Multi-Task Dynamic Joint Training for Aspect-Base-Sentiment-Analysis

Pingchuan Ma[1,2], Bo Zhao[3], Xianxun Liu[3], Peng Sun[2,3]*

[1] School of Computing and Artificial Intelligence, Southwest Jiaotong University, China
[2] Hebi Institute of Engineering and Technology, Henan Polytechnic University, China
[3] Henan Chuitian Technology Co., Ltd, China
371027612@qq.com, guigushen@163.com, 403321565@qq.com, 223582872@qq.com

Abstract

In Aspect-Base-Sentiment-Analysis (ABSA), a single sentence may contain multiple aspect words, making it challenging for the model to find appropriate sentiment words for each aspect word from the text. Therefore, this paper proposes a Multi-Task Dynamic Joint Training (MDJT). To learn better feature representations, we apply a text feature extractor and a graph feature extractor to extract semantic features and graph-structured features of text, respectively. In this paper, we use the multi-task frame to joint two tasks dynamically: the main task is the sentiment classification task (Polarity Task), which is used to classify the sentiment of each aspect word; the auxiliary task is the opinion word task (Opinion Task), whose purpose is to extract text. The features are most likely to become opinion words and the accuracy of the main task sentiment classification are further improved by optimizing the loss value of this task. Experiments on several benchmarking collections illustrate that our proposed model has comparable effectiveness to a range of state-of-the-art models, and they further demonstrate that both syntactical information and multi-task join dynamically and properly.

Keywords: Multi-task dynamic joint training, Aspect-based sentiment classification, Opinion word attention module, Weight dynamic allocation strategy

1 Introduction

Nowadays, sentiment classification has entered our life and plays a vital role in the field of natural language processing (NLP) [1]. Aspect-Based Sentiment Analysis (ABSA) [2-3] is a sub-task of sentiment classification. Unlike the general classification of samples in general sentiment, ABSA aims to judge the sentiment polarity of one or more aspect words within a single sentence. Aspect-based sentiment classification tasks are now active in various industrial fields [4].

HelloBike is a platform focusing on local travel and life services. In addition to shared services (bikes, rides), there are hotels, shopping malls, communities, etc. Mining useful user feedback information can help companies improve their product experience. We mine effective information from the following three perspectives. **(1) Opinion Word Angle:** For example, given a comment "The place is small and cramped but the food is fantastic", make emotional judgments for the aspect words *place* and *food* respectively. The aspect word *place* is a negative emotion, and the aspect word *food* is a positive emotion. The effective judgment of the aspect words *place* and *food* mainly comes from the information provided by the words *small* and *fancy*, which are called opinion words. Therefore, how to extract the features that are most likely to become opinion words from the text is the key to improving the ABSA task. **(2) Feature Angle:** When using neural networks (such as CNN, LSTM, etc.) [5-7] to extract features from the text, the part-of-speech dependencies between tokens in the text are not considered. Both cases in Figure 1 are sentiment classifications for the aspect word *recipe*. In case1, *like* is used as a verb, and expresses the positive sentiment towards the aspect word *recipe*. In this example, ordinary neural networks can be used to classify it successfully. In case 2, *like* is used as a preposition and does not contain any sentiment, and using traditional methods can lead to misclassification. Therefore, learning better feature representations is beneficial for sentiment classification. **(3) Model Angle:** With the increase in the types of neural network models, when solving our downstream tasks, using a single model cannot achieve good results. Model fusion is often used to improve task indicators. Different models will extract features at different levels; for example, global features and local features are extracted by two different models. Therefore, for different tasks, the importance of the features extracted by each model will be different, and it is unreasonable to use the conventional feature fusion method (fixed weight). Determining how to dynamically assign different weights to different models will also further improve the performance of the model.

***case1**: I **like** the **recipe** here.*

***case2**: The **recipe** includes some chinese some **like** dumplings.*

Figure 1. The influence of part of speech on ABSA

Based on the above analysis, this paper presents the following innovations:

*Corresponding Author: Peng Sun; Email: 223582872@qq.com
DOI: https://doi.org/10.70003/160792642026052703010

1. Opinion-Aware Attention Mechanism for ABSA

We introduce opinion words into the Aspect-Based Sentiment Analysis (ABSA) task by constructing an opinion-aware attention mechanism. This mechanism extracts words in the text that are most likely to be opinion words, thereby enhancing the accuracy of sentiment classification through the use of Multi-Task Dynamic Joint Training (MDJT).

2. Enhanced Feature Representation through Dual Encoders

Our model improves text feature representation by learning both semantic features and graph-structured representations. This is achieved through the construction of a Text Encoder and a Graph Encoder.

3. Dynamic Weight Allocation Strategy

To address the subjective judgment regarding the importance of specific modules, we propose a dynamic weight allocation strategy that adapts based on the contributions of each module.

4. Comprehensive Evaluations on Benchmark Datasets

Extensive evaluations conducted on the Restaurant and Laptop datasets demonstrate that our proposed model significantly outperforms baseline models.

2 Related Works

Deep neural networks and graph neural networks have achieved great success in the ABSA task. Today, most deep neural networks are based on traditional neural networks (CNN, LSTM, etc.) or pre-trained models (BERT, etc.). Most graph neural networks are based on GNN, GAT, etc. Below, we present important work based on them in the ABSA task.

2.1 Semantic Feature Extractor

To better solve the ABSA task, researchers have made further explorations into deep neural networks, such as Dong; Tang; Chen et al. [8-12] who used neural network models to extract sentence feature representations. On this basis, Wang; Ouyang et al. [13-14] combined lexical resources with neural networks to achieve state-of-the-art performance on the ABSA task. Due to the lexical and syntactic complexity of language, for two different aspect words in the same sentence to express opposite emotions, it is inappropriate to use neural networks to assign only sentence-level sentiment polarity. Therefore, some researchers strive to use the syntactic structure of sentences to establish connections and then use dependency-based parse trees to provide more comprehensive syntactic information, such as Li; Nguyen et al [15-17]. However, this kind of method cannot extract the features of the words around the aspect words very well. With the birth of the pre-trained model Bert [18], it has achieved success in natural language processing (NLP) tasks including ABSA. For example, Li et al. [19] exploited Bert's end-to-end approach to explore the ABSA task. Sun et al. [20] transformed the ABSA task into a sentence pair classification task by constructing auxiliary sentences. Chen et al. [21] used Bert as a feature extractor and adopted a machine reading comprehension approach to solving the ABSA task.

Several other efforts have also been explored in the ABSA mission. In neural networks, some models used RNN variants such as LSTM and GRU to encode representations of sentences. Gu et al. [22] improved the performance by considering the location of aspect words. Zheng and Xia [23] used LSTMs to extract sentence contextual semantic features and target word feature representations while considering the interaction between target words and context. Majumder et al. [24] used a GRU and an attention mechanism to learn feature representations of sentences. However, these neural network-based methods ignore the syntactic dependencies between aspect words and opinion words.

2.2 Graph Structure Feature Extractor

Recently, the combination of graph neural networks and dependency trees has achieved good results in the ABSA task. Zhang et al. [25] used graph convolutional neural networks (GCNs) to learn dependency tree node feature representations and used them along with other features for sentiment classification. Huang and Carley [26] used a graph attention network (GAT) to build dependencies between words. Sun et al. [27] proposed a method based on dependency tree convolution to reduce the distance between aspect words and opinion words so that dependencies can be efficiently stored in long sentences. Wang et al. [28] proposed an aspect-oriented dependency tree structure, which reconstructs the original dependency tree and builds a tree with aspect words as the root node.

3 Model

The structure of our model is shown in the Figure 2 below, which consists of four parts: the Encoder Layer, the Feature Fusion Layer, the Opinion Attention Layer, and the Output Layer. Suppose the input sentence is: $X = \{x_1, x_2, x_{i-1}, x_i, x_{i+1}, x_N\}$. The goal of our main task is three-category sentiment polarity: to classify sentiment for each aspect word in sentence X. The goal of the auxiliary task is multi-classification of opinion words: judging whether that sentence X belongs to one or several opinion word categories.

3.1 Encoder Layer

The word embedding layer (Encoder Layer) includes two modules: Text Encoder and Graph Encoder, which function to extract the semantic features of text and the graph-structured features of text, respectively.

3.1.1 Text Encoder

For a comment text consisting of N words $X = \{x_1, x_2, ..., x_N\}$, the Text Encoder mainly extracts the semantic features of each word according to the contextual information of the text. BERT, a pre-trained model proposed by Google in 2018, has been experimentally verified to achieve good results in natural language processing tasks. Subsequently, researchers have proposed various pretrained models based on BERT improvements, such as Roberta, Albert, etc., and have achieved good results. Inspired by the successful practice of many NLP tasks, this chapter adopts the initial pre-trained model BERT when extracting text semantic features, which is composed of 12-layer Transformer blocks and has a strong feature extraction ability. Of course, other pre-trained models can also be used. A transformer with 12 layers, along with the hidden layer features of the last layer of the Transformer, are used as the feature representation of the entire sentence. The Text Encoder part is as follows:

$$H^b = TextEncoder([x_1, x_2, \cdots, x_N]) \tag{1}$$

where $H^b \in \mathbf{R}^{N\times d}$ and d represents the dimension of the hidden layer vector. The Text Encoder mainly extracts the semantic features of each word (including context information) according to the context information of the text. It can be replaced by traditional neural network models, such as CNN and LSTM. It can also be replaced by pre-trained models, such as Roberta and Albert.

3.1.2 Graph Encoder

The Graph Encoder mainly uses a graph neural network to extract graph-structured features of the text. Given a graph $\mathcal{G}$, the graph-structured feature extractor mainly learns the word vector representation of the graph structure $g = \mathcal{G}(g, \{x_v\})$, where $\mathcal{G}(\cdot)$ represents the graph structure feature extractor and $\{x_v\}$ represents the initial node representation of the input graph structure feature extractor. The Graph Encoder part is as follows:

$$H^g = GraphEncoder([H_1^t, H_2^t, \cdots, H_N^t]) \tag{2}$$

where $H^g \in \mathbf{R}^{N\times f}$ and f represents the feature dimension of each node. Similar to the Text Encoder, the Graph Encoder does not have a fixed model. The function of this part is only to extract the graph structured features of the text. GCN network, GAT network, or R-GAT network can also be used. Our model uses the GAT network to extract the graph-structured features between words and the R-GAT network to extract the graph-structured features of the dependencies between words.

Graph Attention Network: The dependency tree can be represented by a graph g with n nodes, where the graph g represents the composition of a sentence with a certain word as the root node. For any root node i, its neighbors N_i constitute a feature on the graph; that is, the structural relationship of the graph, where the edges of g represent the dependencies between words. For each root node i, the GAT method is used to learn the feature representation between the vertex and its adjacent vertices:

$$e_{ij} = a([Wh_i \,\|\, Wh_j]), j \in \mathcal{N}_i \tag{3}$$

$$\alpha_{ij} = \frac{exp(LeakyReLU(e_{ij}))}{\sum_{k\in\mathcal{N}_i} exp(LeakyReLU(e_{ik}))} \tag{4}$$

$$h_i^{'} = \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{ij} W h_j) \tag{5}$$

where W is a shared parameter, h_{ij} represents the initialized feature representation of node i and node j, respectively, and $[\cdot\|\cdot]$ represents the splicing of the two features. a represents the shared attention mechanism: $\mathbf{R}^F \times \mathbf{R}^F \rightarrow \mathbf{R}$ and $h_i^{'}$ represents the new feature (combined with neighbor information) of each node output by GAT.

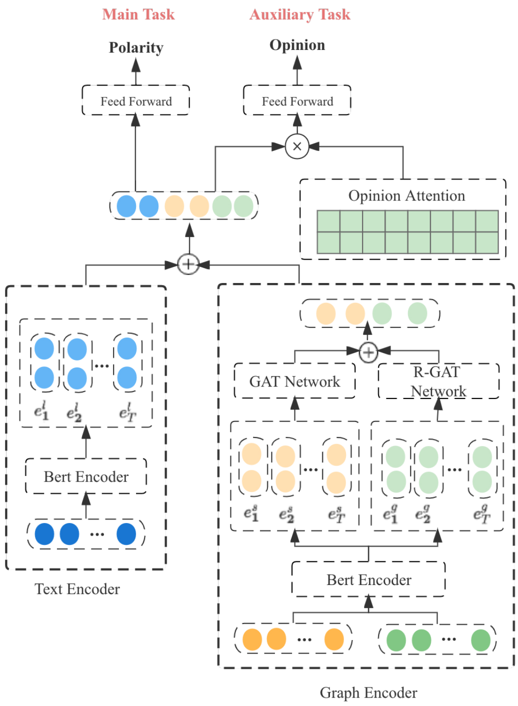


Figure 2. Framework of MDJT

Relational Graph Attention Network: In the dependency tree, GAT only learns the relationship between nodes and their neighbors and does not consider the dependencies between nodes, which may lose some important dependency information. The importance of dependencies between different nodes is not the same. For example, given a text "*The food in this restaurant is delicious*", the dependency *nsubj* between node [*food*] and node [*delicious*] is more important than the information contained in the dependency *det* between node [*food*] and node [*The*]. This paper utilizes the R-GAT network to learn the weight distribution of different dependencies among nodes. First, map the dependencies into a vector representation, and then compute the relational attention as:

$$g_{ij} = a(Wr_{ij} + b) \tag{6}$$

$$\beta_{ij} = softmax(g_{ij}) = \frac{exp(g_{ij})}{\sum_{j=1}^{N_i} exp(g_{ij})} \tag{7}$$

$$h_r^{'} = \sigma(\sum_{j \in \mathcal{N}_i} \beta_{ij} W h_j) \quad \textbf{(8)}$$

Among them, W and b are trainable parameters, r_{ij} represents the word vector expression of the dependency between node i and node j, and $h_r^{'}$ represents the new feature of each vertex i that is output by using R-GAT to fuse the dependency information between nodes.

3.2 Feature Fusion Layer

To solve different tasks (the main task is the three-category task of aspect words and sentiment; the auxiliary task is the multi-classification task of opinion words), different neural network models need to be built, and each model will include multiple modules. Different modules extract different semantic features, and for different tasks, the importance of each module is also different. This paper proposes three weight dynamic allocation strategies to dynamically assign weights to different modules.

3.2.1 Weight Dynamic Allocation Strategy Based on Gating Mechanism

The strategy is mainly to map the feature between 0 and 1 through the gating mechanism, and this value is the weight of the relative importance of the feature. The weights are dynamically assigned to the semantic feature H^b and the graph-structured feature H^g through the gating mechanism: the weight of H^b is: $\alpha = sigmoid(H^b \cdot W_b)$. Then, the weight of H^g is: $\beta = 1.0 - \alpha$. So, the final output based on the gating mechanism is: $output = \alpha \cdot H^b + \beta \cdot H^g$.

3.2.2 A Weight Dynamic Assignment Strategy Based on Parameter Adaptive

This strategy is an improvement on the gating mechanism, which solves the problem that gating only considers the importance of unilateral features, and its weight is the percentage of each module to the total modules. Using the parameter adaptation mechanism to dynamically assign weights to the semantic feature H^b and the graph-structured feature H^g:

$$weight_b = sigmoid(H^b \cdot W_b) \quad \textbf{(9)}$$

$$weight_g = sigmoid(H^g \cdot W_g) \quad \textbf{(10)}$$

$$\alpha = weight_b / (weight_b + weight_g) \quad \textbf{(11)}$$

$$\beta = 1.0 - \alpha \quad \textbf{(12)}$$

Therefore, the final output based on the parameter adaptation mechanism is: $output = \alpha \cdot H^b + \beta \cdot H^g$.

3.2.3 Dynamic Weight Allocation Strategy Based on Kl Divergence

This strategy is to calculate the distance between modules, and the distance value is the weight of the module. Inputting each batch of data into different modules will get different features, and the features obtained by better data through different feature extractors should have a greater degree of discrimination. The degree of discrimination is measured by the distance between modules. The larger the distance, the more obvious the degree of discrimination between features; on the contrary, the degree of discrimination between features is relatively insignificant. That is, the weight dynamic allocation based on KL divergence is important to filter batch data. Using the KL divergence mechanism to dynamically assign weights to the semantic feature H^b and the graph-structured feature H^g, we calculate softmax and logSoftmax separately for H^b:

$$P_b = \frac{exp(H^b)}{\sum_0^{batch} exp(H^b)} \quad \textbf{(13)}$$

$$Q_b = log(P_b) \quad \textbf{(14)}$$

Calculate softmax and logSoftmax separately for H^g:

$$P_g = \frac{exp(H^g)}{\sum_0^{batch} exp(H^g)} \quad \textbf{(15)}$$

$$Q_g = log(P_g) \quad \textbf{(16)}$$

The final weights are as follows:

$$\alpha = \sum_{i=1}^{N} [P_b \odot Q_b - P_b \odot Q_g] \quad \textbf{(17)}$$

$$\beta = \sum_{i=1}^{N} [P_g \odot Q_g - P_g \odot Q_b] \quad \textbf{(18)}$$

$$\alpha + \beta = 1.0 \quad \textbf{(19)}$$

Where N represents the number of batches. Therefore, the final output based on the parameter adaptation mechanism is: $output = \alpha \cdot H^b + \beta \cdot H^g$.

3.3 Opinion Attention Layer

We extract the opinion words from the Restaurant and Laptop datasets for deduplication as the label of the auxiliary task, denoted as y_o. First, map the labels to the same feature space as the output of the Feature Fusion Layer, which is: $output \mapsto \mathcal{R}^f$ and $y_o \mapsto \mathcal{R}^f$. Through the word embedding layer, each label is encoded into the corresponding word vector representation c_i through Glove technology. Then, label y_o is encoded into a two-dimensional matrix as: $C = \{c_1, c_2, \ldots, c_k\}$. where K represents the number of labels.

We use the Opinion Attention Layer to extract the features that are most likely to be opinion words in the text. That is, each token in this article is weighted, and the closer to the opinion word, the greater the weight of the token. The weight can be obtained by Formula (20):

$$\beta = softmax((C^T \cdot output) \oslash M) \quad \textbf{(20)}$$

Where M is a normalized matrix of size $K \times f$, and each element in this matrix is the L2 norm product of the k-th label and the t-th word, $m_{kt} = |c_k| \cdot |output_t|$. The softmax value of the l-th element is: $\beta_l = \frac{exp(a_l)}{\sum_{l^{'}=1}^{L} exp(a_{l^{'}})}$.

The final output after Opinion Attention Layer is:

$$O = \sum_{n=1}^{N} \beta_n \cdot output_n \quad \textbf{(21)}$$

3.4 Output Layer

In this paper, MDJT is used to improve the accuracy of sentiment classification. The Polarity Task is used as the main task to classify the sentiment of aspect words; the Opinion Task is used as an auxiliary task to extract the features that are most likely to be opinion words in a piece of text. By optimizing the loss value of this task, the main task can achieve a better classification effect.

3.4.1 Main Task—Polarity Task

After Section 3.1.1 Text Encoder and Section 3.1.2 Graph Encoder, the semantic feature H^b and graph structural feature H^g of the text can be extracted, and the output is obtained through the Section 3.2 Feature Fusion Layer. After linear activation by tanh, the probability of mapping to different emotional polarities through the softmax layer is as follows:

$$P(p) = softmax(W_{p2} \cdot tanh(W_{p1} \cdot output + b_{p1}) + b_{p2}) \quad \textbf{(22)}$$

Use cross-entropy to calculate the loss value for the main task:

$$\mathcal{L}_p(\theta_1) = -\sum_{(S,A)\in\mathcal{D}} \sum_{pin\mathcal{A}} logP(p) \quad \textbf{(23)}$$

Among them, $\mathcal{D}$ means that it contains all sentence-aspects pairs, and $\mathcal{A}$ means that all the aspect words appearing in the sentence S are set. θ_1 represents the parameters for which the model can be trained.

3.4.2 Auxiliary Task—Opinion Task

Through the Opinion Attention Layer in Section 3.3, the output is O, and after the tanh linear activation, the softmax layer is also used to extract the features that are most likely to become opinion words in each data; that is, multi-classify each text, as shown below:

$$P(o) = softmax(W_{o2} \cdot tanh(W_{o1} \cdot O + b_{o1}) + b_{o2}) \quad \textbf{(24)}$$

Use cross-entropy to calculate loss values for auxiliary tasks:

$$\mathcal{L}_o(\theta_2) = -\sum_{(S,O)\in\mathcal{D}} \sum_{oin\mathcal{B}} logP(o) \quad \textbf{(25)}$$

Among them, $\mathcal{D}$ indicates that all sentence-aspects pairs are included, and $\mathcal{B}$ indicates that all opinion words appearing in sentence S are set. θ_2 represents the model trainable parameters.

3.4.3 Joint Learning

Joint training is the connection between the main task and the auxiliary task. The main task is the three-classification task of aspect words and sentiment, and the auxiliary task is the multi-classification task of opinion words. By optimizing the loss value of the auxiliary task, the model can learn a better feature representation, thereby improving the accuracy of the main task sentiment classification.

$$\mathcal{L} = \mathcal{L}_p(\theta_1) + \mathcal{L}_o(\theta_2) \quad \textbf{(26)}$$

4 Experiment

4.1 Experimental Settings

4.1.1 Datasets Introduction

To verify that the proposed model has good effectiveness, this paper uses two base datasets, Restaurant and Laptop, for model training. It is derived from the SemEval2014 tasks, which are review texts for laptops and restaurants. The dataset for each domain is divided into the training set and test set. Table 1 summarizes the statistics of the dataset.

Table 1. Details of Restaurant and Laptop datasets

Dataset	Positive		Neutral		Negative	
	Train	Test	Train	Test	Train	Test
Restaurant	994	341	870	128	464	169
Laptop	2164	728	807	196	637	196

The experiments in this paper are mainly based on the above two datasets. Since these two datasets are widely used, we modify the datasets accordingly to suit the experiments on this task. The main modifications include: (1) Add the opinion field and convert the two-tuple {aspect, sentiment} to the triple {aspect, sentiment, opinion}. (2) The amount of data is smaller than the size of the original data set.

4.1.2 Implementation Details

The experimental parameters involved in this model are shown in Table 2. Among them are sets according to experience: GloVe_emb, gcn_emb, hidden_size, batch_size, dep_relation_emb. Tuning according to the experimental results are max_hop, num_heads, num_gcn_layers, and learning_rate.

Table 2. Model parameter settings

Parameter name	Parameter value	Parameter name explanation
GloVe_emb	300	Dimensions of GloVe word vectors
gcn_emb	300	The dimension of the GCN encoded-word vector
hidden_size	300	The dimension of the hidden layer vector
batch_size	32	Batch size
dep_relation_emb	300	Dimensions of Dependency Embedding
max_hop	4	Maximum number of neighbor hops
num_heads	7	Gat headcount
num_gcn_layers	1	The number of layers of gcn
gcn_dropout	0.2	Drop rate for GCN
learning_rare	1e-3	Learning rate

4.2 Experimental Results and Analysis

The model in this paper is mainly for experimental analysis of fine-grained sentiment classification. The model is mainly designed based on a semantic feature extractor and graph-structured feature extractor. Therefore, some mainstream models are selected for comparison on Restaurant and Laptop datasets, including Text Encoder and Graph Encoder.

Text Encoder

LSTM: A long-short-term memory network is used to extract contextual semantic features in texts and further classify them.

RAM [11]: Adopts a multi-attention mechanism to capture distant emotional features and is more robust to irrelevant information.

BERT [18]: Use a pre-trained model to solve downstream tasks.

Graph Encoder

ASGCN [25]: To resolve syntactically irrelevant contextual words incorrectly judged as aspect sentiment cues, Zhang et al. proposed to use graph convolutional networks to build sentence dependency trees.

CDT [27]: Sun et al. proposed a method based on dependency tree convolution to reduce the distance between aspect words and opinion words so that dependencies can be efficiently stored in long sentences.

GAT [29]: To learn the importance of different neighbor nodes in the graph structure, an attention mechanism is introduced.

R-GAT [13]: Wang et al. proposed an aspect-oriented dependency tree structure to rebuild the original dependency tree by constructing a tree with aspect words as root nodes.

Table 3. Model experiment comparison results

Category	Model	Restaurant		Laptop	
		Accuracy	Macro-F1	Accuracy	Macro-F1
Text encoder	LSTM	79.10	69.00	71.22	65.75
	RAM	80.23	70.80	74.49	71.35
	BERT	85.62	78.28	77.58	71.38
Graph encoder	ASGCN-DT	81.39	72.19	75.33	70.95
	ASGCN-DG	81.78	73.47	74.24	70.70
	CDT	81.83	72.54	71.30	65.89
	GAT	78.21	67.17	73.04	68.11
	R-GAT	89.52	80.22	77.68	63.61
Ours (Polarity task)	model1	89.52	78.10	79.77	63.29
Ours (Joint task)	model2	90.02	80.91	82.24	71.84

This article deals with the Restaurant and Laptop datasets, adding the opinion field and reducing the amount of data. Running the above models on the new dataset found that the results were slightly lower than the results of the original paper on the original dataset. For the sake of fairness, the better results in the original paper were used as a comparison with the performance of the model in this paper. The following two conclusions can be drawn from Table 3: (1) Our model is better than any other baseline model on either the Laptop dataset or the Restaurant dataset. On the evaluation metrics, it can be seen that our model performs better on the Laptop dataset, but on the Restaurant dataset, there is only a relatively small improvement in the ACC and F1 evaluation metrics. This phenomenon is mainly due to the difference in the number of aspect words and opinion words between datasets. Secondly, the domain differences of the datasets will also affect the model effect. (2) Comparing the performance of the single-task (Polarity Task) and multi-task (Joint Task) models, it is found that the effect of the multi-task model is better than that of the single-task model, which further verifies that the introduction of the opinion word field in this paper can be better. Connect the relationship between aspect words and sentiment words so that the model can achieve better results.

4.2.1 Comparison with Text Encoder

In this subsection, we compare our model with a semantic feature extractor model. LSTM further makes emotional judgments based on the contextual semantic features of the text; on this basis, RAM uses the fusion of LSTM and attention mechanism to capture distant emotional features, thereby improving the accuracy of emotional classification. LSTM and RAM both are based on LSTM as the underlying feature extractor, while BERT is based on transform as the underlying feature extractor. The feature extraction capability of transform is due to LSTM, so the effect of BERT is due to LSTM and RAM. Our model is based on BERT as the underlying feature extractor; it extracts the graph-structured features of the text by constructing a graph-structured feature extractor and performs feature fusion through a weight dynamic allocation strategy. It is seen in Table 3 that its effects are better than Text Encoder.

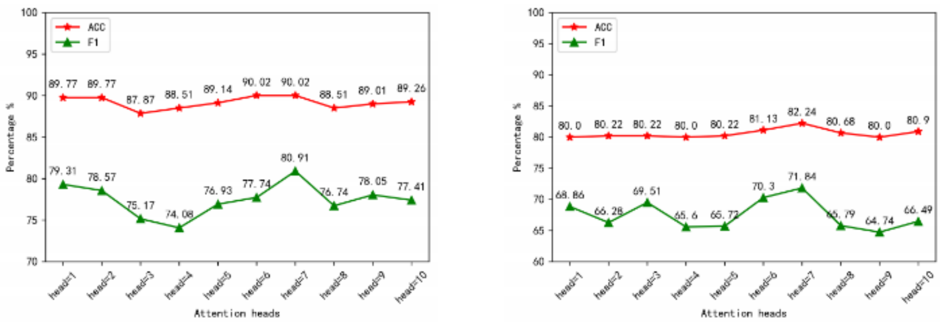
4.2.2 Comparison with Graph Encoder

In this subsection, we compare our model with a graph-structured feature extractor model. As shown in Table 3, it can be concluded that through the comparative analysis of ASGCN, CDT, GAT, and RGAT, the graph structure features constructed by the syntactic dependencies between aspect words and other words can indeed improve the effectiveness of the model. However, comparing the above model with the model in this section, it is found that not only are the graph-structured features of the text considered, but also the semantic features of the text can be further improved in the classification effect of the model.

4.2.3 Discussion

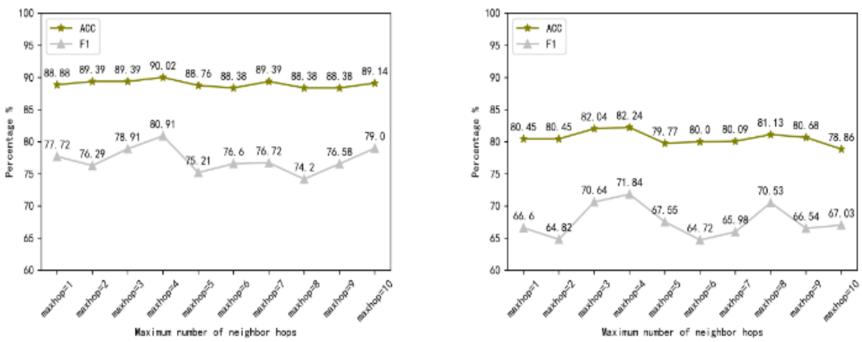
The effect of attention headcount: In this paper, the graph-structured features of the text are extracted by Graph Encoder. Our model uses GAT and R-GAT to extract graph-structured features between tokens in text and graph-structured features to extract dependencies between tokens in text, respectively. This paper uses the multi-head attention mechanism to extract features from different angles. This part discusses the effect of different numbers of attention heads on the experimental results. Define a set H={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} of the number of attention heads, select the value of the number of attention heads of the model in this paper from this set, and observe its performance on Restaurant and Laptop datasets. As shown in Figure 3, when head=7, the ACC and F1 evaluation metrics achieve the best results on the Restaurant and Laptop datasets, respectively.

Impact of Maximum Neighbor Hop Count: Similarly, when using GAT and R-GAT to compose text, the maximum number of neighbor hops may also affect the effectiveness of the model. This section discusses the effect of the maximum number of neighbor hops on the experimental results. Similarly, define a set H={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} with the largest number of neighbor hops, and observe the ACC and F1 evaluations of the experiments on the Restaurant and Laptop datasets. The results are shown in Figure 4. When maxhop = 4, the ACC and F1 values achieve the best results on the Laptop and Restaurant datasets, respectively.



(*a*) Restaurant (*b*) Laptop

Figure 3. The effect of the number of attention heads on the experimental effect

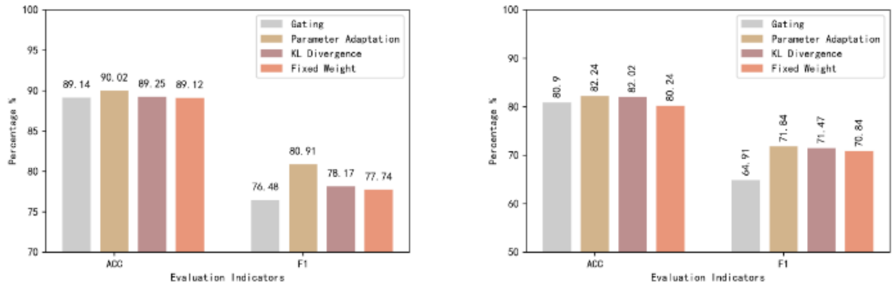


(*a*) Restaurant (*b*) Laptop

Figure 4. The influence of the maximum number of hops on the experimental effect

Influence of Feature Fusion Strategies: Section 3.2 Feature Fusion Layer proposes three weight dynamic allocation strategies: (1) weight dynamic allocation based on gating mechanism (2) weight dynamic allocation based on parameter adaptation (3) weight dynamic allocation based on KL divergence. This part mainly discusses the experimental effect comparison between the weight dynamic allocation strategy and the fixed weight.

From the Figure 5, it can be found that the effect of parameter adaptation and KL divergence for weight dynamic allocation is better than that of fixed weight, and the effect of the gating mechanism is worse than that of fixed weight.



(*a*) Restaurant (*b*) Laptop

Figure 5. Comparison of weight dynamic allocation strategy and fixed weight effect

5 Conclusion

This paper proposes the Multi-Task Dynamic Joint Training (MDJT) framework to address the Aspect-Based Sentiment Analysis (ABSA) task. Firstly, given the grammatical complexity of the text and the interrelationships between aspect words and other words, we employ a combination of BERT Encoder and Graph Encoder to extract both semantic features and graph-structured features of the text. Secondly, to fully leverage the features of opinion words, our model adopts a multi-task learning approach based on the triplet {aspect, sentiment, opinion}. The primary task focuses on classifying each aspect word, while the auxiliary task aims to extract the most significant words in the text that are likely to represent opinion words. By optimizing the loss value of this auxiliary task, we can further enhance the accuracy of the primary sentiment classification task. Finally, as abstract feature extractors, the BERT Encoder and Graph Encoder open avenues for future work, where different models may be employed for feature extraction. Potential alternatives include utilizing RoBERTa or ALBERT for text feature extraction and employing Graph Isomorphism Networks for extracting graph-structured features. At the same time, our method still has some shortcomings, such as the consumption of computing resources which can be further optimized, and its performance in other fields needs to be verified.

References

[1] G. Yi, Z. Chen, Z. Chen, W. E. Wong, N. Chau, Exploring the capability of ChatGPT in test generation, *2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion*, Chiang Mai, Thailand, 2023, pp. 72-80. https://doi.org/10.1109/QRS-C60940.2023.00013

[2] K. K. Yusuf, E. Ogbuju, T. Abiodun, F. Oladipo, A technical review of the state-of-the-art methods in aspect-based sentiment analysis, *Journal of Computing Theories and Applications*, Vol. 1, No. 3, pp. 287-298, February, 2024. https://doi.org/10.62411/jcta.9999

[3] X. Chen, H. Xie, S. J. Qin, Y. Chai, X. Tao, F. L. Wang, Cognitive-inspired deep learning models for aspect-based sentiment analysis: A retrospective overview and bibliometric analysis, *Cognitive Computation*, Vol. 16, No. 6, pp. 3518-3556, November, 2024. https://doi.org/10.1007/s12559-024-10331-y

[4] W. Jin, B. Zhao, Y. Zhang, J. Huang, H. Yu, WordTransABSA: Enhancing aspect-based sentiment analysis with masked language modeling for affective token prediction, *Expert Systems with Applications*, Vol. 238, Article No. 122289, March, 2024. https://doi.org/10.1016/j.eswa.2023.122289

[5] Y. Chen, Y. Shi, J. Ren, J. Li. An Improved Adaptive Angle Weakly Supervised Learning Object Detection, *2024 IEEE 24th International Conference on Software Quality, Reliability and Security*, Cambridge, United Kingdom, 2024, pp. 494-503, 2024. https://doi.org/10.1109/QRS62785.2024.00056

[6] H. Abdelhakim, T. Zied, A Hybrid Ensemble Learning Approach for Detecting Bots on Twitter, *International Journal of Performability Engineering*, Vol. 20, No. 10, pp. 610-620, October, 2024. https://doi.org/10.23940/ijpe.24.10.p3.610620

[7] N. Mishra, F. Khan, A. Mishra, Revolutionizing Text Summarization: A Breakthrough in Content Compression, *International Journal of Performability Engineering*, Vol. 20, No. 1, pp. 40-47, January, 2024. https://doi.org/10.23940/ijpe.24.01.p6.4047

[8] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Baltimore, MD, USA, 2014, pp. 49-54. https://doi.org/10.3115/v1/P14-2009

[9] D. Tang, B. Qin, T. Liu, Document modeling with gated

recurrent neural network for sentiment classification, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 1422-1432.
https://doi.org/10.18653/v1/D15-1167

[10] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based LSTM for aspect-level sentiment classification, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, USA, 2016, pp. 606-615.
https://doi.org/10.18653/v1/D16-1058

[11] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017, pp. 452-461.
https://doi.org/10.18653/v1/D17-1047

[12] J. Cheng, S. Zhao, J. Zhang, I. King, X. Zhang, H. Wang, Aspect-level sentiment classification with HEAT (hierarchical attention) network, *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore, 2017, pp. 97-106.

[13] B. Wang, W. Lu, Learning latent opinions for aspect-level sentiment classification, *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 5537-5544.

[14] Z. Ouyang, J. Su, Dependency parsing and attention network for aspect-level sentiment classification, *Natural Language Processing and Chinese Computing: 7th CCF International Conference*, Hohhot, China, 2018, pp. 391-403.
https://doi.org/10.1007/978-3-319-99495-6_33

[15] L. Li, Y. Liu, A. Q. Zhou, Hierarchical attention based position-aware network for aspect-level sentiment analysis, *Proceedings of the 22nd conference on computational natural language learning*, Brussels, Belgium, 2018, pp. 181-189.
https://doi.org/10.18653/v1/K18-1018

[16] T. H. Nguyen, K. Shirai, PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp. 2509-2514.
https://doi.org/10.18653/v1/D15-1298

[17] W. Wang, S. J. Pan, D. Dahlmeier, X. Xiao, *Recursive neural conditional random fields for aspect-based sentiment analysis*, arXiv: 1603.06679, March, 2016.
https://arxiv.org/abs/1603.06679

[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, 2019, pp. 4171-4186.
https://doi.org/10.18653/v1/N19-1423

[19] X. Li, L. Bing, W. Zhang, W. Lam, Exploiting BERT for end-to-end aspect-based sentiment analysis, *Proceedings of the 5th Workshop on Noisy User-generated Text*, Hong Kong, China, 2019, pp. 34-41.
https://doi.org/10.18653/v1/D19-5505

[20] C. Sun, L. Huang, X. Qiu, Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA, 2019, pp. 380-385.
https://doi.org/10.18653/v1/N19-1035

[21] S. Chen, Y. Wang, J. Liu, Y. Wang, Bidirectional machine reading comprehension for aspect sentiment triplet extraction, *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Virtual Event, 2021, pp. 12666-12674.

[22] S. Gu, L. Zhang, Y. Hou, Y. Song, A position-aware bidirectional attention network for aspect-level sentiment analysis, *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, NM, USA, 2018, pp. 774-784.
https://aclanthology.org/C18-1066/

[23] S. Zheng, R. Xia, *Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention*, arXiv: 1802.00892, February, 2018.
https://arxiv.org/abs/1802.00892

[24] N. Majumder, S. Poria, A. Gelbukh, M. S. Akhtar, E. Cambria, A. Ekbal, IARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 34023411.
https://doi.org/10.18653/v1/D18-1377

[25] C. Zhang, Q. Li, D. Song, Aspect-based sentiment classification with aspect-specific graph convolutional networks, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China, 2019, pp. 45674577.
https://doi.org/10.18653/v1/D19-1464

[26] B. Huang, K. Carley, Syntax-aware aspect level sentiment classification with graph attention networks, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China, 2019, pp. 5469-5477.
https://doi.org/10.18653/v1/D19-1549

[27] K. Sun, R. Zhang, S. Mensah, Y. Mao, X. Liu, Aspect-level sentiment analysis via convolution over dependency tree, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China, 2019, pp. 56795688.
https://doi.org/10.18653/v1/D19-1569

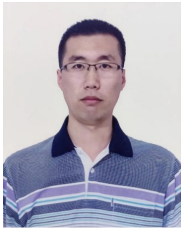
[28] K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational graph attention network for aspect-based sentiment analysis, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 3229-3238.
https://doi.org/10.18653/v1/2020.acl-main.295

[29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *Graph attention networks*, arXiv: 1710.10903, October, 2017.
https://arxiv.org/abs/1710.10903

Biographies

Pingchuan Ma is currently pursuing a Ph.D. degree in transportation, Southwest Jiaotong University, Chengdu, China. At the same time, he worked at the Hebi Institute of Engineering and Technology, Henan Polytechnic University. His current research interests include few-shot learning, natural language processing and imbalanced data.

Bo Zhao obtained a bachelor's degree from Changchun Institute of Technology in 2008. Currently, he is employed by Henan Chuitian Technology Co., Ltd. and holds the position of R&D Director. His research focuses on areas such as software engineering, data analysis, and the Internet of Things.



Xianxun Liu received his Bachelor's degree from Shengda Economics, Trade & Management College of Zhengzhou University in 2007. He currently serves as an Electronic Information Engineer at Henan Chuitian Technology Co., Ltd., China. His research focuses include cloud computing and the Internet of Things.

Peng Sun earned his Ph.D. from the University of Electronic Science and Technology of China (UESTC) in 2017 and served as an assistant professor in UESTC, China. Now he worked as an education expert on a concurrent basis at Hebi Institute of Engineering and Technology, Henan Polytechnic University. His research focus encompasses areas such as reliability modeling, cloud computing, and the Internet of Things.