Attention Capsule Network for Aspect-Level Sentiment Classification

Yu Deng[1], Hang Lei[1], Xiaoyu Li[1*], Yiou Lin[1], Wangchi Cheng[2], and Shan Yang[3]
[1] School of Information and Software Engineering
University of Electronic Science and Technology of China
Chengdu 610054, China
[e-mail: 411180435@qq.com, hlei@uestc.edu.cn, xiaoyuuestc@uestc.edu.cn, lyoshiwo@gmail.com]
[2] Institute of Logistics Science and Technology
Beijing 100166, China
[e-mail: chengwc_szph@163.com]
[3] Department of Chemistry, Physics, and Atmospheric Sciences
Jackson State University
Jackson, MS 39217, USA
[e-mail: Shan.Yang@jsums.edu]
[*]Corresponding author: Xiaoyu Li

Received February 8, 2021; revised March 5, 2021; March 24, 2021; published April 30, 2021

Abstract

As a fine-grained classification problem, aspect-level sentiment classification predicts the sentiment polarity for different aspects in context. To address this issue, researchers have widely used attention mechanisms to abstract the relationship between context and aspects. Still, it is difficult to effectively obtain a more profound semantic representation, and the strong correlation between local context features and the aspect-based sentiment is rarely considered. In this paper, a hybrid attention capsule network for aspect-level sentiment classification (ABASCap) was proposed. In this model, the multi-head self-attention was improved, and a context mask mechanism based on adjustable context window was proposed, so as to effectively obtain the internal association between aspects and context. Moreover, the dynamic routing algorithm and activation function in capsule network were optimized to meet the task requirements. Finally, sufficient experiments were conducted on three benchmark datasets in different domains. Compared with other baseline models, ABASCap achieved better classification results, and outperformed the state-of-the-art methods in this task after incorporating pre-training BERT.

Keywords: Capsule Network, Convolutional Neural Network, Aspect-level Sentiment Classification, Natural Language Processing, Attention Mechanism

This work was partially supported by the National Key R&D Program of China（Grant No.2018YFA0306703）.

1. Introduction

Due to the rapid development of the Internet and mobile communication technologies, social networks and electronic commercial websites have become huge public information distribution centers. The use of massive amounts of data to analyze people's emotions and opinions has significant scientific and social value. Sentiment analysis or opinion mining is a computational study of people's opinions, emotions, evaluations, and attitudes about products, services, organizations, individuals, problems, events, topics, and their attributes [1]. As a sub-task of sentiment analysis, aspect-level fine-grained sentiment analysis can effectively explore the deep emotional features in the context of specific objects. It has recently gained much popularity in this field [2].

Since deep learning has been widely used in natural language processing (NLP) tasks, it has achieved great success in the NLP field. Compared with traditional machine learning algorithms, deep learning does not rely on artificially constructed features and has feature self-learning capabilities [3-5]. It is very suitable for the abstract, high-dimensional, and complex characteristics of language text. Various models based on deep learning framework are emerging to deal with problems in text sentiment analysis [6-8]. Especially, the attention-based deep learning model is not only effective but also has good interpretability in aspect-level sentiment analysis [9-11].

Previous studies also showed some remaining gaps. First, although the attention mechanism can give additional weight information to input and hidden features for different sentiment aspects, the mining of the inline relationship between aspects and the context remains insufficient, creating difficulty for the model to effectively obtain a deeper semantic representation. The situation is more critical with the presence of multiple aspects in the same context. Second, in aspect-level sentiment analysis, all contexts are usually inputted indiscriminately to get a sentence representation, while the context far away from the aspect may negatively affect the classification result.

In this paper, we proposed an attention capsule network for aspect-level sentiment classification (ABASCap) to solve the above problems. ABASCap combined multi-attention mechanism with capsule network to effectively model the internal correlation of context and the relationship between aspects and local context. Finally, the model was evaluated on the semeval2014 and Twitter datasets. The experiment proved the effectiveness of ABASCap. After the integration with the pre-training BERT, the model outperformed the state-of-the-art methods in the task.

The main contributions in this work are listed below.

- We proposed a special capsule network for aspect-level sentiment analysis, clarifying input and output of the model and its intermediate data processing. In this model, the multi-head self-attention mechanism was improved to capture the internal semantic structure of short texts and the relationship between aspects and the context features;
- The local context window (LCW) was defined to clarify the local context related to the aspect, and a local context mask (LCM) mechanism based on LCW was designed to model the strong relevance between the aspect and local context;
- Capsule network was used to classify the polarity of aspect-level sentiment; the routing algorithm and activation function were improved according to the characteristics of the task, so that the model could obtain richer text semantic information;
- Comparative experiments were conducted with various baselines and the latest methods.

The experimental data was used to qualitatively analyze the structure of ABASCap and verify its effectiveness on each data set.

2. Related Work

In the early days, sentiment analysis tasks were mostly processed by traditional machine learning methods relying on feature engineering, thus taking a long time to collect, sort, and abstract background knowledge. After its emergence, the artificial neural network has rapidly replaced machine learning and become the mainstream of the NLP field. The following is a focused discussion on aspect-level sentiment analysis based on deep learning.

2.1 Attention Mechanism

Attention mechanism was first proposed in image recognition research, which allowed the model to effectively focus on specific local information and mine deeper feature information [12, 13]. Subsequently, in NLP, the attention mechanism was verified to make feature extraction more efficient. At present, many researchers have applied the attention mechanism to aspect-level sentiment classification and achieved good results. In research [14], the intermediate states of target content and sequence were spliced in the LSTM network, and attention-weighted output was calculated, which effectively solved the sentiment polarity problem of context based on different aspects. Tang et al. [15] proposed a multi-hop attention memory network model. It calculated attention value based on content and location, used external storage units to save the weight information of aspects, and obtained deeper emotional semantic information through overlay calculation. Chen et al. [16] used a Bi-direction LSTM network to construct a memory unit for improving the multi-hop attention network. Meanwhile, the memory content was weighted to capture sentiment features and eliminate noise interference. Ma et al. [17] proposed an interactive attention network (IAN), which used the attention mechanism to obtain important information from the context according to the aspect and interactive information in context to supervise the modeling process, thereby improving the accuracy of sentiment polarity prediction.

Innovative structures have continually been emerging to optimize the performance of attention mechanism in NLP tasks and make the model more interpretable. Vaswani et al. [18] proposed a Transformer framework to replace CNN and RNN architecture, which achieved state-of-art results in machine translation. Multi-head attention mechanism and self-attention were proposed for the first time in the Transformer structure. It exclusively used the attention mechanism to model the global dependence of input and output so that the model can learn feature information in different representation subspaces, thereby generating more semantically relevant text representations. Ambartsoumian et al. [19] analyzed the characteristics of the self-attention network model, proposed two ways of combining multi-head attention and self-attention, and discussed its effectiveness in sentiment analysis. Letarte et al. [20] proposed a flexible and interpretable text classification model based on a self-attention network, which could effectively improve the accuracy of sentiment classification. Song et al. [21] applied multi-headed self-attention to aspect-level sentiment analysis and proposed an attention encoding network (AEN) to obtain the interaction and semantic information between each word and the context.

2.2 Capsule Network

In 2017, Sabour et al. [22] first proposed a capsule network in image processing, which attracted great attention and provided a new research direction. The capsule is a group of neurons that capture various parameters of specific features, including the possibility of outputting features. The capsule network uses vector capsules as input and output and dynamic routing algorithms to aggregate lower capsules to higher ones. The output vector of capsule is

called the activity vector. The probability of feature detection is represented by the length of the activity vector, and the direction of the vector represents classification attribute. Yang et al. [23] used the capsule network for cross-domain text classification. It achieved the acceleration of model training by improving the dynamic routing algorithm and compressing the capsule, and for the first time, verified the transfer learning ability of capsule network in text classification. Wang et al. [24] designed a capsule network model, which could complete target detection while solving sentiment classification task. The capsules in the model communicated with each other through the RNN network and the model obtained the most advanced classification results on the selected benchmark data set. Chen et al. [25] proposed a capsule network model based on transfer learning, which could transfer knowledge in other corpus to aspect-level sentiment classification. The model used an aspect routing algorithm to encapsulate sentence-level representations into primary semantic capsules. The model extended the dynamic routing algorithm to adaptively merge semantic and class capsules under the framework of transfer learning. Kim et al. [26] proposed a capsule network used for text classification, and simplified the dynamic routing algorithm, which effectively reduces the computational complexity.

3. Hybrid Capsule Network based on Attention Mechanism

We fuse attention mechanism with the capsule network to construct ABASCap model, which can learn the deep interactive relationship between context and aspects. The overall structure of ABASCap is shown in **Fig. 2**, including embedding layer, feature extraction layer, attention coding layer, primary capsule layer, and classification capsule layer. This section will describe the implementation of ideas and details in the model.

3.1 Task Definition

Given a context sequence $\boldsymbol{s} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n\}$ composed of n words, and an aspect sequence $\boldsymbol{t} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k\}$ composed of k aspects, where $\boldsymbol{a}_i = \{\boldsymbol{w}_i, \boldsymbol{w}_{i+1}, \ldots, \boldsymbol{w}_{i+m-1}\}$ is subsequences of $\boldsymbol{s}$. The aspect-level fine-grained sentiment analysis is to classify sentences based on different aspects, expressed as (1), where f_{polar} denotes a nonlinear transformation function.

$$\text{polarity} = f_{\text{polar}}(\boldsymbol{s}, \boldsymbol{a}_i) \tag{1}$$

Example 1: good location but the environment is terrible!

Example 2: I bought a mobile phone. The screen definition is amazing but the screen size is too small.

Fig. 1. Examples of short text comment text

The two example sentences in **Fig. 1** are short texts of customer reviews on products. For Example1, there are two aspect entities, “location” and “environment”. It is clear that “good” expresses the customer’s positive emotions for “location”, while “terrible” expresses the customer’s negative feelings for “environment”. Example 2 contains two aspect entities composed of two words, “screen definition” and “screen size”. Customers also express the opposite emotional polarity for these two entities through “amazing” and “small”. In the same context, people may have different sentiment expressions for various aspects, making the aspect-level sentiment analysis more complicated and difficult.

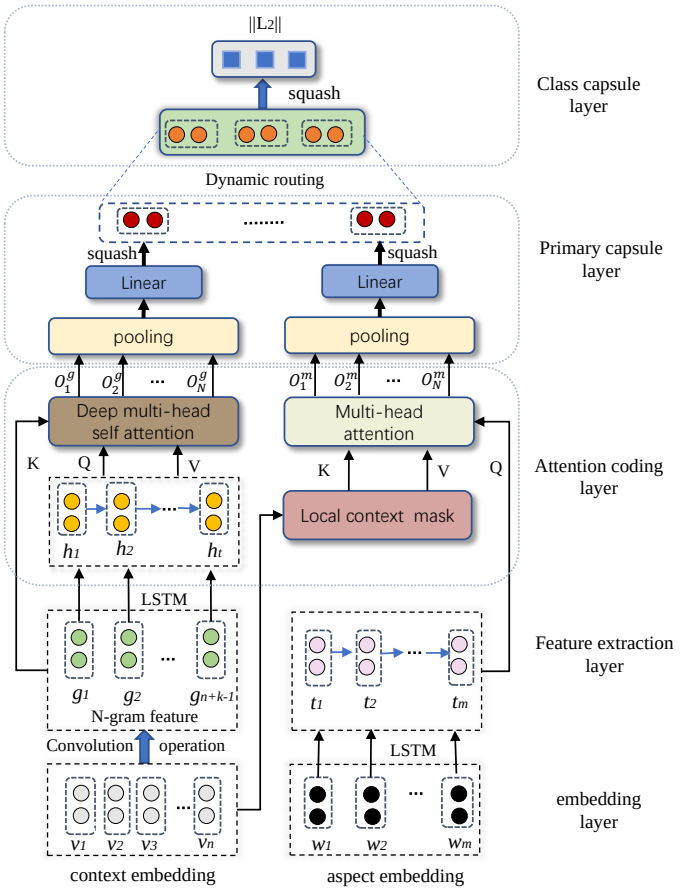

Fig. 2. The architecture of ABASCap

3.2 Embedding Layer

In this layer, a context sequence containing n words can be transformed into $\boldsymbol{S}$= {$\boldsymbol{v}_1$, $\boldsymbol{v}_2$, …, $\boldsymbol{v}_n$}, where $\boldsymbol{v}_i \in \mathbb{R}^d$ is the d-dimensional vector representation of the i-th word, and $\boldsymbol{S}$ is the input word vector matrix of the sentence, which is called context embedding. Correspondingly, the aspect containing m words in the sentence is mapped to $\boldsymbol{T}$ = {$\boldsymbol{v}_\alpha$, $\boldsymbol{v}_{\alpha+1}$, …, $\boldsymbol{v}_{\alpha+m-1}$}, namely aspect embedding, where $\boldsymbol{v}_j \in \boldsymbol{S}$ is a d-dimensional vector representation of the j-th word in aspect. This model uses two pre-training models, Glove, and Bert, as alternatives in the embedding layer.

3.3 Feature Extraction Layer

In this layer, the input features are further abstracted and processed. By using the N-gram model and introducing phrase features, the model input can be transformed from shallow to deep feature, which will have more semantic information and mine more deep interaction characteristics of the context. Generating N-gram features through CNN can effectively deal with the local relevance of the context while avoiding many probability statistics calculations for the feature weights in N-grams.

This layer applies multiple convolution operations to the input word vector matrix (context embedding) of sentences to obtain the corresponding N-gram features and generate a new

feature vector matrix $\boldsymbol{G} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_{n-k+1}\}$, where $\boldsymbol{G} \in \mathbb{R}^{(n-k+1)\times d_p}$, k is the size of one-dimensional convolution operation window, and d_p is the number of convolution kernels.

Besides, the LSTM network is applied to aspect embedding to model each word's dependence on the aspect, so as to dig out its implicit semantics. Finally, the hidden state $\boldsymbol{T}_h = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_m\}$ obtained by the LSTM network is used as the high-level feature representation of aspect embedding, where $\boldsymbol{T}_h \in \mathbb{R}^{m\times d_q}$, q is the hidden layer dimension of the LSTM network.

3.4 Attention Coding Layer

Based on the standard multi-head attention, we proposed a deep self-attention mechanism and a local context mask mechanism in this layer, so as to generate two kinds of output features, which could abstract higher-level feature representation for the next layer.

Specifically, the input matrices $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ correspond to the three important components of attention, namely query, key, and value, where $\boldsymbol{Q} \in \mathbb{R}^{n\times d_k}, \boldsymbol{K} \in \mathbb{R}^{m\times d_k}, \boldsymbol{V} \in \mathbb{R}^{m\times d_v}$. The standard attention calculation method in the general framework is as follows:

$$attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{soft}\max(f_{att}(\boldsymbol{Q}, \boldsymbol{K}))\boldsymbol{V} \tag{2}$$

Where f_{att} is the probability alignment function. In this work, the scaled dot product is used:

$$f_{att}(\boldsymbol{Q}, \boldsymbol{K}) = \frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}} \tag{3}$$

In multi-head attention, input is linearly mapped to different information subspaces through different weight matrices, and the same attention calculation is completed in each subspace to thoroughly learn the potential structure and semantics of the text. The i-head attention calculation process is as follows:

$$\boldsymbol{O}_i = attention(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V) \tag{4}$$

Where $\boldsymbol{W}_i^Q \in \mathbb{R}^{d_k\times \hat{d}_k}, \boldsymbol{W}_i^K \in \mathbb{R}^{d_k\times \hat{d}_k}, \boldsymbol{W}_i^V \in \mathbb{R}^{d_v\times \hat{d}_v}$. Finally, all heads are merged to produce multi-head attention output:

$$MHAtt(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = concat(\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{O}_3, \ldots \boldsymbol{O}_N) \tag{5}$$

Self-attention is to calculate the attention and find the inner relevance in the sequence. Assuming that the input sequence is $\boldsymbol{X}$, the multi-head self-attention calculation process is defined as follow:

$$MHSAtt(\boldsymbol{X}) = MHAtt(\boldsymbol{X}, \boldsymbol{X}, \boldsymbol{X}) \tag{6}$$

3.4.1 Deep Multi-head Self-Attention

Inspired by the work of Hao et al. [27], we proposed a deep self-attention mechanism by combining N-gram features with a multi-head self-attention model. The introduction of semantic features formed by the combination of adjacent words enables the multi-head attention to extract more hidden features in multi-dimensional information space, so as to obtain better prediction of the aspect sentiment polarity.

In deep multi-head self-attention, the input feature sequence is first abstractly transformed, and the obtained high-level representation is added to the model to extend the standard self-attention mechanism. In our model, the LSTM network is used to further abstract the input N-

gram feature sequence G. The specific calculation process of deep multi-head self-attention is as follows:

$$DMHSAtt(\boldsymbol{G}) = MHAtt(\boldsymbol{G}, \boldsymbol{H}, \boldsymbol{H}) \tag{7}$$

$$\boldsymbol{H} = LSTM(\boldsymbol{G}) \tag{8}$$

$$\boldsymbol{O}^{g} = DMHSAtt(\boldsymbol{G}) \tag{9}$$

Where $\boldsymbol{O}^{g} \in \mathrm{R}^{N\times(n-k+1)\times\hat{d}_{v}}$ is the output.

3.4.2 Local Context Mask

In aspect-level sentiment analysis, the semantic relationship between context sequence and aspects is closely related to its relative position. To emphasize the final impact of local context on sentiment polarity, a local context mask (LCM) mechanism was proposed to weight the context sequence. LCM mechanism can strengthen the influence of local context and weaken the noise of non-local context far away from the aspect.

In order to clarify the range of the local context in the input sequence, we proposed a local context window (LCW) to determine the local context boundary for a specific aspect. It is defined as follows:

$$LCW = \left|\beta - P_{\alpha}\right| \tag{10}$$

$$P_{\alpha} = \frac{1}{m}\sum_{i=\alpha}^{\alpha+m-1} i \tag{11}$$

Where β is the position of the specific word $\boldsymbol{v}_{\beta}$ on the boundary of the local context window, α is the position of the first word in the corresponding aspect sequence, and m is the length of the aspect sequence.

First, we constructed the mask matrix $\boldsymbol{W}^{\boldsymbol{m}} = \{\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots \boldsymbol{M}_n\}$:

$$\boldsymbol{M}_i = \begin{cases} \boldsymbol{E}, & |i - P_{\alpha}| \le LCW \\ \boldsymbol{0}, & |i - P_{\alpha}| > LCW \end{cases} \tag{12}$$

Where $\boldsymbol{E},\boldsymbol{0} \in \mathrm{R}^{d}$. Then the input context sequence $\boldsymbol{S}$ and the mask matrix $\boldsymbol{W}^{\boldsymbol{m}}$ are used to perform the vector element-wise product operation to implement the LCM mechanism, so as to change the feature vectors outside the local context window into zero vectors:

$$LCM(S) = S \odot W^{m} \tag{13}$$

This layer applies the LCM mechanism to the input context sequence for generating a weighted input feature sequence:

$$\boldsymbol{V}^{m} = LCM(\boldsymbol{S}) \tag{14}$$

Finally, it is combined with the upper layer input $\boldsymbol{T}_{\boldsymbol{h}}$ through the multi-head attention, so as to generate high-level feature representation:

$$\boldsymbol{O}^{m} = MHAtt(\boldsymbol{T}_{h}, \boldsymbol{V}^{m}, \boldsymbol{V}^{m}) \tag{15}$$

Where $\boldsymbol{O}^{m} \in \mathrm{R}^{N\times n\times\hat{d}_{v}}$.

3.5 Primary Capsule Layer

This layer is responsible for encapsulating the two parts of the multi-head attention output $\boldsymbol{O}^{g}$ and $\boldsymbol{O}^{m}$, so as to convert them into a vector capsule set used by the parent capsule layer. Global max pooling is used to compress the upper layer input in the horizontal direction, which can aggregate the multi-head attention output features in each corresponding subspace:

$$\boldsymbol{v}_i^o = global\ \max\ pooling(\boldsymbol{O}_i^c) \tag{16}$$

Where $\boldsymbol{O}_i^c \in \boldsymbol{O}^g \cup \boldsymbol{O}^m$, $\boldsymbol{v}_i^o \in \mathrm{R}^{\hat{d}_v}$. Then the compressed output is transformed linearly:

$$\boldsymbol{p}_i = squash(\boldsymbol{v}_i^o \boldsymbol{W}^c + \boldsymbol{b}^c) \tag{17}$$

Where $\boldsymbol{W}^c \in \mathrm{R}^{\hat{d}_v \times d_c}$, $\boldsymbol{b}^c \in \mathrm{R}^{d_c}$, $\boldsymbol{p}_i \in \mathrm{R}^{d_c}$, d_c is the dimension of capsules in primary capsule layer. The Squash function can compress the module length of the capsule vector to less than 1, which is used to represent the existence probability for a certain class. It is defined as follow:

$$squash(\boldsymbol{x}) = \frac{\|\boldsymbol{x}\|^2}{0.5 + \|\boldsymbol{x}\|^2} \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|} \tag{18}$$

In order to obtain rich features to model the sentence structure and semantic information of context sequence, the model adopts a variety of granularity lexical combinations (2-gram, 3-gram, and 4-gram) to expand the scale of multi-head attention information subspace and enrich semantic expression (as shown in Fig. 3). Finally, this layer outputs the primary capsule set $\boldsymbol{P}^c \in \mathrm{R}^{4N \times d_c}$ at the bottom of the capsule network, where d_c is the dimension of the primary capsule.

$$\boldsymbol{P}^c = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \ldots \boldsymbol{p}_{4\times N}\} \tag{19}$$

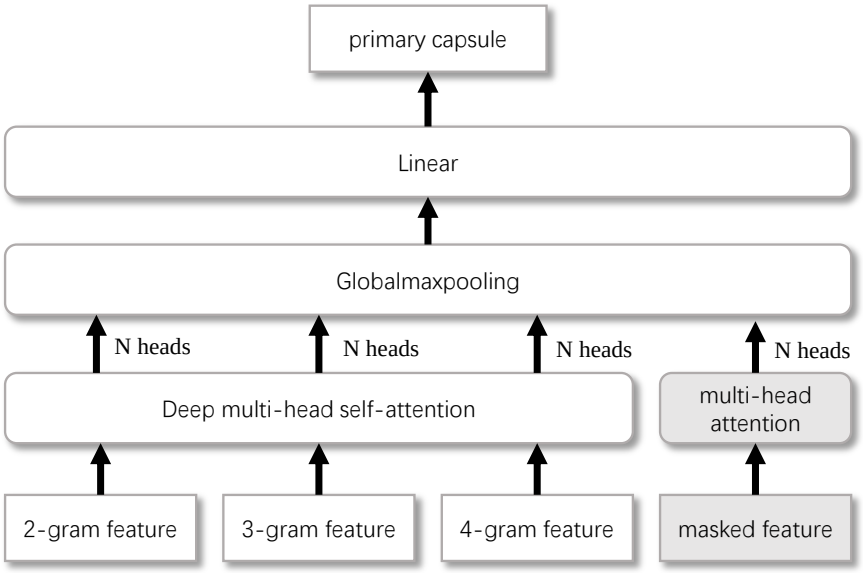

Fig. 3. The detailed architecture of ABASCap

3.6 Class Capsule Layer

To make the dynamic routing protocol more effective, the weight transformation matrix is introduced between the adjacent layers in the capsule network, so as to enhance the model's feature abstraction and combination ability. **Fig. 4** is a schematic diagram of the transformation matrix structure used in ABASCap.

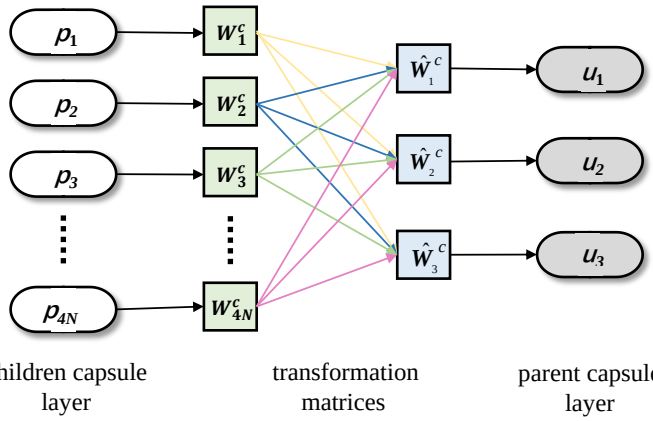

Fig. 4. Schematic diagram of ABASCap transformation matrix structure

The details of the algorithm for specific dynamic routing are shown in **Algorithm 1**. Specifically, before starting the iteration, the capsule $\boldsymbol{p}_i$ in the child capsule layer generates a prediction vector $\hat{\boldsymbol{u}}_{j|i}$ for the capsule $\boldsymbol{u}_j$ in the parent capsule layer through the transformation matrix:

$$\hat{\boldsymbol{u}}_{j|i}=\boldsymbol{p}_i \boldsymbol{W}_i^c \hat{\boldsymbol{W}}_j^c \tag{20}$$

Where $\boldsymbol{W}_i^c \in \mathrm{R}^{d_c \times \hat{d}_c}$ is the weight transformation matrix corresponding to $\boldsymbol{p}_i$, $\hat{\boldsymbol{W}}_j^c \in \mathrm{R}^{\hat{d}_c \times d_o}$ is the weight transformation matrix corresponding to $\boldsymbol{u}_j$, and d_o is the dimension of the output capsule.

All the prediction vectors corresponding to the class capsule $\boldsymbol{u}_j$ are weighted and summed to obtain the new vector representation of the class capsule, so as to enter the next iteration:

$$\boldsymbol{u}_j = \sum_i c_{ij} \hat{\boldsymbol{u}}_{j|i} \tag{21}$$

Where c_{ij} is the coupling coefficient, which is obtained by using SoftMax function for inner product of prediction vector and the corresponding vector in high-level capsule. It represents the aggregation strength of the low-level capsule $\boldsymbol{p}_i$ to the high-level capsule $\boldsymbol{u}_j$:

$$c_{ij}=\frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{22}$$

$$b_{ij} =< \hat{\boldsymbol{u}}_{j|i}, \boldsymbol{u}_j > \tag{23}$$

When all the iterative processes have been finished, $\boldsymbol{u}_j$ is substituted into the squash function, and the final output representation $\boldsymbol{u}_j^{\,o} \in \mathrm{R}^{d_o}$ of the j-th class capsule is generated. The

modulus length is limited to the range of [0,1], which represents the activity probability of the class capsule j:

$$\boldsymbol{u}_j^{\,o} = squash(\boldsymbol{u}_j) \tag{24}$$

Algorithm 1 Dynamic Routing Algorithm

procedure ABASCap_Routing ($\hat{\boldsymbol{u}}_{j|i}$, r_num)

begin

for capsule *i* in layer *l* and capsule *j* in layer *l* + 1:

Initialize the coupling coefficients: $c_{ij} \leftarrow 1/k$

for r_ num iterations **do**

$\boldsymbol{u}_j \leftarrow \sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i}$

$\boldsymbol{u}_j \leftarrow \boldsymbol{u}_j / \| \boldsymbol{u}_j \|$

$c_{ij} \leftarrow \underset{j}{softmax}(< \hat{\boldsymbol{u}}_{j|i}, \boldsymbol{u}_j >)$

$\boldsymbol{u}_j^{\,o} \leftarrow squash(\boldsymbol{u}_j)$

return $\boldsymbol{u}_j^{\,o}$

end

3.7 Margin Loss

Unlike ordinary deep learning networks, when used for classification, the capsule model will finally output multiple vector capsules. Each capsule represents a category. The modulus length of the capsule vector represents the existence probability for a certain category, which should be greater when the category is active. Therefore, the capsule network detects various classes, equivalent to transforming a multi-classification problem into multiple binary classification problems. Therefore, it is not appropriate to use common cross-entropy for the loss function. Instead, the margin loss should be chosen to evaluate each class of capsule networks:

$$L_j = T_j max\left(0, m^+ - \left\|\boldsymbol{u}_j^o\right\|\right)^2 + \lambda\left(1 - T_j\right) max\left(0, \left\|\boldsymbol{u}_j^o\right\| - m^-\right)^2 \tag{25}$$

If the final classification result exists in the *j*-th class capsule, then T_j is 1, otherwise it is 0. Set λ to 0.5 to reduce the loss weight of inactive capsules, set m^+ to 0.8 and m^- to 0.2 respectively, and the total loss of the model is the sum of the loss of each capsule.

4. Experiments

4.1 Datasets

In aspect-level sentiment classification, the restaurant and laptop review datasets in task 4 of semeval2014 [2] and the ACL 14 Twitter dataset [28] are often used as standard evaluation datasets. For different aspect entities, the data in the dataset is classified into three categories, i.e., negative, neutral, and positive, according to sentiment polarity. The experiment in this

paper was also conducted on the above three datasets, and the specific details are shown in **Table 1**.

Table 1. Statistics for three datasets

Dataset		Sentiment polarity		
		Positive	Neutral	Negative
Restaurant	Train	2164	637	807
	Test	728	196	196
Laptop	Train	994	464	870
	Test	341	169	128
Twitter	Train	1561	3127	1560
	Test	173	346	170

4.2 Experiment Settings

When using pre-training Glove [29], the word vector was fixed, and the dimension was set to 300. The learning rate was set to 1e-3. When using pre-training BERT [30], the word vector was fine-tuned along with the model training, and the dimension was set to 768. The learning rate should not be set too high during update process. To ensure performance, set it to 2e-5. Other general hyperparameter settings are shown in **Table 2**. The model was finally run on NVIDIA RTX 2080Ti GPU, and its performance was evaluated using accuracy and Macro-F1 values.

Table 2. Hyperparameter settings

hyperparameter	value
Dropout rate	0.1
Batch size	32
Max sequence length	100
L2 regularization	1e-4
Hidden dimension	300
Capsule output dimension	16
Dynamic routing Iterations	7
Attention head number	8
optimizer	Adam

4.3 Model Comparison

To evaluate the ABASCap model's performance on the three datasets, various typical models were introduced for comparisons, including certain baseline performance methods and the latest pre-training BERT. All comparison models are introduced as follows:

1) ATAE-LSTM[14]: this model combines the attention mechanism with the LSTM network. First, the aspect vector and input features are used to splice, and then the attention weight information of the hidden layer state sequence is calculated.

2) MemNet[15]: this model combines the attention mechanism with the deep memory network and stably optimizes the classification accuracy through the superposition of multiple computing layers.

3) IAN[17]: this model designs an interactive attention network model. The context and target are embedded into the two LSTM networks, and an aspect-based attention mechanism is proposed to obtain important features from context.

4) RAM[16]: based on MemNet, this model uses a Bi-direction LSTM network to improve the memory structure, and combines a multi-attention mechanism with recurrent neural networks to capture long-distance sentiment relationship.

5) TransCap[25]: this model realizes a novel transfer capsule network. The model adopts an aspect routing method, which can adapt the dynamic routing method to the transfer learning framework. It transfers semantic knowledge from other domains to aspect-level sentiment classification tasks.

6) BERT-PT[31]: based on the Bert pre-training model, this model explores an improved general post-training method and builds a self-built data set to improve the performance of Bert fine-tuning to adapt to the target sentiment analysis task.

7) AEN[21]: this model is an attention coder network, which applies multi-head self-attention to aspect-level sentiment analysis, and uses an attention-based encoder to model the relationship between aspects and context for obtaining the interaction and semantic information.

8) BAT[32]: this model proposes a new architecture that combines adversarial training with the Bert. By adding adversarial examples, Bert's performance is further improved in aspect-level sentiment classification and target extraction.

Table 3. Experimental results of performance

	Model	Laptop		Restaurant		Twitter	
		Accuracy	**Macro-F1**	**Accuracy**	**Macro-F1**	**Accuracy**	**Macro-F1**
Baseline	ATAE-LSTM	0.6870	-	0.7720	-	-	-
	MemNet	0.7237	-	0.8095	-	-	-
	IAN	0.7210	-	0.7860	-	-	-
	RAM	0.7449	0.7135	0.8023	0.7080	0.6936	0.6730
	TransCap	0.7387	0.7010	0.7955	0.7141	-	-
	AEN-GloVe	0.7351	0.6904	0.8098	0.7214	0.7283	0.6981
	ABASCap-GloVe	**0.7616**	**0.7178**	**0.8174**	**0.7266**	**0.7292**	**0.7023**
BERT	BERT-PT	0.7807	0.7508	0.8495	0.7696	-	-
	BAT	0.7935	0.7650	0.8603	0.7924	-	-
	AEN-BERT	0.7993	0.7631	0.8312	0.7376	0.7471	0.7313
	ABASCap-BERT	**0.8142**	**0.7831**	**0.8667**	**0.8053**	**0.7628**	**0.7492**

Note: "-" represents unreported experimental results.

To effectively evaluate the performance of ABASCap, all models were divided into two groups according to whether pre-training BERT word vectors were used. **Table 3** shows the main experimental results. It indicated that ABASCap performed well on the three datasets, especially better on the restaurant and laptop datasets.

The classification performance of ATAE-LSTM with the shallowest model depth was the worst, and the performance advantages of other models were apparent. Both MemNet and RAM used a multi-hop attention structure to stack and deepen the network recursively. After improving the memory structure, RAM could improve the classification performance of the laptop dataset. TransCap adopted a multi-level capsule structure, and the aspect-based routing method showed limited improvement in model performance. Both AEN-GloVe and ABASCap-GloVe used a multi-head attention mechanism and performed better on the three datasets, especially the Twitter dataset. Obviously, the introduction of the multi-head attention mechanism can optimize the model for the fine-grained sentiment classification task. ABASCap-GloVe had absolute advantages in the classification performance of laptop and

restaurant datasets.

From the experimental results, pre-training BERT increased the classification accuracy by more than 5 percentage points. But it must be emphasized that the knowledge representation in pre-training BERT is trained through a large general dataset and not targeted at any specific field. In the BERT models, AEN-BERT and ABASCap-BERT had significantly improved the performance compared to BERT-PT and BAT. It showed that the whole model's performance could be further enhanced by reasonably designing the high-level network for specific tasks and fine-tuning its parameters in training. Finally, the classification performance of ABASCap-BERT on each dataset had been substantially improved, which indicated that the capsule network could abstract the context features at a higher-level. By improving the multi-head attention mechanism from two aspects: deep self-attention design and local context feature optimization, BERT could release greater capabilities in fine-grained aspect-level sentiment classification.

4.4 Performance Analysis of Model Structure

To analyze each component's effectiveness in ABASCap, the ablation experiment was conducted by adjusting and replacing different parts of each layer structure. The four ablation models are described below. The specific experimental results are shown in **Table 4**.

Table 4. Classification accuracy of each ablation

Model	Laptop	Restaurant	Twitter
	Accuracy	Accuracy	Accuracy
ABASCap-BERT w/o Conv	0.8014	0.8453	0.7548
ABASCap-BERT w/o DMHSA	0.8102	0.8507	0.7610
ABASCap-BERT w/o LCM	0.8025	0.8411	0.7515
ABASCap-BERT w/o Capsule	0.8016	0.8492	0.7566
ABASCap-BERT	**0.8142**	**0.8667**	**0.7628**

Note: "w/o" means "without."

1) "w/o Conv": the convolution operation was removed from the feature extraction layer, and the N-gram feature was replaced by the original input sequence feature;

2) "w/o DMHSA": in the attention coding layer, the deep multi-head self-attention mechanism proposed in this work was replaced by the standard multi-head self-attention mechanism;

3) "w/o LCM": the local context mask mechanism was removed from the attention coding layer so that the local context weight information in the input sequence was not considered in the model;

4) "w/o Capsule": all the capsule network structures in the class capsule layer were replaced by fully connected multilayer perceptron, and the multi-classification output was performed by SoftMax.

The experimental results showed that the performance of the modified models on the three datasets had been significantly reduced, compared with the original ABASCap-BERT model. The modules of each layer in ABASCap played an essential role in improving the performance.

Specifically, *ABASCap-BERT w/o LCM* had the worst overall classification effect. The local context weighting mechanism made the model's semantic understanding more accurate, with a particularly significant effect in short text tasks. *ABASCap-BERT w/o DMHSA* had the best

classification effect, indicating that deep self-attention was weaker than other designs in mining hidden relationships of text. Still, the improvement over the standard self-attention mechanism remained obvious. Regarding the *ABASCap-BERT w/o Conv*, the multi-dimensional combination of N-gram features could provide a more abstract and accurate representation of text semantics and structure, which were more effective than original sequence features in NLP tasks. Finally, compared with ABASCap-BERT, the classification accuracy of *ABASCap-BERT w/o Capsule* on the three datasets was significantly reduced, proving that the capsule network could express more abundant text sentiment information and improved the model's overall abstraction ability.

4.5 Analysis of Local Context Window Setting

To further verify the LCM's effectiveness in improving the model performance and investigate the influence of the LCW's size on the classification accuracy, a series of comparative experiments were carried out to evaluate the optimal LCW on different domain datasets. According to the principle of expanding the local context scope, the default range of LCW was set from 1 to 10, corresponding to the setting of the local semantic region from small to large. The experimental results are shown in **Fig. 5**-**Fig. 7**.

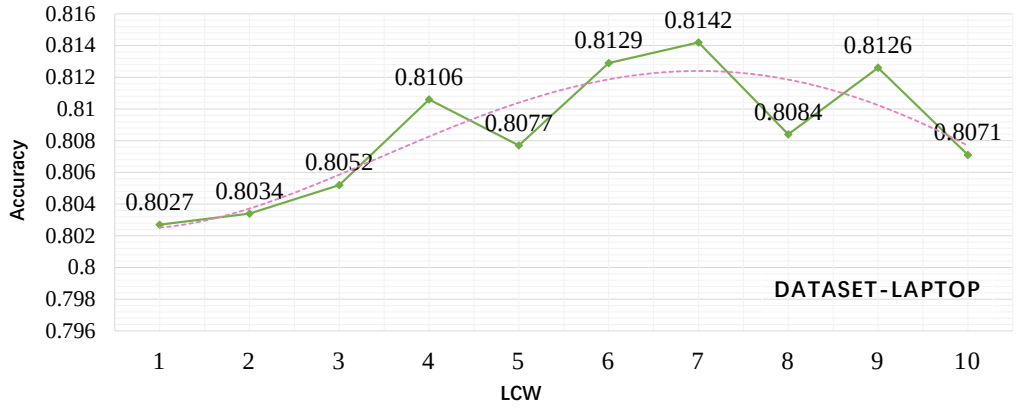

Fig. 5. Classification accuracy of ABASCap-BERT on the laptop dataset with different LCW settings

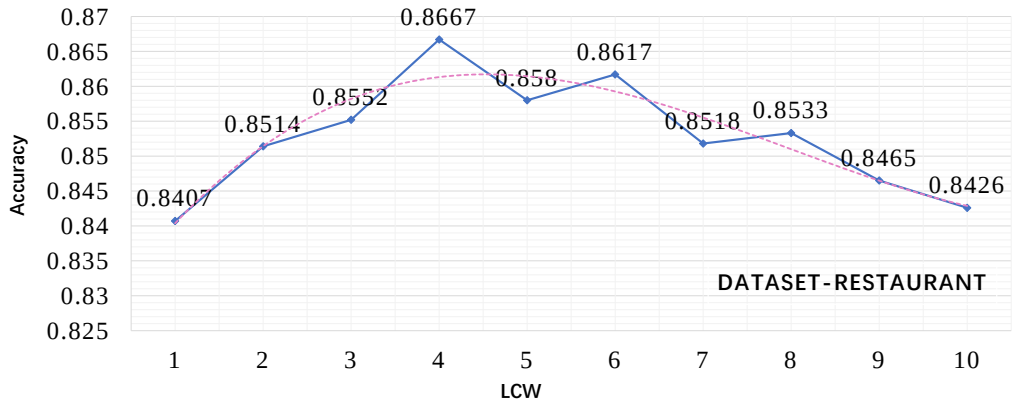

Fig. 6. Classification accuracy of ABASCap-BERT on the restaurant dataset with different LCW settings

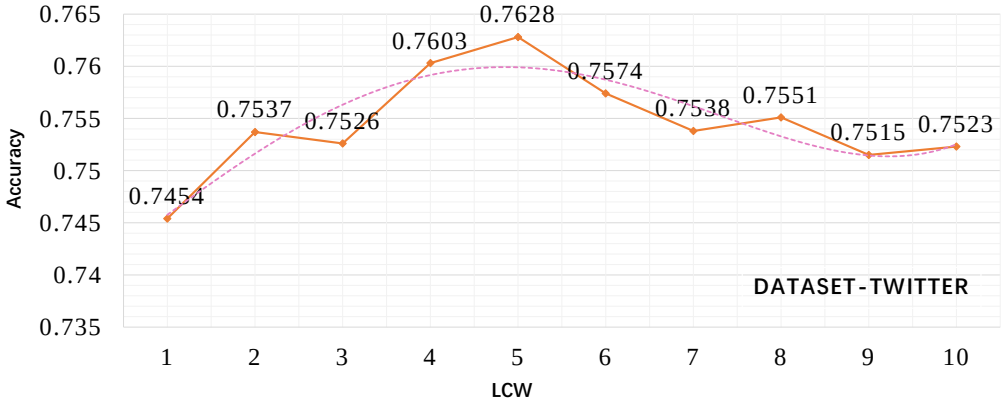

Fig. 7. Classification accuracy of ABASCap-BERT on the Twitter dataset with different LCW settings

The experimental results showed that the optimal local correlation semantic area of the laptop dataset was the largest, and the best corresponding LCW value was 7. As the LCW value increased, the classification accuracy increased steadily. The model performance decreased slightly after the peak. In the restaurant dataset, the best LCW value was 4, indicating that the reviews were more likely to express emotional opinions directly. When the local context range was set beyond the optimal relevance area, the classification accuracy dropped rapidly. Ambiguity and redundant semantic features were apparently present as noise in the model. For the Twitter dataset, the best LCW value was 5. When the local context setting exceeded the optimal relevance region, the classification accuracy decreased significantly. However, with the increase of LCW, the performance did not fluctuate significantly. It can be found that social media text tended to be more consistent in emotional expression.

A comprehensive analysis based on the experimental results showed that the design and use of local context features could improve the task model's performance. Simultaneously, the indiscriminate use of all contextual features in the sentiment analysis task was proven to introduce disturbing sentiment noise to the model with a risk of overfitting.

5. Conclusion

In this paper, a hybrid attention capsule network was proposed for the fine-grained sentiment classification problem. The model used the improved multi-head self-attention mechanism to extract the internal context coloration effectively while introducing the concept of local context association semantic region. Moreover, a capsule network with richer semantic expression was used to process the high-level abstract features and output the classification results. The routing algorithm and activation function were optimized according to the sentiment analysis task. We thoroughly evaluated the model on the semeval2014 and Twitter datasets. The experimental results showed that ABASCap outperformed the popular baseline models and the latest pre-training BERT model. Besides, other comparative experiments not only verified the critical role of each module but also proved that the local context features had more abundant and accurate sentiment semantic information for the aspect.

In the future, we hope to use more flexible and diverse methods for local context feature extraction, especially dynamic and adaptive weighting, to make local context semantic information modeling more efficient and reasonable. Additionally, taking advantage of the

capsule network's scalability, we can try to use position information, part-of-speech tagging information, and prior knowledge as supplements to achieve the extension of the whole task feature space.

References

[1] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1-167, May 2012. Article (CrossRef Link)

[2] M. Pontiki, D. Galanis, I. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "Semeval-2014 task 4: Aspect Based Sentiment Analysis," in *Proc. of the 8th International Workshop on Semantic Evaluation*, pp. 27-35, Aug. 2014. Article (CrossRef Link)

[3] H. Peng and Q. Li, "Research on the automatic extraction method of web data objects based on deep learning," *Intelligent Automation & Soft Computing*, vol. 26, no. 3, pp. 609-616, 2020. Article (CrossRef Link)

[4] F. Bi, X. Ma, W. Chen, W. Fang, H. Chen, J. Li, and B. Assefa, "Review on video object tracking based on deep learning," *Journal of New Media*, vol. 1, no.2, pp. 63-74, 2019. Article (CrossRef Link)

[5] Q. Ye, Z. Li, L. Fu, Z. Zhang, W. Yang, and G. Yang, "Nonpeaked discriminant analysis for data representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3818-3832, 2019. Article (CrossRef Link)

[6] P. Kalaivaani and R. Thangarajan, "Enhancing the classification accuracy in sentiment analysis with computational intelligence using joint sentiment topic detection with MEDLDA," *Intelligent Automation & Soft Computing*, vol. 26, no. 1, pp. 71-79, 2020. Article (CrossRef Link)

[7] M. Cao, S. Zhou, and H. Gao, "A recommendation approach based on product attribute reviews: improved collaborative filtering considering the sentiment polarity," *Intelligent Automation & Soft Computing*, vol. 25, no. 3, pp. 595-604, 2019. Article (CrossRef Link)

[8] G. Zhu, W. Liu, S. Zhang, X. Chen, and C. Yin, "The method for extracting new login sentiment words from Chinese micro-blog based on improved mutual information," *Computer Systems Science and Engineering*, vol. 35, no. 3, pp. 223-232, 2020. Article (CrossRef Link)

[9] D. J. Zeng, Y. Dai, F. Li, J. Wang, and A. K. Sangaiah, "Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 3971-3980, May 2019. Article (CrossRef Link)

[10] A. Feng, Z. Gao, X. Song, K. Ke, T. Xu, and X. Zhang, "Modeling multi-targets sentiment classification via graph convolutional networks and auxiliary relation," *Computers, Materials & Continua*, vol. 64, no. 2, pp. 909-923, 2020. Article (CrossRef Link)

[11] J. Zhou, J. X. Huang, Q. Chen, Q. V. Hu, T. Wang, and L. He, "Deep learning for aspect-level sentiment classification: Survey, vision, and challenges," *IEEE Access*, vol. 7, pp. 78454-78483, May 2019. Aritcle (CrossRef Link)

[12] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," in *Proc. of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2204-2212, Dec. 2014. Article (CrossRef Link)

[13] D. Zheng, Z. Ran, Z. Liu, L. Li, and L. Tian, "An Efficient Bar Code Image Recognition Algorithm for Sorting System," *Computers, Materials & Continua*, vol. 64, no. 3, pp. 1885-1895, June 2020. Article (CrossRef Link)

[14] Y. Wang, M. Huang, X. Zhu, and, L. Zhao, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606-615, Nov. 2016. Article (CrossRef Link)

[15] D. Tang, B. Qin, and T. Liu, "Aspect Level Sentiment Classification with Deep Memory Network," in *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 214-224, Nov. 2016. Article (CrossRef Link)

[16] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 452-461, Sep. 2017. Article (CrossRef Link)
[17] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. of the 26th International Joint Conference on Artificial Intelligence*, pp. 4068-4074, Aug. 2017. Article (CrossRef Link)
[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, Aidan N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Proc. of the 31st International Conference on Neural Information Processing*, pp. 6000-6010, Dec. 2017. Article (CrossRef Link)
[19] A. Ambartsoumian and F. Popowich, "Self-Attention: A Better Building Block for Sentiment Analysis Neural Network Classifiers," in *Proc. of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 130-139, Oct. 2018. Article (CrossRef Link)
[20] G. Letarte, F. Paradis, P. Giguère, and F. Laviolette, "Importance of self-attention for sentiment analysis," in *Proc. of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 267-275. Nov. 2018. Article (CrossRef Link)
[21] Y. Song, J. Wang, T. Jiang, Z. Liu, and Y. Rao, "Attentional encoder network for targeted sentiment classification," *arXiv preprint arXiv:1902.09314*, 2019. Article (CrossRef Link)
[22] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 3859-3869, 2017. Article (CrossRef Link)
[23] M. Yang, W. Zhao, L. Chen, Q. Qu, Z. Zhao, and Y. Shen, "Investigating the transferring capability of capsule networks for text classification," *Neural Networks*, vol. 118, pp. 247-261, Oct. 2019. Article (CrossRef Link)
[24] Y. Wang, A. Sun, M. Huang, and X. Zhu, "Aspect-level sentiment analysis using as-capsules," in *Proc. of the World Wide Web Conference*, pp. 2033-2044, May 2019. Article (CrossRef Link)
[25] Z. Chen and T. Qian, "Transfer capsule network for aspect level sentiment classification," in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 547-556, July 2019. Article (CrossRef Link)
[26] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214-221, Feb. 2020. Article (CrossRef Link)
[27] J. Hao, X. Wang, S. Shi, J. Zhang, and Z.Tu, "Multi-Granularity Self-Attention for Neural Machine Translation," in *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 886-896, Nov. 2019. Article (CrossRef Link)
[28] L. Dong, F. R. Wei, C. Q. Tan, D. Y. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent twitter sentiment classification," in *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 49-54, June 2014. Article (CrossRef Link)
[29] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532-1543, Oct. 2014. Article (CrossRef Link)
[30] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers)*, vol. 1, pp. 4171-4186, June 2019. Article (CrossRef Link)
[31] H. Xu, B. Liu, L. Shu, and P. Yu, "BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis," in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers)*, vol. 1, pp. 2324-2335, June 2019. Article (CrossRef Link)
[32] A. Karimi, L. Rossi, and A. Prati, "Adversarial training for aspect-based sentiment analysis with BERT," *arXiv preprint arXiv:2001.11316*, 2020. Article (CrossRef Link)

Yu Deng received the B.E. and M.S. degrees in computer engineering from Radar Academy, China, in 2004 and 2007, respectively. He is currently a Ph.D. candidate in the School of Information and Software Engineering, University of Electronic Science and Technology, China. His research interests include deep learning, natural language processing.

Hang Lei received the M.S. and Ph.D. degrees in computer engineering from University of Electronic Science and Technology, China, in 1988 and 1997, respectively. He is currently a professor with the School of Information and Software Engineering, University of Electronic Science and Technology, China. His research interests include data mining, embedded real time system, big data analysis, image processing, formal verification.

Xiaoyu Li received the M.S. and Ph.D. degrees in computer engineering from University of Electronic Science and Technology, China, in 2009 and 2014, respectively. She is currently an assistant professor with the School of Information and Software Engineering, University of Electronic Science and Technology, China. Her research interests include data mining, natural language processing, quantum machine learning.

Yiou Lin received the B.E. degrees in computer engineering from University of Electronic Science and Technology, China, in 2013. He is currently a Ph.D. candidate in the School of Information and Software Engineering, University of Electronic Science and Technology, China. His research interests include deep learning, natural language processing.

Wangchi Cheng received the M.S. and Ph.D. degrees in operations research from the Institute of Operations Research and Analysis (IORA), China, in 2003 and 2007, respectively. He is currently an associate researcher of the Institute of Logistics Science and Technology, China. His research interests include emergency logistics, emergency management and data mining.

Shan Yang is currently an associate professor with the Department of Chemistry, Physics, and Atmospheric Sciences, Jackson State University, USA. His research interests include biomedical optics, image processing, data mining.