IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 28, 2020, accepted July 20, 2020, date of publication July 22, 2020, date of current version August 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011318

Hierarchical Gated Deep Memory Network With Position-Aware for Aspect-Based Sentiment Analysis

ZEBING JIA, XIUXIU BAI, (Member, IEEE), AND SHANMIN PANG, (Member, IEEE)
School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Xiuxiu Bai (xiubai@xjtu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61802297.

ABSTRACT Aspect-based sentiment analysis aims at identifying the sentiment polarity of specific aspect in the sentence. Previous work has realized the importance of the information interaction between aspect term and context. However, most existing information interaction methods are coarse-grained, which results in a certain loss of information. In addition, most methods ignore the role of position information in identifying the sentiment polarity of the aspect. To better address the two problems, we propose a novel approach, called hierarchical gated deep memory network with position-aware. Our approach has two characteristics: 1) it has fine-grained information interaction attention mechanism which models the word-level interaction between aspect and context. The sentence-to-aspect attention is used to capture the most indicative sentiment words in context. And the aspect-to-sentence attention is used to capture the most important word in the aspect term. 2) The position information is embedded as a feature in the sentence representation. Finally, we conduct sentiment classification comparative experiment on laptop and restaurant datasets. The experimental results show that our model achieves state-of-the-art performance on aspect-based sentiment analysis.

INDEX TERMS Natural language processing, aspect-based sentiment analysis, attention mechanism, position-aware, memory network.

I. INTRODUCTION

Sentiment analysis is also known as opinion mining. It is an important branch of natural language processing [1]. It can be found applications in carious and companies, large and small, which include the analysis of emotions and as part of their mission [2]. An Aspect-based sentiment analysis is a fine-grained task in sentiment analysis which aims to identify the polarity of aspects in their context. For example, given the mentioned aspect words: "food" and "service", and the sentence is "Great food but the service is terrible.". For aspect word "food", the sentiment polarity is positive, but for aspect word "service", the sentiment polarity is negative. Aspect-based sentiment analysis has attracted much attention from researches in recent years because it can provide complete and in-depth results [3].

In the early days, aspect-based sentiment analysis is solved by constructing classifiers with traditional machine learning methods [4]. Typical example is the feature-based Support Vector Machine (SVM) proposed by Kiritchenko *et al.*[5]. However, such kind of contact feature engineering work is labor-intensive and model generalization performance is not very well. As we all know, deep learning has the advantage of automatically extracting text features [6]. Therefore, the application of neural network to deal with aspect-based sentiment analysis has become a hot topic in recent years [7]. As previous work point out that 40% of sentiment classification errors are caused by not considering targets in sentiment classification [8]. So recent works focused on utilizing the interaction between target and context to capture the most indicative sentiment words in them. Tang *et al.* propose TDLSTM model [9]. The model divides the sentence into two parts with aspect and use two long short-term network (LSTM) to model the hidden state of the two parts. The two parts are then combined to produce a target-specific representation which is fed into the softmax function for sentiment classification. Wang *et al.* propose ATAELSTM model [10]. The model embeds aspect term into each context word and then generates aspect-special sentence representations through LSTM and attention mechanism. Tang *et al.* propose deep memory network (DMN) [11]. The model uses a content-based attention mechanism to model the results

The associate editor coordinating the review of this manuscript and approving it for publication was Ramakrishnan Srinivasan.

of the information interaction between target and context. Thus the generated sentence representation contains more information related to the target. Chen *et al.* propose recurrent attention network on memory (RAM) [12]. RAM uses recurrent attention mechanism to build the relationship between context and target words, and update the memory unit through gate recurrent unit (GRU). Ma *et al.* propose Interactive attention network (IAN) [13]. IAN uses two LSTM to calculate the hidden state of the context and the target, respectively. And then use interactive attention mechanism to generate aspect-based context representation and context-based target representation respectively. Finally, the target representation and context representation are concatenated as a vector for a classifier. Tay *et al.* propose AF-LSTM [14]. AF-LSTM uses circular convolution and circular correlation to conduct word level fine-grained modeling of context and aspect terms. Then the target information is embedded into the sentence representation through the information fusion operation. Huang *et al.* propose Attention-over-Attention (AOA) [15] model. AOA model captures word-level interactions between aspect terms and context through two fine-grained attention mechanisms, enabling the generated sentence representation to automatically focus on the parts of the sentence that are most relevant to the expression of aspect terms. Li *et al.* propose Multi-Granularity Alignment Network (MGAN) for ABSA task, which aims to leverage knowledge learned from a rich-resource domain of the coarse-grained aspect category task, to improve the learning in a low-resource target domain of the fine-grained aspect term task [16].

However, the previous studies have almost neglected the role of position information between context and target when identify the sentiment polarity of the target. For example, consider the sentence "The price is reasonable although the service is poor". For the aspect "price", "reasonable" plays a more important role than the other words when identifying sentiment polarity about "price". In a similar way, "poor" plays a more important role than the other words when identifying sentiment polarity about "service". In this case, if we encode the contextual position information about the target when inferring sentiment polarity, it will greatly improve the accuracy of the sentiment classification. Thus, we introduce position embedding in the word embedding layer of our model and further generate the position-aware word vector. That is to say, we consider not only semantic information but also position information when inferring sentiment polarity of target.

Based on analysis above, we first propose a fine-grained information interaction attention mechanism which models the word-level interaction between aspect and context, then we embed position information of the context relative to the target as a feature in sentence representation. In addition, we use BiLSTM [17] to calculate the hidden state of the sentence and aspect term, and update the memory unit through the GRU network. Based on these, we propose a novel deep memory network called hierarchical gated deep memory network with position-aware.

II. THE PROPOSED APPROACH

In this section, we introduce the proposed model called hierarchical gated deep memory network with position-aware for aspect-based sentiment analysis. The overall architecture of our model is shown in Figure 1.

In Figure 1, suppose that a sentence consists of n words $s = \{w_i\}_{i=1}^{n}$, and an aspect term consists of m words $a = \{w_j^a\}_{j=i}^{j=m+i-1}$. The objective of our model is to predict the sentiment polarity of the sentence over the aspect. As shown in Figure 1, our model primarily includes three modules: encoder module, memory update module and output module. In this section, we will detail the internals of each module.

A. POSITION-AWARE WORD EMBEDDING LAYER

The word embedding layer has two parts: the word embedding and position embedding. Suppose $L \in R^{d\times|V|}$ is an embedded lookup matrix generated by an unsupervised method such as Glove [18] or Word2Vec [19], where d is the dimension of word embedding and V is the size of word vocabulary. After the embedded lookup operation, the word vector sequences of sentence $v = \{v_1, v_2, \ldots, v_n\}$ and the word vector sequence of aspect terms $v^a = \{v_1^a, v_2^a, \ldots, v_m^a\}$ are obtained respectively. The position embedding operation of the context with respect to aspect term is as follow. If a word appears in an aspect term, its positional index is marked as 0, while positional index of other word is represented as the relative distance from the current aspect term. The relative offset between contextual words and target is defined by the follow equation:

$$pos_i = \begin{cases} |i - j_s|, & i < j_s \\ 0, & j_s \le i \le j_e \\ |i - j_e|, & i > j_e \end{cases} \tag{1}$$

where j_s is the index of the first word of aspect term, j_e is the index of the last word of aspect term. pos_i is relative offset between the *i-th* word of the sentence and the current aspect term. Then the position embedding vector of the word is obtained by looking randomly initialized position embedding lookup matrix $P \in R^{d_p\times N}$. Where d_p is the dimension of position embedding and N is length of the sentence, position embedding vector is noted as $p = \{p_i\}_{i=1}^{n}$. Finally, the position embedding vector and the word vector are connected to get the word vector based on position-aware. It is noted as $x_i = [v_i; p_i]$.We then run two independent Bi-LSTM layer to get the hidden state representation of the aspect term and the hidden state representation of the context, respectively. Context hidden state consists of forward hidden state $\vec{h}_i \in R^{d_h}$ and backward hidden state $h_i \in R^{d_h}$. The hidden state of aspect term is similar to the hidden state of sentence.

B. INFORMATION INTERACTION LAYER

The word embedding layer has two parts: sentence-to-aspect attention mechanism and information fusion operation. For the former part, we can obtain the different attention weights of the words in aspect term based on context. For the later

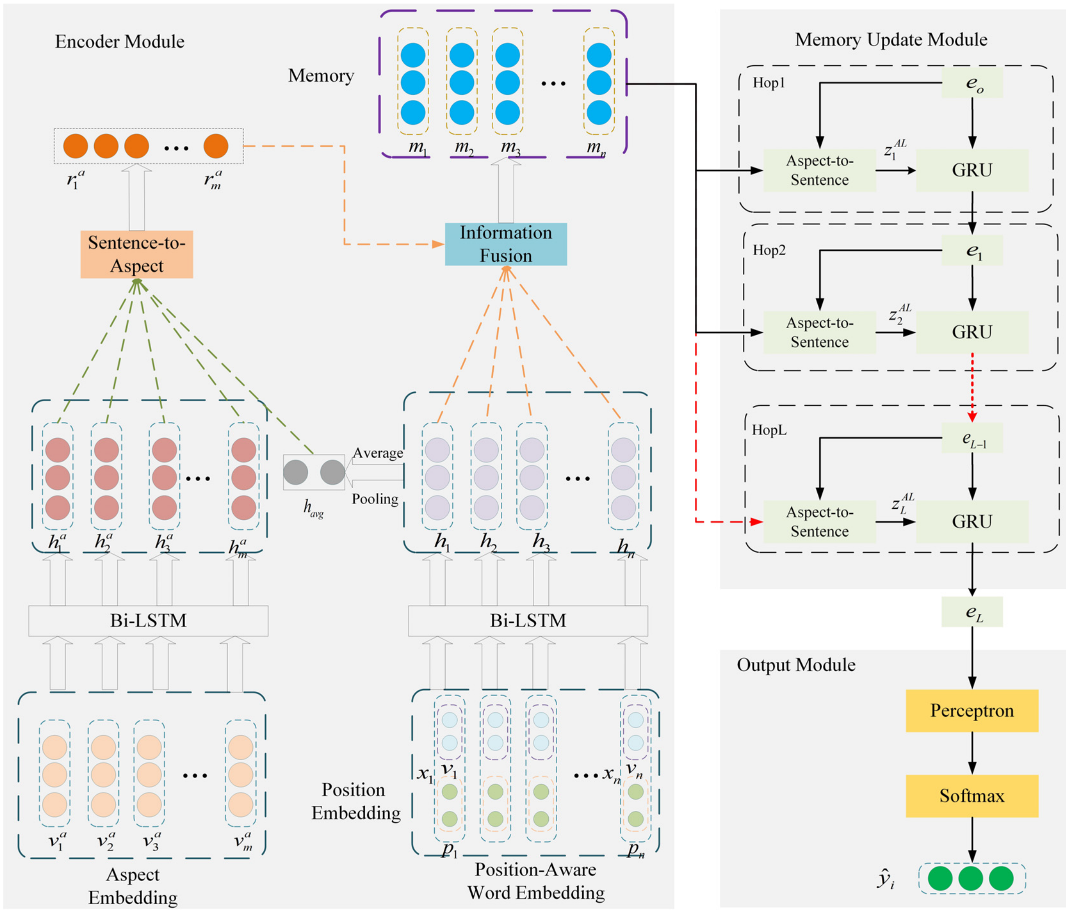

FIGURE 1. The architecture of our model.

part, we fuse aspect information into the context so that the generated sentence representation contains more information related to the target.

Sentence-to-aspect attention: we make use of an average representation of the context to obtain better representation of aspect terms. Firstly, the average value of hidden state of sentence is obtained by average pooling. Then, we define a correlation function f through the aspect term hidden state and the average value of sentence's hidden state. The correlation function is used as a weight that denotes the importance of a word in aspect term. Then, with the correlation function as the input, every word's attention weight in the aspect term is calculated by the softmax function. Finally, the fine-grained representation of aspect term is obtained by weighted sum of hidden states of aspect words. The process can be formulated as follows:

$$\bar{h} = pooling([h_1, h_2, \ldots, h_n]) \tag{2}$$

$$f(h_i^t, \bar{h}) = \tanh(W_t \cdot h_i^t \cdot \bar{h} + b_t) \tag{3}$$

$$\alpha_i^t = \frac{\exp(f(h_i^t, \bar{h}))}{\sum_{j=1}^{m} \exp(f(h_j^t, \bar{h}))} \tag{4}$$

$$r^t = \sum\nolimits_{i=1}^{m} \alpha_i^t \cdot h_i^t \tag{5}$$

where, W_t is the weight matrix and b_t is the bias. r^t is the semantic vector that represents the input aspect term.

Information Fusion: How to focus on the parts of the sentence that are more relevant to a given target of sentiment expression. Many previous models choose to embed the target information directly into the word vector or hidden state sequence of the sentence, such as ATAELSTM [10], RAM [12], AF-LSTM [14], and HAPN [20]. The experimental results also prove that this method of embedding target information can enrich the semantic expression of sentences. The process of information fusion is as follows:

$$m_i = [h_i; h_i \odot r^t] \tag{6}$$

where [;] represents vector connection, $\odot$ represents Hadamard product. m_i is the final output of the model encoder module, called external memory $M = \{m_1, m_2, \ldots, m_n\}$.

C. MEMORY UPDATE MODULE

The module consists of several computation layers. Each layer contains an aspect-to-sentence attention mechanism and a recurrent gated memory update mechanism. Attention mechanism computes the semantic encoding of external memory and memory update mechanism obtains the final representation of the sentence for a given aspect.

Aspect-to-sentence attention: as shown in Figure 2, the inputs to the attention mechanism includes external memory $M = \{m_1, m_2, \ldots, m_n\}$ and upper memory unit e_{l-1} and the

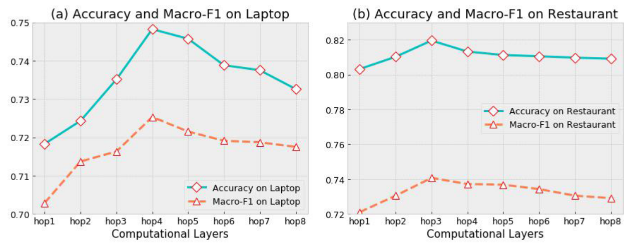

FIGURE 2. **Effects of computation layers.**

attention representation of the aspect term. The calculation process is similar to that of calculating the attention representation of the aspect term. The calculation process is as follows:

$$\beta_i^l = \tanh(W_l^{AL}[m_i, e_{l-1}, r^t]) + b_l^{AL} \tag{7}$$

$$\alpha_i^l = \frac{\exp(\beta_i^l)}{\sum_{j=1}^{n}\exp(\beta_j^l)} \tag{8}$$

$$z_l^{AL} = \sum\nolimits_{i=1}^{n} \alpha_i^l m_i \tag{9}$$

where, W_l^{AL}, b_l^{AL} are the weight matrix and bias, respectively. z_l^{AL} is the current information attended from external memory $M = \{m_i\}_{i=1}^{n}$(AL is acronym of attention layer).

Recurrent gated memory update mechanism: in this paper, GRU network is used as the gate mechanism of deep memory network to control the update of memory unit at each computing layer. This process is mainly accomplished by using two gates of GRU, namely the update gate and the reset gate. The purpose of the reset gate is to control how much irrelevant information is discarded from the hidden state at the previous time. The update gate determines how much information is entered into the current time memory unit between the previous time memory unit and the current time candidate memory unit. There are two main advantages: 1) correctly extract the information related to target sentiment expression from its external memory; 2) appropriately produces input information for sentiment classification. The process of updating current layer memory unit e_l is as follows:

$$r = \sigma(W_r z_l^{AL} + U_r e_{l-1}) \tag{10}$$

$$z = \sigma(W_z z_l^{AL} + U_z e_{l-1}) \tag{11}$$

$$\tilde{e}_l = \tanh(W_e z_l^{AL} + W_g(r \odot e_{l-1})) \tag{12}$$

$$e_l = (1 - z) \odot e_{l-1} + z \odot \tilde{e}_l \tag{13}$$

where, $W_r, W_z, W_e, W_g \in R^{H\times d}$, $U_r, U_z \in R^{H\times H}$ and H is the hidden dimension of GRU. The initial memory unit is zero vector. It can be seen from the above formula: the update of the current layer memory unit is closely related to the previous layer memory unit and the attention representation of the external memory. Finally, the memory unit e_l is passed to the next computation layer and calculated according Formulas 10-13 until the last computation layer is completed. The output memory unit e_{last} at this time is the final sentence representation.

D. MEMORY UPDATE MODULE

After N-time update on the memory unit, the final memory unit e_{last} serves as the feature and is fed into a perceptron and softmax function to predict the distribution probability of the emotional polarity of a given target. And take the maximum probability as the predicted sentiment category. The calculation process is as follows:

$$p = W_p e_{last} + b_p \tag{14}$$

$$y_i = \frac{\exp(p_i)}{\sum_{j=1}^{C}\exp(p_j)} \tag{15}$$

where C is the number of sentiment category. y_i denotes the probability of predicting the i-th sentiment category.

The model is trained by minimizing the cross entropy plus an L2 regularization term:

$$L = -\sum\nolimits_i^D \sum\nolimits_j^T y_i^j \log \hat{y}_i^j + \lambda \left\|\theta\right\|^2 \tag{16}$$

where D is the total number of training data y_i^j represents the true probability that j-th sample belongs to the i-th sentiment category and y^i represents the prediction probability that the j-th sample belongs to the i-th sentiment category. λ is the coefficient of $L2$ regularization term and θ is all parameters to be trained in our model. In this paper, adaptive moment estimation (Adam) [21] is used to optimize the model. We also adopt dropout strategy [22] and early stopping to ease overfitting.

III. EXPERIMENTS

A. EXPERIMENT SETTING

We conduct experiments on two datasets. The two datasets are from SemEval-2015 task [23], one is the laptop datasets and another is the restaurant datasets. The datasets consists of only the training set and the test set. It's worth noting that the original datasets contains a total of four emotion categories {conflict, neutral, positive, negative}, conflict means that a certain aspect of a sentence, and the sentence sentiment can express both positive and negative. Due to too little sample data of this category, data imbalance will occur. Therefore, in the stage of data preprocess, we remove the sentences labeled as conflict in the datasets. The statistical distribution of laptop and restaurant datasets after preprocessing is shown in Table 1.

The evaluation metric is accuracy and macro-f1, which are similar to Ma's paper [24].

In our work, the dimension of word embedding vectors and hidden state vector are 300. We use the pretrained Glove to initialize the word embedding. For out-of-vocabulary words [25] and weight matrices are randomly initialized by a uniform distribution U (-0.1, 0.1). The initial value of all bias are zero. All parameters in the model are randomly initialized by uniform distribution U (-0.1, 0.1). Learning rate and L2 regularization are set as -0.001. Dropout rate is set to 0.5.In addition, the max epoch and batch size are set to 25. The computation layers of the model are set to 3(Restaurant) and 4(Laptop), respectively.

TABLE 1. Units for magnetic properties.

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

B. COMPARED METHODS

We compare our model with the following methods:

1) Majority: assign the majority sentiment polarity of the training dataset to each instance of the test dataset.

2) LSTM [9]: the model is used to calculate the hidden state of the sentence, and the last hidden state is taken as the expression of the sentence, which is taken into input by a softmax function for sentiment classification.

3) TDLSTM [9]: the model employs two direction LSTM networks to abstract the information before and after the target. Then, the vector average is used to splicing the hidden state of the last step LSTM network output as the final expression of the sentence about the aspect.

4) ATAELSTM [10]: the model first embeds aspect information into each of the word vectors, then employs LSTM network to calculate the hidden state of sentence, and finally employs attention mechanism to calculate the semantic representation of the sentence.

5) DMN [11]: the model explicitly captures the importance of each context word about a given aspect through multiple computation layers, each computation layers is a neural attention model over an external memory.

6) IAN [13]: the model learns interaction between aspect and context through coarse-grained attention mechanisms.

7) RAM [12]: the structure of the model is similar to DMN. The difference is that the model first calculates the external memory through BILSTM, and then recurrent attention mechanism is used to nonlinear update the memory unit.

8) PosATT-LSTM [26]: the model not only takes into account the importance of each context word about aspect but also takes into account the importance of position information between the aspect and context.

9) AOA [15]: the model learns word-level interaction between aspect and context through fine-grained attention mechanisms.

C. MAIN RESULTS

The sentiment classification results of our model compared with other baseline methods are shown in Table 2. In our work, we conduct five experiments with all parameters unchanged, and take the average of accuracy and macro-f1 value as the final experimental result.

It can be seen from Table 2, our model achieves the best results in both accuracy and macro-f1 on restaurant and laptop datasets. On the laptop datasets, the accuracy and macro-f1 are 74.82% and 72.53%, respectively. On the restaurant datasets, the accuracy and macro-f1 are 81.95% and 74.07%, respectively. It obvious that Majority method gets the worst result, because it's just a simple statistic on datasets. It is obvious that all models based on LSTM network (LSTM, TDLSTM, ATAELSTM, IAN, RAM, AOA, PosATT-LSTM) have better sentiment classification ability than the Majority method. In addition, we find that all models with attention mechanisms significantly improve the ability of sentiment classification. Furthermore, AOA achieves the best performance among all comparison models, because AOA considers word-level interactions between the target and the context. This is similar to the attention mechanism used in our model. It is worth noting that the RAM model also achieves very good classification performance. This is related to RAM using multiple layers of attention to improve the ability of sentiment classification. In addition, we can see that considering the position information between the target and the context in aspect-based sentiment classification task can also improve the ability of the model to classify emotions, such as PosATT-LSTM and the model proposed in this paper.

TABLE 2. Experiment results on restaurant and laptop datasets.

Model	Laptop		Restaurant	
	Accuracy	F1	Accuracy	F1
Majority	53.45	33.33	65.00	33.33
LSTM[9]	66.45	62.79	74.28	64.71
TDLSTM[9]	68.13	65.43	75.60	66.73
ATAELSTM[10]	68.70	65.20	77.20	69.25
DMN[11]	70.33	64.10	78.16	70.23
IAN[13]	72.10	66.40	78.60	71.34
RAM[12]	74.49	71.35	80.23	70.80
PosATT-LSTM[26]	72.80	70.29	79.40	69.73
AOA[15]	74.50	72.10	81.20	73.40
Our-Model	74.82	72..53	81.95	74.07

TABLE 3. The performance of models with or without position embedding.

Datasets	Restaurant (%)	Laptop (%)
No-Position	81.32	74.43
Our-Model	81.95	74.82

D. EFFECTS OF POSITION EMBEDDING

In order to verify the efficiency and advantage of position embedding, we design the following model for comparison:

No-position: the model is a simplified version of our model, which does not consider the positional relationship between the target word and the context during the word embedding process.

In Table 3, we report the performance comparison of our model and No-Position. We use accuracy as an experimental indicator.

From Table 3, we can observe that our model performs better than No-Position model. Our model achieves improvement of 0.63% and 0.39% on accuracy respectively on the two datasets. This indicates that position information can effectively improve the performance of sentiment classification of the model.

E. EFFECTS OF ATTENTION MECHANISM

In order to verify the efficiency and advantage of the two attention mechanisms proposed in this paper, we design the following model for comparison:

TABLE 4. The performance of models with or without sentence-to-aspect and aspect-to-sentence attention mechanism.

Datasets	Restaurant (%)	Laptop (%)
No-SA-Attention	81.28	74.38
No-AS-Attention	80.37	73.96
Our-Model	81.95	74.82

No-SA-Attention: the model is a simplified version of our model, where the result of sentence-to-aspect attention is replaced with averaging the hidden state of aspect terms.

No-AS-Attention: the model is a simplified version of our model, where the result of aspect-to-sentence attention is replaced with semantic representation of external memory.

In Table 4, we report the performance comparison of our model, No-SA-Attention and No-AS-Attention. We use accuracy as an experimental indicator.

Table 4 presents the performance comparison of Our-Model, No-SA-Attention and No-AS-Attention. From Table 4, we can find that No-AS-Attention model performs the worst. And the accuracy of No-AS-Attention model is 80.37% and 73.96% on the two datasets. This indicates that the Aspect-to-Sentence attention in our model is very important to this task. In addition, we can see that No-SA-Attention achieves the accuracies of 81.78% and 74.58% on restaurant and laptop dataset respectively, which are 0.67% and 0.44% lower than the proposed model. This indicates that Sentence-to-Aspect attention mechanism in our model is effective to ABSA task.

F. EFFECTS OF COMPUTATION LAYERS

In order to explore the influence of the number of computation layers on the ability of sentiment classification of models. With other parameters unchanged, we increment the number of computation layers of the model from one to eight. And we record the accuracy and Macro-F1 on Restaurant and Laptop datasets, respectively. The results are shown in Figure 2.

It can be seen from Figure 2 multiple computation layers can effectively improve the sentiment classification ability of our model. However, the performance of our model doesn't increase monotonously with the increase of the number of computational layers. In addition, the number of computation layers required by the model to achieve the best classification performance on the laptop and restaurant datasets is different. For example, in the restaurant datasets, when the number of computation layers is 3, the model achieves the best classification performance, while on laptop datasets, when the number of computation layers is 4, the classification performance of the model is the best. Therefore, reasonably setting the number of computation layers can effectively improve the sentiment classification ability of our model.

G. CASE STUDY

In order to explore whether the attention mechanism proposed in this paper can effectively focus on the most indicative words of the aspect terms and context. We visualize attention weights of the sentence-to-aspect attention mechanism (Formula 4) and aspect-to-sentence attention mechanism (Formula 8). In this paper, we visualize the attention weight of the sentence "This is one great place to eat pizza but not a good place for take-out pizza". The results of attention weights visualization are shown in Figure 3, where each color block in the Figure 3 represents the weight of the word. The deeper of the color, the more importance of the word in sentiment classification.

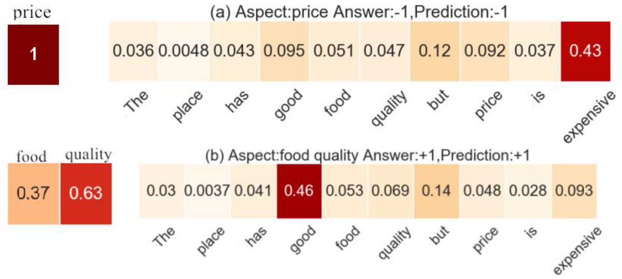

FIGURE 3. Visualization of attention weight.

Figure 3 shows the attention weights of aspect terms on the left and the attention weights of context on the right. It can be found in Figure 3(a), that the word "expensive" has the greatest weight when the aspect is "price". Therefore, the model infers that the sentiment polarity of the sentence about "price" is negative. In Figure 3(b), for the aspect term "food quality", "quality" is assigned more attention than "food", indicating that "quality" is more important in the aspect term. Furthermore, in Figure 3(b), that the word "good" is more important than other contextual words when inferring the sentiment polarity, from which it can be inferred that the sentence's sentiment polarity about "food quality" is positive. From the above analysis, it can be concluded that the sentence-to-aspect attention mechanism and aspect-to-sentence attention mechanism proposed in this paper can effectively focus on the most indicative parts of the aspect term and context that are related to sentiment expression.

H. COMPARISION EXPERIMENT OF MEMORY UNIT UPDATE MECHANISM

In order to explore whether the recurrent gated memory update mechanism can effectively improve the sentiment classification ability of the model. We use the other three memory unit update mechanisms to conduct comparison experiments of sentiment classification. The comparison method is as follows:

1) Linear memory unit update mechanism:

$$e_l = e_{l-1} + z_l^{AL} \tag{17}$$

2) Weighted summation memory unit update mechanism:

$$e_l = W_z z_l^{AL} + W_e e_{l-1} \tag{18}$$

3) Long short-term gated memory unit update mechanism:

$$e_l = LSTM(z_l^{AL}, e_{l-1}) \tag{19}$$

4) Recurrent gated memory unit update mechanism

$$e_l = GRU(z_l^{AL}, e_{l-1}) \tag{20}$$

TABLE 5. The sentiment classification accuracy of our model with four memory unit update mechanisms.

Memory unit update mechanism	Laptop	Restaurant
Linear update mechanism	73.92	80.57
Weight summation update mechanism	74.08	81.05
Long short-term gated update mechanism	74.89	82.03
Recurrent gated update mechanism	74.82	81.95

where z_l^{AL} is the semantic representation of the current layer external memory computed by the attention mechanism and e_{l-1} is memory unit of upper layer.

Table 5 shows the accuracy rate of sentiment classification on laptop datasets and restaurant datasets with our model and four memory unit update mechanisms.

From Table 5, we can get that linear memory unit update mechanism performs worst in sentiment classification. Its accuracy is only 73.92% and 80.57%. This is because the linear method cannot effectively utilize the results of multiple attention computations. Weighted summation memory unit update mechanism is only better than the simple linear update mechanism. It just gives different weights to the memory unit and external memory, which leads to a slight improvement in classification accuracy. The long short-term gated update mechanism has the best accuracy, which is 74.89% and 82.03%, respectively. And the accuracy of recurrent gated update mechanism is close to long short-term memory unit update mechanism, which is 74.82% and 81.95%, respectively. However, since GRU has fewer parameters and does not need to introduce additional unit states, it is a better choice to use the recurrent gated memory unit update mechanism from the perspective of saving computing space and time.

I. ERROR ANALYSIS

We carry out an error analysis of our model on test datasets, and find that most of the errors could be summarized as follows. Firstly, when the sentiment polarity of the target is neutral, the accuracy of sentiment classification is not too high. There are usually two factors for this error: 1) The sentiment polarity of the target is affected by the sentiment polarity of the other targets in the sentence. 2) Since the proportion of neutral samples in the training datasets is smaller than other polarity, when the predicted sentence is an objective expression, the model tends to be biased towards other emotional polarities. The second error is non-compositional sentiment expression. For example, ‘‘sushi is to die for’’, where the aspect is ‘‘sushi’’ and the sentiment word is ‘‘die for’’. However, our model employs a single context word as the basic unit of computation, therefore, the erroneous judgment of the sentiment polarity of sushi is negative. The third factor is the uncommon idiom problem. For example, ‘‘the service was on point’’ where the aspect is ‘‘service’’ and the sentiment expression is ‘‘on point’’. However, the model in this paper cannot understand the meaning of ‘‘on point’’.

IV. CONCLUSION AND FUTURE WORK

In this paper, we propose a hierarchical gated deep memory network with position-aware for aspect-based sentiment analysis. One of the core ideas of our model is to embed the position information as a feature in the sentence representation. In addition, we further propose sentence-to-aspect attention mechanism and aspect-to-sentence attention mechanism to take into account the fine-grained interaction between aspect terms and contexts, so as to better represent aspect terms and contexts. The experiment results on restaurant and laptop datasets show that our model has better sentiment classification performance than the comparison models. We also demonstrate the validity of the internal structure of our model by other experiments.

Although the model proposed in this paper has great potentials for aspect-based sentiment analysis. However, this paper ignores the influence of different language models in generating sentence representation, such as ELMo [27], Bert [28], fastText [29] and other language models. Therefore, in the future, this is a worthy research direction. In addition, by observing the misclassified samples, we find that the accuracy of the model to predict neutral polarity is not high. The second error is non-compositional sentiment expression. The third error is idiomatic expressions. Therefore, we will introduce special methods to solve these three problems in the future, such as increasing the number of neutral sample or adding a label smoothing regularization, and introducing the syntactic structure. Finally, in the future we should pay attention to the work of word polarity disambiguation.

REFERENCES

[1] B. Pang and L. Lee, ‘‘Opinion mining and sentiment analysis,’’ *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, pp. 1–135, 2008.
[2] E. Cambria, ‘‘Affective computing and sentiment analysis,’’ *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.
[3] S. Kim and F. Flavius, ‘‘Survey on aspect-level sentiment anaysis,’’ *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Oct. 2016.
[4] B. Pang, L. Lillian, and V. Shivakumar, ‘‘Thumbs up?: Sentiment classification using machine learning techniques,’’ in *Proc. ACL Conf. Empirical Methods Natural Lang. Process.*, vol. 10, Jul. 2002, pp. 79–86.
[5] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, ‘‘NRC-Canada-2014: Detecting aspects and sentiment in customer reviews,’’ in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 437–442.
[6] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, ‘‘Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis,’’ *Cognit. Comput.*, vol. 10, no. 4, pp. 639–650, Aug. 2018.
[7] H. H. Do, P. Prasad, A. Maag, and A. Alsadoon, ‘‘Deep learning for aspect-based sentiment analysis: A comparative review,’’ *Expert Syst. Appl.*, vol. 118, pp. 272–299, Mar. 2019.
[8] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, ‘‘Target-dependent twitter sentiment classification,’’ in *Proc. 49th Annu. Meeting Assoc. Comput. Lingustics.*, Jun. 2011, pp. 151–160.
[9] D. Tang, B. Qin, X. Feng, and T. Liu, ‘‘Effective LSTMs for target-dependent sentiment classification,’’ in *Proc. COLING 26th Int. Conf. Comput. Linguistics*, Dec. 2016, pp. 3298–3307.
[10] Y. Wang, M. Huang, X. Zhu, and L. Zhao, ‘‘Attention-based LSTM for aspect-level sentiment classification,’’ in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Nov. 2016, pp. 606–615.
[11] D. Tang, B. Qin, and T. Liu, ‘‘Aspect level sentiment classification with deep memory network,’’ in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Nov. 2016, pp. 214–224.
[12] P. Chen, Z. Sun, L. Bing, and W. Yang, ‘‘Recurrent attention network on memory for aspect sentiment analysis,’’ in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 452–461.
[13] D. Ma, S. Li, X. Zhang, and H. Wang, ‘‘Interactive attention networks for aspect-level sentiment classification,’’ in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4068–4074.

[14] Y. Tay, L. Tuan, and S. Hui, "Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 5956–5963.
[15] B. Huang, Y. Qu, and K. Carley, "Aspect level sentiment classification with attention-over-attention neural networks," in *Proc. Conf. Social Comput.*, 2018, pp. 197–206.
[16] Z. Li, Y. Wei, Y. Zhang, X. Zhang, and X. Li, "Exploiting coarse-to-fine task transfer for aspect-level sentiment classification," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Sep. 2018, pp. 4253–4260.
[17] K. Chen and Q. Huo, "Training deep bidirectional LSTM acoustic model for LVCSR by a context-sensitive-chunk BPTT approach," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 7, pp. 1185–1193, Jul. 2016.
[18] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[20] L. Li, Y. Liu, and A. Zhou, "Hierarchical attention based position-aware network for aspect-level sentiment analysis," in *Proc. 22nd Conf. Comput. Natural Lang. Learn.*, 2018, pp. 181–189.
[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[22] G. Hinton, N. Srivastava, A. Krizhevsky, R. R. Salakhutdinov, and I. Sutskever, "Improving neural networks by preventing co-adaptation of feature detectors," *Comput. Sci.*, vol. 3, no. 4, pp. 212–223, Jul. 2012.
[23] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval)*, 2015, pp. 486–495.
[24] Y. Ma, H. Peng, and E. Cambria, "Target aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5876–5883.
[25] I. Bazzi and R. James, "Modeling out-of-vocabulary words for robust speech recognition," Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep., Jan. 2002.
[26] J. Zeng, X. Ma, and K. Zhou, "Enhancing attention-based LSTM with position context for aspect-level sentiment classification," *IEEE Access*, vol. 7, pp. 20462–20471, Feb. 2019.
[27] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., (Long Papers)*, vol. 1, 2018, pp. 2227–2237.
[28] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, May 2019, pp. 4171–4186.
[29] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, Short Papers*, vol. 2, 2017, pp. 427–431.

ZEBING JIA received the B.E. degree in computer science and technology from Sun Yat-sen University, China, in 2015. He is currently pursuing the M.S. degree with the School of Software Engineering, Xi'an Jiaotong University. His main research interests include natural language processing, sentiment analysis, and data analysis.

XIUXIU BAI (Member, IEEE) received the B.S. and Ph.D. degrees from Xi'an Jiaotong University, in 2009 and 2016, respectively. She visited Edinburgh University, from 2017 to 2018. She is currently an Assistant Professor with the School of Software Engineering, Xi'an Jiaotong University. Her research interests include computer vision and visual neuroscience.

SHANMIN PANG (Member, IEEE) received the B.S. degree from Shaanxi Normal University, in 2006, the M.S. degree from Capital Normal University, in 2009, and the Ph.D. degree from Xi'an Jiaotong University, in 2015. He is currently an Associate Professor at the School of Software Engineering, Xi'an Jiaotong University. His research interests include pattern recognition, computer vision, and image processing. He won the Best Application Paper Award at ACCV 2012 Conference.