applied sciences

MDPI

Article

Multiple Interactive Attention Networks for Aspect-Based Sentiment Classification

Dianyuan Zhang, Zhenfang Zh, Qiang Lu, Hongli Pei, Wenqing Wu and Qiangqiang Guo

School of Information Science and Electrical Engineering, Shandong Jiao Tong University, Jinan, 250357, China; csdzdy@163.com (D.Z.); lqsdjtu@163.com (Q.L.); peihongli@sdjtu.edu.cn (H.P.); 19021001@stu.sdjtu.edu.cn (W.W); gqqq777@163.com (Q.G.)

* Correspondence: zhuzf@sdjtu.edu.cn; Tel.: +86-6 13793100702

Received: 28 December 2019; Accepted: 12 March 2020; Published: 18 March 2020

Abstract: Aspect-Based (also known as aspect-level) Sentiment Classification (ABSC) aims at determining the sentimental tendency of a particular target in a sentence. With the successful application of the attention network in multiple fields, attention-based ABSC has aroused great interest. However, most of the previous methods are difficult to parallelize, insufficiently obtain, and fuse the interactive information. In this paper, we proposed a Multiple Interactive Attention Network (MIN). First, we used the Bidirectional Encoder Representations from Transformers (BERT) model to pre-process the data. Then, we used the partial transformer to obtain a hidden state in parallel. Finally, we took the target word and the context word as the core to obtain and fuse the interactive information. Experimental results on the different datasets showed that our model was much more effective.

Keywords: pre-trained BERT; natural language processing; aspect-based sentiment classification; attention mechanism

1. Introduction

With the development of social networks, more and more users are willing to share their opinions on the Internet, comment information is rapidly expanding, and it is difficult to process large amounts of information with sentiment analysis alone. Therefore, as reviews accumulate, in-depth analysis of Aspect-Based (also known as aspect-level) Sentiment Classification (ABSC) task becomes more important. ABSC [1,2] is a sub-task of text sentiment classification, and it is different from the traditional sentiment classification of document-based and sentence-based. It aims to predict sentiment polarities for different aspects within the same sentence or document. For example, in the sentence “Granted space is smaller than most, it is the best service you will find in even the largest of restaurants.”, the sentiment polarity of the target word "space" is negative, but another target word “service” is positive.

Many statistical-based methods have been applied in ABSC and obtained good experimental results. For example, Support Vector Machines (SVM) [3], it is a few support vectors that determine the final result, which not only helps us to grasp key samples, eliminates a large number of redundant samples but also has good robustness. But SVM excessively relies on handcrafted features in multiple classifications [4] . In recent years, the neural network for processing sequence data, such as Recurrent Neural Network (RNN) [5], is designed to automatically learn useful low dimensional representations from targets and contexts. However, they are difficult to implement a parallel operation, and there is a gradient disappearance problem.

In recent years, the attention mechanism with RNN has been successfully used for machine translation [6], and these methods have been extensively used in other fields. Using these methods,

we can make the sentiment analysis model, selectively balancing the weight of context words and target words [7]. However, these models simply average the aspect or context vector to guide learning the attention weight on the context or aspect words. Therefore, these models are still in the preliminary stage in dealing with fine-grained sentiment analysis.

In conclusion, there are two problems with previous approaches. The first problem is that previous approaches are difficult to obtain the hidden state interactively in parallel. Another problem is to insufficiently obtain and fuse contextual information and aspect information.

This paper proposed a model named Multiple Interactive Attention Network (MIN) to address these problems. To address the first problem, we took advantage of Multi-Head Attention (MHA) to obtain useful interactive information. To address another problem, we adopted target-context pair and Context-Target-Interaction (CTI) in our model.

The main contributions of this paper are presented as follows:

1. We took advantage of MHA and Location-Point-Wise Feed-Forward Networks (LPFFN) to obtain the hidden state interactively in parallel. Besides, we applied pre-trained Bidirectional Encoder Representations from Transformers (BERT) [8] in our model.
2. We used the CTI and target-context pair to help us obtain and fuse useful information. We also verified the effectiveness of these two methods.
3. We experimented on different public authoritative datasets: restaurant reviews and laptop reviews of the SemEval-2014 Task 4 dataset, the ACL(Annual Meeting of the Association for Computational Linguistics) 14 Twitter dataset, SemEval-2015 Task 12 dataset, SemEval-2016 Task 5 dataset. The experimental results showed our model outperformed state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 gives a detailed description of our model. Afterward, our model is compared with other recent methods of the ABSC task in Section 4. Section 5 summarizes the conclusions and envisions the future direction.

2. Related Works

In this section, we have introduced the main research methods of ABSC, including traditional machine learning methods and neural networks methods.

Traditional machine learning methods [9] focus on text representation and feature extraction. It can extract a series of features, such as sentiment lexicon features and bag-of-words features, to train the sentiment classifier. The most commonly used classification methods include K-Nearest Neighbor (KNN) [10], naive Bayesian model [11], SVM. However, these methods rely heavily on the characteristics of manual extraction and require a lot of manpower and may be unable to achieve satisfactory results when the dataset changes. Therefore, the generality of these methods is poor, and they are difficult to be applied to other datasets. The semi-supervised model solves these problems to a large extent. It is successfully used to detect the Internet of Things (IoT) distributed attack [12] and plays a significant role in maintaining social network security [13].

Recent work is being combined with neural networks because of the ability to capture the original features, which can be mapped to continuous and low-dimensional vectors without the need for feature engineering. Because of these advantages, much more structured neural networks have been derived, such as Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), and Generative Adversarial Networks (GAN), which are used to solve problems in NLP(Natural Language Processing). These methods based on neural networks have attracted much attention, especially Long Short-Term Memory (LSTM). In order to effectively model the semantic relevance of the target words and the context in the sentence, Tang et al. [14] proposed Target-Dependent Long Short-Term Memory (TD-LSTM). TD-LSTM uses two LSTMs to model around the target from the left and right to take the context as the feature representation of sentiment classification. Then, Tang et al. proposed Target-Connection Long Short-Term Memory (TC-LSTM) based on TD-LSTM to enhance the interaction between target words and context.

Attention mechanism [15], which obtains good results in machine translation tasks, guides the model to get a small amount of important information from a large amount and focuses on it. Wang

et al. [16] proposed a method called ATAE-LSTM(Attention-based Long Short-Term Memory with Aspect Embedding), which combines attention-based LSTM with aspect embedding; in this method, it combines the attention mechanism with LSTM to model sentences semantically; by doing this, it could be addressing the problem of aspect-based sentiment analysis. Tang et al. [17] proposed a model using a deep memory network and attention mechanism on the ABSC, and this model is composed of a multi-layered computational layer of shared parameters; each layer of the model incorporates positional attention mechanism, and this method could learn a weight for each context word and use this information to calculate the text representation. Ma et al. [18] proposed Interactive Attention Networks (IAN), which interactively learn attention in the context words and target words. They used the attention mechanism to link the target word to the context word for multi-level semantic classification. Gu et al. [19] proposed a Position-Aware Bidirectional Attention Network (PBAN), which not only concentrates on the position information of aspect terms but also mutually models the relation between aspect terms and sentence by employing bidirectional attention mechanism. Tang et al. [20] addressed the problems of strong mode over-learning and weak mode under-learning in the neural network learning process and proposed a progressive self-supervised attention mechanism algorithm. They properly constrained the attention mechanism.

Our MIN model used a different method, which made use of multiple attention networks to obtain contextual interactive information; meanwhile, we used the Bidirectional Gated Recurrent Unit (BI-GRU) [21] to obtain the target information. Target and context information was obtained from the target-context pair. Then, we used CTI to dynamically combine the target-context information with the context word. Finally, we used Convolutional Neural Networks (CNN) [22,23] to extract features for classification.

3. Model Description

3.1. Task definition

Given a context sequence $x^c = \{x_1^c, x_2^c, ..., x_n^c\}$ and the target sequence $x^t = \{x_1^t, x_2^t, ..., x_m^t\}$, x^t is a sub-sequence of x^c .The goal of the task is to use the information in x^c to predict the sentiment polarity of x^t (e.g., positive, neutral, negative).

In Figure 1, we have given the structure of our model. It consists of the input embedding layer, attention encoding layer, target-context-interaction layer, Context-Target-Interaction layer, select convolution layer.

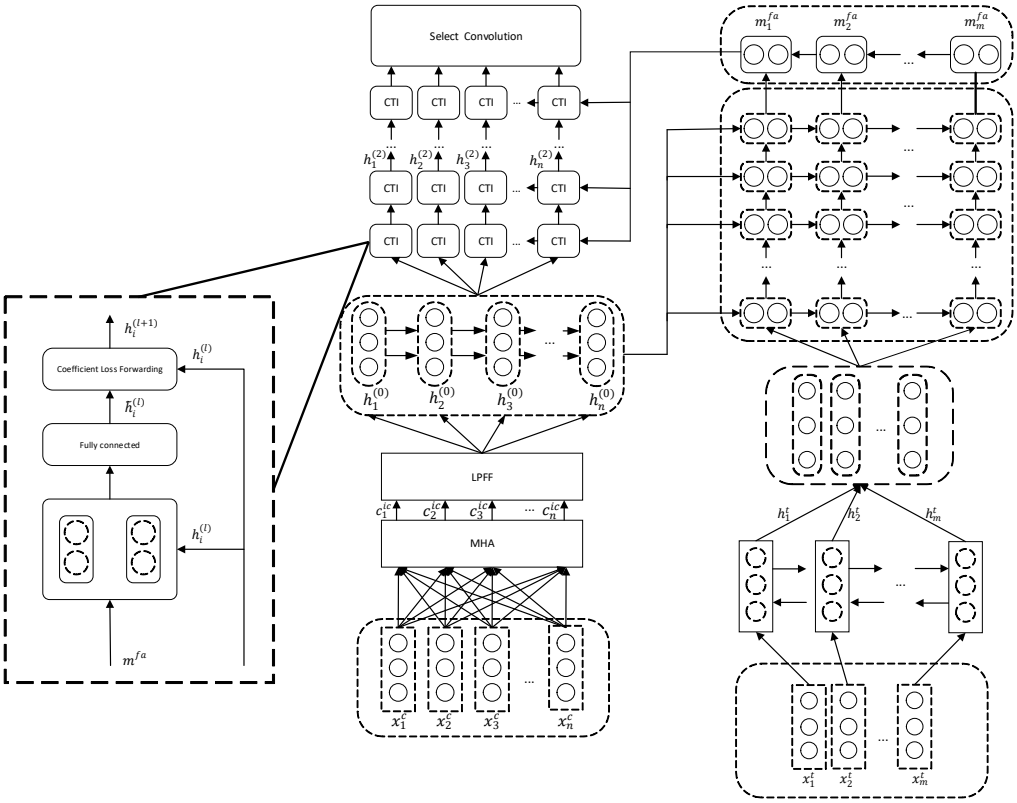

Figure 1. The overall architecture of the Multiple Interactive Attention Network (MIN) design.

3.2. Input Embedding Layer

The embedding layer converts words in a sentence into vectors and maps them to a high-dimensional vector space; we use pre-trained BERT word vectors to obtain pre-trained fixed word embedding for each word.

The BERT model uses two new unsupervised predictive tasks for pre-training—, Masked Language Model and next sentence prediction—and they obtain the representation of word level and sentence level, respectively. The structure of the BERT model is shown in Figure 2.

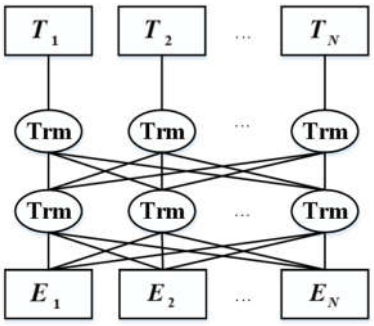

Figure 2. The structure of Bidirectional Encoder Representations from Transformers (BERT) Model.

3.2.1. Masked Language Model

In order to train the deep bidirectional transformer representation, a simple method is adopted. we mask some of the input words randomly, and then predicting those concealed words, this method is called the "Masked Language Model" (MLM). In the process of training, 15% of the tokens in each sequence are randomly masked, and not every word is predicted like Continuous Bag of Words

Model (CBOW) in word2vec. MLM randomly masks words from the input; the goal of MLM is to predict the original words of the masked words based on the context. Unlike the left-to-right pre-training language model, MLM merges context on the left and right side, which allows pre-training of the deep bidirectional transformer. The transformer encoder does not know which words will be predicted, or which words have been replaced by random words, so it must maintain a distributed context representation for each input word. Besides, since random replacement occurs only 1.5% in all words, it does not affect the model's understanding of the language.

3.2.2. Next Sentence Prediction

Many sentence-based tasks, such as Automatic Question and Answer (QA) [24] and Natural Language Inference (NLI) [25], need to understand the relationship between two sentences. Therefore, in the above Masked Language Model task, after the first step, 1.5% of the words are covered. In this task, there is a need to randomly divide the data into two parts of equal size. Two statement pairs in the first part of the data are context-constrained, and the two statement pairs in the other part of the data are not continuous in context. Then, let the transformer model identify if these statements are continuous. Its purpose is to deepen the understanding of the relationship between two sentences. This allows the pre-trained model to adapt to more tasks better.

3.3. Attention Encoding Layer

The attention encoding layer interactively obtains the hidden state in parallel from the input embedding layer between each context word and other context words. It contains two parts, the Multi-Head Attention (MHA) and the Location Point-Wise Feed-Forward Networks.

3.3.1. Multi-Head Attention

We use multiple independent attention mechanisms for introspective context word modeling. Its essence is a collection of multiple attentional attention mechanisms. The purpose is to learn the word dependence within the sentence and capture the internal structure of the sentence. Given a context embedding x^c, we can obtain the introspective context representation $c^{ic} = \{c_1^{ic}, c_2^{ic}, ..., c_n^{ic}\}$ as follows:

$$c^{ic} = \mathrm{MHA}(x^c, x^c) \tag{1}$$

$$\mathrm{MHA}(x^c, x^c) = [O_1^c; O_2^c; \ldots; O_{n_{head}}^c]W_O \tag{2}$$

$$O^C = \mathrm{Attention}(x^c, x^c) \tag{3}$$

where [;] is the vector concatenation，W_O is a linear variable weight matrix, O_h^c is the output of the $h-th$ head attention, and $h \in [1, n_{head}]$.

$$\mathrm{Attention}(x^c, x^c) = \mathrm{softmax}(\frac{f(x^c, x^c)}{\sqrt{d_k}})x^c \tag{4}$$

$$f(x_i^c; x_j^c) = \tanh([x_i^c; x_j^c] * W_s) \tag{5}$$

where W_s is a weight matrix, $\sqrt{d_k}$ acts as a regulator so that the inner product is not too large, the goal of f is to evaluate the semantic relevance of x_i^c and x_j^c.

In this part, we use the Multi-Head Attention mechanism to evaluate the semantic relevance between each context word and other context words, then the output obtained at the Multi-Head Attention part is used as the input to the next part.

3.3.2. Location Point-Wise Feed-Forward Networks

Point-Wise Feed-Forward Networks use two linear transformations to transform the information obtained by MHA as follows:

$$\mathrm{PFFN}(c^{ic}) = \sigma(W_0 c^{ic} + b_0)W_1 + b_1 \tag{6}$$

$$c^{ic-FNN} = \mathrm{PFFN}(c^{ic}) \tag{7}$$

where W_0 and W_1 are the learnable weight. b_0, b_1 are the biases, σ is the ELU(Exponential Linear Unit) activation. PFNN (Point-Wise Feed-Forward Networks) will obtain a sequence $c^{ic-FNN} = \{c_1^{ic-FNN}, c_2^{ic-FNN}, ..., c_n^{ic-FNN}\}$.

Then, we consider the effects of the location message. Context words in different positions may have different effects on the hidden states. For example, in "the price is reasonable although the service is poor!", the word "poor" should be used to describe the "service" rather than "price". But we use MHA to obtain the hidden state, and the location information is not fully utilized. So, we combine location information into the output of MHA. Location weight is defined as follow:

$$\begin{cases} w^{\mathrm{i}} = 1 - \dfrac{l+1}{M+N} & l > 0 \\ w^{\mathrm{i}} = 0 & l \le 0 \end{cases} \tag{8}$$

where l is the distance from context word to the target word, N is the number of the target word, M is the number of context word. In this part, we let $N = M = n$ to obtain the interactive information between each context word and other context words, while $x_i^c = x_j^c, W^i = 0$; this is to avoid the impact of the word itself.

Then, we can obtain the final outputs of context words $H^{(0)} = \{h_0^{(0)}, h_1^{(0)}, ..., h_n^{(0)}\}$;, $h_i^{(0)}$ is defined as follow:

$$h_i^{(0)} = c_i^{ic\text{-}FNN} * W^i \tag{9}$$

3.4. Target-Context Interaction Layer

We use the attention encoding layer to compute the hidden states of the input embedding. In order to better obtain the target-centered interactive information, we employ Bi-GRU to obtain the target word representation first, and then we selectively obtain the interactive information between the target word and the context word by the target-context pair.

3.4.1. Gated Recurrent Neural Networks (GRU)

Gated Recurrent Neural Networks (GRU) is a variant of RNN, and it addresses the problem of gradient disappearance to a certain extent by delicate gate control, like Long Short-Term Memory (LSTM). Although the difference in performance between LSTM and GRU is small, GRU is more lightweight, which is shown in Figure 3.

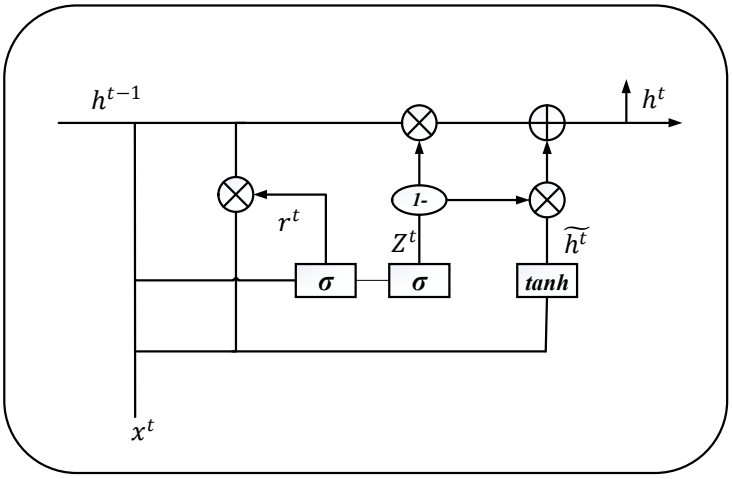

Figure 3. The structure of the Gated Recurrent Neural Networks (GRU) Model.

As Figure 3 shows, there is a current input x^t, and the hidden state h^{t-1} is passed by the previous node; this hidden state h^{t-1} contains information about the previous node. Combined with x^t and h^{t-1}, GRU will obtain the output h^t, which is a hidden state passed to the next node. The process can be formalized as follows:

$$r^t = \sigma(W^r[h^{t-1}; x^t]) \tag{10}$$

$$Z^t = \sigma(W^Z[h^{t-1}; x^t]) \tag{11}$$

$$\widetilde{h^t} = \tanh(W^{\tilde{h}}[r^t * h^{t-1}; x^t]) \tag{12}$$

$$h^t = Z^t * \widetilde{h^t} + (1 - Z^t) * h^{t-1} \tag{13}$$

where r^t is the gated control of reset, Z^t is the gated control of update. The $\widetilde{h^t}$ here mainly contains the data currently input x^t and adds $\widetilde{h^t}$ to the current hidden state in a targeted manner; this process is similar to the selected memory phase of LSTM. $W^r, W^Z, W^{\tilde{h}}$ are the learnable weight matrixes. σ is the activation function. Then, we employed Bi-GRU to obtain the target word representations:

$$h_j^t = [\overrightarrow{GRU(x_j^t)}; \overleftarrow{GRU(x_j^t)}] \tag{14}$$

3.4.2. Target-Context-Interaction

We employ the target-context pair to obtain interactive information, and it interacts with each target and all other context words to obtain the hidden information.

$$U_{ij} = W_u([h_i^{(0)} * h_j^t]) \tag{15}$$

$$a_{ij}^{fa} = \frac{\exp(U_{ij})}{\sum_{k=1}^{n} \exp(U_{kj})} \tag{16}$$

where W_u is the weight matrix, U_{ij} is the target-context pair between $i-th$ context word and $j-th$ target word, a_{ij}^{fa} is the attention weights on $j-th$ target word to $i-th$ context word. For these reasons, we can obtain the output vector $m^{fa} = \{m_0^{fa}, m_1^{fa}, ..., m_m^{fa}\}$ as follow:

$$m_j^{fa} = (\sum_{i=1}^{n} h_i^{(0)} * a_{ij}^{fa}) * h_j^t \tag{17}$$

In this section, we use the target-context pair to obtain the contextual interactive information with the target word as the core. But this method still has shortcomings, for example, in a long sentence, there will be many context words that will affect the target word, and when these context words are too many, the weight of some context words will be small or even ignored. So, we use the Context-Target-Interaction layer to address this problem.

3.5. Context-Target-Interaction Layer

Because we determine the number of CTI based on the number of context words, the context word is at the core of obtaining interactive information. After that, we input the target-context pair information obtained into each CTI. Then, in order to address the problem of insufficient fusion of context information and interactive information, we propose a coefficient loss forwarding mechanism.

3.5.1. Context-Target Interaction

In Context-Target- Interaction layer, we use the target word as the core and generated a target-context pair sequence m^{fa} of length m. In this layer, we combine target-context pairs with context words dynamically. In a long sentence, each context word has a different degree of influence on the target word. So, each context word $h_i^{(l)}$ should have different attention to the target-context pair m_j^{fa}.

$$R_i^t = \sum_{j=1}^{m} m_j^{fa} * \mathcal{F}(h_i^{(l)}, m_j^{fa}) \tag{18}$$

In this function, R_i^t means that we combine $h_i^{(l)}$ with m^{fa} dynamically; the function $\mathcal{F}$ represents the correlation between the $j-th$ target-context pair m_j^{fa} and the $i-th$ word-level representation $h_i^{(l)}$ as follows:

$$\mathcal{F}\left(h_i^{(l)}, m_j^{fa}\right) = \frac{\exp(h_i^{(l)\top} m_j^{fa})}{\sum_{k=1}^{m} \exp(h_i^{(l)\top} m_k^{fa})} \tag{19}$$

Then, we feed R_i^t and $h_i^{(l)}$ to a fully-connected layer to obtain the interactive information representation $\tilde{h}_i^{(l)}$ between the target word and the context word.

$$\tilde{h}_i^{(l)} = g(W^t[h_i^{(l)}; R_i^t] + b^t) \tag{20}$$

where $[;]$ is the vector concatenation, g is a nonlinear activation function, W^t, b^t are the weights of this part.

3.5.2. Coefficient Loss Forwarding Mechanism

Although we use the context word as the core of the Context-Target-Interaction layer, when we combine it with the target-context pair in CTI, the context information may be lost due to too many CTI layers. So, we use a simple but effective strategy to save context information as follow:

$$h_i^{(l+1)} = h_i^{(l)} * \tilde{h}_i^{(l)} * a + \tilde{h}_i^{(l)} * (1-a) \tag{21}$$

where $h_i^{(l)}$ is the input of the $l-th$ layer, $\tilde{h}_i^{(l)}$ is the output of the fully connected layer of the $l-th$ layer. a is the proportional coefficient value and its value range is (0.1, 0.9), and the output of

each layer will contain contextual representations. Because the context representation is fused with the target-context interaction information in a certain proportion, this strategy is called the "coefficient loss forwarding mechanism".

3.6. Select Convolution Layer

We feed the output of Context-Target-Interaction layer to the convolutional layer to generate the feature map c^i as follow:

$$c^i = \text{Relu}(w_{conv}^{\top} h_{i:i+s-1}^{(l)} + b^{conv}) \tag{22}$$

where $h_{i:i+s-1}^{(l)}$ is the concatenated vector of $h_i^{(l)}, ..., h_{i+s-1}^{(l)}$, $w_{conv}^{\top}$ and b^{conv} are weights of the convolutional kernel, s is the kernel size. Then, we use average pooling to obtain the sentence representation by employing n^k kernels:

$$z = [\frac{\sum_{i=1}^{n^k} c^i}{n^k}]^{\top} \tag{23}$$

Finally, we use a fully connected layer to determine the sentimental tendency:

$$p(y \mid x^t, x^c) = \text{Softmax}(W^f z + b^f) \tag{24}$$

where W^f and b^f are learnable parameters.

4. Experiment

4.1. Experimental datasets

We conducted our experiments on different datasets; these datasets included restaurant reviews (Rest 14) and laptop reviews (Laptop) of SemEval-2014 Task 4 dataset [26], ACL 14 Twitter dataset (Twitter) [27], SemEval-2015 Task 12 dataset (Rest 15) [28], and SemEval-2016 Task 5 dataset (Rest 16); the data in these datasets were classified into positive, neutral, and negative. The details of the datasets are shown in Table 1.

Table 1. The details of the datasets.

Dataset	Positive		Neutral		Negative	
	Train	Test	Train	Test	Train	Test
Laptop	994	341	870	128	464	169
Restaurant	2164	728	807	196	637	196
Twitter	1561	173	3127	346	1560	173
Restaurant15	912	326	36	34	256	182
Restaurant16	1240	469	69	30	439	117

4.2. Experimental Settings

The target word vector and the context word vector in our experiment were initialized using the pre-trained BERT [29] word vectors. The dimension of word embedding was 768, the learning rate was 2e-5, all the biases were set to zero, the dropout rate was 0.2, the optimizer was Adam, the number of epochs was 30, and the kernel size s was 1.We used Accuracy and Macro-F1 [30] to judge the performance of a model. The experimental environment is shown in Table 2.

Table 2. Experimental environment.

Experimental environment	Environmental configuration
Operating system	Windows10
GPU	GeForce RTX 2080
Programing language	Python 3.6
PyTorch	1.1.0
Word embedding tool	BERT

4.3. Model Comparisons

To evaluate the performance of our model, we compared it with the following baseline models:

ATAE-LSTM: Traditional LSTM models cannot obtain important semantic information from text, and they have proposed AT-LSTM(Attention-based Bidirectional Long Short-Term Memory) to address this problem. Then, they have proposed ATAE-LSTM to make full use of aspect word information. ATAE-LSTM combines aspect embedding and word embedding on the input layer and also introduces aspect information when calculating the weight.

IAN: IAN interactively learns attention in the context words and target words. It uses the LSTM to obtain the hidden state of context and target embedding. IAN then combines the average output of the hidden layer with the attention mechanism to generate attention weight. The final target attention weight and context attention weight are connected in series as the input of the SoftMax function to obtain the classification result.

PBAN: PBAN considers the influence of context words in different positions on the sentiment polarity of the target word, and the context words closer to the aspect words will make a greater impact on the target words. Then, they use a bidirectional attention mechanism to model target words and sentences.

TNet-LF(Target Specific Transformation Networks-Lossless Forwarding): This model employs a CNN layer to extract features and a bi-directional RNN layer to obtain a hidden state. Between these two layers, it proposes a component to generate target-specific representations of words in the sentence; meanwhile, it proposes a mechanism to retain contextual information selectively [31].

TNet-ATT(Transformation Network with An Attention Mechanism): Traditional attention mechanisms tend to focus too much on high-frequency words, and it has strong sentiment polarity in the data, but it ignores words that are less frequent. This model proposes a progressive self-supervised attention learning algorithm that can automatically and progressively mine important supervised information in the text, thereby constraining the learning of the attention mechanism during model training.

BERT-PT(Bidirectional Encoder Representation from Transformers Post-Training): This model combines review reading comprehension with ABSC, and it uses the form of QA to serve to address the ABSC problem. The author believes that as a pre-trained language model, just fine-tuning during training is not enough. The author believes that before using BERT, BERT can be adjusted from two aspects—the domain and task. So, they have proposed BERT post-training [32].

CDT(Convolution over a Dependency Tree): This model verifies the possibility of integrating dependency trees with neural networks for representation learning. CDT exploits a Bi-LSTM(Bidirectional-Long Short-Term Memory) to learn representations for features of a sentence. They further enhance the embeddings with a Graph Convolutional Network (GCN), which operates directly on the dependency tree of the sentence [33].

ASGCN(Aspect-specific Graph Convolutional Networks): This model proposes to exploit syntactical dependency structures within a sentence and resolves the long-range multi-word dependency issue for Aspect-Based Sentiment Classification. ASGCN posits that GCN is suitable for Aspect-Based Sentiment Classification and proposes a novel aspect-specific GCN model [34].

We compared MIN with other models on different datasets; the results are shown in Table 3.

Table 3. The experimental results.

Models	Rest14		Laptop		Twitter		Rest15		Rest16	
	Acc[1]	F1[2]	Acc	F1	Acc	F1	Acc	F1	Acc	F1
ATAE-LSTM	0.7720	-	0.6870	-	-	-	-		-	
IAN	0.7860	-	0.7210	-	0.7250	0.7081	0.7854	0.5265	0.8474	0.5521
PBAN	0.8116	-	0.7412	-	-	-	-		-	
TNet-LF	0.8079	0.7084	0.7601	0.7147	0.7468	0.7336	0.7847	0.5947	0.8907	0.7043
TNet-ATT	0.8339	0.7566	0.7536	0.7202	0.7861	0.7772	-		-	
BERT-PT	0.8495	0.7696	0.7807	0.7508	-	-	-		-	
CDT	0.8230	0.7402	0.7719	0.7299	0.7466	0.7366				
ASGCN	0.8086	0.7219	0.7262	0.6672	0.7105	0.6945	0.7834	0.6078	0.8833	0.6748
MIN	0.8268	0.7405	0.7978	0.7549	0.7384	0.7237	0.8284	0.6926	0.8912	0.6867

[1] Accuracy; [2] Macro-F1.

As shown in Table 3, the ATAE-LSTM model performed the worst among all the above models; one of the important reasons was that it relied on the LSTM to obtain the hidden state. Although ATAE-LSTM used the attention mechanism, it still could not comprehensively context analytical modeling. Our model also used a lighter GRU than LSTM, and MIN did not depend entirely on GRU.

IAN performed better than ATAE-LSTM. IAN interactively learned attention in the context words and target words. But its target word and context word interactions were still coarse-grained, which might cause loss of interactive information. So, our model used a more fine-grained way to obtain interactive information. PBAN had better results than the model above because they noticed that the context word in different positions would have different effects on the target word, and our model took this into account.

ASGCN proposed a novel aspect-specific GCN model, and it was the first to use the GCN for emotion classification. CDT performed better than ASGCN; CDT exploited a Bi-LSTM to learn representations for features of a sentence and further enhanced the embeddings with a GCN, which operated directly on the dependency tree of the sentence. It propagated both contextual and dependency information from opinion words to aspect words, offering discriminative properties for supervision.

TNet-ATT proposed a progressive self-supervised attention mechanism algorithm based on TNet-LF. In these models, TNet-ATT performed best on the Twitter dataset, probably because many sentences in the TWITTER dataset were not standardized, and TNet-ATT could design different supervision signals for different situations.

BERT-PT performed best in the restaurant reviews dataset, probably because they proposed BERT post-training based on pre-training BERT, and BERT post-training could better adjust BERT from both the domain and the task. Besides, the model connected questions and comments as input, predicted the probability of each word in the comment at the beginning and end of the answer, and then calculated the loss with the position of the real answer.

4.4. Analysis of model

4.4.1. Analysis of CTI

The essence of CTI was the information interactive fusion component. The information we obtained from the previous layer and the information from this layer need to be weighted in the interactive fusion process. This was the origin of the coefficient a. We conducted related experiments on the laptop reviews dataset to find the optimal coefficient a. We only changed the coefficient a and kept the other parameters unchanged to conduct the experiment. We recorded changes in Acc and Macro-F1 in Figure 4.

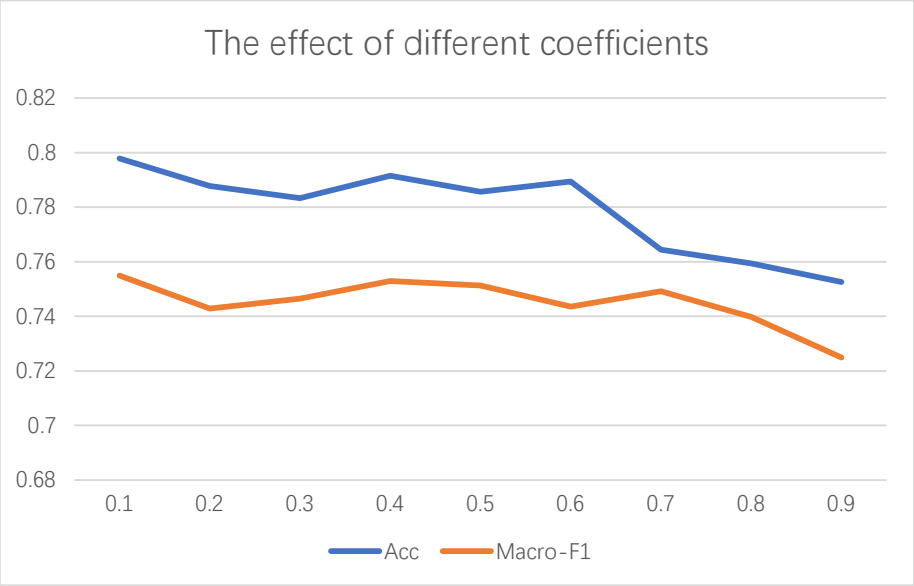

Figure 4. The effect of different coefficients.

From Figure 4, we could observe that Acc and Macro-F1 showed a downward trend with increasing coefficients, and the most suitable coefficient was 0.1. Since MIN involved multiple CTI layers, we needed to study the effect of the number of layers and further prove the necessity of the coefficient. We conducted comparative experiments and recorded the result in Figure 5 and Figure 6.

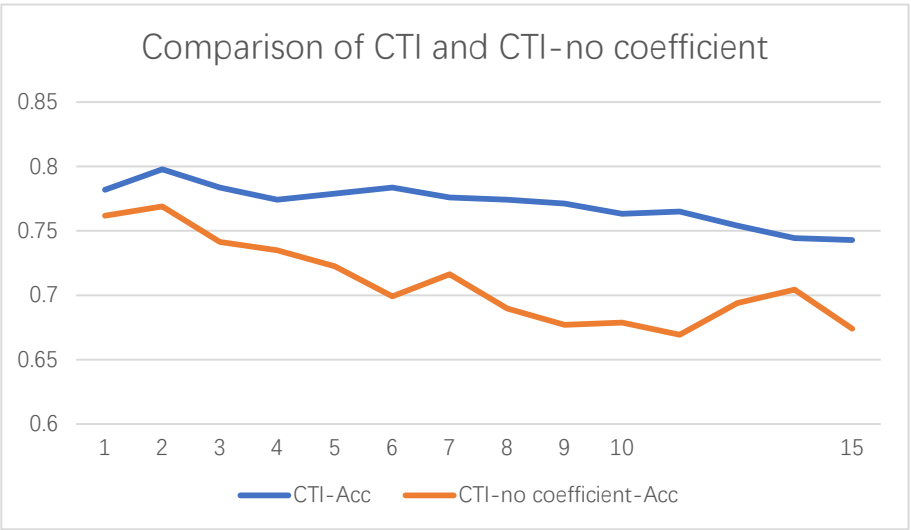

Figure 5. Comparison of CTI-Acc and CTI-no coefficient-Acc.

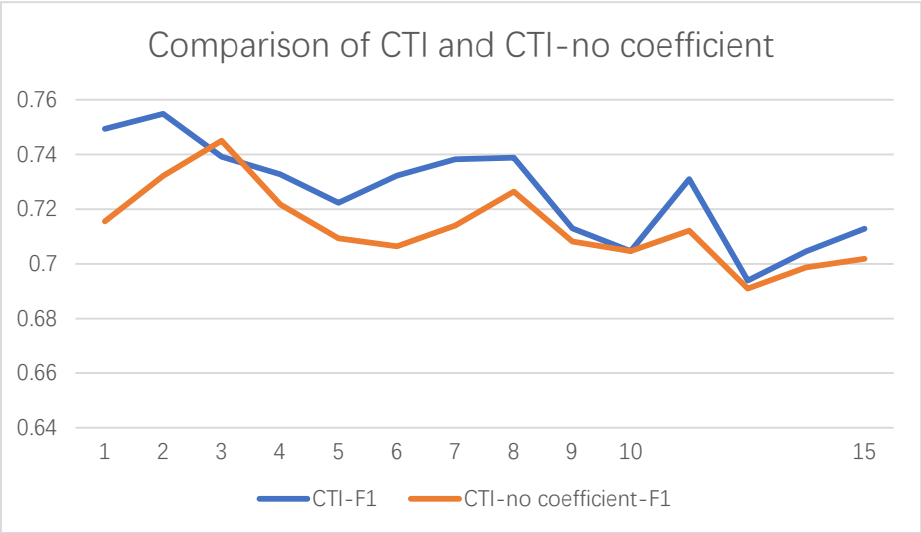

Figure 6. Comparison of CTI-F1 and CTI-no coefficient-F1.

Figure 5 and Figure 6 show the experimental result. In CTI, when the number of layers $L < 2$, the Acc and F1 were increasing. When $L > 2$, the accuracy and F1 showed a downward trend and obtained the best results when the number of layers was equal to 2. It was probably because as the number of layers increased, the model might focus more on context information and ignore a large part of the target-context interactive information. Experimental results showed that the overall performance of the CTI-no coefficient was significantly worse than CTI in our model. This difference was mainly focused on the choice of the coefficient. This further validated the importance of the proper selection of coefficients for CTI.

4.4.2. Case Study

In order to obtain a deeper understanding of MIN, we visualized the focus of the target words and context words in Figure 7; the deeper the color, the higher the attention.

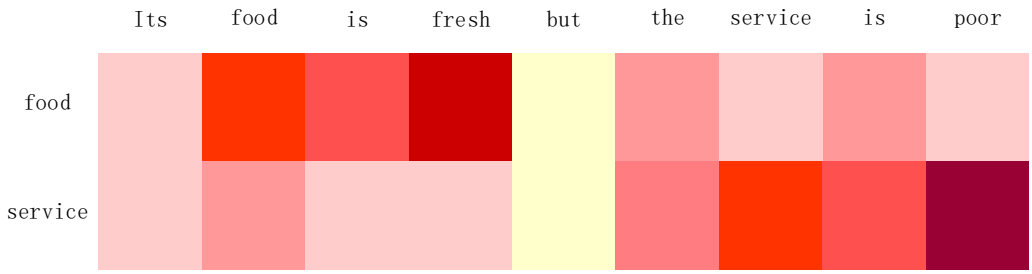

Figure 7. The visualized attention weights for the sentence and aspect term by MIN.

As shown in Figure 7, the sentence "Its food is fresh, but the service is poor" contains two aspects "food" and "service", but the weight of the context word is different for each aspect word. It could be clearly observed that MIN placed great emphasis on the words that needed attention as we expected. For example, in terms of the aspect "food", "fresh" received the highest attention, "poor" got a lower level of attention. It effectively avoided the influence of other sentiment words on itself and paid attention to the sentiment words related to itself. This was the result of our model MIN using multiple attention mechanism and dynamic combined context information.

5. Conclusions and Future Work

In this paper, we proposed a model MIN based on current approaches for ABSC. Through in-depth analysis, we first pointed out the shortcomings of current approaches: difficult to parallelize and insufficiently obtain and fuse the interactive information. In order to address the first problem, we used Multi-Head Attention and location information to obtain contextual hidden information. In order to address the problem to insufficiently obtain and fuse the interactive information, we used BI-GRU to obtain target information, and then we used target-context pair and CTI for obtaining effective information and its fusion. The target-context pair took the target word as the core, and CTI took the context words as the core. Target-context pair and CTI were more fine-grained methods than current approaches. In the end, substantial experiments conducted on different datasets demonstrated that our approach was effective and robust compared with several baselines.

Although it was validated that our proposal showed great potentials for ABSC, there were still shortcomings in our method; for example, we ignored syntactic dependencies within sentences when obtaining interactive information. We thought syntactic dependencies within sentences could help us to improve experimental results for ABSC. In fact, this is a challenging research direction, and we believe it will play an important role in the field of sentiment analysis.

Author Contributions: Writing—original draft, D.Z.; Writing—review and editing, Z.Z.; Data curation, Q.L.; Methodology, D.Z.; Software, Q.G.; Funding acquisition, D.Z.; Project administration, D.Z.; Investigation, H.P.; Validation, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Social Science Foundation (19BYY076); Science Foundation of Ministry of Education of China (14YJC860042); Shandong Provincial Social Science Planning Project (19BJCJ51, 18CXWJ01, 18BJYJ04).

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

References

1. 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.
2. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Mohammad, A.S.; 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.
3. Vinodhini, G.C.; handrasekaran, R.M. Sentiment analysis and opinion mining: A survey. *Int. J.* **2012**, *2*, 282–292.
4. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA 9–15 July 2016; pp. 2873–2879.
5. Mikolov, T.; Karafiát, M.; Burget, L.Š.; Černocký, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
6. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2014**, arXiv:1409.0473.
7. Firat, O.; Cho, K.; Bengio, Y. Multi-way, multilingual neural machine translation with a shared attention mechanism. 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. 866–875.
8. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pretraining of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, 2–7 June2019; Volume 1,_pp. 4171–4186.
9. Read, J. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of the ACL Student Research Workshop, Ann Arbor, MI, USA, 27 June 2005; pp. 43–48.

10. Duwairi, R.M.; Qarqaz, I. Arabic sentiment analysis using supervised classification. In Proceedings of the 2014 International Conference on Future Internet of Things and Cloud, Barcelona, Spain, 27–29 August 2014; pp. 579–583.
11. Narayanan, V.; Arora, I.; Bhatia, A. Fast and accurate sentiment classification using an enhanced Naive Bayes model. In Proceedings of the Intelligent Data Engineering and Automated Learning—IDEAL 2013, Hefei, China, 20–23 October 2013, pp. 194–201.
12. Rathore, S.; Park, J.H. Semi-supervised learning based distributed attack detection framework for IoT. *Appl. Soft Comput.* **2018**, *72*, 79–89.
13. Rathore S, Sharma P K, Loia V, et al. Social network security: Issues, challenges, threats, and solutions. *Inf. Sci.* **2017**, *421*, 43–69.
14. Tang, D.; Qin, B.; Feng, X.; Liu, T. Effective LSTMs for Target-Dependent Sentiment Classification. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–17 December 2016; pp. 3298–3307.
15. Zheng, J.; Cai, F.; Shao, T.; Chen, H. Self-interaction attention mechanism-based text representation for document classification. *Appl. Sci.* **2018**, *8*, 613.
16. Wang, Y.; Huang, M.; 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.
17. Tang, D.; Qin, B.; Liu, T. Aspect Level Sentiment Classification with Deep Memory Network. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 214–224.
18. 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.
19. Gu, S.; Zhang, L.; Hou, Y.; Song, 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.
20. Tang, J.; Lu, Z.; Su, J.; Ge, Y.; Song, L.; Sun, L.; Luo, J. Progressive Self-Supervised Attention Learning for Aspect-Level Sentiment Analysis. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics(ACL 2019), Florence, Italy, 28 July–2 August 2019; pp. 557–566.
21. Chung, J.; Gulcehre, C.; Cho, K.H.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv: 1412.3555.
22. Kim, H.; Jeong, Y.S. Sentiment Classification Using Convolutional Neural Networks. *Appl. Sci.* **2019**, *9*, 2347.
23. Lee, J.; Park, J.; Kim, K.; Nam, J. Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification. *Appl. Sci.* **2018**, *8*, 150.
24. Zhang, S.; Zhang, X.; Wang, H.; Cheng, J.; Li, P.; Ding, Z. Chinese medical question answer matching using end-to-end character-level multi-scale CNNs. *Appl. Sci.* **2017**, *7*, 767.
25. Parikh, A.P.; Täckström, O.; Das, D. Uszkoreit. A decomposable attention model for natural language inference. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016), Austin, TX, USA, 1–4 November 2016; pp. 2249–2255.
26. 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.
27. Nakov, P.; Rosenthal, S.; Kiritchenko, S.; Mohammad, S.M.; Kozareva, Z.; Ritter, A.; Stoyanov, V.; Zhu, X. Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts. *Lang. Resour. Eval.* **2016**, *50*, 35–65.
28. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; Androutsopoulos, I. 2015.Semeval-2015 task 12: Aspect based sentiment analysis. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, USA, 4–5 June 2015; pp. 486–495.
29. 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.
30. Lipton, Z.C.; Elkan, C.; Naryanaswamy, B. Optimal thresholding of classifiers to maximize F1 measure. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in

Databases(ECML PKDD 2014), Nancy, France, 15–19 September 2014; Springer: Berlin/Heidelberg, Germany, 2014. pp. 225–239.

31. 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.
32. Xu, H.; Liu, B.; Shu, L.; Philip, S.Y. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. 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. 2324–2335.
33. Sun, K.; Zhang, R.; Mensah, S.; Mao, Y.; Liu, X. Aspect-Level Sentiment Analysis via Convolution over Dependency Tree. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp 5679–5688.
34. Zhang, C.; Li, Q.; Song, D. Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp 4568–4578.