Aspect-Based Sentiment Analysis via BERT and Multi-scale CBAM

Qingsong Wang and Nianyin Yang(✉)

School of Information, Liaoning University, Shenyang, China
2054289205@qq.com

Abstract. Aspect-based sentiment analysis (ABSA), as a fine-grained sentiment analysis task, predicts polarities for given aspects in one text. However, in the process of aspect-based sentiment analysis, there may be multiple perspectives of sentiment-prone text information, as well as implicit sentiment expressions, which can cause greatly influence on the accuracy of sentiment analysis results. To address the above problems, in this paper, we propose an aspect-based sentiment analysis model BERT-MSCBAM based on BERT and Multi-Scale Convolutional Block Attention Module (MSCBAM). Our model firstly uses BERT to encode the input text, then uses the MSCBAM module, which extracts the deep semantic and key features by its structure of two Multi-Scale Channel Attention Modules (CAMs) with different scales and one Spatial Attention Module (SAM) between them, combining with ResNet, and finally obtains the prediction results through a fully connected layer. To validate the effectiveness of our BERT-MSCBAM model, we conduct experiments on the Restaurant and Laptop datasets of SemEval2014 Task 4, and the Twitter dataset for our model in comparison with several current mainstream models. The experimental results validate the effectiveness of our model.

Keywords: Aspect-based sentiment analysis · BERT · CBAM

1 Introduction

Traditional coarse-grained sentiment analysis of product reviews identifies the sentiment expressed in the whole review. However, in some cases, one review message may cover multiple perspectives, and the sentiment from different perspectives may be inconsistent. For example, a review on one product may express the view that the quality of the product is good, but the environment of the store is average while its location is not good, on which the coarse-grained sentiment analysis cannot cover the complete sentiment tendency and draw an inaccurate conclusion. To address this situation, Aspect-based sentiment analysis (ABSA) [1] can extract the different sentiment tendencies of corresponding aspects of one review text to reach a relatively comprehensive and accurate sentiment analysis conclusion. Aspect level sentiment analysis can identify the sentiment tendency of each given aspect word in one text, and thus draw a more effective sentiment analysis conclusion to avoid information loss.

X. Yang et al. (Eds.): ADMA 2023, LNAI 14179, pp. 211–225, 2023.
https://doi.org/10.1007/978-3-031-46674-8_15

However, when performing ABSA for one certain aspect, there maybe multiple other perspectives of sentiment-prone textual information, as well as implicit sentiment expressions, which can lead to information missing during training and cause great interference in the accuracy of sentiment analysis results.

To address the above problems, in this paper, we propose a deep learning model based on BERT and the Multi-Scale Convolutional Block Attention Module (MSCBAM). The main contributions of our work are as follows:

- A new aspect-based sentiment analysis model BERT-MSCBAM is proposed. The model firstly encodes the text with BERT to obtain a matrix with contextual semantic information; then processes the matrix with the MSCBAM module we propose in this paper, which extracts the deep semantic and key features by its structure of two Multi-Scale Channel Attention Modules (CAMs) with different scales and one Spatial Attention Module (SAM) between them, combining with ResNet; finally uses a fully connected layer to obtain the prediction results.
- The performance of BERT-MSCBAM is evaluated using the Restaurant and Laptop datasets from SemEval 2014 Task 4, and the Twitter dataset, which are widely used in the field of ABSA tasks. The experimental results show that our BERT-MSCBAM model achieves better results on all three datasets.

2 Related Work

Early solutions for ABSA tasks mainly used feature engineering-based approaches, which performed sentiment analysis through supervised learning. Machine learning models such as SVM [2] and decision trees [3] achieved good classification results in ABSA tasks, which are simple in structure and effective in operation. However, they rely too much on complicated pre-processing and feature engineering.

In recent years, as deep learning continues to make breakthroughs, various studies that using deep learning models for ABSA have also achieved good results. Dong et al. [4] conveyed the sentiment of words to the corresponding aspect words by combining Recurrent Neural Network (RNN) with syntactic analysis. Tang et al. [5] spliced the aspect words with sentence on their left as well as right, respectively, and input the results into Long Short-Term Memory Network (LSTM) to capture the connection between aspect words and the context, and then spliced the two together to obtain the integrated textual information. Convolutional Neural Networks (CNNs) can also be useful in aspect-based sentiment analysis tasks. Xue et al. [6] proposed a model Gated Convolutional network with Aspect Embedding (GCAE) based on CNN and gating mechanisms, which enabled information filtering and improved computational efficiency through gated Tanh-ReLU units. Zhang et al. [7] used a multilayer CNN to continuously reinforce the contextual feature information associated with aspect words.

With the wide application of attention mechanism, many researchers have also applied attention mechanism to ABSA tasks. Wang et al. [8] proposed ATAE-LSTM (Attention-based LSTM with Aspect Embedding) model that combined attention mechanism with LSTM, which gave different weights to different words when determining the sentiment tendency of different aspect words in a sentence through the attention mechanism, thus obtaining superior results than the traditional LSTM. Zhao et al. [9]

proposed a classification model based on a bidirectional attention mechanism and Graph Convolutional Network (GCN), which improved the model performance by capturing the sentiment dependencies among multiple aspects of words in a sentence.

Pre-trained language models such as Bert [10] are also effective for aspect-based sentiment analysis tasks. Xu et al. [11] proposed an ABSA method using BERT pre-trained models, which was jointly trained to achieve reading comprehension and aspect-based sentiment analysis. Dai et al. [12] demonstrated by experiments that trees induced by RoBERTa performed better than syntactic dependency trees, thus proving the effectiveness of RoBERTa for ABSA tasks.

3 Method

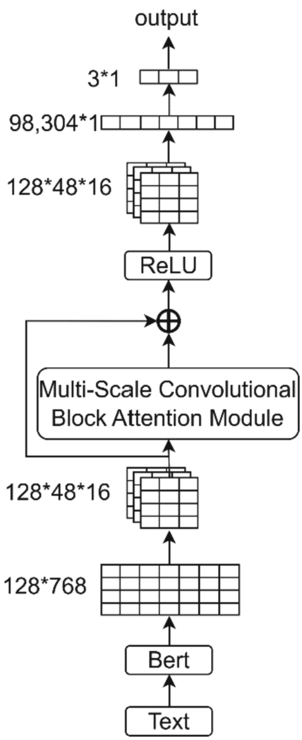

Fig. 1. The structure of BERT-MSCBAM

BERT-MSCBAM model consists of a text encoding layer, an attention focus layer and a fully connected layer, as shown in Fig. 1. The text encoding layer is responsible for converting the short text into a vector matrix by a BERT encoder; the attention focus layer extracts the deep semantic and key features from the vector matrix through the MSCAMs and the SAM of the MSCBAM, combining with ResNet; then the fully connected layer obtains the prediction results from the previous key features.

3.1 Text Encoding Layer

The dataset of ABSA task is $D = \{(s_i, a_i, y_i)\}_{i=1}^{N}$ and its tag set is $C = \{c_1, \ldots, c_k\}$. In the dataset, there is a total of N pieces of comment texts, and this ABSA task belongs to k classification problem. s_i is the i th comment text, a_i is its corresponding aspect word, and $y_i \in C$.

We first perform pre-processing by splicing each piece of comment text with each of its corresponding aspect words respectively to obtain texts with the same length d in the form of $[CLS] + s_i + [SEP] + a_i + [SEP]$, which are used as the input of the model. The specific input form is represented as $e_i = \{[CLS], w_1, w_2, \ldots, w_x, [SEP], z_1, \ldots, z_y, [SEP]\}$, e_i consists of d words.

Then each piece of text is converted into a vector matrix X^{d*h} by a BERT encoder with h hidden layers.

3.2 Attention Focus Layer

In the attention layer, we make improvement on the Convolutional Block Attention Module (CBAM) [13] to better perform deep semantic and key feature extraction of the text, enabling the model to better capture the key information in the sentence for a given aspect word.

CBAM, as shown in Fig. 2, is a lightweight general-purpose module that can be incorporated into various CNNs for end-to-end training. The module is simply and effectively designed, consisting of a CAM module and an SAM module.

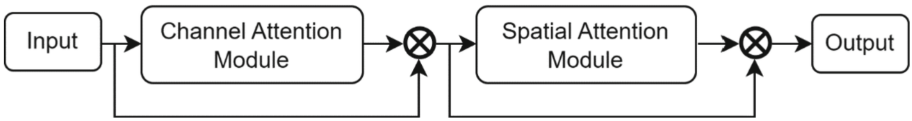

Fig. 2. The structure of CBAM

Like the original CBAM module, the attention focus layer of our model also contains the SAM, the similar CAM, and the ResNet mechanism; however, our attention focus layer improves the CBAM module performance in two aspects and we name this improved CBAM as MSCBAM. The improvement can be summarized in two aspects:

- We adjust the shared MLP layer in the original CAM module to three shared MLP layers of different sizes in parallel for better feature extraction and name the improved CAM as MSCAM;
- Compared with the original CBAM module consisting of one CAM and one SAM module to extract features serially, we add one MSCAM module after the SAM, and the scale of the Shared MLP layer of this added one is different from that of the first MSCAM.

Multi-scale Channel Attention Module. The original CAM module uses one Shared MLP layer for feature extraction, as shown in Fig. 3; while our MSCAM in this paper uses three parallel Shared MLPs instead of the original one Shared MLP layer, as shown in Fig. 4. The specific structure is as follows.

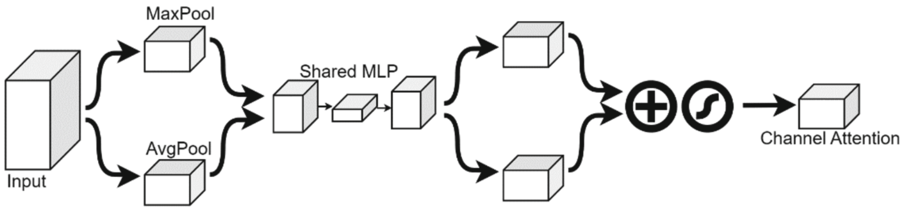

Fig. 3. The structure of CAM

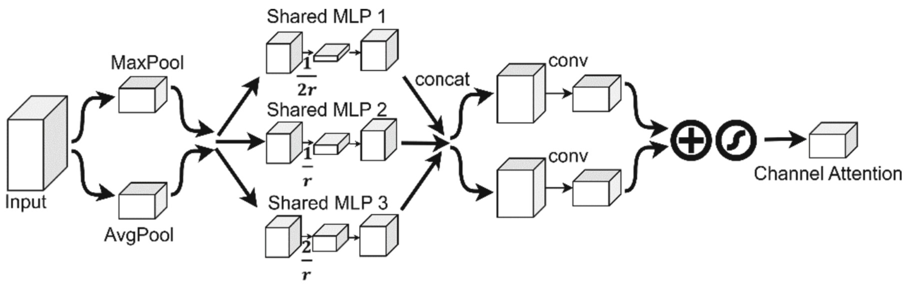

Fig. 4. The structure of MSCAM

Firstly, the input matrix $F_{c1}^{x_1*x_2*x_3}$ is passed through parallel MaxPool and AvgPool layers to get two output matrixes, in which process the size of the feature matrix is changed from $x_1 * x_2 * x_3$ to $x_1 * 1 * 1$; and then the two matrixes are passed through three Shared MLP modules of different scales in parallel, where the weights of the two are shared. In the three Shared MLP modules, the x_1 dimension is compressed into $\frac{1}{2r}$, $\frac{1}{r}$ and $\frac{2}{r}$ times from origin, respectively, and then expanded back; then the three output matrixes of MaxPool and the three of AvgPool are processed by ReLU activation function and stitched together respectively, and then the two output matrixes are processed by convolution layer. Then the two obtained output matrixes are summed element by element, passed through a dropout layer, and then passed through a Sigmoid activation function to get the final output result $F_{c2}^{x_1*1*1}$ of MSCAM.

Finally, the output $F_{c2}^{x_1*1*1}$ is multiplied by the original matrix to obtain $F_{c3}^{x_1*x_2*x_3}$, transforming back to the size of $x_1 * x_2 * x_3$.

The equation of MSCAM is as follows:

$$\begin{aligned}\mathbf{M_c}(\mathbf{F}) = \sigma\big(f^{7*7}\big[&MLP_1(AvgPool(\mathbf{F})) + MLP_2(AvgPool(\mathbf{F}))+\\ &MLP_3(AvgPool(\mathbf{F}))\big] + f^{7*7}[MLP_1(MaxPool(\mathbf{F}))+\\ &MLP_2(MaxPool(\mathbf{F})) + MLP_3(MaxPool(\mathbf{F}))])\\ = \sigma\Big(f^{7*7}\Big[&MLP_{Group_r}\Big(\mathbf{F^c_{avg}}\Big) + MLP_{Group_r}\big(\mathbf{F^c_{max}}\big)\Big]\Big)\end{aligned} \tag{1}$$

In the MSCAM, x_1, which is the word vector dimension of the text, is compressed and then recovered to the original size, while the dimensions x_2 and x_3, which are corresponding to the hidden layer size $x = x_2 * x_3$, is compressed to $1 * 1$. This module focuses on the meaningful information in each word vector.

Spatial Attention Module. The structure of the SAM is shown in Fig. 5, and its specific structure is as follows.

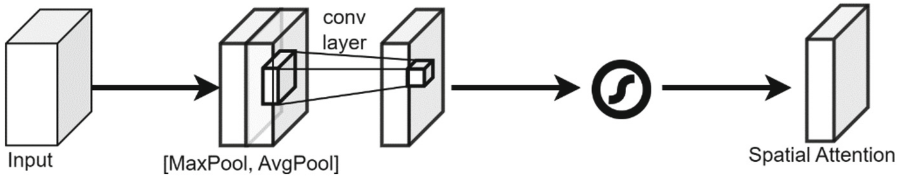

Fig. 5. The structure of SAM

Firstly, the input matrix $F_{s1}^{x_1 * x_2 * x_3}$ is pooled by MaxPool and AvgPool to obtain the feature matrixes with dimension $1 * x_2 * x_3$ respectively, then the two feature matrixs are stitched together into a matrix with dimension $2 * x_2 * x_3$ and convolved into a feature matrix with size 1 in the first dimension by a convolution kernel of size $m * m$. Then the output matrix is processed by a Sigmoid activation function to obtain the final feature matrix $F_{s2}^{1 * x_2 * x_3}$.

Finally, the output $F_{s2}^{1 * x_2 * x_3}$ is multiplied by the original matrix $F_{s1}^{x_1 * x_2 * x_3}$ to obtain $F_{s3}^{x_1 * x_2 * x_3}$, transforming back to the size of $x_1 * x_2 * x_3$.

The equation of SAM is as follows:

$$\begin{aligned} \mathbf{M_s}(\mathbf{F}) &= \sigma\left(f^{7*7}\left(\left[AvgPool(\mathbf{F}); MaxPool(\mathbf{F})\right]\right)\right) \\ &= \sigma\left(f^{7*7}\left[\mathbf{F}_{\mathbf{avg}}^{\mathbf{s}}; \mathbf{F}_{\mathbf{max}}^{\mathbf{s}}\right]\right) \end{aligned} \tag{2}$$

In the SAM, the dimensions related to hidden layer keep their sizes while the dimension of the word vectors are compressed. This module focuses on the different attention weights corresponding to the different words in each piece of text.

Multi-scale Convolutional Block Attention Module. The structure of our proposed MSCBAM is shown in Fig. 6, which contains two MSCAMs of different scales and one SAM in the order of r_0 scale MSCAM, SAM, and $2 * r_0$ scale MSCAM.

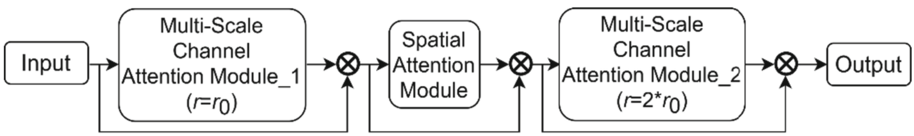

Fig. 6. The structure of MSCBAM

In the attention focus layer of our model, the text encoding matrix X^{d*h} is first transformed into $A_1^{d*h_1*h_2}(h = h_1 * h_2)$, and then input into the MSCBAM. Then the output of the module $A_2^{d*h_1*h_2}$ is added with the previous module input using ResNet mechanism and $A_3^{d*h_1*h_2}$ is obtained. Finally, the output of the attention focus layer $A_4^{d*h_1*h_2}$ is obtained by a ReLU activation function layer.

3.3 Output Layer

The key information obtained from the attention focus layer $A_4^{d*h_1*h_2}$ is transformed into $M^a(a = d * h_1 * h_2)$ and input into a fully connected layer to obtain the final output vector of dimension p, where p represents the number of different sentiment polarities of the ABSA task. The specific equation is as follows:

$$\hat{y} = argmax\left(M^a * W^{a*p} + b^p\right) \tag{3}$$

where W^{a*p} is a trainable parameter, b^p is the bias term; and $\hat{y}$ is the predicted sentiment polarity of the model.

ABSA is a classification task, so we choose Cross-Entropy as the loss function of our model, which is calculated as:

$$loss = -\sum_{i=1}^{c} y_i log\widehat{y_i} \tag{4}$$

where y_i is the actual classification label.

4 Experiment

4.1 Experimental Datasets

To validate the effectiveness of our model, we experiment on the Laptop and Restaurant dataset from the publicly available dataset SemEval-2014 task4 [14] and the Twitter dataset [4], which contain comments and corresponding aspect words data, including Positive, Negative and Neural emotional polarities. Table 1 gives the statistics of the datasets used in our experiments (Tables 2 and 3).

Table 1. Statistical table of datasets

Dataset	Positive	Negative	Neutral
Laptop-train	994	870	464
Laptop-test	341	128	168
Restaurant-train	2164	807	637
Restaurant-test	728	196	196
Twitter-train	1561	1560	3127
Twitter-test	173	173	346

4.2 Experimental Environment

Table 2. Experimental environment

Experimental environment	Configuration
Operating System	Linux-5.15.65 + -x86_64-with-debian-bullseye-sid
Memory	13 GB
Programming Language	Python 3.7.12
GPU	Tesla P100-PCIE-16 GB
Deep Learning Framework	PyTorch 1.11.0
CUDA	11.0

Table 3. Experimental hyperparameters

Parameters	Value
BERT model	bert-base-uncased
Optimizer	adam
Word vector dimension d	128
BERT hidden layer dimension h	768
Number of Transformer layers	12
MLP descending coefficient r_0	16
SAM convolution kernel size $m * m$	3 * 3
Number of emotional polarity categories p	3
Dropout factor	0.5
Learning Rate	2e–5
Number of iterations	30
Batch Size	16

4.3 Experimental Hyperparameters

The strategy for training and optimizing our model is to use Adam optimizer to train this model. And in the optimization process, we use Cross-Entropy as the loss function.

4.4 Experimental Evaluation Indicators

We choose Acc and F1 as evaluation indicators for the experiments, and the equations for both are as follows:

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{7}$$

$$F1 = \frac{2PR}{P + R} \tag{8}$$

4.5 Comparison Models

To verify the validity of this model, we compare our model with the following models in experiments on the Laptop14, Restaurant14 and Twitter datasets, using Acc and F1 values as evaluation metrics.

- feature-based SVM: The model uses traditional feature engineering for contextual feature extraction, and then uses support vector machine as a classifier for sentiment polarity classification.
- TD-LSTM: The model splices aspect words with their left and right sentences respectively, and then inputs the spliced texts into the LSTM network for encoding in order to capture the connection between the aspect words and their contexts, and then splices the two encoded texts to get the integrated text information.
- ATAE-LSTM: The model combines LSTM model with attention mechanism to capture the correlation between aspect words and each context word, and then makes integration to get the classification results.
- GCAE: The model is based on CNN and gating mechanism, in which gated Tanh-ReLU units are used to selectively extract sentiment information from the context based on aspect words.
- ASGCN: In the model, the syntactic dependency tree knowledge is used to construct the adjacency graph, and then the syntactic feature information of the context is extracted by the GCN network.
- AOA [15]: The model jointly extracts information on interaction features between context and aspect words through an attention-over-attention neural network structure.
- AEN-BERT [16]: The model is based on BERT, the semantic information between context and aspectual words is encoded by the attention mechanism.
- BERT-SPC [17]: The model uses BERT to conduct ABSA tasks.
- BERT4GCN [18]: The model integrates the syntactic order features of BERT PLM and the syntactic knowledge of dependency graphs.

4.6 Main Results

The main experimental results are shown in Table 4.

The models of this comparison experiment contain traditional machine learning models, LSTM models, CNN models, GCN models, attention models, and BERT models. In the above three datasets, the metrics of our model BERT-MSCBAM have all improved to a certain extent compared with the baseline models. Specifically, the accuracy of our model reaches 87.41%, 81.19%, and 75.14%, respectively, which increases by 2.95%, 1.26%, and 0.43% compared with the highest values in the baseline model, respectively;

Table 4. Model comparison results

Models	Restaurant14 dataset		Laptop14 dataset		Twitter dataset	
	Acc	F1	Acc	F1	Acc	F1
feature-based SVM	0.8106	–	0.7049	–	0.6340	0.6330
TD-LSTM	0.7563	0.6795	0.6813	0.6521	0.7080	0.6900
ATAE-LSTM	0.7720	0.7080	0.6870	0.6393	0.6864	0.6660
GCAE	0.7935	0.7052	0.7278	0.6710	0.7080	0.6766
ASGCN	0.8077	0.7202	0.7555	0.7105	0.7215	0.7040
AOA	0.7997	0.7042	0.7262	0.6752	0.7230	0.7020
AEN-BERT	0.8312	0.7376	0.7993	0.7631	0.7471	0.7313
BERT-SPC	0.8446	0.7698	0.7899	0.7503	0.7355	0.7214
BERT4GCN	0.8475	0.7711	0.7749	0.7301	0.7473	0.7376
BERT-MSCBAM	0.8741	0.8115	0.8119	0.7684	0.7514	0.7427

and the F1 values reaches 81.15%, 76.84%, and 74.27%, which increases by 4.17%, 0.53%, and 1.15% compared with the highest values in the baseline models, respectively.

Firstly, the results of our experiments prove the rationality and effectiveness of our BERT-MSCBAM model in this paper. Secondly, the results demonstrates that the SAM and CAM modules in CBAM are not only effective in the image processing field, but also can achieve good results in the natural language processing field. In our comparison experiment, the ASGCN model and the BERT-MSCBAM model both applies the combination of CNN model and attention mechanism. According to the experimental results, however, it can be revealed that the performance of our model BERT-MSCBAM significantly surpasses that of ASGCN, which shows that the combination of attention mechanism and CNN in our MSCBAM can more accurately extract the deep semantic information.

4.7 Model Analysis

Influence of Parameters. The MLP descending coefficient r_0 and the value of SAM convolution kernel size m are two important parameters of BERT-MSCBAM, so that their value selection can greatly influence the performance of the model. To explore the most suitable values of r_0 and m for BERT-MSCBAM in this paper, we conduct experiments on the Restaurant and Laptop datasets, and finally obtain the results shown in Fig. 7 and Fig. 8.

It can be found from the experimental data that the best results are achieved when the value of r_0 is 16 and when the value of m is 3. Therefore, in our model, the value of r_0 is taken as 16 and the value of m is taken as 3.

Influence of Scale Sizes of MSCAM. To explore the effect of the scale sizes of MSCAM, we conduct experiments on the Restaurant and Laptop datasets. We select a total of 5 sets of combinations of different scales for the experiments, and the specific downscaling data are shown in Table 5.

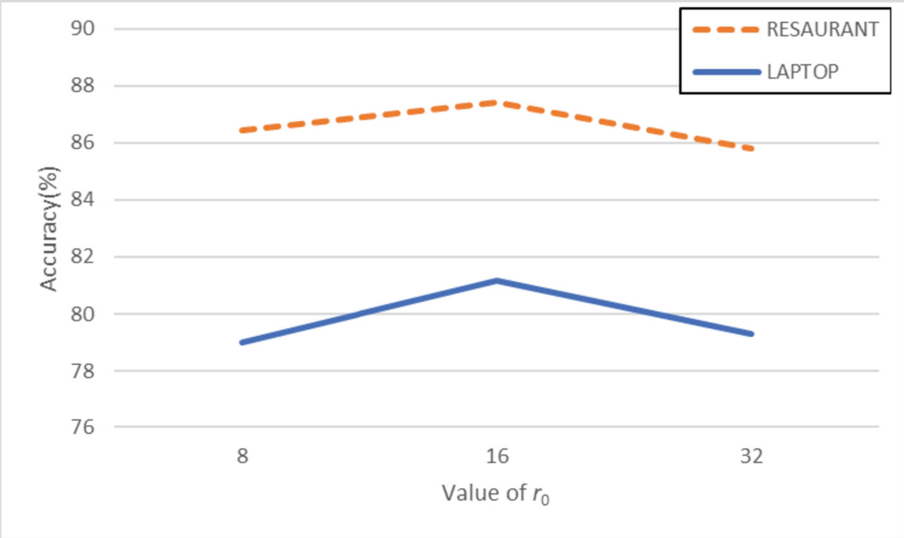

Fig. 7. Comparison experiment results with different values of r_0

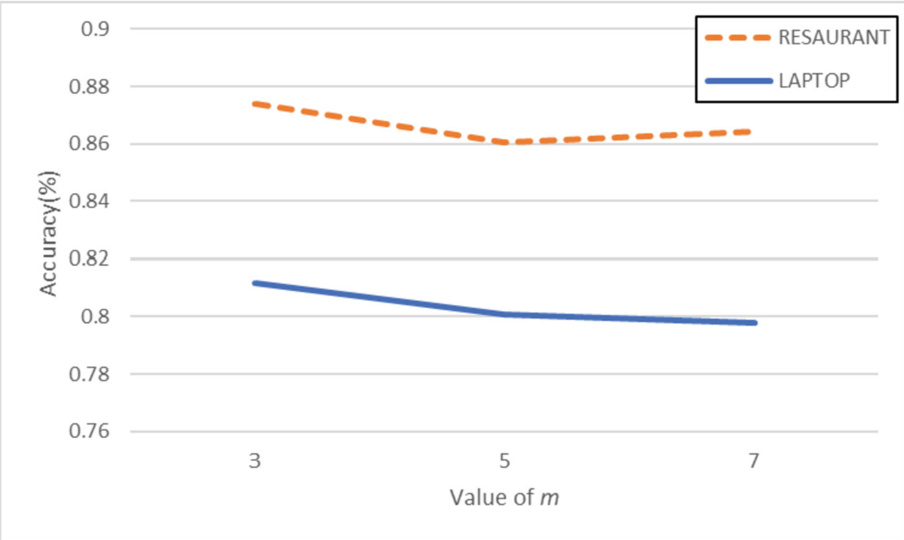

Fig. 8. Comparison experimental results with different values of m

The results of the experiments are shown in Fig. 9.

The experimental data shows that our model performs best when the number of scales is 3. Too few scales can result in insufficient information extraction of texts, while too many scales can lead to overfitting and prolong the training time. Therefore, we select 3 as the number of scales for our model MSCAM.

Table 5. Scale data for MSCAM

Number of scales	MSCAM_1 MLP downscaling	MSCAM_2 MLP downscaling
1	$\left(\frac{1}{r}\right)$	$\left(\frac{2}{r}\right)$
2	$\left(\frac{1}{r}, \frac{2}{r}\right)$	$\left(\frac{2}{r}, \frac{4}{r}\right)$
3	$\left(\frac{1}{2r}, \frac{1}{r}, \frac{2}{r}\right)$	$\left(\frac{1}{r}, \frac{2}{r}, \frac{4}{r}\right)$
4	$\left(\frac{1}{2r}, \frac{1}{r}, \frac{2}{r}, \frac{4}{r}\right)$	$\left(\frac{1}{r}, \frac{2}{r}, \frac{4}{r}, \frac{8}{r}\right)$
5	$\left(\frac{1}{4r}, \frac{1}{2r}, \frac{1}{r}, \frac{2}{r}, \frac{4}{r}\right)$	$\left(\frac{1}{2r}, \frac{1}{r}, \frac{2}{r}, \frac{4}{r}, \frac{8}{r}\right)$

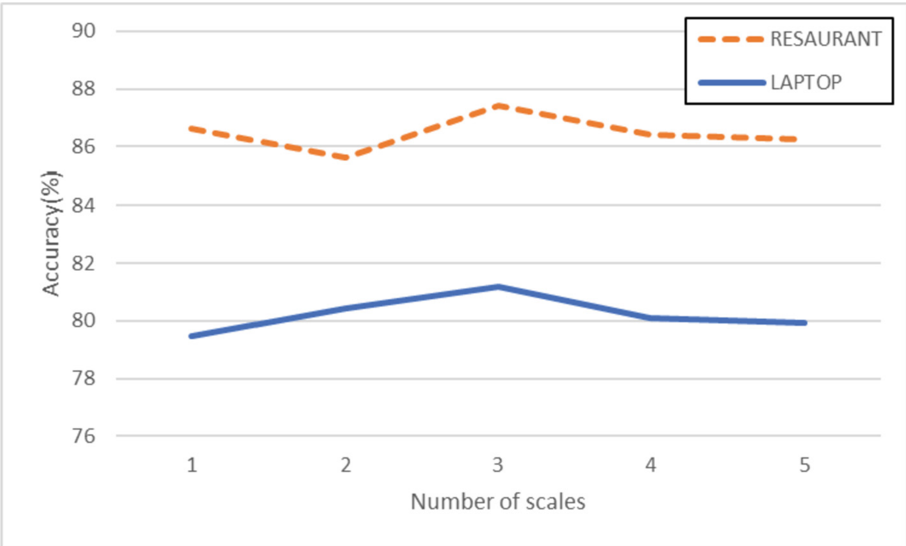

Fig. 9. Comparison experiment results of MSCAM with different scales

Ablation Experiment. In order to verify the effect of SAM and MSCAM modules in our MSCBAM of the model, we make comparison on the model BERT-SPC without SAM and MSCAM modules, the model BERT-CBAM with the original CBAM module, and the BERT-MSCBAM model with SAM and MSCAM modules of different orders. The experimental results are as shown in Table 6.

From Table 6, there are several conclusions can be drawn. Firstly, by comparing the results of BERT-CBAM model with the BERT-SPC model, it can be concluded that the introduction of CBAM module can make a small improvement in the performance of BERT model in ABSA tasks. Then, the small improvement obtained from the introduction of MSCAM module proves that our MSCAM can improve the performance of CBAM module in the ABSA tasks. And it can be demonstrated that the BERT-MSCBAM model in this paper achieves the best results compared with the BERT baseline, the baseline of the combination model of BERT and CBAM modules and all other combinations of BERT, MSCAM and SAM modules with different orders, which can indicate that our MSCBAM module can greatly improve the performance of the ABSA tasks, and that BERT-MSCBAM model can extract the deep semantic meaning of the text corresponding to the specified aspects more effectively and obtain more accurate results.

Table 6. Results of ablation experiments of BERT-MSCBAM model

Models	Restaurant14 dataset		Laptop14 dataset		Twitter dataset	
	Acc	F1	Acc	F1	Acc	F1
BERT-SPC	0.8446	0.7698	0.7899	0.7503	0.7355	0.7214
BERT-CBAM	0.8509	0.7912	0.7968	0.7461	0.7370	0.7163
BERT-MSCAM-SAM	0.8652	0.8072	0.7978	0.7531	0.7312	0.7186
BERT-MSCAM-MSCAM	0.8598	0.7947	0.7947	0.7589	0.7370	0.7268
BERT-SAM-MSCAM	0.8696	0.8118	0.7994	0.7630	0.7428	0.7367
BERT-MSCBAM	0.8741	0.8115	0.8119	0.7684	0.7514	0.7427

5 Conclusions

In this paper, we prove the effectiveness of the combination model of BERT and CBAM in ABSA tasks, propose an improvement on the CAM module in CBAM, which is named as MSCAM, and finally propose an ABSA model BERT-MSCBAM, which makes effective improvement to original CBAM module in ABSA tasks and combines the improved module MSCBAM with BERT. To verify the effectiveness of our model, we conduct extensive experiments in this paper using Restaurant14, Laptop14 and Twitter datasets, and the accuracy of the BERT-MSCBAM model on these three datasets reaches 87.41%, 81.19%, and 75.14%, respectively, with F1 values of 81.15%, 76.84%, and 74.27%, respectively, demonstrating that our model can effectively address specific ABSA tasks by comparing it with the mainstream methods of ABSA tasks in experimental results.

Although our model has achieved good results, the datasets we use in the current experiments are limited to the English language. Therefore, we will investigate the effectiveness of our model on ABSA tasks in other languages in the future works. And because the lengths of the sentences in the ABSA tasks vary greatly, key information may be lost in the pre-processing process, which may cause some impact on the ABSA results. In the future, we will further study on how to make improvement on text pre-processing methods. And in the process of experiments, we find that despite the use of dropout and other methods to prevent the overfitting of the model, a certain amount of overfitting still occurs in the training process. Therefore, we will also study on how to better avoid model overfitting in the future in order to further improve the performance of BERT-MSCBAM.

References

1. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Mining Text Data, pp. 415–463. Springer, Heidelberg (2012). https://doi.org/10.1007/978-1-4614-3223-4_13
2. Luo, F., Li, C., Cao, Z.: Affective-feature-based sentiment analysis using SVM classifier. In: 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design, pp. 276–281. IEEE, New York (2016)
3. Chauhan, C., Sehgal, S.: Sentiment classification for mobile reviews using KNIME. In: 2018 International Conference on Computing. Power and Communication Technologies, pp. 548–553. IEEE, New York (2018)
4. 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), pp. 49–54. Association for Computational Linguistics, Baltimore (2014)
5. Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 3298–3307. The COLING 2016 Organizing Committee, Osaka (2016)
6. Xue, W., Li, T.: Aspect based sentiment analysis with gated convolutional networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (volume 1: Long Papers), pp. 2514–2523. Association for Computational Linguistics, Melbourne (2018)
7. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. In: Proceedings of the 2019 Conference on Empircal Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 4568–4578. Association for Computational Linguistics, Hongkong (2019)
8. 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, pp. 606–615. Association for Computational Linguistics, Stroudsburg (2016)
9. Zhao, P., Hou, L., Wu, O.: Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. Knowl.-Based Syst. **193**, 105443 (2020)
10. Vaswani, A., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6000–6010. Curran Associates Inc., New York (2017)
11. Xu, H., Liu, B., Shu, L., Yu, P.: 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, pp. 2324–2335. Association for Computational Linguistics, Minneapolis (2019)
12. Dai, J., Yan, H., Sun, T., Liu, P., Qiu, X.: Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1816–1829. Association for Computational Linguistics, Online (2021)
13. Woo, S., Park, J., Lee, J., Kweon, I.: Cbam: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision, pp. 3–19. Springer, Munich (2018)
14. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos I., Manandhar, S.: SemEval-2014 task 4: aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 27–35. Association for Computational Linguistics, Dublin (2014)

15. Huang, B., Ou, Y., Carley, K.M.: Aspect level sentiment classification with attention-over-attention neural networks. In: Thomson, R., Dancy, C., Hyder, A., Bisgin, H. (eds.) SBP-BRiMS 2018. LNCS, vol. 10899, pp. 197–206. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93372-6_22
16. Song, Y., Wang, J., Tao, J., Liu, Z., Rao, Y.: Attentional encoder network for targeted sentiment classification. In: 28th International Conference on Artificial Neural Networks, pp. 93–103. Springer, Cham (2019)
17. 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: Human Language Technologies, pp. 4171–4186. Association for Computational Linguistics, Minneapolis (2019)
18. Xiao, Z., Wu, J., Chen, Q., Deng, C.: BERT4GCN: using BERT intermediate layers to augment GCN for aspect-based sentiment classification. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 9193–9200. Association for Computational Linguistics, Punta Cana (2021)