applied sciences

MDPI

Article

Targeted Sentiment Classification Based on Attentional Encoding and Graph Convolutional Networks

Luwei Xiao [1,†], Xiaohui Hu [1,†,*], Yinong Chen [2], Yun Xue [1], Donghong Gu [1], Bingliang Chen [1] and Tao Zhang [3]

[1] School of Physics and Telecommunication Engineering, Guangdong Provincial Key Laboratory of Quantum Engineering and Quantum Materials, South China Normal University, Guangzhou 510006, China; louisshaw008@gmail.com (L.X.); xueyun@scnu.edu.cn (Y.X.); aagudh123@gmail.com (D.G.); bingliangchen97@gmail.com (B.C.)

[2] School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, USA; yinong@asu.edu

[3] Guangdong China Construction Pulian Technology Co., Ltd, Guangzhou 510000, China; taozhangsos@126.com

* Correspondence: huxh@scnu.edu.cn

† These authors contributed equally to this work.

Received: 6 January 2020; Accepted: 21 January 2020; Published: 2 February 2020

Abstract: Targeted sentiment classification aims to predict the emotional trend of a specific goal. Currently, most methods (e.g., recurrent neural networks and convolutional neural networks combined with an attention mechanism) are not able to fully capture the semantic information of the context and they also lack a mechanism to explain the relevant syntactical constraints and long-range word dependencies. Therefore, syntactically irrelevant context words may mistakenly be recognized as clues to predict the target sentiment. To tackle these problems, this paper considers that the semantic information, syntactic information, and their interaction information are very crucial to targeted sentiment analysis, and propose an attentional-encoding-based graph convolutional network (AEGCN) model. Our proposed model is mainly composed of multi-head attention and an improved graph convolutional network built over the dependency tree of a sentence. Pre-trained BERT is applied to this task, and new state-of-art performance is achieved. Experiments on five datasets show the effectiveness of the model proposed in this paper compared with a series of the latest models.

Keywords: targeted sentiment classification; attentional encoding; graph convolutional network; pre-trained BERT

1. Introduction

Natural language processing is an important part of the new generation of artificial intelligence, particularly in human–machine interaction [1]. All the major computing companies have integrated or are integrating natural language processing capacity in their systems. Targeted sentiment classification [2,3] is a basic task of natural language processing that has attracted a great deal of attention in recent years. It is a fine-grained task in sentiment analysis, and it aims to predict the emotional polarity of each target within a sentence. For example, in the sentence "the price is reasonable while the service is poor", the emotional polarities of two targets "price" and "service" are positive and negative, respectively. A specific target is usually an entity or an aspect term.

Usually, researchers use machine learning algorithms to classify the sentiment of the given targets in a sentence. Some early work used handcraft features, such as sentiment lexicon and language bag-of-words features to train classifiers for a specific target sentiment classification [4,5]. However, these methods are highly dependent on the quality of the selected features and require a large amount of manual feature engineering. In later studies, various neural-network-based methods became popular [6,7], which did not need manual feature engineering. Most of them are based on long short-term memory (LSTM) neural networks [8], and some of them are convolutional neural networks (CNNs) [9]. Many of these neural-network-based methods embed specific target information into the sentence representation via an attention mechanism [7]. Some studies have applied attention mechanisms to generate target specific sentence representations [10,11], or to transform sentence representations according to the target words [12]. However, these studies rely on complex recurrent neural networks (RNNs) as sequence encoders to infer the hidden semantics of the context.

The first problem of the previous studies is that the semantic modeling only uses RNNs combined with the traditional attention mechanism. Each output state of RNNs depends on the previous state, while in semantic modeling, long-distance semantic information may be lost and the parallel computing of input data cannot be carried out [13]. In addition, the traditional attention mechanism is prone to introduce excessive noise because the distribution of weight values is too scattered, and thus it is difficult to accurately extract enough contextual sentiment information related to a specific target. Self-attention [14] is a novel attention mechanism. In sequence-to-sequence (Seq2Seq) tasks, experimental results have demonstrated that this method performs more satisfactorily than traditional RNNs in capturing semantic information.

Another problem in previous research is that these methods largely ignore the syntactic structure of the sentence, while in fact the syntactic structure helps to identify the emotional characteristics directly related to the specific target. When a specific target term is separated from its affective phrase, it is difficult to find related affective words in its surrounding words. The CNN-based models perceive multi-word features as continuous words by convolution of word sequences, whereas it is not sufficient to determine the sentiment expressed by multiple words that are not adjacent to each other [15]. Take the following sentence as an example: “The hotpot, though served with poor service, is actually delicious”. As “delicious” and the target word “hotpot” in the word sequence are a little farther away from each other, the CNN models cannot capture the remote word dependency. On the other hand, in the syntactic dependency tree, the word “delicious” is closer to the target “hotpot” (see Figure 1). In addition, the use of syntactic dependency trees also helps to solve the potential ambiguity in word sequences [16]. In the simple sentence “nice beef terrible juice”, nice and terrible can be used interchangeably. It is difficult to distinguish which word “nice” or "terrible" is related to the target word “beef” or “juice” if only the traditional attention-based method is applied. However, if a person has a good knowledge of grammar, she can easily realize that “nice” is the adjective modifier of “beef”, and “terrible” is the modifier of “juice”. Since the structure of the syntactic dependency tree is similar to the graph structure and the graph convolutional network (GCN) [17] is an effective convolutional neural network that is able to directly operate on graphs, this paper proposes an improved GCN to better extract and integrate the syntactic information displayed in the syntactic dependency tree of the sentences. Overall, semantic information and syntactic information are both crucial for determining the sentiment polarity of a specific target and this paper tries to embed the abundant semantic information and syntactic information into the word representation and specific aspect representation.

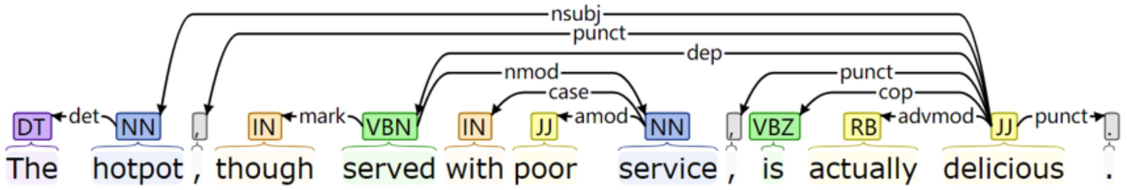


Figure 1. An example of a syntactic dependency parse tree. NN: noun. DT: determiner. IN: preposition or conjunction, subordinating. JJ: adjective or numeral, ordinal. nsubj: nominal subject.

The main contributions of this paper are as follows:

- This paper proposes the novel attentional-encoding-based graph convolutional network (AEGCN) model, which leverages the syntactic structure of a sentence and utilizes multi-head self-attention combined with LSTM to capture context features and specific target features concurrently. The AEGCN model combines semantic information and syntactic information to predict the sentiment polarity of a targeted aspect.
- Syntactic information has not attracted enough attention in many related studies. This paper builds an improved graph convolutional network with point-wise convolution over the dependency tree of a sentence to extract syntactic information and utilize a multi-head self-attention to obtain the syntactic information encoding.
- This paper evaluates the proposed method on five datasets. Experiments show that the AEGCN achieved competitive performance over the state-of-the-art approaches. This paper applies pre-trained BERT

The rest of this paper is organized as follows. Section 2 gives a brief review of the related work. Section 3 describes the AEGCN model. Section 4 shows the experimental results. Finally, Section 5 concludes the paper.

2. Related Works

In this part, we briefly review the specific target sentiment classification and graph convolution network.

2.1. Targeted Sentiment Classification

Targeted sentiment classification is an important research topic as well as a fine-grained task in emotion analysis which is also known as *opinion mining* [18]. The early works on specific targeted sentiment classification mainly focus on extracting features to train sentiment classifiers [19], such as bag-of-word features and sentiment dictionary features. Most of these methods are rule-based [20] and statistical methods [4]—all of which are extremely dependent on feature engineering. Feature engineering is a labor-intensive task. In recent years, recurrent neural networks (RNNs) have achieved great success in this task because the deep learning model is able to utilize distributed representation to automatically learn and obtain the relevant features of targets. In addition, the use of attention mechanisms also makes sentence representation more focused on important information given a specific target [21]. ATAE-LSTM [7] combines LSTM and an attention mechanism. The model embeds specific targets into the calculation of attention weights. RAM was proposed by Chen et al. [11]; this work improves upon Mem-Net by representing memory with bidirectional LSTM (Bi-LSTM) and using a gated recurrent unit network to combine the multiple attention outputs for sentence representation. Ma et al. [10] designed a model with a bi-directional attention mechanism, which learned the attention weights of the contexts and the specific target words in an interactive way. AEN [22] avoids recurrence, and multiple multi-head attention was applied between the contexts and the specific targets. Li et al. [23] developed a new direction named coarse-to-fine task transfer, which aims to use bidirectional LSTM and multiple attention layers to accomplish this task. However, these studies do not take the syntactic information into account and ignore the syntactic interdependence between words, which may lead to ambiguity when identifying the sentiment polarity of a specific target.

2.2. Application of Graph Convolution Networks in NLP

GCNs [24] are very good at processing graph data with rich related information. To begin with, many studies are dedicated to extending GCN for image-related tasks [25,26]. Qi et al. [27] propose a 3D graph neural network (3DGNN) that builds a k-nearest neighbor graph on top of a 3D point cloud and each node, in the graph corresponds to a set of points which allows the model to directly learn its representation from 3D points. In recent years, GCN has attracted increasing attention in NLP, in applications such as semantic role labeling [28] and relationship classification [29]. In the semantic role labeling task, the GCN was applied to the NLP field for the first time, and the experimental results proved that the GCN is very suitable for this task. This has inspired many NLP scholars to explore the application of GCNs in their own research. Some researchers have explored the use of graph neural networks in text classification. Peng et al. [30] first converted texts to graphs-of-words, and then used graph convolution operations to convolve the word graphs. The graph-of-words representation of texts is a novel idea in this field, which has the advantage of capturing non-consecutive and long-distance semantics. There are also some works that have successfully applied GCNs in sentiment classification [15,31]. In [32], Zhao et al. propose a novel aspect-level sentiment classification model which can effectively capture the sentiment dependencies between multiple aspects in one sentence. They consider the sentiment dependencies between aspects in one sentence for the first time. The above studies show that GCNs can effectively capture the relationship between nodes. Inspired by [15], this paper improves the GCN and combines a GCN with a multi-head attention mechanism for targeted sentiment classification, achieving comparable experimental results to state-of-the-art methods.

3. The Attention-Encoding-Based Graph Convolutional Network Model (AEGCN)

The overall architecture of the AEGCN model is illustrated in Figure 2.

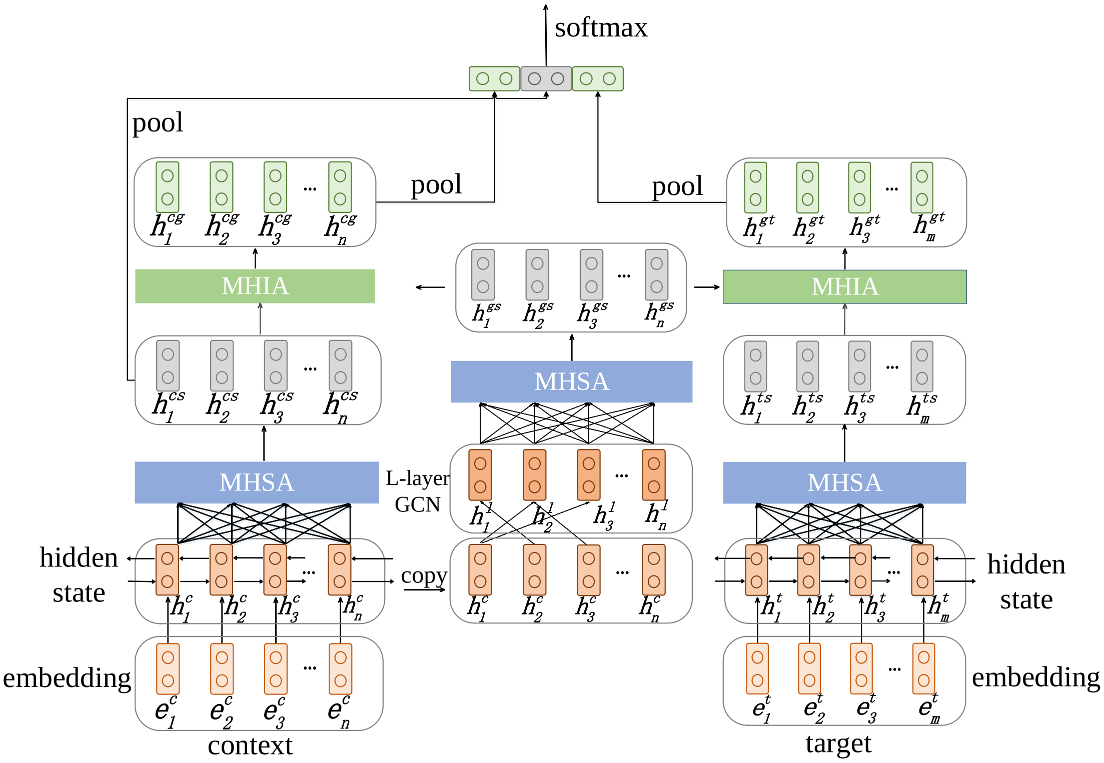

softmax
pool
pool
pool
h_1^{cg} h_2^{cg} h_3^{cg} ... h_n^{cg}
MHIA
h_1^{gs} h_2^{gs} h_3^{gs} ... h_n^{gs}
MHIA
h_1^{gt} h_2^{gt} h_3^{gt} ... h_m^{gt}
h_1^{cs} h_2^{cs} h_3^{cs} ... h_n^{cs}
MHSA
h_1^{ts} h_2^{ts} h_3^{ts} ... h_m^{ts}
MHSA
L-layer GCN
h_1^{l} h_2^{l} h_3^{l} ... h_n^{l}
MHSA
hidden state
h_1^{c} h_2^{c} h_3^{c} ... h_n^{c}
copy
h_1^{c} h_2^{c} h_3^{c} ... h_n^{c}
h_1^{t} h_2^{t} h_3^{t} ... h_m^{t}
hidden state
embedding
e_1^{c} e_2^{c} e_3^{c} ... e_n^{c}
context
e_1^{t} e_2^{t} e_3^{t} ... e_m^{t}
embedding
target

Figure 2. The overall architecture of the attentional-encoding-based graph convolutional network (AEGCN). MHIA: multi-head interactive attention; MHSA: multi-head self-attention.

In the figure, "embedding" denotes GloVe embedding or pre-trained BERT embedding; "hidden state" represents the Bi-LSTM; MHSA refers to multi-head self-attention; MHIA refers to multi-head interactive attention; L-layer GCN represents the layers in the GCN;"pool" indicates average pooling. First, a Bi-LSTM is used for preliminary semantic modeling of contextual and specific targets. After getting the hidden state of the context, the GCN and MHSA are combined to encode the syntactic information. Meanwhile, we exploit MHSA for further attentional encoding of the hidden state of the context and the specific targets to obtain richer semantic information. Then, contextual semantic encoding and target-specific semantic encoding interact with syntactic information encoding by utilizing MHIA. Average pooling is applied to the interactive information and contextual semantic encoding, and finally they are concatenated together to obtain the final feature representation that is used to predict the sentiment polarity.

3.1. Semantic Coding

In this part, AEGCN encodes the semantic information of an n-word sentence containing the m-word aspect term $c = \{e_1^c, e_2^c, \ldots, e_\tau^c, e_{\tau+1}^c, \ldots, e_{\tau+m-1}^c, \ldots, e_n^c\}$ and the targets $\mathrm{t} = \{e_1^t, e_2^t, e_3^t, \ldots . e_{\mathrm{m}}^t\}$ with the combination of Bi-LSTM and multi-head self-attention. τ denotes the start token of the aspect term. This paper applies GloVe embedding and BERT embedding in our model. Accordingly, the models are named AEGCN-GloVe and AEGCN-BERT.

3.1.1. Word Embedding

1. GloVe Embedding
   Provided that $L \in R^{d_e \times |V|}$ is the embedding matrix of pretrained GloVe [33], d_e is the dimension of the word vector. $|V|$ is the vocabulary size. Then, each word $w^i \in R^{|V|}$ is mapped to its corresponding embedding vector $e_i \in R^{d_e \times 1}$, where $R^{d_e \times 1}$ denotes the column of the embedding matrix.
2. BERT Embedding
   This paper uses pre-trained BERT [34] to generate word vectors of sequence as BERT embedding. In order to better facilitate the training and fine-tuning of the BERT model, this paper transforms the formation of the given context and given target to "[CLS] + context + [SEP]" and "[CLS] + target + [SEP]" respectively.

3.1.2. Bi-Directional LSTM

After obtaining the context sequence $c = \{e_1^c, e_2^c, e_3^c, \ldots, e_n^c\}$ and the target sequence $\mathrm{t} = \{e_1^t, e_2^t, e_3^t, \ldots . e_m^t\}$, this paper builds a Bi-LSTM to generate the hidden state vector of the contexts $H^c = \{h_1^c, h_2^c, h_3^c, \ldots, h_n^c\}$ and the targets $H^t = \{h_1^{\mathrm{t}}, h_2^t, h_3^t, \ldots, h_m^t\}$ respectively, where $\left(h_i^c, h_i^t\right) \in R^{2d_h}$ represents the hidden state vector at time i in Bi-LSTM, and d_h is the dimension of the hidden state vector output by undirected LSTM.

3.1.3. Multi-Head Attention

Multi-head attention (MHA) [22] is an attention function that can be performed in parallel subspaces. In this paper, multi-head self-attention and multi-head interactive attention are employed to model different goals. This paper defines a Key sequence $k = \{k_1, k_2, \ldots, k_n\}$ and a Query sequence $q = \{q_1, q_2, \ldots, q_n\}$ according to our specific task. The attention value is obtained by calculating the attention distribution with Key and attaching it to Value. Since keys and values are often the same in the application field of NLP, here Key = Value. Then, an attention function projects Key and Query to an output sequence:

$$Attention(k, q) = soft\max(f_m(k, q))k \tag{1}$$

f_m is the function used to calculate and study the semantic relevance between q_j and k_i:

$$f_m(k_i, q_j) = \tanh([k_i; q_j] \cdot W_a) \tag{2}$$

$W_a \in R^{2d_{hid}}$ is the learning weight matrix. MHA is able to learn different scores of *n_head* in parallel subspaces, and the parameters between heads are not shared because the values of q and k

are constantly changing. The outputs of n_{head} are concatenated and projected to the specific hidden dimension d_{hid} by:

$$MHA(k,q) = [o^1 \oplus o^2 \oplus ... \oplus o^{n_{head}}] \cdot W_m \tag{3}$$

$$o^h = Attention^h(k,q) \tag{4}$$

where "$\oplus$" represents the vector concatenation, $W_m \in R^{d_{hid} \times d_{hid}}$, $o^h = \left\{o_1^h, o_2^h, \ldots, o_m^h\right\}$ is the output of the h-th head attention, and $h \in [1, n_{head}]$. Multi-head self-attention (MHSA) is a special situation of MHA whose input $q = k$. Given the context hidden state H^c and the target word hidden state H^t, AEGCN can derive semantic encoding of the context and the target words H^{cs}, H^{ts} as follows:

$$H^{\text{cs}} = MHA(H^c, H^c) \tag{5}$$

$$H^{\text{ts}} = MHA(H^t, H^t) \tag{6}$$

where $H^{cs} = \{h_1^{cs}, h_2^{cs}, ..., h_n^{cs}\} \in R^{d_h \times n}$, $H^{ts} = \{h_1^{ts}, h_2^{ts}, ..., h_m^{ts}\} \in R^{d_h \times m}$, and d_h is the dimension of MHSA.

3.2. Syntactic Information Encoding

In this section, the architecture of graph convolution network is improved. The modified graph convolution network is used to better integrate syntactic information into each word representation, followed by a multi-head self-attention that encodes the final syntactic information.

3.2.1. Graph Convolution Network

Graph convolution networks [24] are particularly skilled at dealing with graph data with rich relational information. Given a graph with k nodes, an adjacency matrix $A \in R^{k \times k}$ can be obtained by listing the graphs. For convenience, A GCN has L layers $l \in [1, 2, \cdots, L]$, where h_i^L is the final state of node i. The graph convolution of a node can be described as:

$$h_i^l = \sigma\left(\sum_{j=1}^{k} A_{ij} W^l h_j^{l-1} + b^l\right) \tag{7}$$

where W^l is the linear transformation weight matrix, b^l is the offset vector, and σ is a nonlinear function, such as ReLU. An example of a GCN layer is shown in Figure 3.

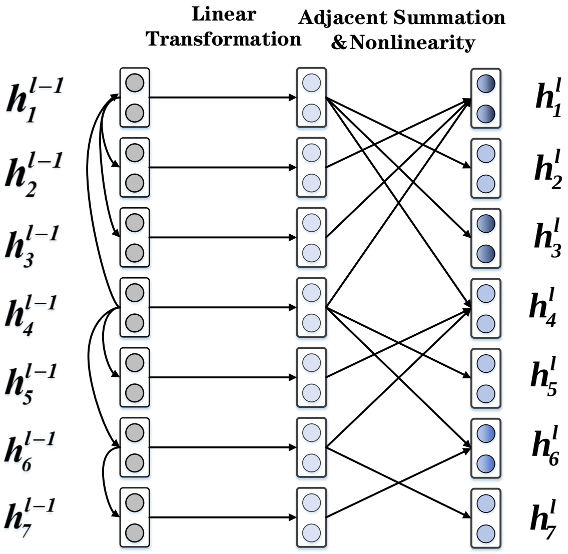


Figure 3. An example of a GCN layer.

First, the hidden state of the previous layer h_i^{l-1} undergoes a linear transformation. Then, each hidden state is shaped by the node information directly related to it to obtain the hidden state of the current layer h_i^l.

Graph convolution over dependency trees [15] was proposed by Zhang et al. In the process of graph convolution, each graph convolution can only encode the information of the immediate neighbor nodes. So, if a GCN has L layers, the information of a node in the graph will only be affected L times by its adjacent nodes. In view of this, syntactic constraints are added to the targets of the sentence by convoluting the syntactic dependency tree of the sentence. Thus, the syntactic distance of the corresponding descriptors can be determined. In addition, when a specific target is described by a non-consecutive word, the method can aggregate the features of non-consecutive words without missing their information. Therefore, this paper puts forward a method to include the syntactic information by combining the graph convolution over dependency trees with point-wise convolution, then encoding the syntactic information obtained by multi-head self-attention to get the final syntactic information coding. The position-aware transformation is made for h_i^l before feeding H^c into continuous GCN layers:

$$s_i^l = F(h_i^l) \tag{8}$$

where $F(\cdot)$ is a position weight distribution function used in [11,12] that is used to enhance the importance of the words close to a specific target. The $F(\cdot)$ function is as follows:

$$q_i = \begin{cases} 1 - \frac{\tau+1-i}{n}, 1 \le i \le \tau+1 \\ 0, \tau+1 \le i \le \tau+m \\ 1 - \frac{i-\tau-m}{n}, \tau+m \le i \le n \end{cases} \tag{9}$$

$$F(h_i^l) = q_i h_i^l \tag{10}$$

where $q_i \in R$ is the position weight of the i-th token.

Next, after constructing the dependency tree of a given sentence, AEGCN first obtains the adjacency matrix $A \in R^{n \times n}$ according to the words in the sentence. Then, following the idea of self looping [17], each word is manually set to be adjacent to itself—that is, the diagonal value of A is 1. The opposite direction of a dependency architecture is also included, which means $A_{ij} = 1$ and $A_{ji} = 1$ if there is an edge going from node i to node j; otherwise, $A_{ij} = 0$ and $A_{ji} = 0$ (see Figures 4 and 5). Finally, the representation of each node is updated with a graph convolution [17] with a normalization factor, as follows:

$$\tilde{h}_i^l = \sum_{j=1}^{n} A_{ij} W^l s_j^{l-1} \tag{11}$$

$$h_i^l = \mathrm{Re}LU\left(\frac{\tilde{h}_i^l}{(d_i+1)} + b^l\right) \tag{12}$$

where $s_j^{l-1} \in R^{2d_h}$ is the representation of the j-th symbol, which is the output of the previous GCN layer. $h_i^l \in R^{2d_h}$ is the output of the current GCN layer, and $d_i = \sum_{j=1}^{n} A_{ij}$ is the degree of the i-th symbol in the tree. W^l and bias b^l are both learnable parameters.

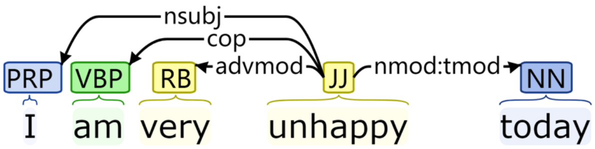
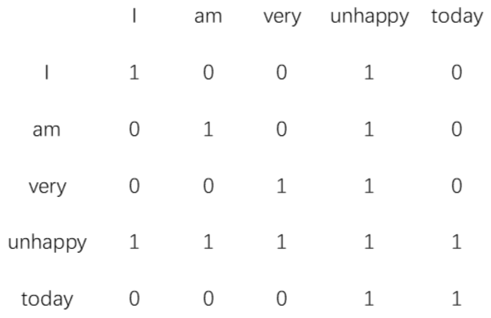


Figure 4. An example of syntactic dependency tree of a given sentence. PRP: pronoun, personal; RB: adverb; cop: copula; advmod: adverbial modifier.



	I	am	very	unhappy	today
I	1	0	0	1	0
am	0	1	0	1	0
very	0	0	1	1	0
unhappy	1	1	1	1	1
today	0	0	0	1	1

Figure 5. The adjacency matrix of a sentence.

3.2.2. Point-Wise Convolution

When the output of the current layer $h^l = \{h_1^l, h_2^l, ..., h_n^l\}$ is derived, point-wise convolution (PWC) is performed. Point-wise means the convolution kernel size is 1. Same operation is applied to each token belonging to the input. Formally, given the input sequence h, PWC is defined as:

$$PWC(h) = \sigma(h \times W_{pwc} + b_{pwc}) \tag{13}$$

where σ represents the activate function ReLU, $*$ is the convolution operation, $W_{pwc} \in R^{2d_h \times 2d_h}$ is the learning weight of the convolution kernel, and $b_{pwc} \in R^{2d_h}$ is the bias of the convolutional kernel. The output of the current layer of GCN h^{l_p} is obtained by exploiting point-wise convolution to h^l:

$$h^{l_p} = PWC(h^l) \tag{14}$$

The final output of the L-th GCN layer is $H^{L_p} = \{h_1^{L_p}, h_2^{L_p}, ..., h_n^{L_p}\}$, $h_i^{L_p} \in R^{2d_h}$. Then, syntactic information encoding H^{gs} is acquired by applying multi-head self-attention encoding to the output of final GCN layer H^{L_p}:

$$H^{gs} = MHA(H^{L_p}, H^{L_p}) \tag{15}$$

$H^{gs} = \{h_1^{gs}, h_2^{gs}, ..., h_n^{gs}\} \in R^{d_h \times n}$, d_h is the dimension of MHSA.

3.3. Information Fusion

In this part, this paper interacts syntactic information with semantic information, and finishes the final concatenation.

3.3.1. Multi-Head Interactive Attention

Multi-head interactive attention (MHIA) is the common form in which q is different from k. Given the syntactic information encoding H^{gs}, the context semantic encoding H^{cs} and the target semantic encoding H^{ts}, the context-perceptive syntactical information H^{cg}, and the syntax-perceptive target information H^{gt} are derived by:

$$H^{cg} = MHA(H^{cs}, H^{gs}) \tag{16}$$

$$H^{gt} = MHA(H^{gs}, H^{ts}) \tag{17}$$

where $H^{cg} = \{h_1^{cg}, h_2^{cg}, ..., h_n^{cg}\} \in R^{d_h \times n}$, and d_h is the dimension of MHIA.

3.3.2. Information Mosaic

The context-perceptive syntactical representation H^{cg}, the syntax-perceptive target H^{gt}, and the context semantic encoding H^{cs} are averaged by utilizing average pooling, then they are concatenated as the final feature representation u as the follows:

$$h_{avg}^{cg} = \sum_{i=1}^{n} h_i^{cg} / n \tag{18}$$

$$h_{avg}^{gt} = \sum_{i=1}^{n} h_i^{gt} / m \tag{19}$$

$$h_{avg}^{cs} = \sum_{i=1}^{n} h_i^{cs} / n \tag{20}$$

$$u = [h_{avg}^{cg} \oplus h_{avg}^{cs} \oplus h_{avg}^{gt}] \tag{21}$$

3.4. Sentiment Classification

After the final feature representation u is obtained, it is fed into the $softmax$ layer, and the probability distribution of the sentiment polarity of the different targets is gained:

$$x = \tilde{W}_u^T \tilde{u} + b_u \tag{22}$$

$$y = soft\max(x) = \frac{\exp(x)}{\sum_{k=1}^{C} \exp(x)} \tag{23}$$

$y \in R^c$ is the predicted distribution of the sentiment polarity, and c is the category of classification. $\tilde{W}_u^T \in R^{1\times c}$ and $b_u \in R^c$ are the learning weight matrix and the bias, respectively.

3.5. Model Training

In the model, the sum of the classification cross entropy and L_2-regularization is introduced as the loss function, and the back propagation algorithm is employed to update the weights and parameters:

$$L = -\sum_i \sum_{j=1}^{c} y_i^j \log \widehat{y}_i^j + \lambda \|\theta\|^2 \tag{24}$$

where i is the subscript of the i-th sample, j is the script of j-th sentiment category; y is the real distribution of sentence sentiment polarity, $\widehat{y}$ is the predicted distribution of sentence sentiment polarity, c is the classification category, and θ is all trainable parameters. λ is the parameter of regularization.

4. Experiments

4.1. Datasets

In order to verify the effectiveness of AEGCN, our experiment was implemented on five datasets: a Twitter dataset used in [35], herein denoted "twitter", rest14 and lap14 (semeval 2014 task 4 [36]), rest15 (semeval 2015 task 12 [37]), and rest16 (semeval 2016 task 5 [38]). The accuracy and macro average F1 were selected for the evaluation. The experimental results were obtained by the average of three random initializations. The experimental datasets are shown in Table 1.

Table 1. Details of the benchmark datasets.

Datasets	TWITTER		LAP14		REST14		REST15		REST16	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Pos.	1561	173	994	341	2164	728	912	326	1240	469
Neu.	3127	346	464	169	637	196	36	34	69	30
Neg.	1560	173	870	128	807	196	256	182	439	117
Total	6248	692	2328	638	3608	1120	1204	542	1748	616

4.2. Hyper-Parameters

In the experiments, for AEGCN-GloVe, the word embeddings were initialized from GloVe with a dimension of 300, and the learning rate was 0.001. The parameter of the regulation was set as 0.00001. The coefficient of the batch size was 32. In order to prevent over-fitting, the dropout rate was 0.5. For AEGCN-BERT, the embedding dimension was set to 768. The learning rate was set to 2×10^{-5}. Regulation was set as 0.001. Dropout was 0.1 and batch size was 16. For both AEGCN design models, the number of multi-head attention heads was 3, the number of GCN layers was set as 2, and the model weights were initialized by uniform distribution. The hidden layer dimension was 300. In addition, the AEGCN models utilized the Adam optimizer.

Hyper-Parameters Analysis

1. The Number of Heads in MHA

This paper explores the influence of the number of heads k on the experimental results. The results over dataset lap14 are shown in Figure 6, and similar results were achieved on the other four datasets.

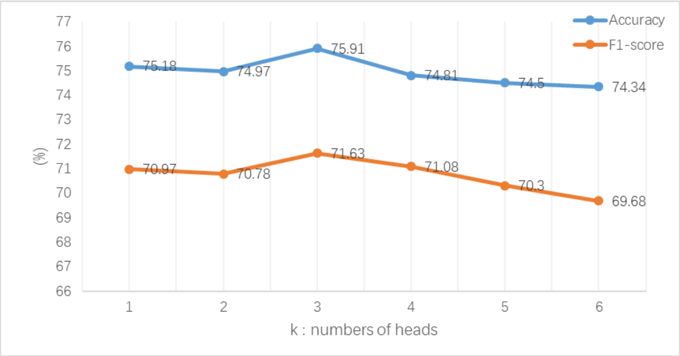


Figure 6. Impact of k over the lap14 dataset.

From Figure 6, it can be observed that the value of accuracy and F1-score fluctuated with increasing k. But when $k = 3$, the highest accuracy and F1 values appear. Then the values of accuracy and F1 decrease with the rising k. We speculate that due to the increase of k, the integration of semantic information from too many context words produces unnecessary interference, confusing the representation of the current words. Therefore, the performance of the model was better when k = 3.

2. Impact of GCN Layer Number

The number of GCN layers is also an important parameter that affects the performance of our model. We tested this on the lap14 dataset with different number of GCN layers R. The results are shown in Figure 7.

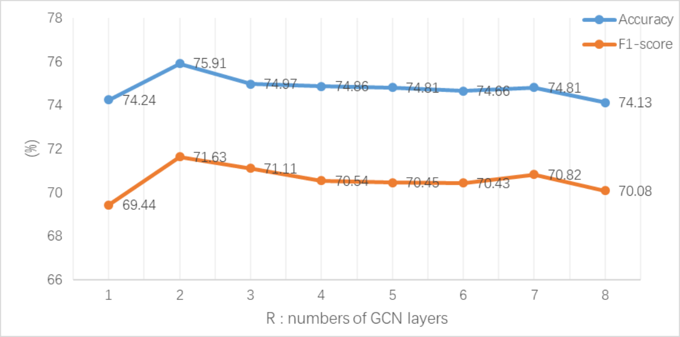


Figure 7. Impact of R over the lap14 dataset.

From the above results, it can be seen that the model achieved the best performance when the number of GCN layers $R = 2$. However, when R was greater than 2, the performance of the model worsened with increasing GCN layers. The possible reason for this is that as the number of GCN layers R increases, the model parameters grow, causing the model to be more difficult to train, and leading to overfitting. In order to avoid too many training parameters and overfitting, this paper set the GCN layer number as 2.

4.3. Experimental Results

Eight baseline models are selected for comparison to evaluate the effectiveness of AEGCN, and the comparison of experimental results is shown in Table 2.

Table 2. Experimental results (%). We use "N/A" to represent unreported experimental results. The top two scores are in bold.

Model	TWITTER		LAP14		REST14		REST15		REST16	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
SVM	63.40	63.30	70.49	N/A	80.16	N/A	N/A	N/A	N/A	N/A
LSTM	69.56	67.70	69.28	63.09	78.13	67.47	77.37	55.17	86.80	63.88
MemNet	71.48	69.90	70.64	65.17	79.61	69.64	77.31	58.28	85.44	65.99
AOA	72.30	70.20	72.62	67.52	79.97	70.42	78.17	57.02	87.50	66.21
IAN	72.50	70.81	72.05	67.38	79.26	70.09	78.54	52.65	84.74	55.21
TNet-LF	72.98	71.43	74.61	70.14	80.42	71.03	78.47	59.47	**89.07**	**70.43**
AEN	72.83	69.81	73.51	69.04	80.98	72.14	N/A	N/A	N/A	N/A
ASGCN-DT	71.53	69.68	74.14	69.24	80.86	**72.19**	79.34	60.78	88.69	66.64
ASGCN-DG	72.15	70.40	75.55	71.05	80.77	72.02	79.89	**61.89**	88.99	67.48
AEGCN-GloVe	**73.16**	**71.82**	**75.91**	**71.63**	**81.04**	71.32	**79.95**	60.87	87.39	68.22
AEGCN-BERT	**75.04**	**73.68**	**78.73**	**74.22**	**82.58**	**73.40**	**82.71**	**69.00**	**89.61**	**73.93**

SVM [39] is a traditional support vector machine method based on complex feature engineering.

LSTM [8] gets the hidden layer output of sentences by LSTM, then obtains the sentiment analysis by $softmax$ classifier.

MemNet [6] regards the contexts as external memory, which makes the model benefit from a multi-hop architecture.

AOA-LSTM [40] obtains the hidden layer output of the contexts and target words through Bi-LSTM, then obtains the corresponding representation of the contexts and target words through the interactive learning of attention over attention, and finally acquires the polarity distribution of sentiment by a $softmax$ classifier.

IAN [10] acquires the hidden output of context and target words through LSTM, and then obtains the expression of the context and target words through the interactive learning of the interactive attention mechanism. It models the relationships between the target words and their contexts interactively. After splicing the expression of context and target words, the polarity distribution of sentiment is received by a $softmax$ classifier.

TNet-LF [12] proposes a method to generate target-specific representations of words in the sentence, incorporating a mechanism for preserving the original contextual information from the RNN layer.

AEN [22] eschews recurrence and uses an attentional encoder network to model the relation between the contexts and the specific targets.

ASGCN [15] puts forward a graph convolution network (GCN) on the dependency tree of sentences to take advantage of syntactic information and word dependency.

From the experimental results in Table 2, we can see that the AEGCN-GloVe model was slightly better than all the other models in the twitter and lap14 datasets. Compared with the baseline model ASGCN-DT, it obtained comparable results in the rest14 dataset. Compared with the baseline model ASGCN-DG, it still obtained comparable results on the rest15 dataset. However, in the rest16 dataset, the results were slightly inferior to those of the baseline model TNet-LF. In particular, AEGCN-BERT obtained new state-of-the-art performances on all datasets.

Based on the deep learning model, the performance of the proposed model was better than the traditional machine learning methods. In Table 2, the SVM model proposed by Kiritshenko uses an SVM for classification, which relies on a large number of artificial feature extractions. There is no artificial feature extraction in AEGCN-GloVe, and its accuracy was 9.76%, 5.42%, and 0.88% higher than SVM on twitter, lap14, and restaurant datasets, respectively. This shows that the deep learning model is suitable for specific aspect sentiment analysis.

It is better to model context semantics by exploiting Bi-LSTM combined with a multi-head attention mechanism rather than by applying only a standard multiple-attention or multi-head attention mechanism. Taking MemNet as an example, it uses multiple hops to combine different attentions linearly, and its accuracy and F1 score were lower than that of AEGCN-GloVe on all five datasets. One possible reason is that once the traditional attention mechanism incorrectly assigns weights to words that are not related to determining the sentiment polarity of a specific target, repeating the traditional attention mechanism multiple times will make it more difficult for the model to predict the correct sentiment polarity of a specific target. In addition, AEN only uses the multi-head attention mechanism to semantically model the contextual and specific targets. Without using Bi-LSTM, it may not be possible to adequately consider the contextual semantics of the entire sentence from front to back and back to front. Except for one index (rest4, F1), AEGCN-GloVe obtained higher values of accuracy and F1 score than AEN in three data sets (twitter, lap14, and rest14). This indicates that the semantic information extracted by combining Bi-LSTM and multi-head self-attention was more abundant.

Due to the combination of syntactic information, the effect of this model was better than the model that does not consider it. Although AOA emphasizes the influence between the contexts and the target words through attention over attention, it only achieved slightly higher accuracy and F1 score over rest16 dataset (0.11% higher) and lower values in the other datasets. Moreover, though IAN improves the interaction between the contexts and the target words through the interaction of the interactive attention mechanism, the accuracy and F1 score of the model in five datasets were lower than those for AEGCN-GloVe. TNet-LF could integrate the specific target information into word representation well, and the context-preserving mechanism could retain semantic information well. In the rest16 dataset, the performance of TNet-LF was slightly higher than AEGCN-GloVe (1.7% and 2.21% respectively). However, it was worse than AEGCN-GloVe in the other four datasets. We suppose that the sentences in the rest16 dataset are particularly dependent on the original semantic information of the text, and the application of syntactic information for targeted sentiment classification was successful.

The performance of GCN with point-wise convolution was better than the GCN with an aspect-specific masking layer. In the ASGCN model, a multi-layered graph convolution structure is implemented on top of the LSTM output, followed by a masking mechanism that filters out non-aspect words and keeps only aspect-specific features. However, the contextual representations with syntactic information are lost. We cannot conclude that some context words with syntactic information are useless in determining the emotional polarity of a particular target, so we chose to keep all the context words with syntactic information and encoded them with multi-head self-attention. We believe that after using the syntactic information to reshape the representation of each context word through GCN, the syntactic information-rich words will be more prominent after the parallel calculation of multi-head self-attention, which is more conducive to the full use of syntactic information to determine the sentiment polarity of specific targets. Among the five data sets, only three results of AEGCN-GloVe were lower than those of ASGCN in rest14, rest15 and rest16 (0.87%, 1.02%, and 1.60% respectively), and the other seven indexes were better than that of ASGCN, which proves the efficiency of AEGCN-GloVe.

Compared with all the models listed on the Table 2, AEGCN-BERT achieved new state-of-the-art results, which demonstrates the power of pre-trained BERT and the huge superiority of the BERT-based model over GloVe-based models.

4.4. Ablation Study

In order to further identify the level of benefit that each component of AEGCN-GloVe contributes to the model performance and the importance of each component, we conducted an ablation study on AEGCN-GloVe. The results are shown in Table 3.

Table 3. Ablation study results (%). PWC: point-wise convolution; w/o: without. The best two results with each dataset are highlighted in bold.

Model	TWITTER		LAP14		REST14		REST15		REST16	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
AEGCN	**73.16**	**71.82**	**75.91**	**71.63**	**81.04**	**71.32**	**79.95**	60.87	87.39	68.22
w/o GCN	**73.55**	**71.88**	74.86	70.65	**80.74**	**70.78**	79.82	**61.12**	**87.66**	**69.57**
w/o PWC	73.06	71.81	75.02	70.55	80.68	70.64	79.70	60.76	**87.71**	**69.77**
w/o MHSA	72.49	70.77	**75.28**	**71.71**	80.17	70.18	**80.13**	**62.40**	87.60	68.24
w/o MHIA	73.12	71.20	72.62	67.57	79.94	70.18	78.16	58.34	86.79	68.65

As we can see from Table 3, most results of AEGCN-GloVe ablations were inferior to AEGCN-GloVe in both accuracy and macro-F1 measure.

4.4.1. Ablate Graph Convolution Network

For AEGCN, a graph convolution network was deployed between the Bi-LSTM and multi-head self-attention in the proposed model, since it can reconstruct the representation of each word using syntactic information. We ablated the graph convolution network to examine the performance of AEGCN without it.

As a result of removing the GCN, the accuracy and F1-score on three datasets (lap14, rest14, and rest15) decreased, while the accuracy and F1-score on the twitter and rest16 datasets increased. This is because the sentences from the twitter dataset are colloquial and less grammatical. Since the syntactic structure of most sentences in the twitter dataset is not perfect, the introduction of syntactic information would interfere with the prediction of the sentiment polarity of a specific target. TNet-LF is able to learn more abstract contextualized word features from deeper networks, and it achieved comparable results on the rest16 dataset in terms of both accuracy and macro-F1 measure (Table 2). We can conclude from this that the twitter and rest16 datasets are less-sensitive to syntactic information. Our experimental results demonstrate that syntactic information is quite helpful for targeted sentiment classification on most datasets.

4.4.2. Ablated Point-Wise Convolution

Point-wise convolution (PWC) is within every layer of the graph convolution network. We improved the GCN by adding PWC to every layer, aiming at better integrating the syntactic information. We ablated the PWC to see what would happen to the results if the GCN ran without PWC.

The AECGN without PWC performed better than baseline AEGCN on the rest16 dataset, while its performance on the lap14 dataset worsened obviously. On the twitter dataset, AEGCN (without (w/o) PWC) achieved an almost equal performance compared to the baseline model. PWC is deployed to better learn and integrate the syntactic information representation of words of the current GCN layer. Table 3 indicates that PWC was very significant for AEGCN, especially on the lap14 dataset.

4.4.3. Ablated Multi-Head Self-Attention

Multi-head self-attention (MHSA) is mainly used to extract richer semantic information and encode the syntactic information. We ablated MHSA to examine the performance of AEGCN without it.

The removal of multi-head self-attention (MHSA) led to poor performance on the twitter dataset and a slight performance increase on the rest15 dataset, which indicates that the twitter dataset contains a great deal of semantic information and that the application of MHSA in our model could effectively capture it. Based on these results, we suppose that the rest15 dataset is more sensitive to syntactic information than the other datasets displayed in Table 3.

4.4.4. Ablated Multi-Head Interactive Attention

Multi-head interactive attention (MHIA) aims at assembling features and interactively learning the correlation between syntactic information and semantic information. Concatenation and pooling can replace MHIA, but then the learning process is no longer interactive.

Without MHIA, the performance of the proposed model on the five datasets was unsatisfactory. This shows that MHIA is crucial for AEGCN and its core architecture. Without the interaction of syntactic and semantic information, the results onthree datasets (lap14, rest14, and rest15) were disastrous. Based on the results, we can say that the interaction of syntactic and semantic information is significant in most datasets, and especially for lap14, rest14, and rest15.

4.4.5. AEGCN Ablations Analysis

According to Table 3, the performance of AEGCN ablations was significantly reduced. Compared to the AEGCN model, the AEGCN ablations of the GCN layer achieved limited performance on lap14 and rest14 datasets, especially on lap14. AEGCN attained inferior performance when PWC was removed since the model lost the ability to better study and integrate the syntactic information. We utilize multi-head self-attention to capture more abundant semantic information and encode syntactic information. Its absence led to poor performance on three datasets (twitter, lap14, rest14). Multi-head interactive attention is applied to interactively study the features of the syntactic information and semantic information. The removal of MHIA was fatal since performance on all datasets dropped—dramatically in some (lap14, rest14, rest15). In conclusion, the experimental results reveal that, for AEGCN, all the components work well and brought a huge improvement in all five datasets. If AEGCN is run without interactive semantic and syntactic information, the performance decreased by 2%–3% on lap14 and rest15 datasets. Experimental results show that each component of AEGCN is indispensable and effective in achieving overall good results on all datasets.

5. Conclusions and Future Work

In this paper, a model based on attention encoding and a graph convolution network is proposed for targeted sentiment classification. In order to solve the problems of losing long-distance emotional words in semantic modeling and parallel computing for input data, this paper proposes a semantic encoding method combining bidirectional LSTM with a multi-head self-attention mechanism. To make use of the syntactic information that most models ignore, we developed a graph convolution neural network that integrated the point-wise convolution and builds the graph convolution network over the syntax dependency tree to encode the syntactic information. It is then interacted with the semantic information through multi-head interactive attention. The experimental results on five datasets—twitter, lap14, rest14, rest15, and rest16—show that the AEGCN model proposed in this paper was significantly better than the models based on traditional machine learning and other deep learning models, proving its effectiveness.

Although the current model achieved good experimental results, there is still a great deal of work to be done. In our future work, we intend to reduce the training parameters of the model to make our model more lightweight. Second, extracting more original contextual semantic information will also be an important part of our future work. Finally, combining domain knowledge with syntactic information can be taken into consideration in future research.

Author Contributions: conceptualization, L.X. and X.H.; methodology, Y.X. and X.H.; investigation, L.X.; formal analysis, Y.X.; software, L.X. and B.C.; validation, D.G. and B.C.; writing, X.H., Y.C. and L.X.; resources, T.Z.; review, X.H., Y.C. and L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Scholarship Council, Guangdong Science and Technology Department under Grant no. 2016A010101020, 2016A010101021, 2016A010101022, Guangzhou Science and Information Bureau under Grant no. 201802010033.

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

References

1. Chen, Y. *Service-Oriented Computing and System Integration: Software, IoT, Big Data, and AI as Services*, 6th ed.; Kendall Hunt Publishing: Dubuque, IA, USA, 2018.
2. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Synth. Lect. Hum. Lang. Technol.* **2008**, *2*, 1–135. [CrossRef]
3. Liu, B. Sentiment analysis and opinion mining. *Found. Trends Inf. Retr.* **2012**, *5*, 1–167. [CrossRef]
4. Jiang, L.; Yu, M.; Zhou, M.; Liu, X.; Zhao, T. Target-dependent twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Portland, OR, USA, 19–24 June 2011; pp. 151–160.
5. Wagner, J.; Arora, P.; Cortes, S.; Barman, U.; Bogdanova, D.; Foster, J.; Tounsi, L. Dcu: Aspect-based polarity classification for semeval task 4. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014); Dublin, Ireland, 23–24 August 2014; pp. 223–229.
6. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. *arXiv* **2016**, arXiv:1605.08900.
7. Wang, Y.; Huang, M.; Zhao, L.; Zhu, X. 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.
8. Tang, D.; Qin, B.; Feng, X.; Liu, T. Effective lstms for target-dependent sentiment classification. *arXiv* **2016**, arXiv:1512.01100.
9. Xue, W.; Li, T. Aspect based sentiment analysis with gated convolutional networks. *arXiv* **2018**, arXiv:1805.07043.
10. Ma, D.; Li, S.; Zhang, X.; Wang, H. Interactive attention networks for aspect-level sentiment classification. *arXiv* **2017**, arXiv:1709.00893.
11. Chen, P.; Sun, Z.; Bing, L.; Yang, W. Recurrent attention network on memory for aspect sentiment analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 452–461.
12. Li, X.; Bing, L.; Lam, W.; Shi, B. Transformation networks for target-oriented sentiment classification. *arXiv* **2018**, arXiv:1805.01086.
13. Li, X.; Song, J.; Gao, L.; Liu, X.; Huang, W.; He, X.; Gan, C. Beyond RNNs: Positional Self-Attention with Co-Attention for Video Question Answering. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, No 01.

14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc.: Long Beach, CA, USA, 2017; pp. 5998–6008.
15. Zhang, C.; Li Q.; Song, D. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. *arXiv* **2019**, arXiv:1909.03477.
16. Huang, B.; Carley, K.M. Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks. *arXiv* **2019**, arXiv:1909.02606.
17. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2017**, arXiv:1609.02907.
18. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
19. Rao, D.; Ravichandran, D. Semi-supervised polarity lexicon induction. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Athens, Greece, 30 March–3 April 2009; pp. 675–682.
20. Ding, X.; Liu, B.; Yu, P.S. A holistic lexicon-based approach to opinion mining. In Proceedings of the 2008 International Conference on Web Search and Data Mining, Palo Alto, CA, USA, 11–12 February 2008; pp. 231–240.
21. Ma, X.; Zeng, J.; Peng, L.; Fortino, G.; Zhang, Y. Modeling multi-aspects within one opinionated sentence simultaneously for aspect-level sentiment analysis. *Future Gen. Comp.* **2019**, *93*, 304–311. [CrossRef]
22. Song, Y.; Wang, J.; Jiang, T.; Liu, Z.; Rao, Y. Attentional encoder network for targeted sentiment classification. *arXiv* **2019**, arXiv:1902.09314.
23. Li, Z.; Wei, Y.; Zhang, Y. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 4253–4260.
24. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv* **2014**, arXiv:1312.6203.
25. Henaff, M.; Bruna, J.; LeCun, Y. Deep convolutional networks on graph-structured data. *arXiv* **2015**, arXiv:1506.05163.
26. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the Advances in Neural Information Processing Systems 29, Barcelona, Spain, 5–10 December 2016; pp. 3837–3845.
27. Qi, X.; Liao, R.; Jia, J.; Fidler, S.; Urtasun, R. 3d graph neural networks for RGBD semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5209–5218.
28. Marcheggiani, D.; Titov, I. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv* **2017**, arXiv:1703.04826.
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.* **2019**, *26*, 262–268. [CrossRef] [PubMed]
30. Peng, H.; Li, J.; He, Y.; Liu, Y.; Bao, M.; Wang, L.; Song, Y.; Yang, Q. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In Proceedings of the 2018 World Wide Web Conference on World Wide Web, Lyon, France, 23–27 April 2018; pp. 1063–1072.
31. Hou, X.; Huang, J.; Wang, G.; Huang, K.; He, X.; Zhou, B. Selective Attention Based Graph Convolutional Networks for Aspect-Level Sentiment Classification. *arXiv* **2019**, arXiv:1910.10857.
32. Zhao, P.; Hou, L.; Wu, O. Modeling Sentiment Dependencies with Graph Convolutional Networks for Aspect-level Sentiment Classification. *arXiv* **2019**, arXiv:1906.04501.
33. Pennington, J.; Socher, R.; Manning, C. 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.
34. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (Long and Short Papers), pp. 4171–4186.

35. 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 (Volume 2: Short Papers), Baltimore, MD, USA, 23–25 June 2014; pp. 49–54.
36. Maria Pontiki, D.G.; John Pavlopoulos, H.P.; Ion Androutsopoulos, S.M. Semeval-2014 task 4: SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 27–35.
37. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; Androutsopoulos, I. Semeval-2015 task 12: Aspect based sentiment analysis. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, Colorado, 4–5 June 2015; pp. 486–495.
38. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Al-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; de Clercq, O.; et al. Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the 10th international Workshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 19–30.
39. 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 (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 437–442.
40. Huang, B.; Ou, Y.; Carley, K.M. Aspect level sentiment classification with attention-over-attention neural networks. *arXiv* **2018**, arXiv:1804.06536.