J Inf Process Syst, Vol.18, No.5, pp.614~627, October 2022
https://doi.org/10.3745/JIPS.04.0254

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Aspect-Based Sentiment Analysis with Position Embedding Interactive Attention Network

Yan Xiang, Jiqun Zhang, Zhoubin Zhang, Zhengtao Yu*, and Yantuan Xian

Abstract
Aspect-based sentiment analysis is to discover the sentiment polarity towards an aspect from user-generated natural language. So far, most of the methods only use the implicit position information of the aspect in the context, instead of directly utilizing the position relationship between the aspect and the sentiment terms. In fact, neighboring words of the aspect terms should be given more attention than other words in the context. This paper studies the influence of different position embedding methods on the sentimental polarities of given aspects, and proposes a position embedding interactive attention network based on a long short-term memory network. Firstly, it uses the position information of the context simultaneously in the input layer and the attention layer. Secondly, it mines the importance of different context words for the aspect with the interactive attention mechanism. Finally, it generates a valid representation of the aspect and the context for sentiment classification. The model which has been posed was evaluated on the datasets of the Semantic Evaluation 2014. Compared with other baseline models, the accuracy of our model increases by about 2% on the restaurant dataset and 1% on the laptop dataset.

Keywords
Aspect-Based Sentiment Analysis, Attention Mechanism, Long Short-Term Memory Network, Position Information, Word Embedding

1. Introduction

There are many major undertakings in natural language processing. Sentiment analysis is considered to be one of them [1,2]. Compared with document level and sentence level sentiment analysis, aspect-based sentiment analysis (ABSA) is more fine-grained. The task of ABSA is to recognize the positive or negative opinions about the entities (also called aspects/features/targets) in a review. Aspect-term sentiment analysis (ATSA) will recognize emotional polarity in the form of multi-word phrases or single words related to the target entity appearing in the comment [3]. We focus on the latter in this paper. There are different emotions classified in a sentence, such as in the sentence “iPhone's voice quality is great, but its battery sucks.” the voice quality of the iPhone is classified as a positive emotion, and the battery is classified as a negative emotion.

Traditional methods emphasize the design of a set of features to solve the ABSA problem [4-9]. These models use conventional machine learning to construct sentiment features and make sentimental polarity

Manuscript received March 25, 2021; first revision January 14, 2022; accepted February 9, 2022.

* **Corresponding Author:** Zhengtao Yu (ztyu@hotmail.com)
Dept. of Information Engineering and Automation, Kunming University of Science and Technology (sharonxiang@126.com, 296901207@qq.com, z388075221@163.com, ztyu@hotmail.com, 20030083@kust.edu.cn)

predictions for specific aspects. Bag-of-words, sentiment lexicon, or rules are used to train classifiers. Their performances depend largely on the quality of features, which has great limitations.

There are three important issues in ABSA. The first is to represent the contextual words of an aspect in the sentence or document. The second is to get aspect representation with contextual interaction. Last but not least, how to distinguish the important emotional words in the specified aspect of the text. Many methods are developed to model the relationship of aspect and context, and to model context more accurately by generating specific representations [10]. However, they do not consider a particular positional relationship of the aspect and its context. In fact, the polarity of an aspect term is mainly represented by its neighboring words in the context. Those words closer to the aspect become more likely to represent the polarity. For instance, in the sentence “The sangria was pretty tasty and good on a hot muggy day,” “sangria” is an aspect term, and “pretty tasty” next to “sangria” can express positive sentiment, not “muggy day” far away. The general attention mechanism calculates the relationship between hidden vectors. It can't discriminate the significant degree between different words, because it does not make explicit use of position information. Therefore, we proposed a model named PEIAN (position embedding interactive attention network) for ABSA, which fully utilizes context position and the interactive attention between aspect and context to obtain the most important sentiment terms for the aspect and to better conduct sentiment prediction.

The major contributions of the dissertation are as the following:

1) We propose three-position embedding methods to represent the relative position information of context and aspect, including random embeddings with relative positions, random embeddings with absolute positions, and word embeddings with weights. We compare the performances of the three methods and find that the method of word embeddings with weights has the best performance.
2) We propose a long short-term memory (LSTM)-based model incorporating the relative position information for ABSA. We study the affection of the relative position information on the input layer and the interactive attention layer. When the location information is explicitly added to the input layer and the interactive attention layer of the model simultaneously, the most effective representation can be generated, and the model performs best.
3) On the premise of adding position information to the network, interactive attention mechanisms are used to model the aspect and the context and simultaneously obtained information of aspect and context. It plays an important role, because this mutual relation between the aspect and the different words in the context can predict the sentimental polarity.

We evaluated the proposed model on the datasets of the Semantic Evaluation 2014 (SemEval2014). The experimental results show that of our model outperform the other state-of-the-art models.

2. Related Works

ABSA is a branch of text classification, which belongs to fine-grained sentiment classification. Related researches include traditional emotion classification methods and neural network methods.

2.1 Conventional Sentiment Classification Methods

In the early stages, sentiment classification consists of rule-based methods [11], SVM-based methods

[5,12], and so on. Manual feature engineering efforts are widely needed, which includes sentiment lexicon [7,8], n-gram and parses tree feature [9]. Feature selection techniques can decrease the less informative features and increase the performance and the speed [13,14]. These methods are widely used, but their results still depend on whether the manual features are effective enough. What's more, the features cannot be extracted automatically, which leads to the time and manpower consumption when processing a large amount of data. In addition, sentimental labels are usually not easy to be obtained. In this case, text clustering methods can be used, such as k-mean clustering, LDA clustering, hierarchical clustering, and so on [15-18].

2.2 Methods based on Neural Networks

Nowadays neural networks are researched mostly and they become common in sentiment analysis [19,20]. They can extract features automatically for this task. We introduce the related works below.

One approach is based on convolutional neural network (CNN) or recursive neural network (RNN) [21]. However, the underlying assumption that the sentences have syntactic rules may not always be correct when online comments and reviews are considered. Chen [22] used CNN to obtain the sentiment of an aspect by recognizing the sentiment of the clause. The neural sequential model, such as LSTM [23], is another way to represent features. They have abilities to represent sequential information.

The hierarchical and bidirectional LSTM model, which proposed by Ruder et al. [24], was utilized the relationship between the words and the sentences. Furthermore, attention mechanism was considered in some sequence-based methods [4,10]. An LSTM network based on aspect embedding is conceived by Wang et al. [9], which focuses on the relevant parts of the sentence. The model used attention mechanism to focus important parts of a sentence. It is adaptive, for the correct words are focused by the model. Tay et al. [25] use aspect information, which is established by the relationship between context and aspect terms, and is incorporated into a neural model. The interactive attention networks (IAN), which were proposed by Ma et al. [4], get the feature representations for aspect terms and context. This attention mechanism takes the sequence representation and external memory as inputs, and generates a probability distribution to generate the focus of each position in the sequence [26]. On the whole, the advantage of CNN-based methods is high efficiency, while LSTM based methods have better performance of classification.

Some methods use the phrase and syntactic structure information to improve performance [25]. What's more, there are also different kinds of joint models for ABSA. The opinion can be refined by adding sentiment polarity so that both opinion expressions and the polarity information can be jointly captured using sequence labeling models [28]. Also, aspect extraction tasks can be added to this joint learning framework [10,28]. These methods get more performance because of additional knowledge, but models are complex and there are many limitations to use.

We propose an LSTM-based model utilizing explicit position information and the interaction attention between the aspect and the context. The hidden vectors encoded by LSTM contain word order information and syntactic information and LSTM-based models have been proved to perform well for ABSA, compared with the CNN-based method. Position information and interaction attention can identify the importance of context words for the given aspect, then get a better representation of context for sentiment classification. Compared with the other methods, our model achieves better results.

3. Proposed Device Discovery Scheme

3.1 Position Embedding of Context

For a sentence, the aspect may have more than one word, which is uniformly expressed as w_a, and the context has N words totally $\{w_1, w_2, \cdots, w_{a-1}, w_{a+1}, \cdots, w_N\}$, shown in Fig. 1. Then the relative position of the context and the aspect is $R_p = \{1-a, 2-a, \cdots, -1, 1, \cdots, N-a\}$. For example, given a sentence "This is some of the worst sushi I have ever tried", the aspect "sushi" is the seventh word, and the position sequence of the context "this is some of the worst I have ever tried" is expressed as $R_p = \{-6, -5, -4, -3, -2, -1, 1, 2, \cdots, N-7\}$.

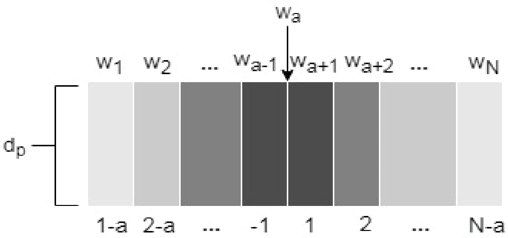


Fig. 1. The diagram of the position embedding.

To encode the relative position of the context and the aspect, we design three position embedding modes, which are expressed by $P_i, i = 1,2,3$:

1) The first mode P_1: For a position value in R_p, a random vector subjected to uniform distribution is generated as position embedding with dimension d_p. If two words in different sentences have the same relative position values, they have the same position embeddings. On the contrary, they are different.
2) The second mode P_2: Firstly, the absolute value of R_p is taken. Then the position embedding is generated according to the mode of P_1. If the absolute value of a relative position in R_p is the same, then the position embedding is the same.
3) The third mode P_3: The relative distance between the certain word w_i of the context and the aspect w_a is $i-a$. First, the word embedding of w_i with dimension d_p can be obtained from the index table of the word vector. Then, the corresponding word embedding is multiplied by the weights $1-(|i-a|-1)/N$, as the position embedding of w_i. The word vector can be Word2vec, Glove, and all that.

3.2 Structure of Our Model

Our model has the structure shown as the Fig. 2.

1) The input layer: Suppose the context is composed of N words $\{w_1^c, w_2^c \cdots w_N^c\}$ and the aspect is composed of M words $\{w_1^t, w_2^t \cdots w_M^t\}$. Firstly, we get the word embedding of the context and the aspect with dimensions d_p from the index table of word vectors. For the context, the position embedding P_{i} is obtained by using the method described above, and it is concatenated with the word embedding of the context to get the input of the context. The input of the aspect is its word embedding.

2) The hidden layer: The inputs of the context and the aspect are fed into LSTM, respectively. If the input vector of a word is e^k, the state of the former cell is c^{k-1}, the state of the previously hidden layer is represented as h^{k-1}. The network was updated when it uses the current cell state c^k and hidden layer state h^k :

$$i^k = \sigma(W_i^e \cdot e^k + W_i^h \cdot h^{k-1} + b_i) \tag{1}$$

$$f^k = \sigma\left(W_f^e \cdot e^k + W_f^h \cdot h^{k-1} + b_f\right) \tag{2}$$

$$o^k = \sigma(W_o^e \cdot e^k + W_o^h \cdot h^{k-1} + b_o) \tag{3}$$

$$\hat{c}^k = \tanh(W_c^e \cdot e^k + W_c^h \cdot h^{k-1} + b_c) \tag{4}$$

$$c^k = f^k \odot c^{k-1} + i^k \odot \hat{c}^k \tag{5}$$

$$h^k = o^k \odot \tanh(c^k) \tag{6}$$

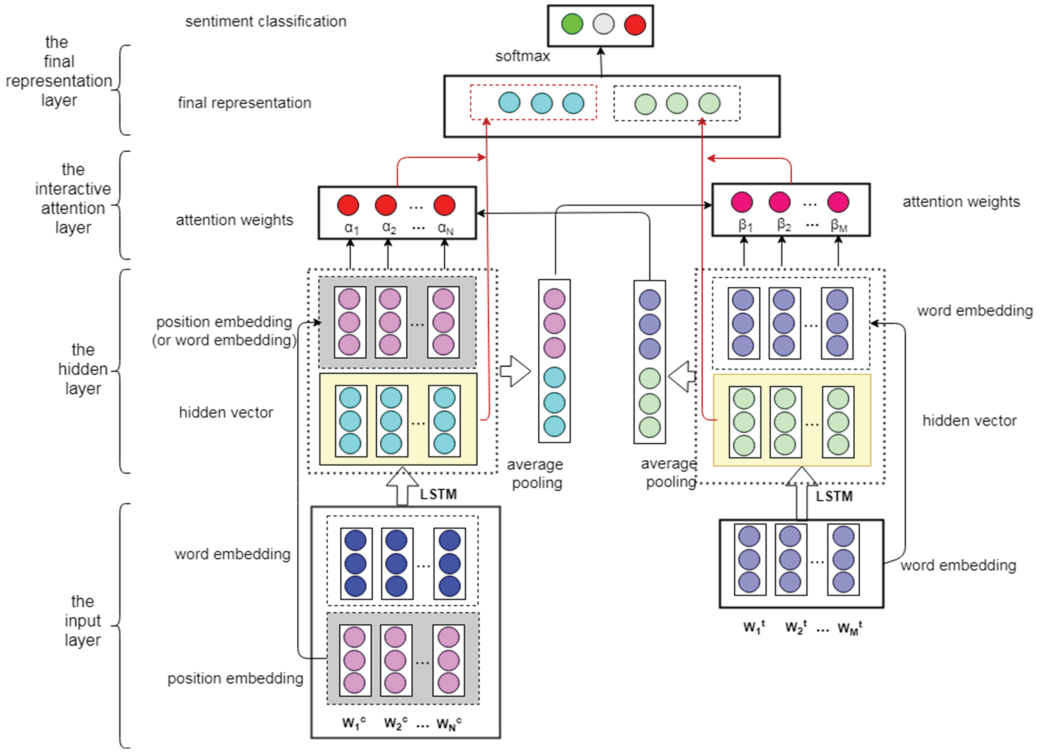


Fig. 2. Overall architecture of PEIAN.

Among them, input gate, forgetting gate, and output gate are represented by $i, f,$ and o, respectively; σ is sigmoid activation functions; W and b denote weights and biases respectively; the symbol $\cdot$ is matrix multiplication, $\odot$ expresses element-wise multiplication. Then we obtain the hidden vector of the context $\{h_1^c, h_2^c \cdots h_N^c\}$ and the hidden vector of the aspect $\{h_1^t, h_2^t \cdots h_N^t\}$, respectively.

3) Acquisition of the new vectors: Concatenate the position embedding of the context to the hidden vector of its corresponding words $\{h_1^c, h_2^c \cdots h_N^c\}$. For example, for the hidden vector h_i^c of the word

w_i in the context, the new vector is $h_i^{cp} = [h_i^c, w_i^p]$, where w_i^p is the position embedding of w_i. What's more, the word embedding of the aspect and its corresponding hidden vector $\{h_1^t, h_2^t \cdots h_M^t\}$ are concatenated. For example, for the hidden vector h_j^t of the word w_j in the aspect, the new vector is $h_i^{tw} = [h_i^t, w_i^t]$,where w_i^t is the word embedding vector of the aspect.

4) The interactive attention layer: New vectors h_i^{cp} and h_i^{tw} are used to calculate interactive attention. Firstly, the average vectors of the context and the aspect are obtained by average pooling:

$$\mathrm{C} = \frac{1}{N}\sum_{i=1}^{N} h_i^{cp} \tag{7}$$

$$\mathrm{T} = \frac{1}{M}\sum_{j=1}^{M} h_j^{tw} \tag{8}$$

Use T to obtain the attention weights of $h_i^{cp}(i = 1,2 \cdots N)$, and use C to obtain the attention weights of $h_j^{tw}(j = 1,2 \cdots M)$:

$$\emptyset\left(h_i^{cp}, \mathrm{T}\right) = \tanh\left(h_i^{cp} \cdot W_{cp} \cdot \mathrm{T}^T + b_{cp}\right) \tag{9}$$

$$\emptyset\left(h_j^{tw}, \mathrm{C}\right) = \tanh\left(h_j^{tw} \cdot W_{tw} \cdot \mathrm{C}^T + b_{tw}\right) \tag{10}$$

where $\tanh(\cdot)$ is an activation function.

5) The final representation layer: final attention $\alpha_i(i = 1,2 \cdots N)$ and $\beta_j(j = 1,2 \cdots M)$ can be calculated by:

$$\alpha_i = \frac{e^{\emptyset(h_i^{cp}, T)}}{\sum_{i=1}^{N} e^{\emptyset(h_i^{cp}, T)}} \tag{11}$$

$$\beta_j = \frac{e^{\emptyset\left(h_j^{tw}, C\right)}}{\sum_{j=1}^{M} e^{\emptyset\left(h_j^{tw}, C\right)}} \tag{12}$$

We multiply the hidden vectors by the attention weight α_i, β_j to get the context representation and the aspect representation:

$$\mathrm{C}' = \sum_{i=1}^{N} \alpha_i h_i^c \tag{13}$$

$$\mathrm{T}' = \sum_{j=1}^{M} \beta_j h_j^t \tag{14}$$

Then the representation C' and $\mathrm{T}'$are concatenated as $\mathrm{S} = [\mathrm{C}', \mathrm{T}']$. We project S into the space of K categories by a non-linear transformation:

$$x = \tanh(W_r \cdot S + b_r) \tag{15}$$

Finally, the probability that an aspect belongs to a sentiment category i $(i = 1,2,\ldots,K)$ is computed by

$$y_i = \frac{e^{x_i}}{\sum_{i=1}^{K} e^{x_i}} \tag{16}$$

According to the maximum probability, the model gets final sentiment category. The training loss is the cross-entropy loss.

4. Experimental Results

4.1 Experimental Data and Parameter Setting

The effectiveness of the model is verified on the SemEval2014 task. SemEval2014 datasets contains data from restaurant and laptop. Sentiment polarities include negative, positive, and neutral. The number of training and test instances of the two datasets is stated categorically in Table 1.

Table 1. Statistics of SemEval2014 datasets

Dataset	Positive		Neural		Negative	
	Train	Test	Train	Test	Train	Test
Restaurant	2,164	728	637	196	807	196
Laptop	994	341	464	169	870	128

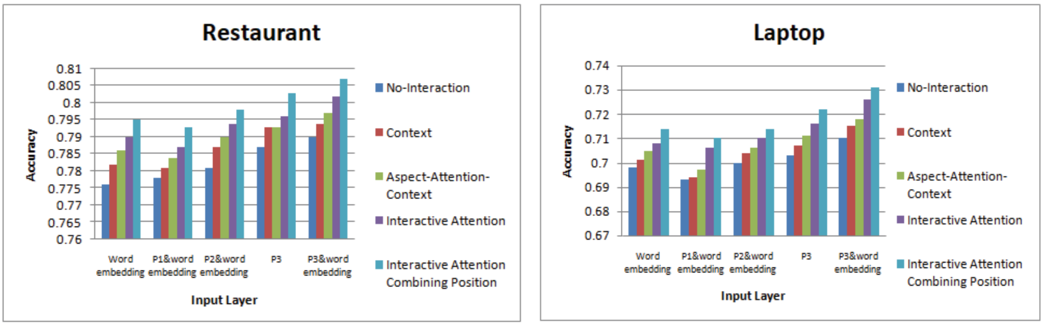
In our model, word embedding is given by Glove [29], and all out-of-vocabulary words are initialized by uniform distribution U(−0.1, 0.1). The initial weight matrix is in a uniform distribution U(−0.1, 0.1). Set the initial biases to be zeros. The dimension of word embedding, position embedding, and LSTM hidden states are set to 300 to compare fairly with IAN and other baseline models. We set the coefficient of L2 normalization to be 10^{-5}, and the dropout rate to be 0.5. The experiment utilizes the Adam optimization strategy with a batch size of 32. The default learning rate is 0.01, and the maximal training epoch is 10.

4.2 Experiments on Real Images

We design a series of experiments to verify the ability of introducing position information into the input layer and attention layer. In Fig. 3, five nodes of horizontal ordinate represent five different input layers of the context. $P_i, i = 1,2,3$ represent the position embedding as mentioned above, and "&word embedding" represents that the position embedding concatenates to the word embedding as input. In addition, we design five different networks by referring to [4], like the following.

1) Context: We do not use aspect information. The attention weights of the context are learned by its own hidden vectors $h_i^c(i = 1,2,IN)$. Finally, we use the summation of the hidden feature vectors to multiply their corresponding attention weights, to represent a sentence.
2) No-Interaction: The attention weights of the context and the aspect are learned by their own hidden vectors, without interactive attention. That is, the aspect and the context are modeled independently.
3) Aspect-Attention-Context: The average pool vector of $\{h_1^t, h_2^t \cdots h_M^t\}$ is used to get the attention weight of h_i^c. The final processing is same as the step 2.

4) Interactive Attention: The average pool vector of $\{h_1^t, h_2^t \cdots h_M^t\}$ is used to get the attention weight of $h_i^c .(i = 1,2, IN)$, and the average pooling vector of $\{h_1^c, h_2^c \cdots h_N^c\}$ to obtain the attention weights of $h_j^t (j = 1,2, IM)$. The hidden vectors earning from context and aspect will multiply their corresponding attention weights, summed and concatenated to represent the sentence.
5) Interactive Attention Combining Position: Use T to obtain the attention weights of $h_i^{cp} (i = 1,2, IN)$, and C to the attention weights of $h_j^{tw} (j = 1,2, IM)$. Then the hidden vectors of the context and the aspect multiplied by their corresponding attention weights are summed and concatenated to represent the sentence.



(a) (b)

Fig. 3. Accuracy of different networks with different input layers on the SemEval2014 dataset: (a) restaurant and (b) laptop.

Fig. 3 illustrates the accuracy of experiments on the restaurant and the laptop datasets. For the different position embedding of the input layer, we can see that P3&word embedding gets the best result, followed by P3 and P2&word embedding. P1&word embedding gets the worst result, even worse than not using position embedding in most cases. Comparing with the input using word embedding directly, P3 increases the accuracy of the five networks by 0.5%–1%. This is owing to that P3 is a kind of weighted word embedding and has both semantic information and position information. What's more, P3 is concatenate to the original word embedding to get P3&word embedding, which further strengthens the combination of semantic and position information, and obtains about 0.5% higher accuracy than P3.

Interactive Attention Combining Position network with input of P3&word embedding (i.e., PEIAN) has the highest accuracy in all methods, which achieves 80.7% and 73.1% in restaurant and laptop, respectively. This is because PEIAN fully use the position information of the input layer and the attention layer, as well as the interaction of the aspect and the context, thus effectively improving the sentiment classification.

4.3 Comparisons of Different Models

In order to evaluate the superiority of our model comprehensively, we compared it with some advanced models as follows:

LSTM: LSTM is a kind of neural network that contains LSTM blocks. It models the context with only one LSTM network. After obtaining the hidden state, we averaged the values as the final representation, and sent it to the softmax function ultimately [10]. The 300-dimensional Glove vector is used and embedded for a word. The hidden states produced by the LSTM is set to 300. The learning rate is set as 0.01.

AE-LSTM: Words are modeled through LSTM. It is a better method, for the final representation of the input sentence generated by using the attention weight for judging the polarity. The attention weight is obtained by connecting various aspects hidden to the context representation [10]. The word vector, hidden states and learning rate that we set is same as the LSTM model.

ATAE-LSTM: ATAE-LSTM is extended by AE-LSTM. Its parameters are the same as those of AE-LSTM. For this model, aspect embedding is added to each word embedding method to represent the input sentence [10].

TD-LSTM: TD-LSTM gives right and left context representation using two LSTM models individually and combines them to predict aspect polarity [26]. The word vector, hidden states and learning rate that we set is same as the LSTM model.

GCAE: GCAE is proposed by combining the convolution layer and gating mechanism. Partly is due to the convolution filter and partly is due to the gating unit on the convolution layer and the maximum pool layer, the model extracts different granularity n-gram features from different embedded vectors in each position, and accurately extracts and selects the related emotional features [23]. The word vector and learning rate, which we fit, is identical to the LSTM model, 100 filter's widths that we set are 3,4,5.

MemNet: More abstract evidence can be selected through external memory, which is provided by the MemNet algorithm. The output of the attention layer is connected the softmax layer, after embedding words with multiple attention [27]. We set dimension of the word vector and learning rate of the model to be the same as those of LSTM model.

IAN: Aspects and contexts are modeled respectively based on the interactively learn attentions, and are combined to predict the sentiment polarity [4]. The word vector, the number of hidden states and the learning rate of the model is same as the LSTM model. Get the initial weights by using uniform distribution U(0.1, 0.1). Set all deviations to be zero.

Table 2 shows the results of different models on SemEval2014 datasets. Our model achieves the best performance in all models. LSTM model was the most inferior. The principal reason is that it only depends on the context information. So it cannot give full information of the aspect to predict the sentimental polarity. Compared with LSTM, TD-LSTM improves the accuracy of restaurant and laptop datasets by 1.3% and 1.6%, respectively. The main contribution comes from the processing of aspects and contexts. It is more effective, for adding the attention mechanism to get important words. The two models of AE-LSTM and ATAE-LSTM are somewhat similar, and the latter is an extension of the former. Both of them have much better performance than TD-LSTM. In particular, ATAE-LSTM enhances the interaction between context and aspect. Therefore, compared with TD-LSTM, ATAE-LSTM improves the accuracy of restaurant and laptop datasets by 1.6% and 0.6%, respectively.

Compared with ATAE-LSTM, the accuracy of IAN in restaurant and laptop datasets are improved by 1.4% and 3.4%. The advantage of MemNet is that it applies multiple attention to word embedding, but it does not pay enough attention to the potential relevance of context and aspect. PEIAN uses context location information in the embedded layer and the attention layer simultaneously, and mines attention weights for the context, which greatly strengthens the important information and weakens the unimportant information. The experimental results show that it achieves best performance, which is 0.8% and 2.8% higher than MemNet, as well as 2.1% and 1% higher than IAN.

Table 2. Experimental results of different models

Model	**Accurate rate of 3-class for SemEval2014 Task 4**	
	Restaurant	**Laptop**
LSTM	0.743	0.665
TD-LSTM	0.756	0.681
AE-LSTM	0.762	0.689
ATAE-LSTM	0.772	0.687
GCAE	0.775	0.694
MemNet	0.799	0.703
IAN	0.786	0.721
PEIAN	**0.807**	**0.731**

Bold font indicates the best performance in each test.

4.4 Statistical Significance Analysis

We further evaluate the results of PEIAN and IAN on ten experiments by *t*-test. Precision, recall, F1, and weighted-average are shown in Tables 3 and 4. We highlight (bold) the better values and *p*-values of t-test.

Table 3. Significance tests of PEIAN and IAN on Restaurant dataset

	Precision			**Recall**			**F1**		
	best	**mean**	***p*-value**	**best**	**mean**	***p*-value**	**best**	**mean**	***p*-value**
Positive			1.17E-01			4.86E-01			**1.62E-03**
IAN	0.885	0.841		0.962	0.925		0.886	0.881	
PEIAN	**0.894**	**0.856**		**0.964**	**0.926**		**0.899**	**0.889**	
Neutral			1.51E-01			1.23E-01			**2.72E-02**
IAN	0.622	0.570		0.490	0.388		0.508	0.459	
PEIAN	**0.693**	**0.597**		**0.541**	**0.428**		**0.559**	**0.499**	
Negative			4.10E-01			1.16E-01			**1.87E-02**
IAN	0.761	0.701		0.745	0.654		0.703	0.673	
PEIAN	**0.779**	**0.705**		**0.770**	**0.692**		**0.724**	**0.695**	
Weighted-average			**3.54E-05**			**3.54E-05**			**3.54E-05**
IAN	0.787	0.769		0.794	0.784		0.788	0.770	
PEIAN	**0.807**	**0.787**		**0.811**	**0.798**		**0.811**	**0.787**	

Bold font indicates the better values and *p*-values of *t*-test (α<0.05).

Table 4. Significance tests of PEIAN and IAN on Laptop dataset

	Precision			**Recall**			**F1**		
	best	**mean**	***p*-value**	**best**	**mean**	***p*-value**	**best**	**mean**	***p*-value**
Positive			1.95E-01			1.89E-01			**1.00E-02**
IAN	**0.878**	0.821		0.894	0.848		0.841	0.833	
PEIAN	0.872	**0.834**		0.894	**0.858**		**0.855**	**0.845**	
Neutral			2.05E-01			**7.14E-02**			**1.50E-02**
IAN	**0.702**	0.648		0.556	0.451		0.591	0.527	
PEIAN	0.702	**0.661**		**0.568**	**0.490**		**0.597**	**0.561**	
Negative			3.31E-01			2.64E-01			3.56E-01
IAN	**0.578**	0.521		0.742	0.674		0.603	0.586	
PEIAN	0.571	**0.526**		**0.773**	0.658		**0.612**	0.582	
Weighted-average			**7.17E-02**			**1.88E-02**			**2.93E-02**
IAN	0.727	0.715		0.724	0.708		0.723	0.703	
PEIAN	**0.749**	**0.726**		**0.735**	**0.720**		**0.726**	**0.717**	

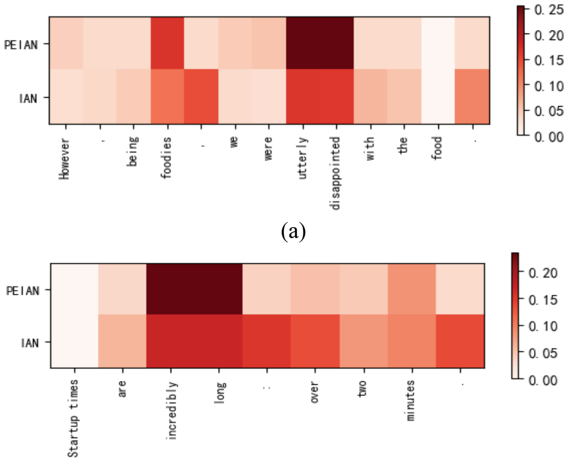
Bold font indicates the better values and *p*-values of *t*-test (α<0.05).

In Table 3, the weighted-average of precision, recall, and F1 of PEIAN and IAN are significantly different. All F1 values of PEIAN have significant improvement. On balance, the experiment suggests that PEIAN outperforms IAN.

Similar with Table 3, the weighted-average of precision, recall, and F1 of PEIAN and IAN in Table 4 are significantly different. F1 values of PEIAN show significant improvement. The experimental result also shows that PEIAN outperforms IAN.

4.5 Analysis of Specific Examples

To intuitively understand the proposed model, we show the attention visualizations for two examples in Fig. 4. The weight of the aspect is set to be zero to specifically compare the context weights of PEIAN and IAN.



(a)

(b)

Fig. 4. Attention visualizations for two examples in PEIAN and IAN: (a) Restaurant and (b) Laptop. The weight of the aspect is set to be zero for visualizations.

In Fig. 4(a), we can observe that sentiment terms “utterly disappointed” closer to the aspect “food” weighs a lot more than other context words in PEIAN. This characteristic also can be observed in Fig. 4(b), which visualizes weights of the negative review “Startup times are incredibly long; over two minutes.” In this example, the aspect is “Startup times.” The sentiment terms “incredibly long” closer to the aspect also have a lot more weights than other context words in PEIAN. This makes many contributions to judging aspect sentiment polarity. So, in both cases, PEIAN correctly judged the polarity of aspects.

5. Conclusion

Position information plays a crucial role in aspect sentiment classification, but previous models do not explicitly use this position information. In this paper, we designed three patterns to represent position information. We proposed a sentiment classification method named PEIAN, which explicitly took advantage of position information in the input layer and the interactive attention layer to generate the most effective representation for the aspect and the context respectively. We tested PEIAN and other seven models on SemEval 2014 datasets. The accuracy of PEIAN achieved 80.7% on the restaurant dataset and 73.1% on the laptop dataset. In comparison with other baselines, PEIAN obtained the best results. At the same time, we calculated p-value of significance tests for PEIAN and IAN. The weighted-average F1-values of PEIAN obtained 81.1% and 72.6% in the two datasets, which were much better than IAN. The attention visualizations also showed that PEIAN can reasonably pay attention to those sentiment terms, and learn effective features of the aspect and the context for judging the aspect sentiment polarity. Overall, PEIAN is a suitable model for the ABSA task. The model can be applied to reviews of other domains in the future.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. 62162037) and the General projects of basic research in Yunnan Province (No. 202001AT070047 and 202001AT0 70046).

References

[1] B. Liu, “Sentiment analysis and opinion mining,” *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1-167, 2012.

[2] A. Yadav and D. K. Vishwakarma, “A Language-independent network to analyze the impact of COVID-19 on the world via sentiment analysis,” *ACM Transactions on Internet Technology (TOIT)*, vol. 22, no. 1, article no. 28, 2022. https://doi.org/10.1145/3475867

[3] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah, “Aspect-based sentiment analysis: a survey of deep learning methods,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 6, pp. 1358-1375, 2020.

[4] D. Ma, S. Li, X. Zhang, and H. Wang, “Interactive attention networks for aspect-level sentiment classification,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, Melbourne, Australia, 2017, pp. 4068-4074.

[5] V. Perez-Rosas, C. Banea, and R. Mihalcea, “Learning sentiment lexicons in Spanish,” in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey 2012, pp. 3077-3081.

[6] D. Rao and D. Ravichandran, “Semi-supervised polarity lexicon induction,” in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL)*, Athens, Greece, 2009, pp. 675-682.

[7] N. Kaji and M. Kitsuregawa, “Building lexicon for sentiment analysis from massive collection of HTML documents,” in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, 2007, pp. 1075-1083.

[8] S. M. Mohammad, S. Kiritchenko, and X. Zhu, “NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets,” in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*, Atlanta, GA, 2013, pp. 321-327.

[9] Y. Wang, M. Huang, X. Zhu, and L. Zhao, “Attention-based LSTM for aspect-level sentiment classification,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, 2016, pp. 606-615.

[10] A. Ranjan, A. Tiwari, and A. Deepak, "A sub-sequence based approach to protein function prediction via multi-attention based multi-aspect network," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021. https://doi.org/10.1109/TCBB.2021.3130923

[11] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "NRC-Canada-2014: detecting aspects and sentiment in customer reviews," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval)*, Dublin, Ireland, 2014, pp. 437-442.

[12] J. C. Lamirel, P. Cuxac, A. S. Chivukula, and K. Hajlaoui, "Optimizing text classification through efficient feature selection based on quality metric," *Journal of Intelligent Information Systems*, vol. 45, no. 3, pp. 379-396, 2015.

[13] D. Agnihotri, K. Verma, and P. Tripathi, "An automatic classification of text documents based on correlative association of words," *Journal of Intelligent Information Systems*, vol. 50, no. 3, pp. 549-572, 2018.

[14] D. Wu, R. Yang, and C. Shen, "Sentiment word co-occurrence and knowledge pair feature extraction based LDA short text clustering algorithm," *Journal of Intelligent Information Systems*, vol. 56, no. 1, pp. 1-23, 2021.

[15] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "Hybrid clustering analysis using improved krill herd algorithm," *Applied Intelligence*, vol. 48, no. 11, pp. 4047-4071, 2018.

[16] L. M. Abualigah and A. T. Khader, "Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering," *The Journal of Supercomputing*, vol. 73, no. 11, pp. 4773-4795, 2017.

[17] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis," *Engineering Applications of Artificial Intelligence*, vol. 73, pp. 111-125, 2018.

[18] S. C. Tseng, Y. C. Lu, G. Chakraborty, and L. S. Chen, "Comparison of sentiment analysis of review comments by unsupervised clustering of features using LSA and LDA," in *Proceedings of 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, Morioka, Japan, 2019, pp. 1-6.

[19] Y. Mao, Y. Shen, C. Yu, and L. Cai, "A joint training dual-MRC framework for aspect based sentiment analysis," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, pp. 13543-13551, 2021.

[20] B. Kane, A. Jrad, A. Essebbar, O. Guinaudeau, V. Chiesa, I. Quenel, and S. Chau, "CNN-LSTM-CRF for aspect-based sentiment analysis: a joint method applied to French reviews," in *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART)*, Virtual Event, 2021, pp. 498-505.

[21] J. Zhou, S. Jin, and X. Huang, "ADeCNN: an improved model for aspect-level sentiment analysis based on deformable CNN and attention," *IEEE Access*, vol. 8, pp. 132970-132979, 2020.

[22] S. Chen, C. Peng, L. Cai, and L. Guo, "A deep network model for specific target sentiment analysis," *Computer Engineering*, vol. 45, no. 3, pp. 286-292, 2019.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[24] S. Ruder, P. Ghaffari, and J. G. Breslin, "A hierarchical model of reviews for aspect-based sentiment analysis," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, TX, 2016, pp. 999-1005.

[25] Y. Tay, L. A. Tuan, and S. C. Hui, "Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 5956-5963, 2018.

[26] J. Yang, R. Yang, H. Lu, C. Wang, and J. Xie, "Multi-entity aspect-based sentiment analysis with context, entity, aspect memory and dependency information," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 4, article no. 47, 2019. https://doi.org/10.1145/3321125

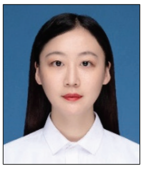
[27] F. Li, C. Han, M. Huang, X. Zhu, Y. Xia, S. Zhang, and H. Yu, "Structure-aware review mining and summarization," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, Beijing, China, 2010, pp. 653-661.

[28] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, MIT Stata Center, MA, 2010, pp. 56-65.

[29] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532-1543.

Yan Xiang **https://orcid.org/0000-0002-6475-638X**

She is an associate professor of the Faculty of Information Engineering and Automation, Kunming University of Science and Technology. She graduated with B.E. degree in engineering and M.S. degree in science from Wuhan University. She presided one general basic research project in Yunnan Province, one sub project of Yunnan provincial major science and technology special plan projects, one project of Yunnan Provincial Department of education. She published over twenty papers as the first author or corresponding author. Her main research interests include text mining and sentiment analysis.



Jiqun Zhang **https://orcid.org/0000-0002-5350-980X**

She is an M.S. candidate at Faculty of Information Engineering and Automation, Kunming University of Science and Technology. Her research interests include natural language processing and sentiment analysis, etc.

Zhoubin Zhang **https://orcid.org/0000-0002-3610-7147**

He is an M.S. candidate at Faculty of Information Engineering and Automation, Kunming university of Science and Technology. His research interests include natural language processing and sentiment analysis, etc.

Zhengtao Yu **https://orcid.org/0000-0002-4012-461X**

He is a professor, Ph.D. supervisor of Faculty of Information Engineering and Automation in Kunming University of Science and Technology. He graduated with Ph.D. in Beijing Institute of Technology. He is holding Chinese Information Processing Society of China (CIPSC), director of the CAA, the director-general of Yunnan Association of Automation title. There are more than 100 published by different international journals and conferences. Possess various copyrights, patents. Have been gained the first prize and the third prize for Yunnan Province Science and Technology Progress Award, the second prize for the Yunnan Province Natural Science Award. His main direction of research is information retrieval, machine translation, natural language processing, etc.

Yantuan Xian **https://orcid.org/0000-0001-6411-4734**

He is currently an associate professor at Kunming University of Science and Technology, China. He graduated from Yunnan Normal University, China, in 2003. He received the M.S. degree from Shenyang Institute of Automation (SIA), China, in 2006. His research interests include pattern recognition, machine learning and information retrieval.