATGCN is always better than ATCNN, which indicates that ATGCN performs better in capturing remote dependence.

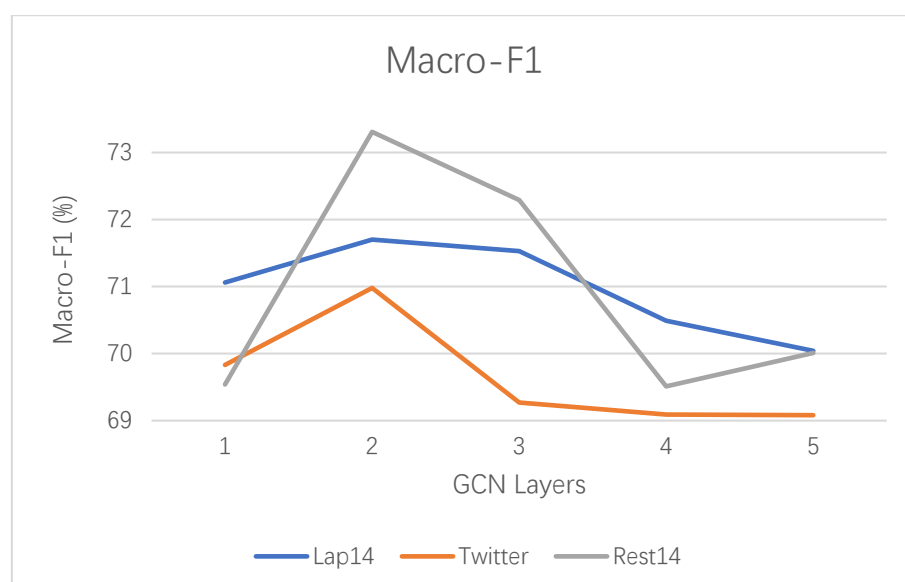
As mentioned above, the performance of ATGCN on the three datasets is better than that of the syntactic-based model proposed before, which proves that ATGCN can capture syntactic information more effectively. Therefore, GCN and attention mechanism have a significant impact on the performance of our model. All discarded components are essential and vital parts of our model.

#### 4.4.3. Impact of GCN Layers

According to the experimental results, we know that the number of GCN layer is an essential parameter affecting the final performance. The number of GCN layers is a very important setting parameter that affects the final performance. In order to study the effect of the number of GCN layers on the final performance, we experimented ATGCN (GloVe) at different GCN layers from 1 to 5. Figure 3 shows the experimental results of accuracy. Figure 4 shows the experimental results of Macro-F1. Basically, we can see from the Figures 3 and 4 that, in most cases, our model will get the best or acceptable results when the number of GCN layers is 2. Specifically, when the number of GCN layers increases, with the increase of the number of GCN layers, the performance indicators of the dataset show a trend of first increasing and then decreasing. With the increase of the number of layers, the training and over fitting of the model become more difficult, which may be the reason for the performance degradation of the model.



**Figure 3.** Accuracy results of GCN at different layers on three datasets.



**Figure 4.** Macro-F1 results of GCN at different layers on three datasets.

From Figures 3 and 4, we can observe that Accuracy and Macro-F1 fluctuate in the same trend on Rest14 and Twitter datasets as the number of GCN layers changes. In the three datasets, Rest14 result is the best, Lap14 is the second, and Twitter is the worst. The reason for this result is caused by the dataset itself. The reason Lap14 has worse results than Rest14 is that the Lap14 has more samples of implicitly expressed sentiment. The more samples of implicit expression of sentiment, the more difficult the task of sentiment classification. As Lap14 has more implicit expressions than Rest14 and Twitter, Lap14 fluctuates less with the number of GCN layers. The Twitter dataset has the worst results because the Twitter dataset samples are more colloquial and the dataset quality is lower.

#### 4.4.4. Case Study

In order to understand more intuitively how ATGCN works, we adopt the example from the restaurant dataset as a case study. The case we chose has two aspects, namely price and service. In particular, we visualize in Figure 5 the attention scores of the IAN and ATGCN models for sentences and aspects, and their predictions for these examples. The attention weight assigned by the attention mechanism is represented by the depth

of the color. The lighter the color, the lighter the weight. From the example, we can see that there are two aspects of “price” and “service” in the sentence. Both visualized models can accurately predict sentiment polarity. In the context, the common words “the” and punctuation marks “.” rarely attract attention. This shows that certain modifiers and punctuation can hardly determine the sentiment polarity of aspect terms. It is worth noting that in the ATGCN prediction, in addition to the word “reasonable”, the conjunction “although” was also noticed. This phenomenon suggests that our model captures both aspects of sentiment dependence through “although”. In other words, it not only pays attention to the corresponding word information for predicting the sentiment polarity of each aspect, but also considers the context information that helps to judge the relationship between different aspects. Therefore, we can model the interaction between context and aspect to capture perceptual dependencies.

Model	Aspect	Sentence										Pred	Ans
IAN	price	The	price	is	reasonable	although	the	service	is	poor	.	pos	pos
	service	The	price	is	reasonable	although	the	service	is	poor	.	neg	neg
ATGCN	price	The	price	is	reasonable	although	the	service	is	poor	.	pos	pos
	service	The	price	is	reasonable	although	the	service	is	poor	.	neg	neg

Figure 5. The case predicted by IAN and ATGCN.

## 5. Conclusions and Future

We designed a novel aspect-based sentiment analysis model (ATGCN). Through research, we pointed out the deficiencies of the existing methods. Firstly, the current methods do not well analyze the role of words and long-distance dependence in sentences. Secondly, there is insufficient interaction and fusion of aspect and context information. In order to solve the first problem, we used graph convolutional networks to obtain syntactic information and long-distance dependencies within sentences. In order to effectively realize the interaction between aspect terms and context, we used multi-headed attention to achieve the first interaction to obtain interactive information, then bidirectional attention is used to further obtain effective interactive information. Finally, we conducted experiments on three open standard datasets. The experimental results showed that ATGCN is effective for aspect-based sentiment classification.

Although our proposed approach has great potential in the field of aspect-based sentiment classification, it still has shortcomings. We ignored the importance of commonsense knowledge for the aspect-based sentiment classification. In future work, we will use graph convolutional networks to apply commonsense knowledge to our tasks. In fact, this is a challenging research hotspot, and we figure that commonsense knowledge will play a key role in the field of sentiment classification in the near future.

**Author Contributions:** Conceptualization, J.L. and P.L.; Funding acquisition, P.L.; Methodology, J.L.; Project administration, Z.Z.; Resources, P.L. and Z.Z. Writing—original draft, J.L.; Writing—review and editing, X.L., G.X.; Visualization, J.L.; Investigation, J.L.; Validation, J.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** Our work was supported by the Science Foundation of the Ministry of Education of China (no. 14YJC860042), the National Social Science Fund (19BYY076), and the Shandong Provincial Social science Planning Project (no. 19BJCJ51/18CXWJ01/18BJYJ04/16CFXJ18).

**Acknowledgments:** Thanks to all commenters for their valuable comments.

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

## References

1. Zhang, L.; Liu, B. Sentiment Analysis and Opinion Mining. *Synth. Lect. Hum. Lang. Technol.* **2016**, *30*, 167.
2. Xue, W.; Li, T. Aspect Based Sentiment Analysis with Gated Convolutional Networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 2514–2523.
3. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for Aspect-level 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.
4. Li, X.; Bing, L.; Li, P.; Lam, W. A Unified Model for Opinion Target Extraction and Target Sentiment Prediction. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 6714–6721.
5. Ruder, S.; Ghaffari, P.; Breslin, J.G. A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 999–1005.
6. Minh, V.; Heess, N.; Graves, A. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems 27, Montreal, QC, Canada, 8–13 December 2014; pp. 2204–2212.
7. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
8. Gu, S.; Zhang, L.; Song, H.; Hou, Y. A position-aware bidirectional attention network for aspect-level sentiment analysis. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 774–784.
9. Andreas, J.; Rohrbach, M.; Darrell, T. Learning to compose neural networks for question answering. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1545–1554.
10. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 4068–4074.
11. Qi, X.; Liao, R.; Jia, J.; Fidler, S.; Urtasun, R. 3D Graph Neural Networks for RGBD Semantic Segmentation. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5209–5218.
12. Li, Y.; Ouyang, W.; Zhou, B.; Shi, J.; Zhang, C.; Wang, X. Factorizable Net: An Efficient Subgraph-based Framework for Scene Graph Generation. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 346–363.
13. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Proceedings of the NIPS 2016, Barcelona, Spain, 5–10 December 2016; pp. 3837–3845.
14. Zhang, M.; Zhang, Y.; Vo, D. Gated neural networks for targeted sentiment analysis. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 3087–3093.
15. He, R.; Lee, W.S.; Ng, H.T.; Dahlmeier, D. Exploiting Document Knowledge for Aspect-level Sentiment Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–12 July 2018; Volume 2, pp. 579–585.
16. Zhou, J.; Chen, Q.; Huang, J.X.; Hu, Q.V.; He, L. Position-Aware Hierarchical Transfer Model for Aspect-Level Sentiment Classification. *Inform. Sci.* **2020**, *513*, 1–16. [[CrossRef](#)]
17. 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.
18. 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.
19. Ma, Y.; Peng, H.; Cambria, E. Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 5876–5883.
20. Jianming, Z.; Fei, C.; Taihua, S.; Chen, H. Self-Interaction Attention Mechanism-Based Text Representation for Document Classification. *Appl. Sci.* **2018**, *8*, 613.
21. Wang, J.; Li, J.; Li, S.; Kang, Y.; Zhang, M.; Si, L.; Zhou, G. Aspect sentiment classification with both word-level and clause-level attention networks. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 4439–4445.
22. Nguyen, H.T.; Nguyen, M.L. Effective Attention Networks for Aspect-level Sentiment Classification. In Proceedings of the 10th International Conference on Knowledge and Systems Engineering, Ho Chi Minh City, Vietnam, 1–3 November 2018; pp. 25–30.
23. Qiang, Y.; Li, X.; Zhu, D. Toward Tag-free Aspect Based Sentiment Analysis: A Multiple Attention Network Approach. In Proceedings of the 2020 International Joint Conference on Neural Networks, Glasgow, UK, 19–24 July 2020.
24. Wang, R.; Tao, Z. Interactive Attention Encoder Network with Local Context Features for Aspect-Level Sentiment Analysis. In Proceedings of the 2020 IEEE/CIC International Conference on Communications in China, Chongqing, China, 9–11 August 2020.
25. Kim, H.; Jeong, Y.S. Sentiment Classification Using Convolutional Neural Networks. *Appl. Sci.* **2019**, *9*, 2347. [[CrossRef](#)]



26. Lee, J.; Park, J.; Kim, K.L.; Nam, J. Samplecnn: End-to-End Deep Convolutional Neural Networks Using Very Small Filters for Music Classification. *Appl. Sci.* **2018**, *8*, 150. [\[CrossRef\]](#)
27. Marcheggiani, D.; Titov, I. Encoding sentences with graph convolutional networks for semantic role labeling. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1507–1516.
28. Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; Simaan, K. Graph convolutional encoders for syntax-aware neural machine translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1947–1957.
29. Li, Y.; Jin, R.; Luo, Y. Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (Seg-GCRNs). *J. Am. Med. Inform. Assoc. JAMIA* **2018**, *26*, 262–268. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Thomas, N.K.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. Available online: <https://arxiv.org/abs/1609.02907> (accessed on 12 December 2020).
31. 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, Hong Kong, China, 3–7 November 2019; pp. 4568–4578.
32. Yao, L.; Mao, C.; Luo, Y. Graph Convolutional Networks for Text Classification. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 7370–7377.
33. Peng, H.; Li, J.; He, Y.; Liu, Y.; Bao, M.; Wang, L.; Song, Y.; Yong, Q. Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1063–1072.
34. Zhang, Y.; Qi, P.; Christopher, D.M. Graph convolutional over pruned dependency trees improves relation extraction. In Proceedings of the EMNLP, Brussels, Belgium, 31 October–4 November 2018; pp. 2205–2215.
35. Marcheggiani, D.; Bastings, J.; Titov, I. Exploiting Semantics in Neural Machine Translation with Graph Convolutional Networks. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 2, pp. 486–492.
36. Li, X.; Bing, L.; Lam, W.; Shi, B. Transformation Networks for Target-Oriented Sentiment Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 946–956.
37. 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, MD, USA, 23–25 June 2014; Volume 2, pp. 49–54.
38. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 task 4: Aspect based sentiment analysis. In Proceedings of the International Workshop on Semantic Evaluation, Dublin, Ireland, 23–24 August 2014; pp. 27–35.
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015.
40. Nguyen, T.H.; Shirai, K. PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2509–2514.
41. Sun, C.; Huang, L.; Qiu, X. Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. 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; Volume 1, pp. 380–385.
42. Wang, Q.; Liu, P.; Zhu, Z.; Yin, H.; Zhang, Q.; Zhang, L. A Text Abstraction Summary Model Based on BERT Word Embedding and Reinforcement Learning. *Appl. Sci.* **2019**, *9*, 4701. [\[CrossRef\]](#)