World Academy of Science, Engineering and Technology
International Journal of Cognitive and Language Sciences
Vol:16, No:10, 2022

Aspect-Level Sentiment Analysis with Multi-Channel and Graph Convolutional Networks

Jiajun Wang, Xiaoge Li

Abstract**—**The purpose of the aspect-level sentiment analysis task is to identify the sentiment polarity of aspects in a sentence. Currently, most methods mainly focus on using neural networks and attention mechanisms to model the relationship between aspects and context, but they ignore the dependence of words in different ranges in the sentence, resulting in deviation when assigning relationship weight to other words other than aspect words. To solve these problems, we propose an aspect-level sentiment analysis model that combines a multi-channel convolutional network and graph convolutional network (GCN). Firstly, the context and the degree of association between words are characterized by Long Short-Term Memory (LSTM) and self-attention mechanism. Besides, a multi-channel convolutional network is used to extract the features of words in different ranges. Finally, a convolutional graph network is used to associate the node information of the dependency tree structure. We conduct experiments on four benchmark datasets. The experimental results are compared with those of other models, which shows that our model is better and more effective.

Keywords**—**Aspect-level sentiment analysis, attention, multi-channel convolution network, graph convolution network, dependency tree.

I. Introduction

THE purpose of aspect-level sentiment analysis (ASA) is to identify the sentiment of each aspect of comments in an automated way. ASA can be summarized into two tasks: (1) to extract the aspects of the target sentence; (2) to classify the sentiment of aspects. In this paper, we mainly focus on the sentiment classification task of aspects.

So far, the problem of sentiment classification has been solved by many methods. The early methods are based on dictionaries, but the construction and updating of dictionaries are very complicated and the generalization ability is poor. Support vector machine [1], decision tree [2] and Naive Bayesian [3] are also used to solve classification problems, but they are limited by the poor processing ability and generalization ability of complex texts. Due to the rise of deep learning and its powerful modeling ability, many people use neural networks to build models to establish the relationship between aspect words and context. Studies [4], [5] show that this method can strengthen the degree of association between aspect and other words. After that, more improved methods appeared on the basis of neural network, such as adding attention mechanism [6]-[9] and integrating the relative position relationship [10], the document knowledge [11] and commonsense knowledge [12] of aspects. These methods make the effect of aspects-level sentiment classification better. However, most of them ignore important features of words and word dependencies in sentences. Although the attention mechanism allows the encoder to obtain associations between aspects and other words during encoding, it is difficult to capture the dependencies between words. These problems can lead to significant deviations in giving weight to words other than aspect words in the sentence relative to aspects words.

In order to solve these problems, the dependency tree structure is introduced into the task. Dependency trees can shorten the path between aspect and evaluation words, and can capture dependencies between words. For example, "I ate the salad in this restaurant, that was too much sauce in it, so it tasted too sweet," using attention mechanism and relative position distance to associate aspect words with evaluation words, it will be difficult to connect salad with its evaluation word (sweet) for the above sentence. However, using syntactic dependency-based analysis can shorten the direct distance between salad and sweet, so as to get better results in the later modeling. Graph convolution neural network [13] can capture the local relationship of nodes in the graph by combining the characteristics of nodes. The dependency tree is used as input and encoded by bidirectional LSTM neural network. The results are used as the characteristics of graph convolution neural network nodes, and then graph convolution neural network is used for aspects-level sentiment classification [14]. After that, there are more and more fusion based on dependency tree and neural network [15]-[18]. However, when encoding node features, they only focus on words and dependencies between words in the same range, lacking dependencies in different ranges. In the process of position fusion, the relative position is used instead of the word and the path between words in the dependency tree, which will give wrong weight to some irrelevant words.

In this paper, we combine multi-channel convolution and graph convolution to solve the problem of word dependence in different ranges, and also solve the optimization problem of assigning weight between words.

The main works of this paper are as follows: We propose an ASA model. The multi-channel convolution neural networks and graph convolution neural networks are combined for aspects-level sentiment classification. Text convolution neural network represents the text in the form of n-gram, so as to extract the local features. The graph convolution neural network is used to calculate the aspect words and context interactively. At the same time, the complex and irregular syntactic structure

Jiajun Wang and Xiaoge Li are with NLP Laboratory, Xi'an University of Posts & Telecommunications, Xi'an, China (e-mail: Jiajunwang1999@163.com, xiaoge.li@gmail.com).

can be processed to obtain the long-distance syntactic information corresponding to the aspect words.

We replace the traditional relative position distance with the path distance of other words corresponding to aspect words in the dependency tree. This method solves the problem that the relative position distance cannot be correctly represented for complex and aspect words depending on long-distance information.

We combine multi-channel convolutional neural network and LSTM in different orders and compared them both experimentally under the same conditions. The results show that the best results are obtained by first using LSTM for context encoding and then using multi-channel convolutional neural networks to extract features in different ranges.

We compare the experimental results of different number of channels and the number of convolution kernels, and also compare the effect of different layers of GCN on the experimental results. These comparative experiments allow us to choose the optimal parameters for the experiments.

II. Related Work

Sentiment classification can be divided into text level, sentence level and aspect level. In contrast, ASA is more widely used and studied than the other two. It belongs to the fine-grained task of sentiment analysis, which aims to analyze the sentiment polarity of specific aspects in sentences. For example, “The food in this restaurant is delicious”. Through the ASA of food, it can be concluded that its sentiment polarity is positive. For ASA, attention mechanism and neural network are used to model in the early stage, and the following models are obtained: ATAE-LSTM [4] splice word embedding and aspect embedding, and then passes the result through attention; MEMNET [5] uses multi-layer computing layers, each layer includes an attention layer and a linear layer; IAN [8] uses the attention mechanism to model aspects and contexts; RAM [6] (Recurrent Attention Network on Memory) is used to extract sentiment information separated by long distance. After that, the basic neural network has been improved and innovated, and many new models have been obtained. PF-CNN [19] uses a new parametric convolutional neural network. MGAN [7] can learn the representation containing sentence and aspect related information, integrate it into the multi granularity sentence modeling process, and finally get a comprehensive sentence representation. TRANSCAP [20] proposes a transfer capsule network model that transforms document level knowledge into aspect-level sentiment classification. IACAPSNET [21] proposes a capsule network with Interactive Attention. Although these models have achieved good results in ASA tasks, they do not make use of the dependency between words.

In order to model the dependency of words in ASA, dependency tree and GCN are introduced. After that, many people modeled on this basis. CDT [14] uses LSTM to represent the characteristics of a sentence, and further improves the embedded GCN to directly operate the dependency tree of the sentence. ASGCN [16] passes the features encoded by LSTM, through GCN layer, and then carries result operation with the initial coding. AEGCN [18] proposed in this paper is mainly composed of multi head attention and an improved GCN based on sentence dependency tree. kumaGCN [22] integrates graph convolution neural network with gate and attention mechanism.

III. The Proposed Model

The architecture of our model (TANGCN) is shown in Fig. 1.

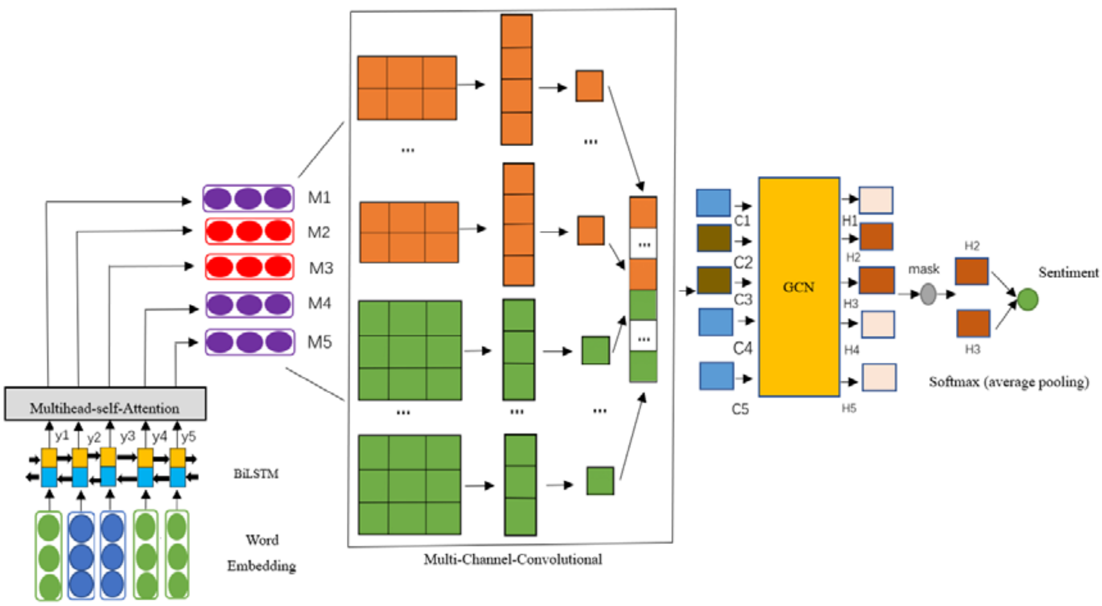

Fig. 1 Model architecture

A. Relative Graph Path of Aspects

In the past, most people, e.g. [10], used relative position to determine the positional relationship between aspects words and other words. For example, “This computer screen is very cool." The relative positional relationship between aspect (screen) and other words is shown in Fig. 2.

This computer screen looks very cool.

-2 -1 0 1 2 3 4

Fig. 2 Relative position of aspects

We propose to replace the previous relative position annotation with the path of aspects and other words in the dependency tree. The above sentence is represented in the form of a graph in the form of a dependency tree as shown in Fig. 3.

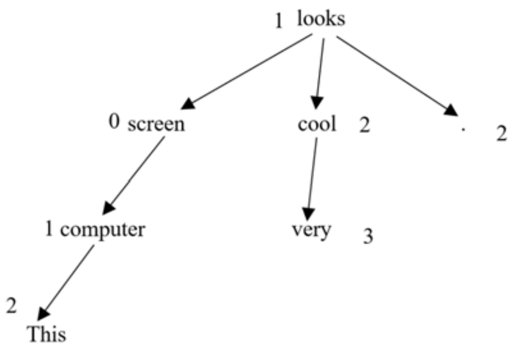

Fig. 3 Dependency tree

From the graph structure of Fig. 3, a new position code can be obtained as shown in Fig. 4. It is embedded into word vector.

This computer screen looks very cool.

2 1 0 1 3 2 2

Fig. 4 Relative graph path of aspects

B. Word Embedding

The 300-dimensional word vector of pre-trained Glove, 50-dimensional part of speech representation and 50-dimensional path representation are used to represent the input sentence. The sentence is represented by $X = \{X_1, X_2, X_3, \cdots, X_{n-1}, X_n\}$. The aspect is represented by $A = \{A_1, A_2, A_3, \cdots, A_{t-1}, A_t\}$. n is the length of the sentence and t is the length of the aspect word. The dimension of word embedding is 400.

C. Bi-LSTM Layer

LSTM can only predict the output of the next time according to the timing information of the previous time. But on most issues, we should pay attention not only to the previous information, but also to the future information, so bidirectional LSTM (BiLSTM) is introduced. The BiLSTM structure is shown in Fig. 5.

The core structure of BiLSTM can be regarded as an ordinary unidirectional LSTM, which is divided into two directions, one is forward with the input timing and the other is reverse with the input timing. As shown in Fig. 5, after the sentence $X = \{X_1, X_2, X_3, \cdots, X_{n-1}, X_n\}$ is input to BiLSTM, the forward LSTM represents the learned hidden state as $\{\overrightarrow{y_1}, \overrightarrow{y_2}, \overrightarrow{y_3}, \cdots, \overrightarrow{y_{n-1}}, \overrightarrow{y_n}\}$ and the hidden state learned by reverse LSTM is $\{\overleftarrow{y_1}, \overleftarrow{y_2}, \overleftarrow{y_3}, \cdots, \overleftarrow{y_{n-1}}, \overleftarrow{y_n}\}$. Finally, the two are spliced to get $Y = \{y_1, y_2, y_3, \cdots, y_{n-1}, y_n\}$. In this way, the context information corresponding to the aspect word is captured.

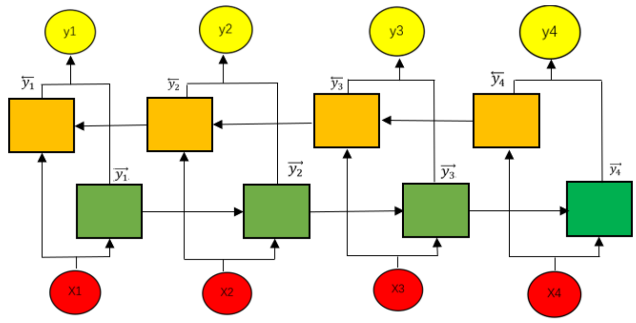

Fig. 5 BiLSTM structure

D. Multi-Head-Self-Attention Layer

Multi-head self attention is an attention mechanism that can operate in parallel in space. In this paper, $Y = \{y_1, y_2, y_3, \cdots, y_{n-1}, y_n\}$ is regarded as an input feature. After passing through the attention layer, a new feature $M = \{M_1, M_2, M_3, \cdots, M_{n-1}, M_n\}$ containing the degree of association between words is obtained. Firstly, the input features Y are multiplied by three matrices (B1, B2 and B2) to obtain three matrices (Q, K and V).

$$Q = YB1 \quad (1)$$

$$K = YB2 \quad (2)$$

$$V = YB3 \quad (3)$$

Secondly, the definition of attention is given.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

The formula (4) is actually the weighted calculation of V. The weight is softmax (*) before V. In softmax, the similarity score is obtained by the dot product operation of Q and K, and then we use $\sqrt{d_k}$ to adjust size. The multi-head-self attention mechanism is obtained by linearly changing the initial Q, K and V matrix to obtain the following matrix.

$$Q = \{Q1, Q2, Q3, \cdots, Q_n\} \quad (5)$$

$$K = \{K1, K2, K3, \cdots, K_n\} \quad (6)$$

$$V = \{V1, V2, V3, \cdots, V_n\} \quad (7)$$

The h is the number of heads. Each head does not share a parameter matrix. We splice the results of attention calculation on h heads to get the final result.

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (8)$$

$$MultiHead(Q, K, V) = Concat(head_1, \cdots, head_h) \quad (9)$$

E. Multi-Channel-Convolutional Layer

We use two types of convolution kernels in the convolutional layer to extract the associations between words in different scopes. Their width is equal to the dimension of the input word vector. Their heights are 2 and 3 respectively. The number of convolution kernels is 150. In this paper, $M = \{M_1, M_2, M_3, \cdots, M_{n-1}, M_n\}$ is regarded as an input feature. After passing through this layer, a new feature $C = \{C_1, C_2, C_3, \cdots, C_{n-1}, C_n\}$ is obtained. Multi-Channel-Convolutional structure is shown in Fig. 6.

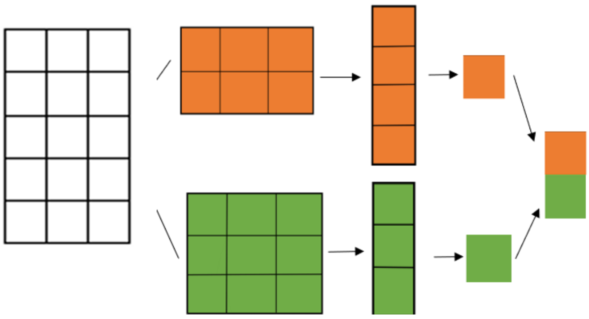

Fig. 6 Multi-Channels-Convolutional structure

We set the number of input features n. The input matrix is $M_{[n \times word\ embedding_{dim}]}$. $A_{[i:j]}$ stands for lines i to j. The convolution operation can be expressed by (10); w is convolution kernel; h is the height of the convolution kernel.

$$o_i = w \cdot A_{[i:i+h-1]}, i = 1,2, \ldots, n-h+1 \tag{10}$$

The obtained result is added with bias b, and then activated using the activation function to obtain the desired feature. Finally, max pooling is used for pooling.

$$c_i = f_{(o_i+b)} \tag{11}$$

F. GCN Layer

GCN is suitable for processing graph structure data with rich correlation information. According to the feature representation $C = \{C_1, C_2, C_3, \cdots, C_{n-1}, C_n\}$ of the previously generated words and the dependency tree generated according to the sentences, the information required by the graph convolution neural network can be obtained. The dependency tree can be seen as a graph $G = (\mathrm{V}, \mathrm{E})$. Nodes V represent words, and edges E represent the relationship between words. $C = \{C_1, C_2, C_3, \cdots, C_{n-1}, C_n\}$ represents the characteristic representation of each word. By constructing $|V| \times |V|$ the adjacency matrix A can be obtained. $A_{\mathrm{ij}} = 1$ if node i is connected to node j, otherwise $A_{\mathrm{ij}} = 0$. In order to make the GCN model and embed nodes more effectively, each node in the graph is allowed to have self-loops.

$$\tilde{A} = A + I \tag{12}$$

where $\tilde{A}$ is the adjacency matrix plus the identity matrix I of the graph. The graph convolution of a node can be described as:

$$\alpha^i = (\textstyle\sum_{j=1}^{n} \tilde{A}_{ij})^{-1} \tag{13}$$

$$h_i^{(k+1)} = \sigma\left(\textstyle\sum_{j=1}^{n} \alpha^i \tilde{A}_{ij}(W^k h_j^{(k)} + b^{(k)})\right) \tag{14}$$

where W^k is the weight matrix, $b^{(k)}$ is the offset vector, σ is a nonlinear function. $h_j^{(k)}$ is the hidden state of node j after passing through the k-1-layer GCN. α^i is the reciprocal of the degree of node i in the graph. After passing through the k-layer GCN, we get the final output $H = \{h_1, h_2, h_3, \cdots, h_{n-1}, h_n\}$ of the k-layer. Then we mask it to get the feature representation $A = \{a_1, a_2, a_3, \cdots, a_{t-1}, a_t\}$ of the aspect word we need where t is the length of aspect word. Mask function is an operation function that multiplies input and mask matrix.

$$\{a_1, a_2, \cdots, a_{t-1}, a_t\} = MASK(\{h_1, h_2, \cdots, h_{n-1}, h_n\}) \tag{15}$$

This feature A represents the fusion of context related information extracted by LSTM, the degree of association between words obtained by attention mechanism, the feature relationship between words in different ranges extracted by multi-channel convolution and the information aggregated by GCN. We use average pooling to average the information in the aspect word vector to obtain the final feature representation.

$$L = Average\ pooling(\{a_1, a_2, a_3, \cdots, a_{t-1}, a_t\}) \tag{16}$$

Finally, the feature representation is input to the softmax layer for sentiment distribution probability calculation where r is the category of classification. W_p and b_p are the learned weight matrix and bias, respectively.

$$S = Softmax(W_p L + b_p) \tag{17}$$

IV. Model Training

This model is trained by the standard gradient descent algorithm with the cross-entropy loss.

$$Loss = -\textstyle\sum_i \sum_{j \in J} y_i^j \log\ \hat{y}_i^j \tag{18}$$

where i represents the subscript of the i-th aspect sentence and j represents the sentiment category of the j-th sample sentence. y represents the true sentiment distribution of the sentence. $\hat{y}$ represents the predicted sentiment distribution of the sentence.

V. Experiments

A. Datasets

In order to prove the superiority and effectiveness of TANGCN, we evaluate the performance of this model on restaurant reviews (Rest14; Rest16 [23]), laptop reviews (Laptop14) [24] and Twitter reviews [25]. The details of the experimental data are shown in Table I.

B. Parameters

In this experiment, we use a 96-dimensional LSTM embedding for each word. The dependency tree structure of sentences is developed by Stanford parser. The batch size is 32. The number of self-multi-head attention heads is 8. Multi-channel convolution uses two convolution kernels, all of which have widths equal to the dimension of the word vector, and heights of 2 and 3 respectively. The number of convolution kernels is 150. The number of GCN layers is 2.

TABLE I
DISTRIBUTION OF SAMPLES BY CLASS LABELS ON BENCHMARK DATASETS

Polarity	Dataset	Rest14	Laptop14	Rest16	Twitter
Positive	Train	2164	976	1657	1507
	Test	727	337	611	172
Neutral	Train	637	455	101	3016
	Test	196	167	44	336
Negative	Train	807	851	748	1528
	Test	196	128	204	169

C. Effect of CNN and LSTM Order

Multi-channel-Convolutional-LSTM: Firstly, the local features of the text are extracted by Multi-Channel-Convolutional Networks, and then the long-distance features of these local features are extracted by LSTM.

LSTM-Multi-Channel- Convolutional Networks: Firstly, the long-distance features of the text are extracted by LSTM to obtain the new text fused with the context, and then the local features of the new text are extracted by Multi-Channel-Convolutional Networks. The results on rest16 and twitter are shown in Figs. 7 and 8

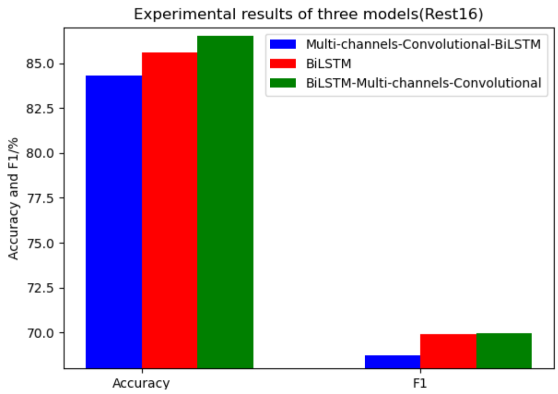

Fig. 7 Model effect comparison (Rest16)

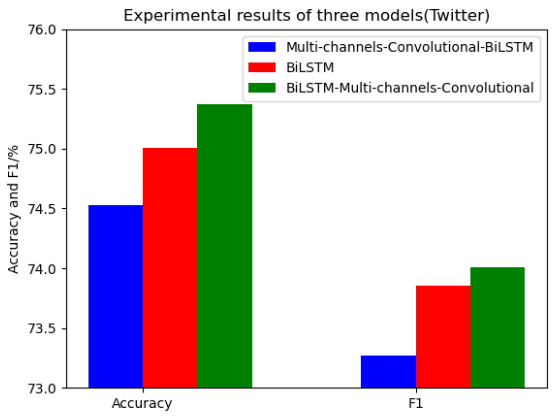

Fig. 8 Model effect comparison (Twitter)

From the experimental data, it can be concluded that the effect of LSTM-Multi-Channel- Convolutional Networks is the best and Multi-Channel-Convolutional-LSTM is the worst.

D. Effect of Different Convolution Kernel Size and Number

The size and number of convolution kernels have a great impact on the text convolution process using multi-channel convolution networks. In this paper, the experimental results of multi-channel convolution kernel and single channel convolution are compared. We use the width of the convolution kernel equal to the dimension of the word vector. We use convolution kernels with heights of 2, 3 and 4 to combine to obtain two multi-channel convolutional layers. One combination is convolution kernel heights 2 and 3 ([2,3]), and the other is convolution kernel heights 2, 3 and 4 ([2,3,4]). For single-channel convolutional layers, we use a convolutional kernel height of 2 ([2]). The experimental data are Twitter and Rest14. In order to compare the effect of different numbers of convolution kernels on our experimental results, in the case of a combination of kernel heights of 2 and 3 ([2,3]), we conduct experimental comparisons with the number of convolution kernels of 50, 100, 150 and 200 respectively. The experimental results are shown in Figs. 9 and 10.

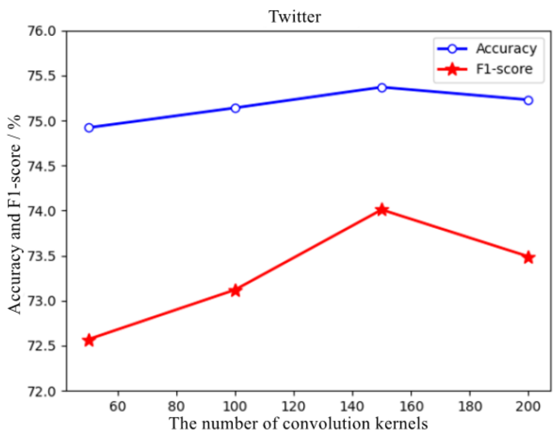

Fig. 9 Effect of different numbers of convolution kernels (Twitter)

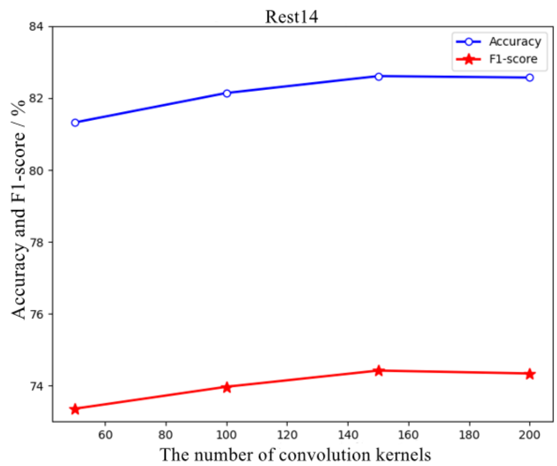

Fig. 10 Effect of different numbers of convolution kernels (Rest14)

It can be seen from Figs. 9 and 10 that under the same number of channels, the best result is to choose 150 convolution kernels for the four types. In order to compare the effect of different size of convolution kernels on our experimental results, in the case of using 150 convolution kernels, we conduct experimental comparisons with the size of convolution kernels of [2], [2,3] and [2,3,4] respectively. The experimental results are shown in Figs. 11 and 12.

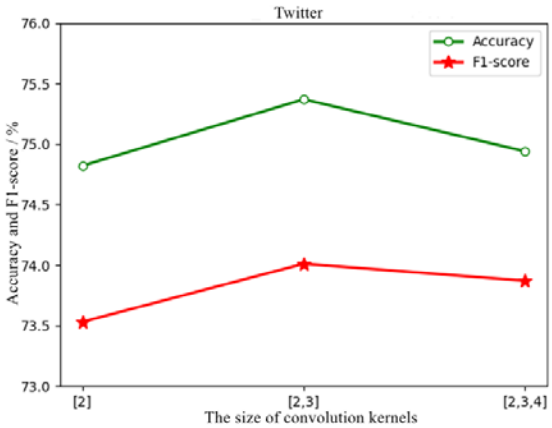

Fig. 11 Effect of different convolution kernel sizes (Twitter)

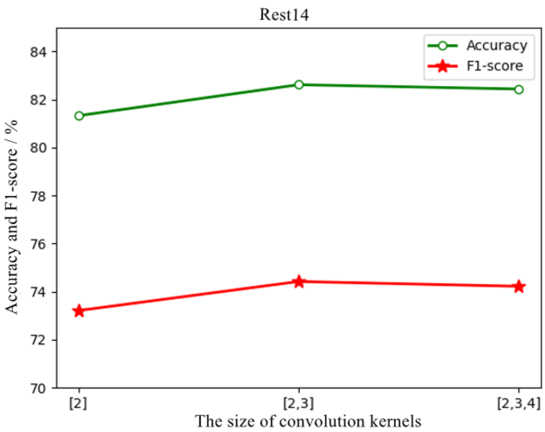

Fig. 12 Effect of different convolution kernel sizes (Rest14)

It can be seen from Figs. 11 and 12 that [2,3] is the best for the three channel numbers under the same number of convolution kernels

E. Effect of GCN Layers

The number of layers in graph convolution neural network is a parameter that has a great influence on the results. Based on the original model, the number of layers of GCN is changed, and the experimental results of Laptop14 are statistically analyzed. The comparison results are shown in Fig. 13.

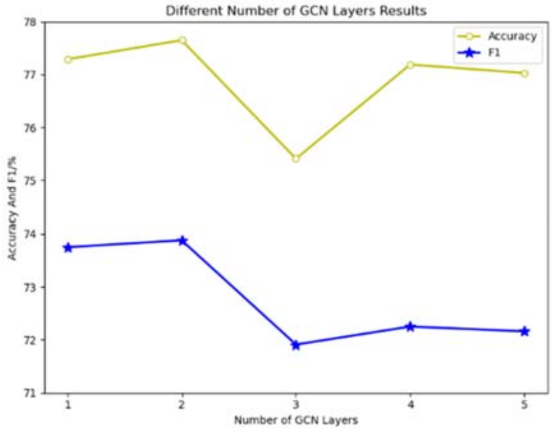

Fig. 13 Effects of the number of GCN layers

It can be seen from the data statistical chart that with the increase of GCN layers, the accuracy of experimental results and F1-Score are changing. Firstly, after increasing from 1 layer to 2 layers, the accuracy and F1-Score increase. Then from 2 layers, the accuracy and F1-Score show a downward trend with the increase of the number of layers. Therefore, the GCN in this model adopts a 2-layer structure.

F. Experimental Results

Table II shows the experimental results of each model on four different datasets. Compared with the model after integrating the dependency tree structure and considering the dependency relationship between words, we can find that the experimental results obtained by using only attention mechanism and neural network for aspect words and context modeling in the early stage are lower, Thus, it is proved that the dependency tree can improve the ASA, and the graph convolution neural network is effective for this task. Compared with the models integrated with the dependency tree, it can be seen that the experimental results of CDT, ASGCN, AEGCN and kumaGCN are basically lower than the TANGCN model in this paper in accuracy and F1-Score, and only the accuracy of ASGCN on rest16 dataset is higher than the model in this paper. Therefore, it is proved that using the path distance of graph as the coding fusion of words and using multi-channel text convolution to extract the dependencies of words in different ranges can make the classification task more accurate.

VI. Conclusion

In this paper, we use dependency trees in ASA to enforce semantic dependencies between words and address long-distance dependencies. We use a multi-channel convolutional neural network to solve the problem of extracting dependencies between words in different ranges. GCN is used to aggregate the information around each word and finally get the sentiment polarity of the aspect word. Compared with other models, our model is more accurate and effective on multiple datasets. In future work, we will pay more attention to the research and improvement of graph structure construction to solve the noise problem of dependency trees.

TABLE II
EXPERIMENTAL RESULTS

Models	Rest14		Rest16		Twitter		Lap14	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1
ATAE-LSTM	77.20	-	-	-	-	-	68.70	-
MEMNET	80.95	-	-	-	-	-	72.21	-
IAN	78.60	-	78.60	-	-	-	72.10	-
RAM	80.23	70.80	-	-	69.36	67.30	74.49	71.35
PF-CNN	79.20	-	-	-	-	-	70.06	-
MGAN	81.25	71.94	-	-	72.54	70.81	75.39	72.47
HSCN	77.80	70.20	-	-	69.60	66.10	76.10	72.50
TRANSCAP	79.55	71.41	-	-	-	-	73.87	70.10
IACAPSNET	81.79	73.40	-	-	-	-	76.80	73.29
ANTM	82.49	72.10	-	-	72.35	69.45	75.84	72.49
CDT	82.30	74.02	85.58	69.93	74.66	73.66	77.19	72.99
ASGCN	81.22	72.94	**88.99**	67.48	72.69	70.59	75.55	71.05
AEGCN	81.04	71.32	87.39	68.22	73.16	71.82	75.91	71.63
kumaGCN	81.43	73.64	-	-	72.45	70.77	76.12	72.42
TANGCN	**82.61**	**74.42**	86.51	**69.97**	**75.37**	**74.01**	**77.65**	**73.88**

REFERENCES

[1] E. Burnaey, D. Smolyakoy, “One-Class SVM with Privileged Information and its Application to Malware Detection,” *IEEE* (2016).

[2] L. H. Zhang, et al. “Factors affecting decision tree classification method over TM image.” in *Forest Research, Beijing*, pp. 1-5, 2014.

[3] S. Mishra, M. Panda, “Histogram of oriented gradients-based digit classification using naive Bayesian classifier.” in *Progress in Computing, Analytics and Networking*, Springer, Singapore, pp. 285-294.

[4] Y. Wang, M. Huang, X. Zhu, and L. Zhao, “Attention-based LSTM for aspect-level sentiment classification,” in *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 606-615, Nov 2016.

[5] D. Tang, B. Qin, X. Feng, and T. Liu, “Effective LSTMs for Target-Dependent Sentiment Classification.” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3298–3307, 2016.

[6] Sun, L. Bing, and W. Yang, “Recurrent attention network on memory for aspect sentiment analysis.” in *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 452-461, Sep 2017.

[7] F. Fan, Y. Feng, and D. Zhao, “Multi-grained attention network for aspect-level sentiment classification.” in *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 3433-3442, 2018.

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

[9] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, “Effective attention modeling for aspect-level sentiment classification.” in *Proceedings of the 27th international conference on computational linguistics*, pp. 1121-1131, Aug 2018.

[10] S. Gu, Zhang, L. Zhang, Y. Hou, and Y. Song, “A position-aware bidirectional attention network for aspect-level sentiment analysis.” in *Proceedings of the 27th international conference on computational linguistics*, pp. 774-784, Aug 2018.

[11] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, “Exploiting document knowledge for aspect-level sentiment classification.” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018.

[12] Y. Ma, H. Peng, and E. Cambria, "Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM.” *AAAI* 2018.

[13] T. N. Kip F, M. Welling, “Semi-supervised classification with graph convolutional networks.” 2016.

[14] K. Sun, Zhang, R. Zhang, S. Mensah, Y. Mao, and X. Liu, “Aspect-level sentiment analysis via convolution over dependency tree.” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5679-5688, Nov 2019.

[15] B. Huang, K. M. Carley, “Syntax-aware aspect level sentiment classification with graph attention networks.” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5472–5480 2019.

[16] C. Zhang, Q. Li, D. Song, “Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks.” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4560-4570, 2019.

[17] K. Wang, W. Shen, Y. Yang, X. Quan, and R. Wang, “Relational Graph Attention Network for Aspect-based Sentiment Analysis.” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3229–3238, Online 2020.

[18] L. Xiao, X. Hu, Y. Chen, Y. Xue, and T. Zhang, “Targeted sentiment classification based on attentional encoding and graph convolutional networks.” Applied Sciences, 10(3), 957, 2020.

[19] B. Huang, K.M. Carley, “Parameterized convolutional neural networks for aspect level sentiment classification” 2019.

[20] Z. Chen, T. Qian, “Transfer capsule network for aspect level sentiment classification.” in *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 547-556, July 2019.

[21] C. Du, H. Sun, J. Wang, Q. Qi, and M. Liu, “Capsule Network with Interactive Attention for Aspect-Level Sentiment Classification.” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[22] C. Chen, Z. Teng, and Y. Zhang, “Inducing target-specific latent structures for aspect sentiment classification.” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online. Association for Computational Linguistics (2020)*, pp. 5596–5607, 2020.

[23] M. Pontiki, et al., “SemEval-2016 Task 5: Aspect Based Sentiment Analysis.” in *International Workshop on Semantic Evaluation 2018*.

[24] M. Pontiki, et al., “SemEval-2014 Task 4: Aspect Based Sentiment Analysis. in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*.

[25] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, “Adaptive recursive neural network for target-dependent twitter sentiment classification.” In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 49–54. Baltimore, MD, USA, 23–25 Jun 2014.