BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 05/06 pp195–207
Volume 4, Number 3, September 2021
DOI: 10.26599/BDMA.2021.9020003

A Multitask Multiview Neural Network for End-to-End Aspect-Based Sentiment Analysis

Yong Bie and Yan Yang*

Abstract: The aspect-based sentiment analysis (ABSA) consists of two subtasks—aspect term extraction and aspect sentiment prediction. Existing methods deal with both subtasks one by one in a pipeline manner, in which there lies some problems in performance and real application. This study investigates the end-to-end ABSA and proposes a novel multitask multiview network (MTMVN) architecture. Specifically, the architecture takes the unified ABSA as the main task with the two subtasks as auxiliary tasks. Meanwhile, the representation obtained from the branch network of the main task is regarded as the global view, whereas the representations of the two subtasks are considered two local views with different emphases. Through multitask learning, the main task can be facilitated by additional accurate aspect boundary information and sentiment polarity information. By enhancing the correlations between the views under the idea of multiview learning, the representation of the global view can be optimized to improve the overall performance of the model. The experimental results on three benchmark datasets show that the proposed method exceeds the existing pipeline methods and end-to-end methods, proving the superiority of our MTMVN architecture.

Key words: deep learning; multitask learning; multiview learning; natural language processing; aspect-based sentiment analysis

1 Introduction

Different from the traditional sentence-level or document-level sentiment analysis, which evaluates the overall sentiment polarity[1], aspect-based sentiment analysis (ABSA)[2] aims to conduct a more fine-grained sentiment analysis task for the specific aspects of a sentence. This fine-grained task generally requires extracting aspects, which are explicitly mentioned in the text first, and then predicting the sentiment polarity of the sentence towards the extracted aspects. For example, in the sentence "*I love Windows7 which is a vast improvement over Vista*", we need to extract the two aspects of "*Windows7*" and "*Vista*", and then accurately predict their sentiment polarity to be positive or negative.

Given that ABSA can be broken into two subtasks, namely, aspect term extraction (AE) and aspect sentiment prediction (ASP), many approaches are used to investigate both subtasks. For AE, traditional machine learning methods such as support vector machine (SVM), hidden Markov model (HMM), and conditional random field (CRF)[3–6], were used in the early stage with an increasing number of deep learning methods[7–11] widely employed now. As for ASP, many existing studies[12–20] claim to be ABSA methods. However, considering that the aspects in their works are given in advance, these methods essentially only perform ASP subtasks, rather than a complete ABSA task. The research status shows that combining the methods of two subtasks in a pipeline manner is often necessary when dealing with a complete ABSA task, which can increase the complexity of practical application. The error extraction results from

• Yong Bie and Yan Yang are with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China. E-mail: autwind_by@163.com; yyang@swjtu.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2021-01-03; accepted: 2021-01-25

AE can inevitably affect the downstream ASP subtask, leading to the continuous propagation and accumulation of errors and thus reducing the overall performance of the whole pipeline model. The pipeline approach also ignores some potential associated information among two subtasks, leading to the waste of useful knowledge.

To overcome the problems above, several recent studies concentrate on establishing a one-stop framework to conduct ABSA in an end-to-end manner. Specifically, these works can be divided into two categories, one of which[21,22] is the unified model and regards the whole ABSA problem as a sequence labeling task, extracting the aspects and predicting their sentiment polarities simultaneously via a unified labeling scheme such as "*B-POS, I-POS, B-NEG, I-NEG, O* ...". The others[23,24], namely, joint models, exploit multitask learning to build an end-to-end model in which AE and ASP are dealt with together. In this way, the interactions between two subtasks can be well utilized. These two kinds of methods have their own advantages and disadvantages. The unified models use the unified labeling scheme, which better reflect the characteristics of an end-to-end architecture and can excavate further potential information from a global view to perform the ABSA task. However, these unified labeling schemes also have problems that easily make learned representations confused[24]. No such problems are encountered in the joint models, as they learn separate representations for AE and ASP, which can respectively be seen as excavating aspect boundary information and sentiment polarity information from two local views. However, the difficulty of these kinds of methods lies in the design of an excellent multitask information interaction mechanism to make full use of the correlation between two subtasks.

Inspired by the two kinds of methods above, we propose a novel multitask multiview neural network (MTMVN) architecture for end-to-end ABSA. By building three network branches on the basis of convolutional neural network (CNN)[25] and gated recurrent unit (GRU)[26] to respectively conduct AE task, ASP task, and the whole ABSA task under the unified labeling scheme (UFT), we can use multitask learning to realize the combination of the unified models and the joint models mentioned above; we can also take AE and ASP as auxiliary tasks to make up for the deficiency from the unified labeling scheme of UFT. Moreover, we regard the representation learned from the branch of UFT as the global view and the representations from AE and ASP as two local views focusing on aspect boundary information and sentiment polarity information, respectively. Based on a multiview fusion method[27], we fuse the two local views to obtain another global view. Then a canonical correlation analysis (CCA) term is introduced into the joint loss function via a deep CCA method proposed by Andrew et al.[28], training to reduce the total loss while increasing the correlation of the two global views to fine-tune each network branch and further improve the overall performance of the model. We evaluate our approach on three benchmark datasets. The experimental results show that our method has certain superior perfomance against some existing pipeline methods and end-to-end methods.

The rest of the paper is organized in the following manner. Section 2 summarizes the related works. Section 3 describes our proposed MTMVN model. Section 4 provides and analyzes the experimental results on the three benchmark datasets. Section 5 draws the conclusion about this work.

2 Related Work

2.1 Aspect term extraction

For AE, early methods were mostly based on traditional machine learning. Jiang et al.[3] extracted entities from product comments by using a method called generalized view-sentiment tree. Jin and Ho[4] integrated the lexical features, including the negative words, into HMM to coextract aspect terms and opinion terms at the same time. Jakob and Gurevych[5] performed this co-extraction via CRF in the form of sequence labeling. Li et al.[6] combined skip-chain CRF and tree CRF to extract aspects and opinions in movie reviews.

In recent years, deep learning has been widely applied in this field. Poria et al.[7] proposed a deep neural network model consisting of seven layers of CNN. Based on the "BIO" labeling scheme, this model can effectively extract the aspects in the text. Wang et al.[8] presented a coupled multilayer attention mechanism. By constructing a couple of attentions for each sentence, it can capture the direct and indirect relations between aspects and opinions, realize the full interaction of such information, and finally address the coextraction of aspects and opinions. Ye et al.[9] put forward a multilayer CNN model on the basis of a dependency tree (DTBCSNN), which can effectively utilize the dependency information of sentences to capture syntactic features and obtain high-

quality aspect extraction results. Further, Luo et al.[10] studied the aspect extraction method on the basis of the representation of the bidirectional dependency tree, which can not only capture syntactic features but also memorize the sequential information of sentences at the same time to fuse them, rather than only using a single representation to address the AE task. Xu et al.[11] introduced domain knowledge into the AE task, obtained text representations by cascading general and domain embeddings, and achieved good results in the stacked CNN network structure.

2.2 Aspect sentiment prediction

In the field of ASP, Wang et al.[12] combined attention mechanism with long short-term memory (LSTM) network[29] to realize the incorporation of aspect information and context information. Tang et al.[13] proposed an attention mechanism which integrates the context's position information relative to the aspects. Combined with the deep memory network[30], their model obtained state-of-the-art (SOTA) at that time. Fan et al.[14] presented a multi-grained attention network, which fuses coarse and fine grained attention mechanisms while using the deep neural network model. Chen et al.[15] investigated a multilayer attention mechanism combined with a GRU network, which is mainly used to capture the sentiment features separated by a long distance. Ma et al.[16] used attention mechanism to model the interactive information between contexts and aspects for making full use of the aspect information. Huang et al.[17] calculated the interaction matrix on the basis of the context representations and aspect representations encoded by the bidirectional LSTM (BiLSTM) and realized an attention-over-attention module[31] through this interaction matrix to calculate the final attention on the target sentence for obtaining the final representation for classification. Li et al.[18] explored a hierarchical attention-based position-aware network. Similar to Ref. [13], this network uses the position information of aspects in sentences through a position-aware encoding layer to learn the position-aware representations of sentences and thus generates the representations of contexts to the specific aspects. Song et al.[19] used attention mechanism as the encoding layer of the model instead of the traditional recurrent neural network (RNN), which can avoid the shortcomings of RNN. Meanwhile, the authors[13] introduced a label smoothing regularization term into the loss function, which solved the label unreliability issue neglected in previous work. Li et al.[20] pointed out that certain defects are found with regard to the use of attention mechanism to extract the semantic association between an aspect and its context; therefore, they used CNN instead to learn important information in the context and proposed a context-preserving transformation module to assist CNN in this operation.

Note that although the above models are all claimed to be approaches for the ABSA task, their models deal with aspects, which have been given in advance without undergoing the AE process. Therefore, we only consider them as methods for ASP, rather than methods for a complete ABSA task.

3 Proposed Method

In this section, we describe the details of our proposed MTMVN model for end-to-end ABSA task. The overall architecture of the model is shown in Fig. 1. It is mainly composed of five parts: embedding layer, shared layer and encoding layer, local view fusion layer, global view interaction layer, and output layer.

3.1 Task definition

Given an input text sequence S with L words, denoted by $S = \{w_1, w_2, \ldots, w_L\}$, the end-to-end ABSA task aims to extract M aspects $A = \{a_1, a_2, \ldots, a_M\}$ explicitly mentioned in the sequence and the sentiment polarities $P = \{p_1, p_2, \ldots, p_M\}$ held by the text to the aspects. Based on unified labeling scheme $T = \{$ B-polarity, I-polarity, O|polarity $\in \{$ POS, NEG, NEU, CONF$\}\}$, which means the "polarity" can be positive, negative, neutral, or conflict, this task can be tackled as a sequence labeling problem, that is, input the text sequence S to obtain the tagging results $O = \{o_i | o_i \in T, 1 \leqslant i \leqslant L\}$. The word labeled as "B-polarity" or "I-polarity" indicates the beginning or the inside of an aspect whose sentiment polarity should be "polarity". Relatively, being labeled as "O" means that the word does not belong to any aspect, namely, other words. We call this ABSA under the unified labeling scheme as UFT task. For auxiliary tasks AE and ASP, we also formulate them as sequence labeling problems under the labeling scheme $T^a = \{\text{B}, \text{I}, \text{O}\}$ and $T^p = \{\text{POS}, \text{NEG}, \text{NEU}, \text{CONF}\}$.

3.2 Embedding layer

In order to learn the initial representation of text, we need to embed the input sequence as vectors and design an embedding layer in our model. In this layer, we adopt the approach described in Ref. [11] to concatenate

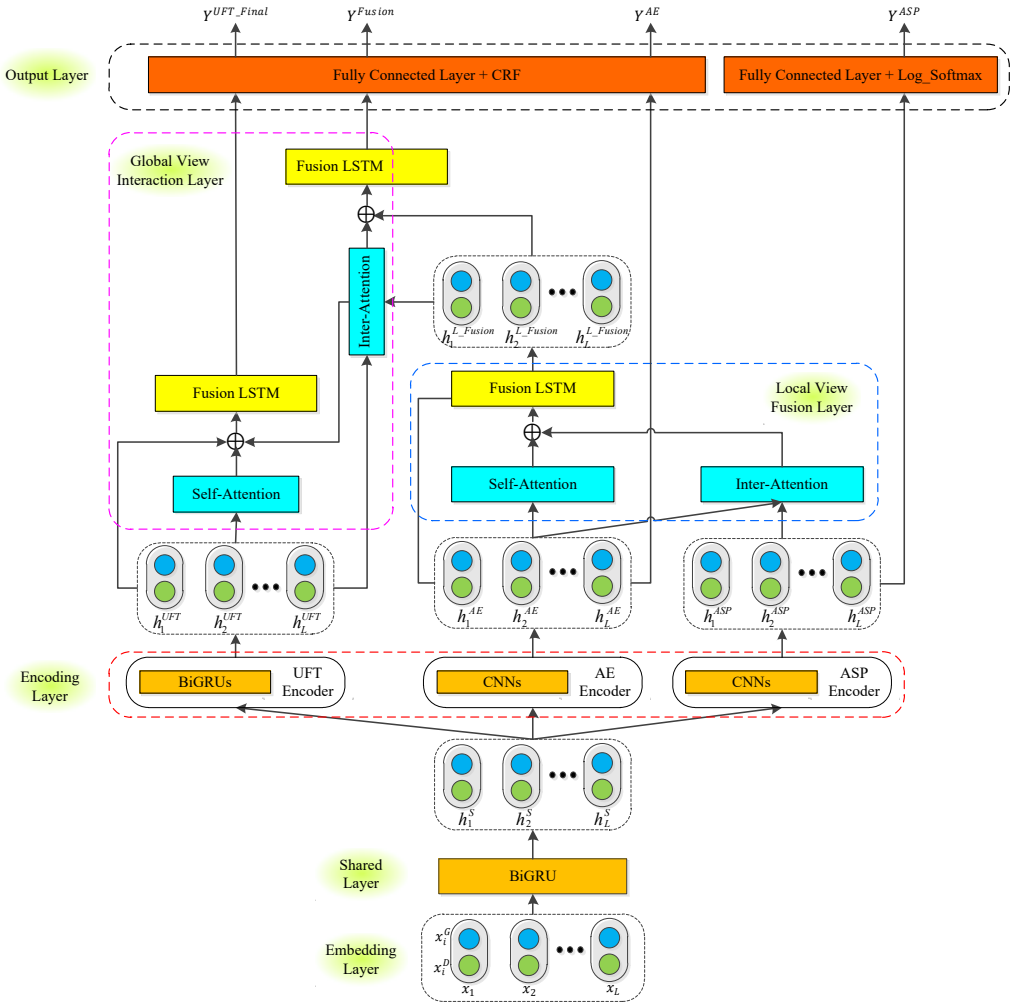

Fig. 1 Architecture of proposed MTMVN model.

the general embedding and the domain embedding with specific domain knowledge as embedding representations of the input text sequence. That is, for input text sequence $S = \{w_1, w_2, \ldots, w_L\}$, the general embedding is obtained as $X^G = \{x_1^G, x_2^G, \ldots, x_L^G\}$ via the pretrained Glove[32] embeddings; on the basis of the pretrained domain-specific embeddings released by Ref. [11], the domain embedding is denoted as $X^D = \{x_1^D, x_2^D, \ldots, x_L^D\}$, where $x_i^G \in \mathbb{R}^{d_G}$, $x_i^D \in \mathbb{R}^{d_D}$. Hence, we obtain the final embedding representation $X = \{x_1, x_2, \ldots, x_L\}$ of sequence S, where $x_i = x_i^G \oplus x_i^D$ and $\oplus$ means the concatenation operation.

3.3 Encoding layer

We use an encoding layer to further extract the latent features from the initial embedding representation X, in which the stacked GRU encoders and stacked CNN encoders are mainly employeed. First, a multitask shared layer composed of a bidirectional GRU (BiGRU) is used for preliminary feature extraction to obtain the latent representation H^S shared among all tasks.

$$\begin{aligned} h_t^S &= [\overrightarrow{GRU}(x_t); \overleftarrow{GRU}(x_t)], \\ H^S &= \{h_1^S, h_2^S, \ldots, h_L^S\} \end{aligned} \tag{1}$$

where h_t^S is the feature vector of t-th timestep of H^S.

Second, three tasks are performed by three network branches from the global view and two local views, that is, the main task UFT and two auxiliary tasks AE and ASP. For UFT, the stacked BiGRUs are utilized as the encoder to deal with the shared representation H^S, by which the representation H^{UFT} containing further global information for the whole ABSA task can be obtained.

$$\begin{aligned} h_t^{UFT} &= [\overrightarrow{GRU}(h_t^S); \overleftarrow{GRU}(h_t^S)], \\ H^{UFT} &= BiGRUs(H^S) \end{aligned} \tag{2}$$

where h_t^{UTF} is the feature vector of t-th timestep of H^{UFT}.

For AE, our proposed method employs the stacked

CNNs as the encoder to perform further feature extraction on the basis of the shared representation H^S. Reference [11] proved that such a structure exhibits a good performance in extracting the aspect terms. For each time step h_i^S of H^S, the local convolution operation performed by each layer of CNN is as follows:

$$h_{i,r}^{l+1} = \max\left(0, \left(\sum_{j=-c}^{c} \omega_{j,r}^{l} h_{i+j}^{l}\right) + b_r^l\right) \quad (3)$$

where h_i^l represents the i-th time step of the input of l-th layer CNN, and $h_i^1 = h_i^s$, which means the time step h_i^s of H^S is accepted as input by the first layer CNN. $\omega_{j,r}^l$ and b_r^l respectively represent the weight and the bias of the r-th convolution kernel of the l-th layer CNN, and each convolution kernel size is fixed at $2c + 1$. ReLU is used as the activation function for each CNN layer. As the output of the stacked CNNs encoder, the final representation H^{AE} for AE is obtained as below:

$$H^{AE} = CNNs(H^S) \quad (4)$$

We believe that H^{AE} can be regarded as a local view with further aspect boundary information generated from input representation X.

For ASP, based on the labeling scheme $T^p = \{\text{POS}, \text{NEG}, \text{NEU}, \text{CONF}\}$, we still treat it as a sequence labelding problem, and label each token with the sentiment polarity of the corresponding aspect term which the token belongs to. Table 1 shows an example with gold AE and ASP labels. Referring to Ref. [23], we ignore the predictions on the tokens without ASP gold labels (i.e., the token labeled as "–") when computing the training loss for ASP.

We also use the stacked CNNs as the encoder, but the number of stacked layers is different from the one in AE. The final representation H^{ASP} for ASP is as follows:

$$H^{ASP} = CNNs(H^S) \quad (5)$$

Similar to H^{AE}, H^{ASP} tends to contain further information on sentiment polarity, which can be regarded as another local view of the whole ABSA task.

3.4 Local view fusion

We obtain two local views H^{AE} and H^{ASP} with different emphases by the encoding layer. Then we fuse them to incorporate aspect boundary information and sentiment polarity information, which is conducive to directly predicting the sentiment polarity of a certain aspect in an end-to-end manner.

By introducing the attention mechanism and a fusion LSTM[27], we denote the fusion operation of two local views as follows:

$$\begin{aligned} H^{Concat} &= [H^{AE}; H^{AE'}; H^{ASP'}], \\ H^{L\_Fusion} &= Fusion\_LSTM(H^{Concat}) \end{aligned} \quad (6)$$

where $H^{AE'}$ is generated from H^{AE} on the basis of the self-attention mechanism as shown below:

$$\begin{aligned} f(H^{AE}) &= \tanh(W_{att\_S}[H^{AE}; H^{AE}]), \\ self\_score &= softmax(f(H^{AE})), \\ H^{AE'} &= self\_score^{\mathrm{T}} H^{AE} \end{aligned} \quad (7)$$

Through the self-attention mechanism, the key contents containing more aspect boundary information in H^{AE} are paid more attention by the model, and these significant parts are always the aspects that the model must identify. Therefore, self-attention can further enhance the mining and utilization of aspect boundary information in H^{AE}.

$H^{ASP'}$ contains the interactive information between the representations of H^{AE} and H^{ASP} captured by the interactive-attention mechanism, which is calculated as follows:

$$\begin{aligned} f(H^{ASP}, H^{AE}) &= \tanh(W_{att\_I}[H^{ASP}; H^{AE}]), \\ Inter\_score &= softmax(f(H^{ASP}, H^{AE})), \\ H^{ASP'} &= Inter\_score^{\mathrm{T}} H^{ASP} \end{aligned} \quad (8)$$

Here, we regard H^{AE} as the query and H^{ASP} as the key and value in the attention model. The calculated attention distribution score *Inter_score* reflects which time steps in H^{ASP} deserve further attention after the aspect boundary information from H^{AE} is integrated. Then, we attach this attention score to H^{ASP}, thus obtaining representation $H^{ASP'}$, which further enhances the sentiment polarity information.

To make H^{Concat}, which is obtained from concatenating the results of the above attention mechanism, well fuses the information from two local views, we employ the fusion LSTM presented in Ref. [27] to further process H^{Concat}.

Representation H^{Concat} is sent into the Fusion LSTM which is a variant of BiLSTM. Taking the forward

Table 1 A labeling example with gold AE and ASP labels.

Gold labels	Input sequence									
	Moules	were	excellent	,	lobster	ravioli	was	very	salty	!
Gold labels for AE	B	O	O	O	B	I	O	O	O	O
Gold labels for ASP	POS	–	–	–	NEG	NEG	–	–	–	–

direction as an example, an information vector is calculated on the basis of each time step when it is processed in an LSTM cell.

$$I_{t-1} = W_i h_{t-1}^{Concat} + b_i \tag{9}$$

where h_{t-1}^{Concat} and I_{t-1} are the feature vector and an information vector corresponding to the $(t-1)$-th time step of H^{Concat}, respectively. Later, with the hidden state $\overrightarrow{h_{t-1}}$ of the $(t-1)$-th time step serving as a signal, the model can be told how much of information vector I_{t-1} should be integrated into the LSTM memory, which is shown as below:

$$z = sigmoid(W_z \overrightarrow{h_{t-1}}) \tag{10}$$

Then, the LSTM memory cell can be updated by

$$\overrightarrow{M_t} = (1-z) \odot \overrightarrow{M_{t-1}} + z \odot I_{t-1} \tag{11}$$

where M_t is the state of the LSTM memory cell corresponding to t-th time step, with its update, the latent information contained by each time step h_t^{Concat} of H^{Concat} is continually integrated into the memory unit via the weight (i.e., signal z) calculated by the current hidden state $\overrightarrow{h_t}$ of LSTM. Last, we use the current memory cell state $\overrightarrow{M_t}$, the hidden state $\overrightarrow{h_{t-1}}$ of the previous time step, and the input feature vector h_t^{Concat} of the current time step to calculate and update hidden state $\overrightarrow{h_t}$ of the current time step as follows:

$$\overrightarrow{h_t} = LSTMCell(h_t^{Concat}, (\overrightarrow{h_{t-1}}, \overrightarrow{M_t})) \tag{12}$$

The calculation of the backward direction is the same as that of the forward direction; therefore, the output of the fusion LSTM is obtained as follows:

$$\begin{aligned} M_t &= [\overrightarrow{M_t}; \overleftarrow{M_t}], \\ h_t &= [\overrightarrow{h_t}; \overleftarrow{h_t}] \end{aligned} \tag{13}$$

Given that the length of the input sequence is L, the situation where M_L, the state of memory cell corresponding to the last time step, fuses the information of each time step of H^{Concat} is reasonable. Inspired by the deep residual network[33], we add M_L to each hidden state H_t to obtain the final fusion result.

$$\begin{aligned} h_t^{L\_Fusion} &= h_t + M_L, \\ H^{L\_Fusion} &= \{h_t^{L\_Fusion} | 1 \leqslant t \leqslant L\} \end{aligned} \tag{14}$$

3.5 Global view interaction

The above representation H^{L_Fusion} generated by fusing two local views can be regarded as another global view for the whole ABSA task because it contains and fuses the aspect boundary information extracted by H^{AE} and the sentiment polarity information extracted by H^{ASP}. A global view H^{UFT}, which contains more global information from UFT under the unified labeling scheme, also exists. The utilization of multiview learning, which can enhance the correlation of the two global views so that they can influence each other during the joint training stage, can help to make the learned representation more robust.

First, we still conduct information interaction and fusion between global views H^{UFT} and H^{L_Fusion} similar to the fusion of local views. For H^{UFT}, the self-attention mechanism can be used to enhance the representation as follows:

$$H^{UFT'} = Self\_Att(H^{UFT}, H^{UFT}) \tag{15}$$

Meanwhile, H^{UFT} and H^{L_Fusion} are respectively regarded as key and query in the interactive-attention mechanism to conduct information interaction below:

$$\begin{aligned} H^{L\_Fusion'} &= Inter\_Att(H^{L\_Fusion}, H^{UFT}), \\ H^{UFT''} &= Inter\_Att(H^{UFT}, H^{L\_Fusion}) \end{aligned} \tag{16}$$

where $H^{L\_Fusion'}$ refers to the representation obtained by treating H^{L_Fusion} as key, and H^{UFT} as query, generated from attaching the attention distribution score on the basis of the information of H^{UFT} to H^{L_Fusion}. $H^{UFT''}$ is just the opposite.

Second, similarly, we use the fusion LSTM to make the results obtained by the attention mechanism fully incorporated into the original representation:

$$\begin{aligned} H^{UFT\_Final} &= Fusion\_LSTM([H^{UFT}; H^{UFT'}; H^{L\_Fusion'}]), \\ H^{Fusion} &= Fusion\_LSTM([H^{L\_Fusion}; H^{UFT''}]) \end{aligned} \tag{17}$$

Note that the computational process of the above attention mechanism and the fusion LSTM is consistent with Eqs. (7) to (14) used in the local view fusion, which is unrepeated here.

So far, we obtain the final representations H^{Fusion} and H^{UFT_Final} after the interaction between the two global views H^{L_Fusion} and H^{UFT}. Such an operation is helpful to further enhance and constrain the process of maximizing the correlation of global views during the training stage.

3.6 Decoding layer and joint loss

Four representations are sent to the output layer to predict the labels of UFT, AE, and ASP tasks. CRF is applied after a linear transformation over local view H^{AE} and two global views H^{UFT_Final} and H^{Fusion}, as shown below:

$$z_t^{\mathcal{Q}} = W^{\mathcal{Q}} h_t^{\mathcal{Q}} + b^{\mathcal{Q}},$$
$$s(X, Y^{\mathcal{Q}}) = \sum_{t=1}^{L} (W_{y_{t-1}^{\mathcal{Q}} y_t^{\mathcal{Q}}} z_t^{\mathcal{Q}} + b_{y_{t-1}^{\mathcal{Q}} y_t^{\mathcal{Q}}}), \quad (18)$$
$$\text{s.t. } \mathcal{Q} \in \{Fusion, UFT\_Final, AE\}$$

where $h_t^{\mathcal{Q}}$ is the feature vector of $H^{\mathcal{Q}}$ at t-th time step, $\mathcal{Q} \in \{Fusion, UFT\_Final, AE\}$ means that $H^{\mathcal{Q}}$ here can be H^{Fusion}, H^{UFT_Final}, or H^{AE}. $s(X, Y^{\mathcal{Q}})$ represents the score of each possible label sequence $Y^{\mathcal{Q}}$ for the input sequence X. Hence, the conditional probability $p(Y^{\mathcal{Q}}|X;\theta)$ over all possible paths $\widetilde{Y}^{\mathcal{Q}}$ given X is calculated as

$$p(Y^{\mathcal{Q}}|X;\theta) = \frac{e^{s(X,Y^{\mathcal{Q}})}}{\sum_{\widetilde{Y}} e^{s(X,\widetilde{Y}^{\mathcal{Q}})}}, \quad (19)$$
$$\text{s.t. } \mathcal{Q} \in \{Fusion, UFT\_Final, AE\}$$

where θ represents the related parameters in the model.

For local view H^{ASP}, we use a linear layer and the softmax function instead of CRF to directly obtain the probability distribution of each time step on the labels.

$$p(Y^{ASP}|X;\theta) = Softmax(W^{ASP} H^{ASP}) \quad (20)$$

The reason is that the labels of the global views under the unified labeling scheme tend to have a strong dependency relationship, which should be considered when decoding. At this point, considering the labels of all tokens as a whole sequence for making predictions is advised. Similarly, for AE, an aspect is usually a chunk composed of multiple tokens. When the aspect is labeled, the correlation between the labels of tokens in its chunk should also be considered. Therefore, for global views H^{Fusion} and H^{UFT_Final} and local view H^{AE}, we adopt CRF to decode all the labels of tokens as a whole sequence and calculate the probability distribution over all possible label sequences to keep the dependency among the labels. For ASP, each label only reflects the sentiment polarity information of the corresponding token; therefore, the softmax function is used to predict the label of each token independently.

The joint loss function of the entire model is denoted as follows:

$$\mathcal{L} = \mathcal{L}^{UTF\_Final} + \mathcal{L}^{Fusion} + \mathcal{L}^{AE} + \mathcal{L}^{ASP} + \sigma CCA\_Loss(Z^{UTF\_Final}, Z^{Fusion}) \quad (21)$$

where $\mathcal{L}^{\mathcal{T}}$, $\mathcal{T} \in \{Fusion, UFT\_Final, AE, ASP\}$, is the negative log-likelihood loss between the label probability distribution $p(Y^{\mathcal{T}}|X;\theta)$ and the ground truth $\widehat{Y}^{\mathcal{T}}$:

$$\mathcal{L}^{\mathcal{T}} = -\sum_i \mathbb{I}(\hat{y}_i^{\mathcal{T}}) \log p(y_i^{\mathcal{T}}|x_i;\theta), \quad (22)$$
$$\text{s.t. } \mathcal{T} \in \{Fusion, UFT\_Final, AE, ASP\}$$

Inspired by Ref. [28], we add a CCA_Loss term to the joint loss function, so that the model can maximize the correlation between two global views while minimizing the loss of each task. CCA_Loss is defined as

$$CCA\_Loss = -corr(Z^{UFT\_Final}, Z^{Fusion}) \quad (23)$$

As mentioned above, UFT finally generates a global view, whereas another global view is generated from the fusion of the local views of AE and ASP. During training, we reduce CCA_Loss, that is, we enhance the correlation between two global views to let them influence each other to fine-tune the entire network structure, which helps the model learn more robust representations.

3.7 Output

Two results of aspect-polarity pairs can be obtained from the predicted labels Y^{UFT_Final} and Y^{Fusion} of the two global views, and the third one is generated by integrating the aspect term and sentiment polarity respectively extracted by AE and ASP auxiliary tasks. Here, we adopt the same method as Ref. [24]. Within the scope of each aspect term extracted by AE, the label of each token from ASP is counted. Then, the sentiment polarity corresponding to the label with the highest frequency is used as the sentiment polarity of the whole aspect. If each sentiment polarity appears the same number of times within the scope of an aspect term, then this aspect will be labeled as the polarity of its first token.

After obtaining the above three results under the unified labeling scheme, we simply employ the voting method to integrate these results and generate the final outputs for the end-to-end ABSA task.

4 Experiment

4.1 Datasets

We conduct experiments on three benchmark datasets comparing our proposed model with other baselines. The details of these datasets are shown in Tables 2 and 3.

D_L and D_R come from SemEval-2014[34], which mainly contain reviews from laptop domains and

Table 2 Numbers of sentences and aspect terms contained by three benchmark datasets.

Dataset	Train		Test	
	Sentence	Aspect terms	Sentence	Aspect terms
D_L	3045	2358	800	654
D_R	3041	3693	800	1134
D_T	2115	2608	235	604

Table 3 Numbers of aspects holding different sentiment polarities in three benchmark datasets.

Dataset		Train	Test	Total
D_L	POS	987	341	1328
	NEG	866	128	994
	NEU	460	169	629
	CONF	45	16	61
D_R	POS	2164	728	2892
	NEG	805	196	1001
	NEU	633	196	829
	CONF	91	14	105
D_T	POS	563	132	695
	NEG	218	48	266
	NEU	1827	424	2251

restaurant domains, respectively. D_T consists of English tweets collected by Ref. [35]. The ground truth of the aspects and corresponding sentiment polarities are also given in these datasets. We keep the official division of D_L and D_R for the training set and testing set. A five-fold cross-validation is used for each dataset to acquire a parameter set as good as possible.

4.2 Baselines

In order to demonstrate the effectiveness of performing the ABSA task in an end-to-end manner and the performance of our proposed end-to-end architecture, we compare our MTMVN model with the following existing pipeline and end-to-end baselines, respectively.

4.2.1 Pipeline methods

We compare our model with the following pipeline methods:

CRF-pipeline. Reference[35] uses CRF to deal with the sequence labeling task in a pipeline manner directly on the basis of semantic features.

NN-CRF-pipeline. Based on the improvements made in Ref. [35], Ref. [36] introduces word embeddings to represent the aspect and its context.

CMLA-ALSTM. CMLA is a multilayer attention network proposed in Ref. [8] for the co-extraction of aspect terms and opinion terms. ALSTM[12] is an attention-based LSTM network model for ASP in the case of known aspect terms. Here, CMLA and ALSTM are combined into a pipeline method. The latter performs ASP based on the aspect terms extracted from the former.

DECNN-ALSTM. DECNN[11] introduces domain knowledge by applying domain and general embeddings and perform AE based on the multi-layer stacked CNNs network structure. Here, it is also combined with ALSTM in the pipeline manner for the whole ABSA task.

4.2.2 End-to-end methods

We also compare our model with the following end-to-end approaches:

Sentiment-Scope. It is an end-to-end CRF model[37] to capture sentiment scopes by expanding the node types of CRF.

LSTM-CRF-LSTMc. This is a named entity recognition (NER) model proposed in Ref. [38] based on BiLSTM for feature extraction and CRF for decoding. Meanwhile, LSTM is used to learn and encode the char-level embeddings. Since the NER task is similar to the end-to-end ABSA task under the unified labeling scheme, both of which can be regarded as sequence labeling problems, this model is also used as an end-to-end approach to directly carry out end-to-end ABSA task.

LSTM-CRF-CNNc. The model[39] has the same structure as Ref. [38], except that the model uses CNN instead of LSTM to encode the char-level embeddings.

MNN. It is an end-to-end ABSA model proposed in Ref. [21]. Taking the concatenation of word-level and char-level embeddings as the output of the embedding layer, the model further encodes them by respectively using CNN and BiLSTM, and fuses the obtained two representations through the attention mechanism. Finally, it realizes the co-extraction of aspects and corresponding sentiment polarities under the unified labeling scheme via the fusion result.

4.3 Parameter setting and training details

4.3.1 Word embedding

Similar to the operation in Refs. [11, 23, 24], for each token in the input text sequence, we concatenate the general embedding and the domain embedding as the representation to be encoded by downstream encoders. The open-source Glove.840B.300D pretraining embeddings[32] are used to generate the general embeddings with 300 dimensions. The domain embeddings with 100 dimensions are initialized via the embeddings provided by Ref. [11], which are pretrained on the domain-specific corpus collected from Amazon reviews and Yelp reviews. Therefore, at the encoding layer, each token in the input text sequence is represented as a 400-dimensional vector.

4.3.2 Parameter settings

The hidden state size of BiGRU in the shared layer is 200, whose forward and backward results are concatenated,

so that the size of the output vector of the shared layer is the same as the embeddings with 400 dimensions. In the encoder for the UFT global task, the hidden state size of two BiGRU is 128. For AE, the stacked CNNs encoder has the same layers and dimension settings as Ref. [11]. Two CNN layers are found in the encoder for ASP, both have 128 filters, and their kernel sizes are 3 and 5. All the outputs of CNN above are processed by batch normalization method and activated by the ReLu function.

For the embedding layer and the encoding layer of UFT, dropout with $p = 0.5$ is employed. For CNN, the dropout rate is also 0.5 but is used after each CNN layer. All the attention in our MTMVN model adopts the multi-head attention mechanism[40] with $N = 8$, whereas the dropout rate is set as 0.25.

The weight matrices involved in the attention mechanism and BiGRU are initialized by Xavier uniform distribution[41], whereas for CNN, the default Kaiming uniform distribution[42] is employed for initialization.

4.3.3 Training settings

We use Adam as the optimizer with a batch size of 54. The maximum number of training epochs is set to 100. During training, a progressively decreasing dynamic learning rate is employed and is initially set to 0.003 for D_L and 0.001 for D_R and D_T. The hyperparameter σ used in the joint loss function to balance the contribution of CCA_Loss is set to 2, 2, and 1.5 for D_L, D_R, and D_T, respectively. The impact of this hyperparameter is further discussed below.

4.4 Experimental results and analysis

4.4.1 Main results

Table 4 shows the comparison results of our proposed MTMVN model with the baseline approaches on the above three benchmark datasets. Here, we calculate the F1 score[23] by considering both the aspect extraction result and the sentiment prediction result to measure the model performance. That is, only when the whole chunk of an aspect and its sentiment polarity are all predicted correctly can it be taken as a correct result.

As indicated in the results, our proposed MTMVN model achieves convincing results on all three datasets, which outperform other baselines on D_L, D_T, and D_R. Specifically, compared with the pipeline methods, our proposed MTMVN outperforms the F1 score of CMLA-ALSTM, whose performance is only below the DECNN-ALSTM in all pipeline baselines by 1.4%, 1.33%, and 1.42% on three benchmark datasets. Although the gap is small, our MTMVN also exceeds the DECNN-ALSTM, which is the best one among pipeline approaches. It indicates that dealing with the entire ABSA task in an end-to-end manner can achieve competitive performance to the pipeline manner.

Compared with the other end-to-end methods, MTMVN respectively improves by 0.37%, 0.17%, and 0.54% over the best performance in them on three benchmark datasets. Note that the improvements of 1.15%, 1.49%, and 1.72% on the three benchmark datasets over the F1 score of MNN are also achieved by our proposed MTMVN. MNN is a representative method for end-to-end ABSA under the idea of multiview fusion similar to our MTMVN. However, MNN performs end-to-end ABSA just under the unified labeling scheme, neglecting the consideration of the interaction between AE and ASP subtasks. Therefore, our model obtains a better performance than MNN. These results show the potential of combining a unified model under a unified labeling scheme with a joint model carrying out AE and ASP subtasks simultaneously via multitask learning.

4.4.2 Ablation study

To further test the effectiveness of the structure of our MTMVN, we conduct an ablation experiment with results shown in Table 5. In the model variants, "Only

Table 4 Comparison results in the F1 scores (%) of all the baselines and our proposed model for the whole ABSA task.

Method	Model	F1 scores (%)		
		D_L	D_R	D_T
Pipeline methods	CRF-pipeline	51.73	54.16	31.35
	NN-CRF-pipeline	53.23	61.03	45.08
	CMLA-ALSTM	53.68	63.87	46.47
	DECNN-ALSTM	54.87	65.13	47.24
End-to-End Methods	Sentiment-Scope	50.27	62.01	45.91
	LSTM-CRF-LSTMc	54.37	65.03	46.82
	LSTM-CRF-CNNc	54.71	64.29	47.35
	MNN	53.93	63.71	46.16
Ours	MTMVN	55.08	65.20	47.89

Table 5 Comparison results in the F1 scores (%) of different model variants.

Model Variants	F1 scores (%)		
	D_L	D_R	D_T
Only UFT	52.32	59.35	44.85
Only Fusion	52.73	61.69	44.20
Fusion+AE+ASP	53.48	62.36	45.64
UFT+Fusion+AE+ASP	54.12	63.77	46.78
The entire MTMVN model	55.08	65.20	47.89

UFT" means the single network branch for UFT, which is just composed of three layers of BiGRU (one for shared layer and the other two for UFT in the original model). And "Only Fusion" means a network structure consisting of the two network branches for AE and ASP, but only the fusion result will be output and involved in the loss function. The overall performance of these two model variants are not as good as the other model variants. This is because they can be directly seen as unified models without AE and ASP participating in the loss function as auxiliary tasks and the interaction between AE and ASP also is ignored under such a situation. Therefore, with the help of AE and ASP via multitask learning, the model variant "Fusion+AE+ASP", which only modifies "Only Fusion" to output the AE and ASP results and add their losses to the entire loss function, achieves better performance.

The comparison results between model variants "UFT+Fusion+AE+ASP" and "Fusion+AE+ASP" prove that the entire model can further benefit from the multitask learning and the information interaction of global views by adding UFT as another task and another global view.

Finally, our proposed MTMVN achieves superior performance. It not only proves the effectiveness of the overall structure of MTMVN but also proves the effectiveness of our introduced *CCA_Loss* term through the comparison with the model variant "UFT+Fusion+AE+ASP" whose only difference is lacking this term in the loss function comparing to the entire MTMVN.

4.4.3 Impact of hyperparameter σ

We investigate the impact of hyperparameter σ of the *CCA_Loss* term in the joint loss function of the model. Based on the observation of the proportion of each part in the whole loss, the experiment is conducted on D_L and D_R by varying σ from 0.5 to 5 with the step of 5. As illustrated in Figs. 2 and 3, we observe that when the value of σ is greater than 3, the performance of

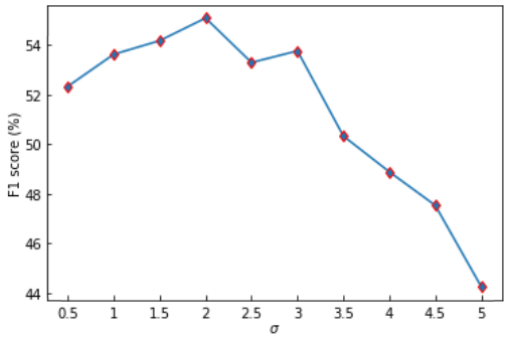

Fig. 2 F1 score (%) on D_L with different σ.

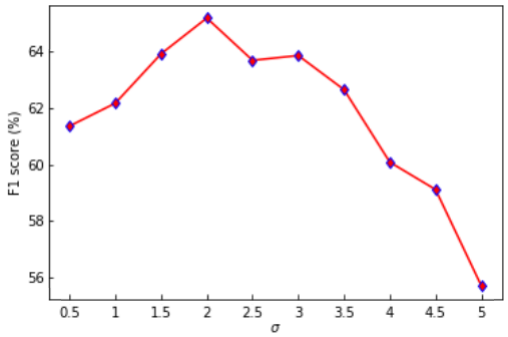

Fig. 3 F1 score (%) on D_R with different σ.

the model on D_L and D_R will decline rapidly with the increase of σ. We also find that 2 is the best choice for the proposed method to balance the contribution of the *CCA_Loss* term to the whole loss function on D_L and D_R. Therefore, σ is set to 2 on these two datasets in our experiment, and the same method is used to obtain the best choice of 1.5 for D_T.

4.5 Case analysis

Table 6 shows some examples with gold labels and the corresponding predictions of CMLA_ASLTM, MNN, and MTMVN. In the first row, we can see that our MTMVN extracts the complete aspect term and predicts its sentiment polarity, whereas the two other methods either obtain the incomplete aspect term or incorrect polarity prediction.

The comparison results in the second, third, and fourth rows all reflect the better performance of our MTMVN, whether in the extraction of multiple aspects contained by one sentence or in the accurate prediction of polarity.

The result in the fifth row catches our attention, in which our MTMVN extracts the correct aspect terms but obtains the wrong prediction on polarity "CONF" even if the two other methods do not obtain the complete aspect chunks. This result indicates that our proposed MTMVN still has room for improvement in predicting aspects with the sentiment polarity of conflict.

Table 6 Case analysis on CMLA_ALSTM, MNN, and MTMVN; ✗ and ✓ mean wrong and correct predictions, respectively.

Example	Model		
	CMLA-ALSTM	MNN	Our MTMVN
No [installation disk (dvd)]$_{\text{NEU}}$ is included.	[installation disk]$_{\text{NEU}}$ (✗)	[installation disk]$_{\text{NEG}}$, [dvd]$_{\text{NEG}}$ (✗)	[installation disk]$_{\text{NEU}}$, [dvd]$_{\text{NEU}}$ (✓)
[Works]$_{\text{POS}}$ well, and I am extremely happy to be back to an [apple os]$_{\text{POS}}$.	[Works]$_{\text{POS}}$ (✓), [apple os]$_{\text{POS}}$ (✓)	None (✗)	[Works]$_{\text{POS}}$ (✓), [apple os]$_{\text{POS}}$ (✓)
Straight-forward, no surprises, very decent [japanese food]$_{\text{POS}}$.	[japanese food]$_{\text{NEG}}$ (✗)	None (✗)	[japanese food]$_{\text{POS}}$ (✓)
Try the [rose roll]$_{\text{POS}}$ (not on [menu]$_{\text{NEU}}$).	[rose roll]$_{\text{POS}}$ (✓), [menu]$_{\text{POS}}$ (✗)	[rose roll]$_{\text{POS}}$ (✓), None (✗)	[rose roll]$_{\text{POS}}$ (✓), [menu]$_{\text{NEU}}$ (✓)
The only task that this computer would not be good enough for be [gaming]$_{\text{NEG}}$, otherwise the [integrated Intel 4000 graphics]$_{\text{CONF}}$ work well for other tasks.	[gaming]$_{\text{NEG}}$ (✓), [graphics]$_{\text{POS}}$, [tasks]$_{\text{NEG}}$ (✗)	[gaming]$_{\text{NEG}}$ (✓), [graphics]$_{\text{POS}}$ (✗)	[gaming]$_{\text{NEG}}$ (✓), [integrated Intel graphics]$_{\text{NEG}}$ (✗)

5 Conclusion

We present an end-to-end neural network model to conduct ABSA by combining the unified approaches and the joint approaches under the ideas of multiview learning and multitask learning. By taking the whole ABSA under a unified labeling scheme as the main task UFT, and taking the two branches AE and ASP of joint models as auxiliary tasks, our proposed MTMVN model realizes the information interaction among different tasks via mutlitask learning. Meanwhile, we regard the representation obtained from UFT as the global view with the representations from AE and ASP as two local views. Based on the operations of view fusion and view correlation enhancement under the idea of multiview learning, our MTMVN model can further improve the above representations and fine-tune the three network branches during the training stage to achieve a good performance. The experimental results on the three benchmark datasets illustrate the effectiveness of MTMVN, which outperforms the baselines on the end-to-end ABSA task. In the future, we will consider introducing the combination of graph neural network[43] and dependency tree into the end-to-end ABSA model to obtain a better performance.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61976247).

References

[1] M. Bouazizi and T. Ohtsuki, Multi-class sentiment analysis on twitter: Classification performance and challenges, *Big Data Mining and Analytics*, vol. 2, no. 3, pp. 181–194, 2019.

[2] B. Liu, *Sentiment Analysis and Opinion Mining*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2012.

[3] P. Jiang, C. X. Zhang, H. P. Fu, Z. D. Niu, and Q. Yang, An approach based on tree kernels for opinion mining of online product reviews, in *2010 IEEE Int. Conf. Data Mining*, Sydney, Australia, 2010, pp. 256–265.

[4] W. Jin and H. H. Ho, A novel lexicalized hmm-based learning framework for web opinion mining, in *Proc.* 26^{th} *Int. Conf. Machine Learning*, Montreal, Canada, 2009, pp. 465–472.

[5] N. Jakob and I. Gurevych, Extracting opinion targets in a single- and cross-domain setting with conditional random fields, in *Proc. 2010 Conf. Empirical Methods in Natural Language Proc.*, Boston, MA, USA, 2010, pp. 1035–1045.

[6] F. T. Li, C. Han, M. L. Huang, X. Y. Zhu, Y. J. Xia, S. Zhang, and H. Yu, Structure-aware review mining and summarization, in *Proc.* 23^{rd} *Int. Conf. Computational Linguistics*, Beijing, China, 2010, pp. 653–661.

[7] S. Poria, E. Cambria, and A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, *Knowl. Based Syst.*, vol. 108, pp. 42–49, 2016.

[8] W. Y. Wang, S. J. Pan, D. Dahlmeier, and X. K. Xiao, Coupled multi-layer attentions for co-extraction of aspect and opinion terms, in *Proc.* 31^{st} *AAAI Conf. Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 3316–3322.

[9] H. Ye, Z. C. Yan, Z. C. Luo, and W. H. Chao, Dependency-tree based convolutional neural networks for aspect term extraction, in *Pacific-Asia Conf. Knowledge Discovery and Data Mining*, H. Ye, Z. Yan, Z. Luo, and W. Chao, Eds. Cham, Germany: Springer, 2017, pp. 350–362.

[10] H. S. Luo, T. R. Li, B. Liu, B. Wang, and H. Unger, Improving aspect term extraction with bidirectional dependency tree representation, *IEEE/ACM Trans. Audio Speech Language Proc.*, vol. 27, no. 7, pp. 1201–1212, 2019.

[11] H. Xu, B. Liu, L. Shu, and P. S. Yu, Double embeddings and

CNN-based sequence labeling for aspect extraction, in *Proc. 56th Ann. Meeting of the Association for Computational Linguistics*, Melbourne, Australia, 2018, pp. 592–598.

[12] Y. Q. Wang, M. L. Huang, X. Y. Zhu, and L. Zhao, Attention-based LSTM for aspect-level sentiment classification, in *Proc. 2016 Conf. Empirical Methods in Natural Language Proc.*, Austin, TX, USA, 2016, pp. 606–615.

[13] D. Y. Tang, B. Qin, and T. Liu, Aspect level sentiment classification with deep memory network, in *Proc. 2016 Conf. Empirical Methods in Natural Language Proc.*, Austin, TX, USA, 2016, pp. 214–224.

[14] F. F. Fan, Y. S. Feng, and D. Y. Zhao, Multi-grained attention network for aspect-Level sentiment classification, in *Proc. 2018 Conf. Empirical Methods in Natural Language Proc.*, Brussels, Belgium, 2018, pp. 3433–3442.

[15] P. Chen, Z. Q. Sun, L. D. Bing, and W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in *Proc. 2017 Conf. Empirical Methods in Natural Language Proc.*, Copenhagen, Denmark, 2017, pp. 452–461.

[16] D. H. Ma, S. J. Li, X. D. Zhang, and H. F. Wang, Interactive attention networks for aspect-level sentiment classification, in *Proc. 26th Int. Joint Conf. Artificial Intelligence*, Melbourne, Australia, 2017, pp. 4068–4074.

[17] B. X. Huang, Y. L. Ou, and K. M. Carley, Aspect level sentiment classification with attention-over-attention neural networks, in *Int. Conf. Social Computing, Behavioral-Cultural Modeling & Prediction and Behavior Representation in Modeling and Simulation* (*SBP-BRiMS*), R. Thomson, C. Dancy, A. Hyder, and H. Bisgin, Eds. Cham, Germany: Springer, 2018, pp. 197–206.

[18] L. S. Li, Y. Liu, and A. Q. Zhou, Hierarchical attention based position-aware network for aspect-level sentiment analysis, in *Proc. 22nd Conf. Computational Natural Language Learning*, Brussels, Belgium, 2018, pp. 181–189.

[19] Y. W. Song, J. H. Wang, T. Jiang, Z. Y. Liu, and Y. H. Rao, Attentional encoder network for targeted sentiment classification, arXiv preprint arXiv: 1902.09314v2, 2019.

[20] X. Li, L. D. Bing, W. Lam, and B. Shi, Transformation networks for target-oriented sentiment classification, in *Proc. 56th Ann. Meeting of the Association for Computational Linguistics*, Melbourne, Australia, 2018, pp. 946–956.

[21] F. X. Wang, M. Lan, and W. T. Wang, Towards a one-stop solution to both aspect extraction and sentiment analysis tasks with neural multi-task learning, in *Proc. 2018 Int. Joint Conf. Neural Networks* (*IJCNN*), Rio de Janeiro, Brazil, 2018, pp. 1–8.

[22] X. Li, L. D. Bing, P. J. Li, and W. Lam, A unified model for opinion target extraction and target sentiment prediction, *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 6714–6721, 2019.

[23] R. D. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, An interactive multi-task learning network for end-to-end aspect-based sentiment analysis, in *Proc. 57th Ann. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 504–515.

[24] H. S. Luo, T. R. Li, B. Liu, and J. B. Zhang, DOER: Dual cross-shared RNN for aspect term-polarity co-extraction, in *Proc. 57th Ann. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 591–601.

[25] Y. Kim, Convolutional neural networks for sentence classification, in *Proc. 2014 Conf. Empirical Methods in Natural Language Proc.*, Doha, Qatar, 2014, pp. 1746–1751.

[26] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 1724–1734.

[27] L. Sha, X. D. Zhang, F. Qian, B. B. Chang, and Z. F. Sui, A Multi-view fusion neural network for answer selection, in *32nd AAAI Conf. Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 5422–5429.

[28] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, Deep canonical correlation analysis, in *Proc. 30th Int. Conf. Machine Learning*, Atlanta, GA, USA, 2013, pp. 1247–1255.

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

[30] J. Weston, S. Chopra, and A. Bordes, Memory networks, arXiv preprint arXiv: 1410.3916, 2014.

[31] Y. M. Cui, Z. P. Chen, S. Wei, S. J. Wang, T. Liu, and G. P. Hu, Attention-over-attention neural networks for reading comprehension, in *Proc. 55th Ann. Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017, pp. 593–602.

[32] J. Pennington, R. Socher, and C. D. Manning, Glove: Global vectors for word representation, in *Proc. 2014 Conf. Empirical Methods in Natural Language Proc.*, Doha, Qatar, 2014, pp. 1532–1543.

[33] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[34] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, Semeval-2014 task 4: Aspect based sentiment analysis, in *Proc. 8th Int. Workshop on Semantic Evaluation* (*SemEval 2014*), Dublin, Ireland, 2014, pp. 27–35.

[35] M. Mitchell, J. Aguilar, T. Wilson, and B. Van Durme, Open domain targeted sentiment, in *Proc. 2013 Conf. Empirical Methods in Natural Language Proc.*, Seattle, WA, USA, 2013, pp. 1643–1654.

[36] M. S. Zhang, Y. Zhang, and D. T. Vo, Neural networks for open domain targeted sentiment, in *Proc. 2015 Conf. Empirical Methods in Natural Language Proc.*, Lisbon, Portugal, 2015, pp. 612–621.

[37] H. Li and W. Lu, Learning latent sentiment scopes for entity-level sentiment analysis, in *Proc. 31st AAAI Conf. Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 3482–3489.

[38] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, Neural architectures for named entity recognition, in *Proc. 2016 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, CA, USA, 2016, pp. 260–270.

[39] X. Z. Ma and E. Hovy, End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF, in *Proc. 54th Annu. Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 1064–1074.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł Kaiser, and I. Polosukhin, Attention is all you need, in *Proc. 31st Conf. Neural Information Proc. Systems*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[41] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proc. 13th Int. Conf. Artificial Intelligence and Statistics*, Sardinia, Italy, vol. 9, pp. 249–256.

[42] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in *2015 IEEE Int. Conf. Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1026–1034.

[43] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, The graph neural network model, *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

Yan Yang received the BS and MS degrees from Huazhong University of Science and Technology, Wuhan, China in 1984 and 1987, respectively. She received the PhD degree from Southwest Jiaotong University, Chengdu, China in 2007. From 2002 to 2003 and 2004 to 2005, she was a visiting scholar with the University of Waterloo, Waterloo, Canada. She is currently a professor and vice dean at the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China. Her current research interests include multi-view learning, big data analysis and mining, ensemble learning, semi-supervised learning, and cloud computing.

Yong Bie received the BS degree from Southwest Jiaotong University, Chengdu, China in 2018 and now he is a master candidate of Southwest Jiaotong University. His current research interests include data mining, natural language processing, and machine learning.