Journal of Computers Vol. 32 No. 3, 2021, pp. 30-42
doi:10.3966/199115992021063203003

A BERT-based Interactive Attention Network for Aspect Sentiment Analysis

Yu-Ting Yang, Lin Feng*, Lei-Chao Dai

Department of Computer Science, Sichuan Normal University, Chengdu, 601101, P.R. China
1539782809@qq.com, fenglin@sicnu.edu.cn, 1012320443@qq.com

Received 3 July 2020; Revised 23 November 2020; Accepted 6 January 2021

Abstract. Aspect-level sentiment classification is a key topic in the field of natural language processing (NLP). The existing models for sentiment classification have some drawbacks, such as the weakness of word-aspect associative perception, and with not strong generalization ability. In this paper, we develop a BERT-based interactive attention network (BIAN) to help improve aspect-level short-text sentiment classification. First, we use BERT model as encoder to extract different type of context features. Next, we create interactive attention networks to learn interactive attentions between context and aspect words. The final attention representations are constructed and the classification results are output. Experiments on the multiple data sets demonstrate that BIAN can achieve the state-of-the-art performance.

Keywords: short-text, aspect-level sentiment classification, BERT model, interactive attention network

1 Introduction

The rapid development of the Internet has led to a large number of comments on the network, such as consumers' evaluation of movies on ticketing platforms, and online products in e-commerce websites. These comments are the feedback of users' satisfaction with business services, shopping experiences, etc. It is crucial to make good use of consumers' feedback to improve businesses' products and services. So, a growing study of sentiment classification for online evaluation is presented.

For online evaluation, when users express their opinions on an entity, they usually evaluate different aspects of the entity with different sentiment polarity. The aspect sentiment analysis model can more specifically mine the sentiment expression of different aspects in the user's opinion, and then predict the independent sentiment polarity of different aspects in the same sentence [1]. For example, Given the mentioned targets: "drivers" and "BIOS update", and their context sentence "drivers updated ok but the BIOS update froze the system up and the computer shut down". The emotional polarity of "drivers" and "BIOS update" are positive and negative respectively. Aspect sentiment analysis not only depends on context information, but also considers the sentiment polarity of different aspect words in context. Therefore, when there are many sentiment targets in the evaluation text, the aspect sentiment analysis can predict the sentiment more accurately.

The early aspect sentiment classification methods are mainly based on traditional machine learning, and carry out text classification in a supervised way [2]. In recent years, deep learning methods have attracted the scholars' wide attention. Because it is no need to extract features manually and can encode sentences in low dimensional word vectors with rich semantic information [3-4]. Among them, attention mechanism has been applied in a large number of deep learning models [5]. And the performance of most neural network-based models have been improved by attention mechanism. The attention mechanism uses the semantic correlation between context and aspect words to calculate the attention weight of context words, and combines with the deep learning network to obtain fine-grained sentiment polarity. In aspect sentiment analysis, recurrent neural network (RNN) or long short-term memory (LSTM) network

* Corresponding Author

is often used to extract context features by combining attention mechanism [6-9]. Tang [10] proposed to integrate aspect information into LSTM to encode sentences for sentiment analysis. Experiments showed that the model without aspect information had poor classification effect. Wang [11] embedded aspect words into LSTM. They designed the attention mechanism to learn sentence weights for responding to special aspect words and achieved better results. Chen [12] used the hidden layer of bidirectional LSTM to construct the location weighted memory, and then used the attention mechanism to capture the distant separated sentiment features from the location weighted memory. They improved the accuracy of classification effectively.

With the further development of deep learning in natural language processing (NLP), the pre-training model has become a hot topic in the research of aspect sentiment analysis tasks. The pre-training model can be applied to a large number of NLP tasks, and improve the performance of each them. Traditional language models have some drawbacks to capture many facets of relevant semantic for downstream tasks. For example, the word2vec [13] is easy to find other words with similar semantics, but unable to distinguish different semantics for polysemes. After transformer proposed by Vaswani [14], Peters [15] proposed ELMO, which used pre-training representation as additional function to resolve the problem of polysemes, and improved the performance of many NLP tasks. However, compared with transformer [14], the feature extraction ability of ELMO is weaker. GPT [16] is similar to ELMO, but its ability of feature extraction is better than Elmo. In addition, GPT uses a one-way language model, which can’t integrate words for the following. After that, BERT was proposed to overcome the shortcomings of Elmo and GPT, applied to the aspect sentiment analysis tasks. The classification effect was significantly improved [17].

The above research results prove that attention mechanism can recognize the long-term dependence between aspect words and context. But it is insufficient to mine deeper and more specific context interaction information, and the classification accuracy needs to be improved. The methods of using RNN, LSTM, CNN models as text modeling tools have better ability to extract text features, but they can’t run in parallel. Moreover, most methods use backward pre-trained models as text encoders such as Word2Vec [13] etc. Those methods cannot obtain the correct semantics of polysemes well. In addition, each training algorithm of RNN is basically truncated backpropagation. This algorithm will affect the model’s ability to capture dependencies on a longer time scale [18].

In this paper, we propose a BERT-based interactive attention network (BIAN). The BIAN is based on the bidirectional encoder representation from transformers (BERT) model [17], which uses transformer [14] as encoder to extract the semantic information of context, aspect and aspect-aware. So It overcomes the problem of polysemes mentioned above. Then, we compute semantic information value of context and aspect words to supervise the generation of attention vectors. The attention mechanism is adopted to capture the important information in the context and aspect words. Finally, BIAN predict the sentiment polarity of aspect words in context through the joint learning of semantic information in aspect-aware and attention output. Experiments on the three datasets show that BIAN can effectively improve the performance of short text sentiment classification.

Our contribution is two-fold:

(a) Different from the previous model focusing on word level sequence, we use BERT model as the encoder to replace RNN and other networks, so that the language model is transformed from one-way to two-way. The interactive information of context and aspect words is extracted, thus avoiding the problem of polysemes.

(b) We design an interactive attention mechanism between context and aspect words to obtain the dependency relationship between them, and combine it with BERT model to improve the accuracy of aspect level sentiment classification.

2 Related Work

Aspect-level sentiment classification is typically regarded as a kind of text classification problem. Majority of existing studies build sentiment classifiers with supervised machine learning approach, such as feature-based Naive Bayes [20] and Supported Vector Machine (SVM) [2, 20]. Some researchers used topic model in text classification and achieved better results [21]. However, these approaches rely on n-gram features or artificially designed, such as the establishment of sentiment lexicon [22]. Rao [23] used a rich sentiment lexicon to design sentiment classification models. Mullen [24] combined SVM and bagof-words model with sentiment lexicon for classification. There are two drawbacks in these methods, one is not every word in the sentiment lexicon, the other is deviation of constructing sentiment lexicon based on the experts' experience. Yi [25] used the positive and negative sentiment value of words to generate sentiment feature vectors, and then applied to twitter dataset after training through machine learning methods. Although these methods work well, the effect of them depends on good feature selection.

With developing of deep learning for NLP, RNN and LSTM are mostly used to extract text features for aspect-level sentiment classification [6, 12]. RNN and LSTM are mostly used to extract text features for aspect-level sentiment classification. Tang [4] proposed the integration of targets and sentence information via LSTM to encode context for sentiment analysis. Experiments showed that the models of sentiment classification without integration of target and context information did not work well. Tai [27] led the Tree-LSTM model into LSTM, and established a tree-like LSTM network topology, which performed well in sentiment classification. Wang [11] embed aspect words into LSTM, and used the attention mechanism to learn sentence weights for responding special targets. This experiment proved that the model with attention mechanism can better recognize the relationship between context and aspect words. But the classification model using RNN and LSTM as encoders cannot be parallelized. At the same time, when faced with different contexts, it is impossible to recognize different semantics of the same word. Besides, it is impossible to recognize different semantics of the same word, when faced with different contexts.

We propose a BERT-based interactive attention network (BIAN). BIAN uses BERT as encoder to overcome the shortcomings of RNN and LSTM models. In addition, we introduce the attention mechanism to calculate the attention vector of the aspect word and context separately, and jointly learn the aspect word closely related to context. Experimental results show that BIAN has better classification ability than other methods.

The rest of the paper is organized as follows. We develop the BIAN in Section 3. We discuss experimental results in Section 4. We summarize the study in Section 5.

3 BIAN Model

Given a context sentence, an aspect sentence, a pair of sentences and a segment id, whereis the length of the context sentence, is the length of the aspect sentence, is a sub-sequence of, is a combination of and. and are fed to single sentence classification model in BERT to capture the hidden states of sentences and aspect words. and are fed to sentence pair classification model in BERT to capture aspect-dependent features. The interactive attention representation is obtained by attention mechanism from the hidden states. Finally, the attention mechanism and BERT model are jointly trained for sentiment classification. Fig. 1 shows the overall architecture of BIAN model, which mainly consists of an input layer, an embedding layer, an interactive attention layer, and an output layer.

3.1 Input Layer

In order to facilitate the training and fine-tuning of BERT model, we transform the given context and aspect word to "[CLS] context [SEP]" "[CLS] aspect word [SEP]" "[CLS] context [SEP] aspect word [SEP]" and "1, 1, ..., 1, 0, 0, ..., 0" respectively. For example, given a sentence "The battery life is excellent.", where the aspect word is "battery life". It is transformed to "[CLS] The battery life is excellent. [SEP]" "[CLS] battery life [SEP]" "[CLS] The battery life is excellent. [SEP] battery life [SEP]" and "1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0".

3.2 Embedding Layer

This layer is composed of a stack of identical layers. Each layer has two sub-layers, that is a multi-head self-attention mechanism (i.e., Self-Attention) and a simple, position wise fully connected feed-forward network (i.e., Feed Forward). There is a residual connection (i.e., Add&Normalize) around each of the two sub-layers, followed by layer normalization. Fig. 2 shows the structure of the encoder.

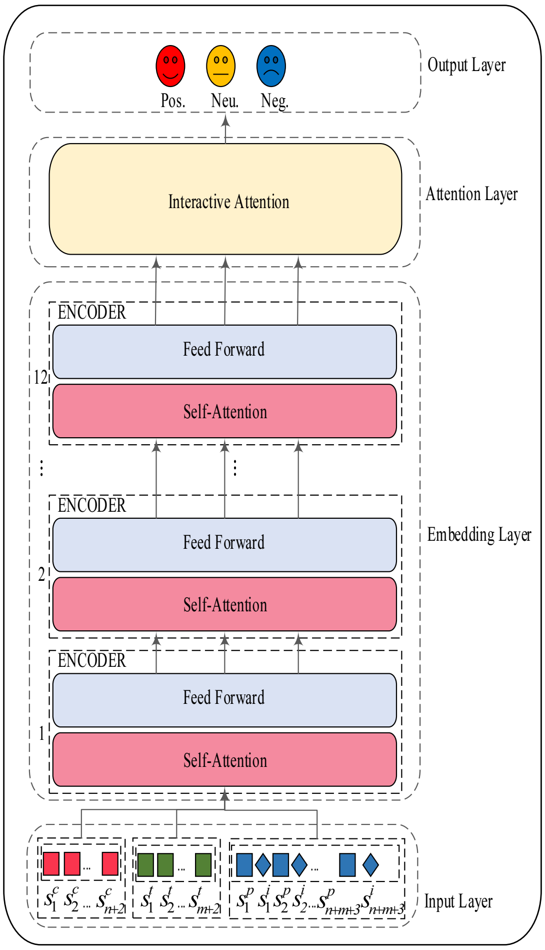


Fig. 1. The overall architecture of BIAN

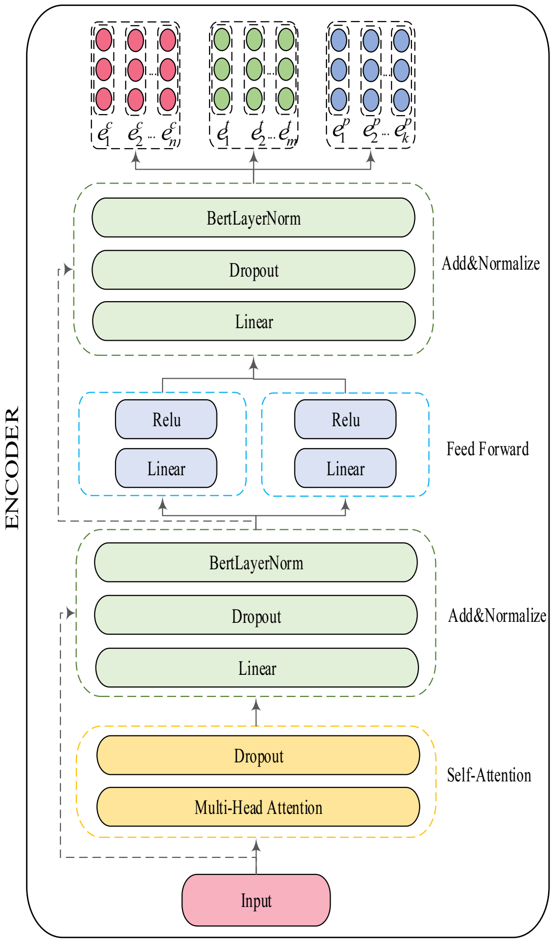


Fig. 2. Structure of encoder

First, we encode the input data initially. we preprocess the input sentences, detects whether the words from the metatext are in the pre-training dictionary[1], replace the symbols and words in the original context with the corresponding ids in the dictionary, and do padding to the length of sentence. After that, express each word as the sum of the three embeddings, i.e., word embeddings, position embeddings and token-type embeddings. After add the three vectors, normalization and dropout are performed. Then, feed the vectors into encoders based on BERT.

Self-Attention contains Multi-Head attention and dropout. Multi-Head Attention performs on word vectors by:

$$MultiHead\,(Q,K,V)=Concat\,(head_1,...,head_h)\,W^O\,, \tag{1}$$

$$head_i=Attention(QW_i^Q,KW_i^K,VW_i^V)\,, \tag{2}$$

$$Attention\,(Q,K,V)=softmax(\frac{QK^T}{\sqrt{d_k}})V\,. \tag{3}$$

Where the projections are parameter matrices Q is the given query matrix, K is the key matrix, and V is the value matrix, W^Q, W^K and W^V are the parameter matrices of Q, K and V, respectively, d_k is dimension of K.

Add & Normalize contains linear, dropout and BertLayerNorm. The output of Self-Attention is $BertLayerNorm\,(x+SelfAttention(x))$, where $SelfAttention\,(x)$ is the function implemented by the Self-Attention itself.

[1] https://github.com/google-research/bert

Feed Forward concludes linear and Relu activation function. The output of Feed Forward is $BertLayerNorm\,(x + FeedForward\,(x))$.

From here, the functions in encoder are finished. The hidden states of context and aspect words from embedding layer are expressed as $\boldsymbol{e^c} = \{e_1^c, e_2^c, ..., e_n^c\}$ and $\boldsymbol{e^t} = \{e_1^t, e_2^t, ..., e_{\mathrm{m}}^t\}$. The aspect-dependent features from embedding layer are expressed as $\boldsymbol{e^p} = \{e_1^p, e_2^p, ..., e_{\mathrm{k}}^p\}$.

3.3 Interactive Attention Layer

After embedding, we employ an interactive attention mechanism to learn more semantics between context and aspect words. Fig. 3 shows the structure of the attention layer.

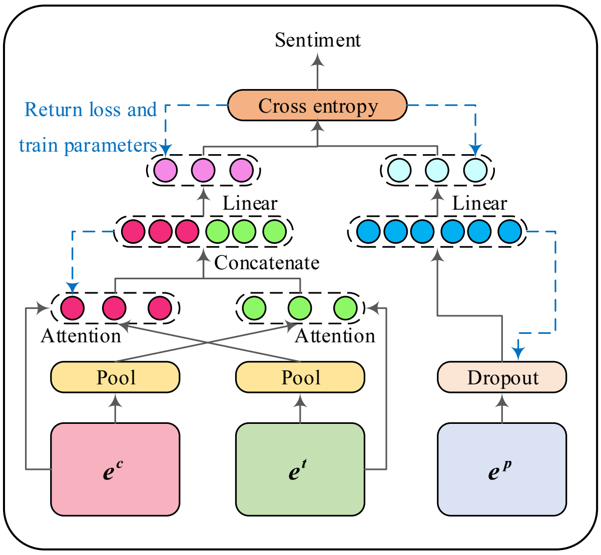


Fig. 3. Structure of attention layer

First, we got the initial representation for context and aspect words by averaging $\boldsymbol{e^c}$ and $\boldsymbol{e^t}$ respectively, that is:

$$\boldsymbol{v^c} = e_i^c / n\,, \tag{4}$$

$$\boldsymbol{v^t} = e_i^t / m\,. \tag{5}$$

Then, we adopt attention mechanism to learn attentions in context and aspect words. With the initial representation (i.e., $\boldsymbol{v^c}$ and $\boldsymbol{v^t}$) and the hidden states (i.e., $\boldsymbol{e^c}$ and $\boldsymbol{e^t}$) as input, we got two interactive attention vectors $\boldsymbol{\alpha_i}$ and β_i by:

$$\alpha_i = \frac{\exp(f_a(e_i^t, \boldsymbol{v^c}))}{\sum_{j=1}^{n} \exp(f_a(e_j^t, \boldsymbol{v^c}))}\,, \tag{6}$$

$$\beta_i = \frac{\exp(f_a(e_i^c, \boldsymbol{v^t}))}{\sum_{j=1}^{m} \exp(f_a(e_j^c, \boldsymbol{v^t}))}\,. \tag{7}$$

Where f_a is a score function which learns the semantic relevance between $\boldsymbol{e^c}$ and $\boldsymbol{v^t}$, $\boldsymbol{e^t}$ and $\boldsymbol{v^c}$. The score function f_a is defined as:

$$f_a(k_i, q_j) = Tanh([k_i, q_j] \cdot W_a)\,. \tag{8}$$

Where Tanh is activation function Tanh , “;” denotes vector concatenation, W_a are learnable weights.

After learning the attention weights, we got the attention representation $\boldsymbol{a_r}$ and $\boldsymbol{b_r}$ for context and aspect word based on the interactive attention vectors (i.e., $\boldsymbol{\alpha_i}$ and $\boldsymbol{\beta_i}$) by:

$$a_r = \sum_{i=1}^{n} \alpha_i e_i^c \ , \tag{9}$$

$$b_r = \sum_{j=1}^{m} \beta_i e_j^t \ . \tag{10}$$

Finally, we concatenate the attention representation a_r and b_r as the final comprehensive representation :

$$h_a = \{a_r ; b_r\} \tag{11}$$

After that, we fuse the output of the interactive attention layer for training. Two full connected layers project the concatenated vectors into the space of the targeted C classes:

$$y_h = W_h{}^T h_a + b_h , \tag{12}$$

$$y_p = W_p{}^T e^p + b_p \ . \tag{13}$$

Where $W_h, W_p \in R^{C \times d_{eb}}$, y_h, $y_p \in R^{d_{bs} \times C}$.

Finally, two projection vectors (i.e., y_h and y_p) are fused by average and expressed as $y^{hp} \in R^{d_{bs} \times C}$:

$$y_{hp} = avg(y_h + y_p) \ . \tag{14}$$

Jointly train the model, return loss, and optimize parameters that can be learned by the fusion layer, interactive attention layer and embedding layer. The output of fusion layer is fed into a cross-entropy loss function, that is:

$$J = \sum_{i=1}^{G} \sum_{j=1}^{C} \hat{y}_i^j \log y_i^j \ . \tag{15}$$

Where G is size of the training set, C is the number of categories, $\hat{y}_i^j$ is the ground truth represented as a one-hot vector and $y_i^j \in R^C$ is the output of fusion layer y_{hp} .

Update the parameters using the back propagation algorithm and dropout before the model outputs to avoid overfitting.

3.4 Output Layer

Complete training and output three classifications and accuracy, namely positive, neutral, negative results.

3.5 Training Strategies

The procedures for training BIAN is summarized in Algorithm 1.

Algorithm 1. The training strategies for BIAN

```
Input: Training set $T_{train} = (x, f, y)$ , test set $T_{test} = (x', f', y')$ . Where $x$ is a sentence, $f$ is an aspect
       word, $y$ is a sentiment label.
Parameter: $d_{eb}$ , $d_h$ , $fold$ , $epoch$ , $bs$. Where $d_{eb}$ is dimension of Bert embedding, $d_h$ is hidden
           dimension, and $bs$ is batch size.
Output: sentiment classification model $\Omega$ .
1. Preprocess the data sets into the corresponding input forms, $s^c$ , $s^t$ , $s^p$ and $s^i$ .
```

```
Initialize all parameters in network layers.
For ($i$ = 0; I< $fold$ ; I++):
    For ($j$ = 0; $j$< $epoch$ ; J++)
        Feed $s^c$ , $s^t$ , $s^p$ and $s^i$ into the embedding layer to get $e^c$ , $e^t$ and $e^p$ ;
        Feed $e^c$ , $e^t$ and $e^p$ into the interactive attention layer to get $y_{hp}$ ;
        $\eta$ is a high-dimensional matrix, where each row represents the similarity of a sample instance in the training set for each category, $F = \{f_1, f_2, f_3\}$ ;
        Calculate $\eta'$ of the test set, each element in each row of which represents the similarity of sample instance in the support set for each category, $F' = \{f_1', f_2', f_3'\}$ ;
        Take the category corresponding to the maximum similarity of each row in $\eta'$ as the predictive label of the sample sentence, and record the predictive label as $Y' = (y_1, y_2, y_3)$ ;
        Calculate loss by formula (15);
        Use the gradient descent method to propagate J to adjust the parameters in model $\Omega$ .
    END For
END For
The classification model is output and the algorithm stops.
```

4 Experiments

4.1 Data Sources

We conduct experiments on SemEval 2014 and Twitter [28] to validate the effectiveness of our model. The three types of reviews are labeled with three categories, positive, neutral and negative, which are represented by 1, 0, -1. And “T” is a label used to cover the aspect words. Table 1 shows the data set instance numbers in each category.

Table 1. Data Sets

Dataset	Positive		Neural		Negative		Total
	Train	Test	Train	Test	Train	Test	
Laptop	994	341	464	169	870	128	2966
Restaurant	2164	728	637	196	807	196	4728
Twitter	1561	173	3127	346	1560	173	6940

4.2 Experimental Settings

The hyper parameters of the models are tuned for training. Xavier Initialization initializes the weights and fine-tune the BERT model. In our experiments, Bert embedding dimensionis 768, learning rateis 2e-5. The mini-batch size was fixed to 32. Regularization was done by weight decay with the parameter 0.01 and by dropout with 0.1. Evaluation metrics are accuracy over positive, negative and neutral categories.

Hardware environment: GPU NVIDIA T4; memory: 30G; processor (core): 4.

Software environment: linux; pytorch.

4.3 BIAN for Sentiment Classification

We list 6 baseline approaches for sentiment classification. The baselines are listed as follows.

SVM is a supervised machine-learning approach to detect aspect terms and aspect categories and to detect sentiment expressed towards aspect terms and aspect categories in customer reviews [2].

GNN adopts a threshold neural network to connect words to represent a given aspect and context in twitters, then integrate this information to generate the final sentence representation for sentiment classification [3].

IAN uses the LSTM network to obtain the hidden states of the targets and context in word-level, and then gets the average value of all hidden states as the final representation, which is fed into the softmax function to estimate the probability of each sentiment label [6].

BMAM uses the sequence learning ability of the recurrent neural networks to obtain a compositional representation of a statement, and then decode it with attention mechanism to extract the affective polarity information with respect to the given aspect [7].

POSP-CNNAM analyzes the importance of each word in context when inferring the sentiment polarity of an aspect and generate a vector for the context and then using the vector for sentiment analysis [9].

RAM uses multiple attention mechanisms to capture distantly separated sentiment features [12].

BIAN (our approach) combines different methods to extract features, and get interactive representation by attention mechanism for sentiment classification.

Experimental results are given in Table 2.

Table 2. Classification results. "—" means not reported

Model	Laptop (%)	Restaurants (%)	Twitter (%)
SVM	70.49	80.16	—
GNN	66.02	76.15	69.11
IAN	72.10	78.60	--
BMAM	74.45	80.44	71.38
POSP-CNNAM	72.47	81.33	—
RAM	74.49	80.23	69.36
BIAN (our approach)	**76.49**	**83.11**	**71.84**

In Table 2, BIAN performs best on the three benchmark data sets, which indicates that the model has better generalization ability and is more effectively than GNN model as a feature extraction. In addition, SVM is better than non-deep networks, such as GNN. Due to IAN uses attention mechanism on the word level, IAN outperforms SVM and GNN on the Laptop. While IAN is not strong at extracting hidden features of sentences, its results are 1.56% lower than SVM on the Restaurant. POSP-CNNAM performs best on the Restaurant among the comparison models, just like RAM on the Laptop. This is why they use the attention mechanism on the word level to get target-context relatedness for improving the sentiment classification accuracy. Considering the depth of the model, RAM also adopted the attention mechanism and increased by 4% and 0.7% compared with SVM. So, we can show that getting target-specific representation by attention mechanism and extracting features by deep network can effectively improve the accuracy of the aspect-level sentiment classification tasks. And RAM is second only to BIAN, and they use multiple layers of self-attention for sentence modeling, indicating that the attention mechanism plays a key role in text information modeling.

The results show that the attention mechanism is applied to aspect words and context for short text sentiment classification, which can effectively identify key sentiment polar words from aspect words in complex sentences to determine sentiment categories. On the other hand, it also shows that our method is superior to other classical methods.

4.4 Performance Analysis of BIAN

In this section, we designed the cross-validation experiments to further reveal the generalization ability of BIAN. Two sets of experiments are designed with 10-fold. The training set is 80% and test set is 20% on Laptop and Restaurant. The percentages of training and test sets on Laptop and Restaurant are both 80% and 20%. On the Twitter, training and test set are 90% and 10%. The training accuracy are shown in Table 3. We compare the test accuracy with the 6 benchmark methods, and the histogram of the difference is shown in Fig. 4. From Table 3, we can observe that BIAN is stable in cross validation. In training accuracy, BIAN are all over 90%, which shows that BIAN can fit the training data well. Moreover, BIAN have achieved the best results on Laptop from Fig. 4. In 5-fold, although the test accuracy of BIAN is lower than that of POSP-CNNAM on Restaurant and BMAM on Twitter, BIAN performs better than other methods cross-validation test results. The cross validation results show that the BIAN can perform well in a variety of datasets, that is, good generalization.

Table 3. Cross validation training results of BIAN model on three data set

cv	Laptop (%)	Restaurant (%)	Twitter (%)
5-fold	95.01	96.56	93.25
10-fold	92.83	94.94	91.06

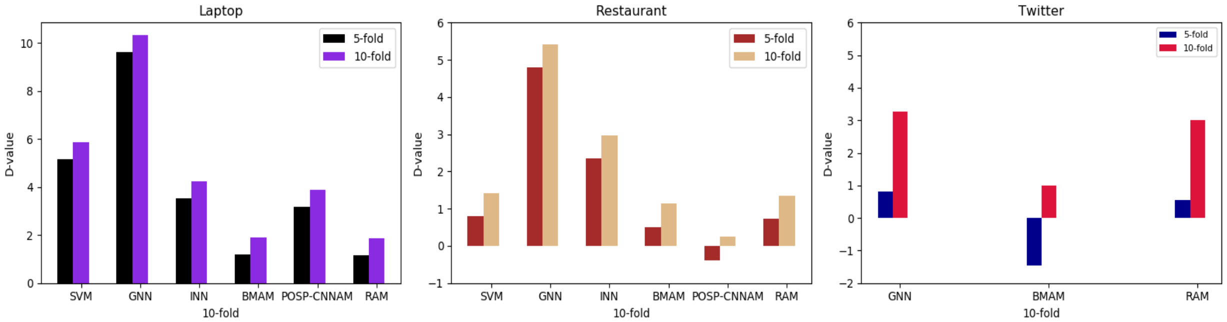
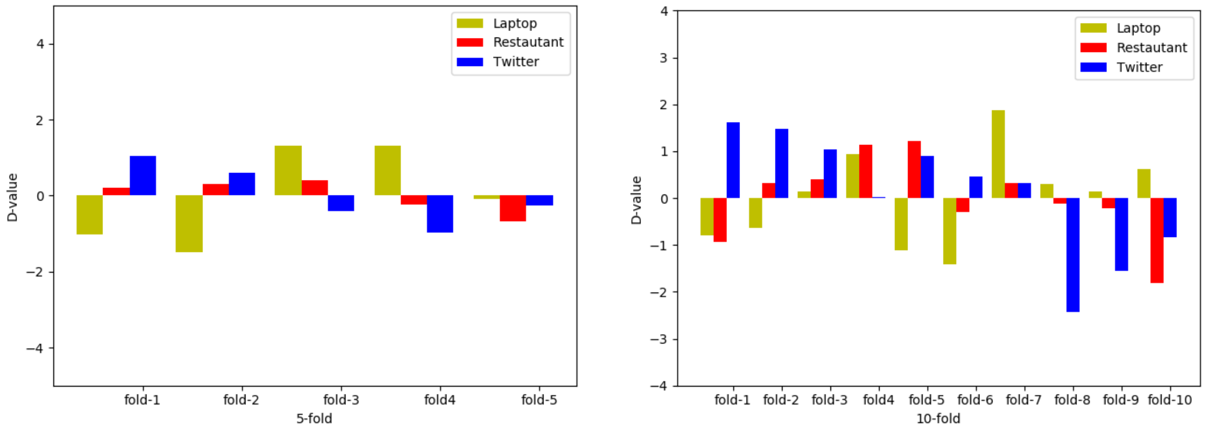


Fig. 4. Histogram of the difference between the accuracy of test and 6 baseline approaches

We compare with the best performing BMAM model in the comparison experiment. The experimental results are shown in Table 4. we also give a histogram of the mean difference of cross validation results in 5-fold and 10-fold for each fold, as shown in Fig. 5. In Table 4, we can see that, the performance of BIAN is stable, the variance of test results is small and below 3%, and most of the results are better than BMAM. In 10-fold, although the variance of BIAN on Laptop is smaller than that of BMAM, the accuracy of them is equal. From Fig. 5, we can see that the mean difference tends to be stable on the three data sets. In Fig. 5(a), the fluctuation range of columns is largest on Laptop, which shows that BIAN performs is relatively stable when the data size is large. But the columns in three colors are close to the x-axis, and the fluctuation range is between - 2% and 2%. As shown in Fig. 5(b), in 10-fold, the D-value on the three data sets are also in a small range, between -3% and 2%, with no significant deviation. In general, BIAN has achieved good results in different data sets, and its performance is still stable. Generally speaking, generalization performance of BIAN is superior to the best model in the existing literature.

Table 4. Generalized performance comparison experiments between BIAN and BMAM. The results of BMAM are retrieved from published papers

cv	Model	Laptop	Restaurant	Twitter
5-fold	BMAM	74.50 ± 1.5	78.29 ± 0.6	69.75 ± 0.5
	BIAN	**75.64 ± 1.7**	**80.95 ± 0.2**	**69.91 ± 0.7**
10-fold	BMAM	**76.50 ± 2.4**	80.01 ± 1.8	70.83 ± 1.3
	BIAN	76.50 ± 1.0	**81.57 ± 0.8**	**72.37 ± 2.4**



(a) (b)

Fig. 5. Histogram of mean difference of cross validation results. Dotted line indicates trend line

4.5 Comparison of Different attention Mechanisms

In order to verify attention mechanism's effectiveness on computing semantic relevance between context and target, we propose three different attention functions to replace formula (5) for training our model: **BIAN-Ⅰ** is BIAN with the general attention function, that is:

$$f_{gl}(K,Q) = QW_{atn}K^{T}. \qquad \textbf{(16)}$$

BIAN-Ⅱ is BIAN with the scaled dot product attention function, that is:

$$f_{sdp}(K,Q) = \frac{QK^{T}}{\sqrt{d_h}}. \qquad \textbf{(17)}$$

BIAN-Ⅲ is BIAN with the dot product attention function, that is:

$$f_{dp}(K,Q) = QK^{T}. \qquad \textbf{(18)}$$

The experimental results are shown in Table 5.

Table 5. Experimental results using different attention functions

	Laptop (%)	Restaurant (%)	Twitter (%)
BIAN-I	76.16	82.10	71.52
BIAN-II	**76.85**	82.61	70.58
BIAN-III	75.50	**83.04**	70.85
BIAN	76.30	82.11	**71.84**

From Table 5, we can see that BIAN performs best on the Twitter and BIAN-Ⅰ is only smaller than BIAN which indicates that in the case of large amount of data, setting an appropriate amount of learnable weights can make the algorithm more accurate. BIAN-Ⅱ performed best on the Restaurant, but worse on the Laptop and Twitter. On the Twitter, the performance of BIAN is the best, indicating that the model in this paper can achieve the best results when the dataset is large. Also, on the other two datasets, the difference is only 0.55% and 0.93% higher than the highest accuracy among the other three models, indicating that the attention function proposed in this paper can be used in the attention mechanism to make the context pay close attention to targets and achieve better results for sentiment classification.

We also give the running time of different attention functions on three datasets as shown in table 6. From Table 5 and Table 6, we can see that BIAN used the shortest running time on Laptop and Restaurant and his accuracy is relatively high. BIAN used the longest running time on Twitter, but his accuracy is the highest, which shows that BIAN performs well in execution efficiency. BIAN-I used the shortest running time on Twitter, but longest on Laptop and Restaurant. From table 5, BIAN-I performs not well on Laptop and Restaurant but relatively well on Twitter, which indicates that its execution efficiency is high only on twitter. The running time of BIAN-II and BIAN-III is almost the same, because BIAN-II has one more matrix scaling than BIAN-III, but there is not a great difference between BIAN-II and BIAN-III in calculation efficiency. In general, BIAN performs better in execution efficiency than using other attention functions on three data sets.

Table 6. Running time of different attention functions on three datasets. The unit is seconds. Minimum time is in bold

	Laptop(s)	Restaurant(s)	Twitter(s)
BIAN-I	2610	3887	**5794**
BIAN-II	2570	3809	5956
BIAN-III	2559	3806	5858
BIAN	**2494**	**3796**	5956

4.6 BIAN-Ablations Analysis

In this section, we designed a series of models to verify the validity of our BIAN model. For the embedding layer, in order to explore the influence of BERT model on BIAN, we designed **LSTM-IAN**, which uses the traditional LSTM network instead of the BERT model as the pre-training method. In addition, we also designed **Glove-IAN**, which uses Glove method to embed context and aspect words. In order to explore the influence of interactive attention mechanism on BIAN, we designed three models (i.e., **BIAN-no-Att**, **BIAN-no-Interaction** and **BIAN-no-Aspect**) for interactive attention layer. Among them, **BIAN-no-Att** only uses BERT model as encoder, without interactive attention, and outputs classification results. **BIAN-no-Interaction** is for context and aspect modeling separately, without interaction, in the interactive attention layer. **BIAN-no-Aspect** is only for context modeling and calculates the interaction attention between contexts.

The experimental results are shown in Table 7. From the results, we can see that the results of **Glove-IAN** and **LSTM-IAN** are not as good as those of BIAN, which indicates that the models without BERT performs not as well as those with BERT model. It also proves that BERT can well establish the semantic and grammatical relationship between sentences and aspect words in sentence coding. Compared with **BIAN-no-Att**, BIAN is improved, which indicates that the classification effect of the model is improved by attention mechanism. It is proved that attention mechanism is effective in recognizing the relationship between aspect words and context. For interactive attention layer, the results of **BIAN-no-Aspect** and **BIAN-no-Interaction** were lower than BIAN. **BIAN-no-Aspect** is the lowest among the three, which proved the importance of aspect word modeling alone, and aspect word expression could contribute to the judgment of emotional polarity. The performance of **BIAN-no-Interaction** is worse than that of BIAN, which shows that the interaction between context and aspect words can obtain more fine-grained semantic information. The experimental results show that BIAN can improve the ability of word-aspect associated perception by combining BERT and interactive attention, so that the emotional polarity of sentences can be accurately classified.

Table 7. Ablation results on three datasets

Layer	Models	Laptop	Restaurants	Twitter
For Embedding layer	Glove-IAN	68.75	78.30	66.32
	LSTM-IAN	71.68	79.26	68.44
For interactive attention layer	BIAN-no-Att	74.48	80.29	70.45
	BIAN-no-Interaction	75.58	83.01	71.66
	BIAN-no-Aspect	75.36	82.95	71.78
	BIAN	**76.49**	**83.11**	**71.84**

5 Conclusion

In this paper, we propose a BERT-based interactive attention networks for aspect sentiment analysis. Which employs the Bert model for modeling context, target and target-dependent context and uses attention mechanism to well generate representations for contexts and aspect words separately. BIAN can pay close attention to the important parts in context by joint learning attention and embedding layer. Experimental results on multiple data sets show that BIAN leads to much better performances than other state-of-the-arts. Our model has large frame, large memory and long training time. In the future, we will try to make the model lightweight and use parallelizable multi-head attention mechanism to shorten the training time.

References

[1] B. Pang, L. Lee. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval 2(1)(2008) 1-135.

[2] S. Kiritchenko, X. Zhu, C. Cherry, S. Mohammad. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In Proceedings of the 8th International Workshop on Semantic Evaluation SemEval 2014(2014, August) 437-442.

[3] M. Zhang, Y. Zhang, & D.-T. VO. Gated Neural Networks for Targeted Sentiment Analysis[C]. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16). AAAI Press (2016) 3087-3093.

[4] D. Tang, B. Qin, X. Feng, et al. Effective LSTMs for Target-Dependent Sentiment Classification, in: International Conference on Computational Linguistics, 2016..

[5] D. Bahdanau, K. Cho, Y. Bengio, et al. Neural Machine Translation by Jointly Learning to Align and Translate, arXiv: Computation and Language, 2014.

[6] D. Ma, S. Li, X. Zhang, et al. Interactive Attention Networks for Aspect-level Sentiment Classification, in: International Joint Conference on Artificial Intelligence, 2017.

[7] Y.-F. Zeng, T. Lan, Z.-F. Wu, Q. Liu, Bi-Memory based attention model for aspect level sentiment classification, Chinese Journal of Computers 42(08)(2019) 1845-1857.

[8] T. Wu, C.-P. Cao, Aspect Level Sentiment Classification Model with Location Weight and Long-short Term Memory based on Attention-over- attention, Journal of Computer Applications 39(08)(2019) 2198-2203.

[9] X.-F. Wang, L. Wang, F.-Y. Miao, C.-X. Shao, Aspect Level Sentiment Analysis with Memory Network with POS, Position and Polarity of Word, In Journal of Chinese Computer Systems 40(02)(2019) 383-389.

[10] D. Tang, B. Qin, X. Feng, et al. Effective LSTMs for Target-Dependent Sentiment Classification, in: Proc of the 26th International Conference on Computational Linguistics (Technical Papers), 2016.

[11] Y. Wang, M. Huang, X. Zhu, et al. Attention-based LSTM for Aspect-level Sentiment Classification, in: Empirical Methods in Natural Language Processing, 2016.

[12] P. Chen, Z. Sun, L. Bing, et al. Recurrent Attention Network on Memory for Aspect Sentiment Analysis, in: Empirical Methods in Natural Language Processing, 2017.

[13] T. Mikolov, K. Chen, G.-S. Corrado, et al. Efficient Estimation of Word Representations in Vector Space, in: International Conference on Learning Representations, 2013.

[14] A. Vaswani, N. Shazeer, N. Parmar, et al. Attention is All you Need, in: Neural Information Processing Systems, 2017.

[15] M.-E. Peters, M. Neumann, M. Iyyer, et al. Deep Contextualized Word Representations, in: North American Chapter of the Association for Computational Linguistics, 2018.

[16] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever. Improving language understanding with unsupervised learning, in: Technical report, OpenAI, 2018.

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL-HLT 1(2019) 4171-4186.

[18] P.-J. Werbos, Backpropagation through time: what it does and how to do it, Proc IEEE 78(10)(1990) 1550-1560.

[19] B. Liu, E. Blasch, Y. Chen, et al. Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier, in: International Conference on Big Data, 2013.

[20] S.-I. Wang, C.-D. Manning. Baselines and Bigrams: Simple, Good Sentiment and Topic Classification, in: Meeting of the Association for Computational Linguistics, 2012.

[21] X. Song, J. Liang, C. Hu, et al. Sentiment Classification: A Topic Sequence-Based Approach, Journal of Computers (2016) 1-9.

[22] A. Neviarouskaya, H. Prendinger, M. Ishizuka, et al. SentiFul: Generating a Reliable Lexicon for Sentiment Analysis, in: Affective Computing and Intelligent Interaction, 2009.

[23] D. Rao, D. Ravichandran. Semi-Supervised Polarity Lexicon Induction, in: Meeting of the Association for Computational Linguistics, 2009.

[24] T. Mullen, N. Collier. Sentiment Analysis using Support Vector Machines with Diverse Information Sources, in: Empirical Methods in Natural Language Processing, 2004.

[25] S. Yi, H. Yi, G.-D Zhou. Twitter sentiment classification with sentimental feature vector, In Journal of Chinese Computer Systems 37(11)(2016) 2454-2458.

[26] D. Tang, B. Qin, X. Feng, et al. Effective LSTMs for Target-Dependent Sentiment Classification, in: International Conference on Computational Linguistics, 2016.

[27] K.-S. Tai, R. Socher, C.-D Manning, et al. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks, in: International Joint Conference on Natural Language Processing, 2015.

[28] L. Dong, F. Wei, C. Tan, et al. Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification, in: Meeting of the Association for Computational Linguistics, 2014.