Aspect-level Sentiment Analysis Based on Convolutional Network with Dependency Tree

Lei Mao[1] Jianxia Chen[1] Shi Dong[2] Liang Xiao[1] Haoying Si[1] Xinyun Wu[1]

[1]*School of Computer Science*
Hubei University of Technology
Wuhan, China
Email: 2219236005@qq.com

[2] *School of Computer Science and Technology*
Zhoukou Normal University
Zhoukou, China
Email: 3751720@qq.com

***Abstract*—Aspect-Based Sentiment Analysis (ABSA) aims to determine the sentiment polarity of certain aspect words in a sentence. Recently, it is a popular approach to fuse the sentences' syntactic information via the dependency tree into the graph neural network. However, how to efficiently utilize the obtained syntactic information is still a challenging problem of this kind of approach. Therefore, this paper proposes a novel Aspect-level Sentiment Analysis model based on Convolutional network with Dependency Tree, named ASAC-DT in short. First, the attention mechanism is utilized to obtain the attention score of the sentence and the aspect word respectively, to improve the connection of the words related to the aspect word in the sentence. Afterwards, by relying on the syntactic information obtained from the dependency tree, the connections of words that are not related to the aspect words are reduced. Finally, the feature information most relevant to the aspect words in the proposed model is extracted through the graph convolutional neural network and the interactive network. Through extensive experimental baselines the proposed ASAC-DT model shows effectiveness in aspect-level sentiment classification and outperforms baselines in accuracy.**

Keywords-Aspect-Based Sentiment Analysis; Attention mechanism; GCN; Interactive network

I. INTRODUCTION

In sentiment analysis research on texts, aspect based sentiment analysis (ABSA) aims to analyze the sentiment polarity of related aspect terms in sentences. The analysis of the emotional polarity of different aspect terms in a sentence is defined as the aspect-level sentiment analysis (ASA), and has been a focus of textual sentiment analysis in recent years.

ASA is constantly evolving with deep learning techniques, and the techniques used are constantly being updated and iterated. One of the key points is to establish a link between aspect and viewpoint words. Among the studies in recent years that have modeled the link between aspect and viewpoint words. The AT-LSTM [1] and RAM [2] models emphasize the use of AM to model the relevance between aspect words and contexts. However, because of the complexity of textual language, the use of the attention mechanism may lead to compromised connections between aspect and viewpoint words.

Due to the specificity of graphical neural networks, the study of syntactic information in combination with sentences has also become a trend in party-level sentiment analysis. The information on the syntactic structure of the sentence enables a better strengthening of the connection between aspect and viewpoint words. For example, the ASGCN [3] model extracts the syntactic information of a sentence after processing it and then extracts the sentence feature information via GNN [4]. The CDT [5] model combines syntactic information with word embeddings by acquiring sentence feature information by the use of a graph convolutional network (GCN) to enhance the presentation of the learned aspect words. However, in both studies, neighboring nodes were given the same weight, without distinguishing between the different importance of neighboring nodes to the current node. Therefore, the noise information in it will have a certain impact on the accuracy of the model.

In addition, many ASA studies based on GNNs incorporate syntactic information about sentences, but don't sufficiently utilize syntactic information. In some complex sentence structures, aspect and viewpoint nodes are linked to each other by different dependencies. In the process, irrelevant node information may be introduced, creating a noisy impact, which is a challenge in the current research.

Therefore, we propose a novel Aspect-level Sentiment Analysis model based on Convolutional network with Dependency Tree, named ASAC-DT in short. First, the sentence representation is obtained by Bi-LSTM to obtain a hidden layer representation of the sentence. Afterwards, the attention mechanism(AM) was utilized to calculate the self-attention scores and the aspect words respectively. The dependency tree generated by the sentence representation is then constructed into a graph-related representation. Both the attention mechanism and the dependency tree representation of sentences can strengthen the links between aspect and viewpoint words in a sentence, while reducing the interference of irrelevant words. Finally, the attention score and sentence dependency tree representation are integrated through the GCN network to maximize the use of effective information in the sentence.

In summary, the main contributions of the proposed model are as follows:

- Construct and use sub-trees of the dependency tree with attention scores, to reduce the effect of noise.
- Applying attention scores directly to the GCN network with the interaction network to ensure maximum extraction of sentence feature information.

DOI reference number: 10.18293/SEKE2023-127

- Experimental results demonstrate that the proposed model achieves better results than the baseline method on the public datasets.

II. RELATED WORK

A. Aspect-level sentiment analysis

With the continuous development of the research process, in the early related research [6], the task of predicting emotional polarity in the ABSA was realized by manually defining the relevant syntactic rules. In addition, some studies [7-9] combined the relevant neural network in deep learning to improve the efficient of the sentence feature information extraction. Among them, Li [7] effectively utilizes aspect words and context words to provide information utilization of the context of the sentence, but does not effectively consider other structural information of the sentence. Wang [8] proposed a hierarchical network that focuses on both the word and the clause level. However, it ignores the sentence text information. Fan [9] proposed a new multi-granularity attention model, but it cannot solve the noise problem generated in the model well.

B. Syntax Analysis

Applying syntactic information to sentence representation is a commonly used research mode recently. The adaptive recurrent neural network (AdaRNN) [10] transferred the emotional tendency to the corresponding target word according to the context and syntactic structure information. However, it is easily affected by noise data during data transmission. Nguyen [11] optimized and expanded AdaRNN, combining the dependency tree and composition tree of the sentence into the structural information of the sentence. The CDT [12] model directly integrates the syntactic information of the dependency tree into the word embedding information. The DSS-GCN [13] model combined syntactic information, semantic information, and structural information of sentences to improve the connection between words in sentences.

Inspired by the CDT model, the syntactic information of the dependency tree is adopted in the proposed ASAC-DT model Different from the CDT, the syntactic information of ASAC-DT is split according to the corresponding dependency tree structure information. Also ASAC-DT constructed the subtree of the dependency tree, so as to capture the syntactic information much better.

C. GCN

Many ASA researches have begun to revolve around GCN with the AM. The relational graph attention (R-GAT) model[14] has achieved good results in prediction through the effective use of AM. However, sentence information is affected by dependencies during the encoding. Chen [15] proposed a model that integrates GCN and common AM, and removes the effect of noise in contextual words, but does not use the dependence on syntactic information.

A model [3] applied GCN on the dependency tree of a sentence to effectively capture syntactic information and remote word dependencies. However, the model does not make sufficient use of the syntactic information captured. The HL-GCN model [16] improved the dependency between sentence representations by dividing different dependency subtrees. The CANN-SSCG model [17] integrated the sentiment knowledge, syntactic information and contextual information of the sentence, followed by a GNN to extract the effective information of the sentence.

The proposed ASAC-DT model incorporates syntactic information and AM into the GCN together to improve the information extraction capability of the GCN as well as to reduce the interference of noisy data.

III. METHODOLOGY

As shown in Fig.1, the ASAC-DT model consists of six parts, each of which has the following roles.

- Input and encoding layer. The word embedding technique obtains a vector representation of the sentence.
- Attention layers. The AM was used to calculate the self-attention scores of the sentences and the attention scores about the aspect words separately.
- Dependency tree layer. The syntax parser is used to obtain information about the dependency tree structure of the sentence.
- Graph convolution layer. The ASAC-DT model integrates the hidden layer information of the sentence, the sentence dependency tree structure information and the attention score for feature extraction. Interaction network layer. Fusion extracts useful information from sentences and reduces the loss of data information during convolution.
- Output layer. Output the final classification result of the model.

A. Input and Encoding Layer

The sentence representation $\mathrm{s} = \{w_1, w_2, \dots, a_1, a_m, \dots, w_n\}$ is obtained by the word embedding technique, where $\{a_1, \dots, a_m\}$ is the aspect word representation of the sentence. Afterwards the data is entered into the encoding area.

Bi-LSTM has been utilized here to extract sentence feature information. The word embedding representation of a sentence is obtained by the GloVe technique with the word embedding representation $\mathrm{x} = \{x_1, \dots, x_{a+1}, x_{a+m}, \dots, x_n\}$. Afterwards the Bi-LSTM extracts the feature information of the sentence to obtain the hidden state vector $\mathrm{H} = \{h_1, h_2, \dots, h_{a+1}, \dots, h_{a+m}, \dots, h_n\}$ of the sentence, where $h_t \in \mathbb{R}^{2d}$ denotes the hidden state at time step t. d denotes the dimension of the hidden state vector in the one-way LSTM.

The word embedding representation x is simply represented by the following equations (1-3).

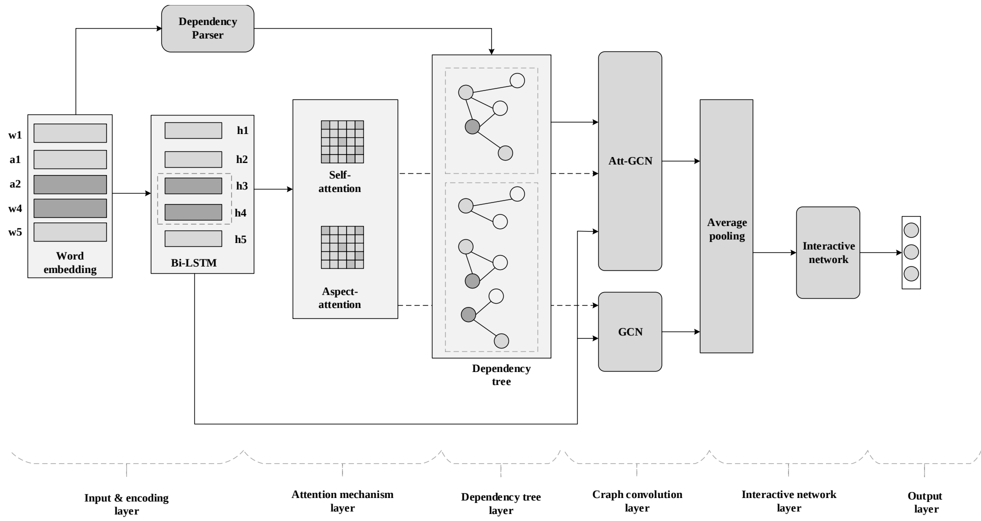

Figure 1. General structure of the ASAC-DT model

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(x_t) \quad (1)$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(x_t) \quad (2)$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (3)$$

Where：

- x_t indicates word embedding representation.
- h_t defines the hidden state at the time t.

B. *Attention Layers*

The ASAC-DT model calculates the self-attention scores of the sentences and the attention scores about the aspect words with the AM[18]. The formula is as follows (4)

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

Where:

- Given a key *K*, a query *Q* and a value *V*.
- d_w indicates the dimension of the word representation in the sentence representation.

1) Calculation of self-concentration score: The sentence hidden state vector H is obtained from the input and encoding layers and then assigned to K, Q and V respectively. The self-attention score of the sentence is obtained by calculating the Attention(Q,K,V)formula shown as (5).

$$self_{att} = Attention(H,H,H) = softmax\left(\frac{HH^T}{\sqrt{d_w}}\right)H \quad (5)$$

2) Calculation of aspect word attention score: The hidden state H_a of the aspect word is obtained from the hidden state representation in the following (6).

$$H_a = mask(H) \quad (6)$$

The mask function obtains the aspect-word representation $H_a = \{0,0,\dots,h_{a+1},\dots,h_{a+m},\dots,0\}$by setting 0. Assign $H_a$to Q and H to K and V to obtain the aspect-word representation's attention score aspect-attention calculated as (7).

$$asp_{att} = Attention(H_a,H,H) = softmax\left(\frac{H_aH^T}{\sqrt{d_w}}\right)H \quad (7)$$

The attention layer acquires the sentence's self-attention score self-attention and the aspect word's attention score aspect-attention.

C. *Dependency Tree Layer*

The sentences form a dependency tree structure through the parser. This dependency structure preserves the dependencies between each word in the sentence. This is shown in Fig. 2 below.

A sentence dependency tree is transformed into a graphical representation of the dependency structure of the sentence d^{syn}. The dependency graph representation of the sentence d^{syn} is partitioned into three sub-tree dependency structure graph representations $\{d_1^{syn}, d_2^{syn}, d_3^{syn}\}$ according to the syntactic distance {1， 2， 3}.

D. *Graph Convolution Layer*

In the proposed ASAC-DT model, a GCN is utilized to process on further extraction of sentence features. The calculation formula is described as the following (8)

$$H^{out} = GCN(H,D) \quad (8)$$

Where:

- *H* is represented as a sentence vector.
- *D* is represented as a sentence adjacency matrix.

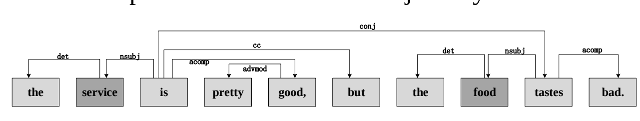

Figure 2. Diagram of the sentence dependency structure

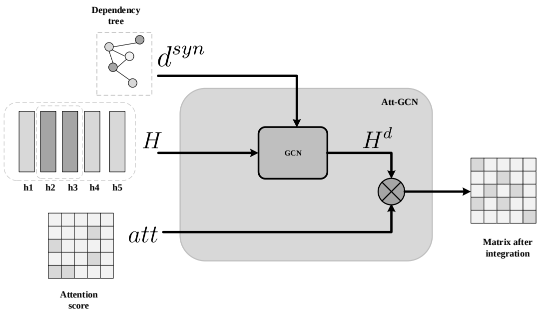

Figure 3. Att-GCN structure diagram

Here the hidden state vector H generated by the encoding layer , the self-attentive score $self_{att}$ of the sentence generated by the attention layer are used as the input of the GCN. The self-attentive scores assign weights based on the weight relationships of the words in the sentence, making it more efficient to extract information about the sentence features. The final output is H^{att}.

At the same time, this model integrates the GCN to make it more suitable for the information extraction ability of the model. As shown in Fig. 3. The integrated calculation formula is as follows (9-10):

$$H^d = GCN(H, D^{syn}) \tag{9}$$

$$A_d = H^d \otimes att \tag{10}$$

Where:

- H is represented as a sentence vector.
- D^{syn} is represented as a sentence dependency graph.
- att represents the attention score.

Calculated according to the self-attention score $self_{att}$ of the sentence and the aspect word attention score asp_{att}. First, the hidden state representation H of the sentence in the encoding layer and the dependency structure graph representation d^{syn} of the sentence in the dependency tree layer are used as the input of $GCN(H, D^{syn})$, and then the self-attention score of the sentence obtained by the attention layer $self_{att}$ to operate. Get an output result $A_d \in \mathbb{R}^{2d}$.

At the same time, the hidden state representation H of the sentence in the encoding layer and the three subtree dependency structure graphs of the sentence represent $\{d_1^{syn}, d_2^{syn}, d_3^{syn}\}$. As the input of $GCN(H, D^{syn})$, it is operated with the aspect word attention score asp_{att} of the sentence obtained by the attention layer respectively, and three output results $\{A_{d1}, A_{d2}, A_{d3}\} \in \mathbb{R}^{2d}$.

E. Interaction Network Layer

Before the graph convolution layer data is transmitted, an average pooling operation will be performed. The calculation method is as follows (11).

$$f^{out} = \text{Average pooling(f)} \tag{11}$$

The output results H^{att}, A_d and $\{A_{d1}, A_{d2}, A_{d3}\}$ obtained by the graph convolution layer are subjected to an average pooling operation. Before pooling, the integration operation is performed on $\{A_{d1}, A_{d2}, A_{d3}\}$. It is calculated as follows (12).

$$A_{asp} = [A_{d1} \oplus A_{d2} \oplus A_{d3}] \tag{12}$$

Perform interactive processing on the pooled results $f^{H^{att}}$, f^{A_d} and $f^{A_{asp}}$ of H^{att}, A_d and A_{asp}. This paper adopts a simple and effective cross-network structure [19]. The calculation formula of interaction is as follows (13).

$$X_{l+1} = X_0\, X_l^T W_l + b_l + X_l \tag{13}$$

Where:

- $X_l \in \mathbb{R}^d$ is the input of the interaction network.
- $X_{l+1} \in \mathbb{R}^d$ is the output.
- $W_l \in \mathbb{R}^d$ is the weight parameter.
- $b_l \in \mathbb{R}^d$ is a bias item.

The calculation process of the simplified interaction in this paper is as follows (14-15):

$$X_A^{d_c} = f^{H^{att}} \, (f^{A_d})^T + f^{A_d} \tag{14}$$

$$X_A^{d_{asp}} = f^{A_{asp}} \, (f^{A_d})^T + f^{A_d} \tag{15}$$

Where:

- $X_A^{d_c} \in \mathbb{R}^a$ and $X_A^{d_{asp}} \in \mathbb{R}^a$ are the output results after interaction.

F. Output Layer

In the output layer, the two obtained outputs $X_A^{d_c} \in \mathbb{R}^a$ and $X_A^{d_{asp}} \in \mathbb{R}^a$ are spliced first, and then sent to the classifier. The output process is shown in formula (16-17).

$$o = \left[X_A^{d_c}, X_A^{d_{asp}}\right] \tag{16}$$

$$o' = softmax\big(Linear(o)\big) \tag{17}$$

Where:

- $o \in \mathbb{R}^{2a}$ is the spliced output.
- $o' \in \mathbb{R}^{out}$ is the final output. out is the number of classifications of the final output.

The loss function of the ASAC-DT model is the cross entropy loss function (CrossEntropy Loss), the specific formula (18) is as follows:

$$H(p,q) = -\sum_x p(x)\log q(x) + \lambda ||\theta|^2 \tag{18}$$

Where $p(x)$ is the true distribution of the sample, and $q(x)$ is the predicted distribution of the sample. At the same time, L_2 regularization is also added for constraints. L_2 regularization controls the complexity of the model and reduces the over-fitting of the model.

IV. Experiment Result and Analysis

A. Datasets

For experimental evaluation, this paper uses three public datasets. Lap14 and Rest14 in Twitter datasets [10] and

TABLE I. DATASET STATISTICS

Dataset	Positive		Neutral		Negative	
	Train	***Test***	***Train***	***Test***	***Train***	***Test***
Rest14	2164	728	637	196	807	196
Lap14	994	341	464	169	870	128
Twitter	1561	173	3127	346	1560	173

TABLE II. HYPERPARAMETER SETTINGS

Hyper-parameter	Description	Value
dropout rate	Word embedding layer	0.7
batch_size	Size of mini-batch	16
r	Initial learning rate	0.001
d_e	Size of embedding layer	300
d_h	Size of hidden layer	300
l_2	L_2-Regularization weight	0.0001

SemEval 2014 Task [20]. The details of the datasets are shown in Table I.

B. Parameter Settings

In this article, GloVe[21] is used to initialize sentences. The relevant hyperparameters in the model are shown in Table II. The Adma optimizer [22] is used. The L_2 regularization weight is set to 0.0001. At the same time, this experiment uses the accuracy rate and Macro-F1 as the evaluation index to evaluate the performance of the model.

C. Model Comparison

The proposed ASAC-DT model has been compared with the following baselines.

- AS-GCN[3]:Applying GCN to sentiment analysis to extract feature information of sentences efficiently.
- CDT[5]: The dependency tree and CNN are modeled to learn sentence feature information.
- BiGCN[23]: Hierarchical modeling is carried out through the syntactic structure and lexical information of the sentence itself.
- kumaGCN[24]: Apply dependency graphs and latent graphs to models to improve the model performance.
- DGEDT[25]: Through the flat representation of the transformer, a dual-transformer model augmented by a dependency graph is designed.

D. Results Analysis of Comparison Experiments

The comparison results of the ASAC-DT model and other models are shown in Table III. As can be seen from the data in the table, the model in this paper outperforms most of the baseline methods in terms of accuracy and Macro-F1. It is shown that the attention mechanism used by the text model and the integration of the dependency tree structure information into the GCN has led to some improvement in the performance of

TABLE III. RESULTS OF COMPARISON EXPERIMENTS

method	Twitter		Lap14		Rest14	
	Accuracy(%)	***Macro-F1(%)***	***Accuracy(%)***	***Macro-F1(%)***	***Accuracy(%)***	***Macro-F1(%)***
AS-GCN	72.15	70.40	75.55	71.05	80.96	72.21
CDT	74.66	73.66	77.19	72.99	82.30	74.02
BiGCN	74.16	73.35	74.59	71.84	81.97	73.48
kumaGCN	72.45	70.77	76.12	72.42	81.41	73.64
DGEDT	74.80	73.40	76.80	72.30	**83.90**	75.10
ASAC-DT	**76.16**	**74.47**	**77.59**	**74.01**	83.39	**75.81**

the model. It also outperforms models using dependency trees and GCNs such as CTD, BiGCN and kumaGCN in terms of effectiveness in capturing aspect and viewpoint words.

E. Number of Att-GCN Layer

In this paper, we conducted a comparative experiment on the effect of the number of layers of Att-GCN on the accuracy of the model, and the results are shown in Fig. 4.

As can be seen in Fig. 4, the number of layers set up for the experiment was 3, 4, 5, 6 and 7, with varying degrees of decrease in the accuracy of the model as the number of layers increased. It shows that an increase in the number of layers can lead to over-fitting of the model.

F. Ablation Experiment

The corresponding ablation experiments were also performed in this paper, as shown in Table IV.

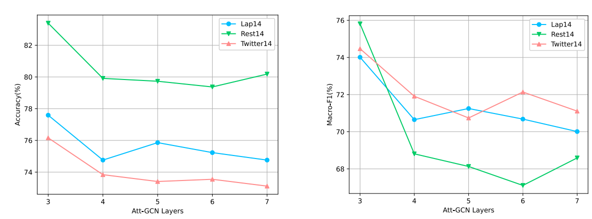

Figure 4 Att-GCN layer count test on three datasets

TABLE IV. RESULTS OF ABLATION EXPERIMENT

method	Twitter		Lap14		Rest14	
	Accuracy(%)	***Macro-F1(%)***	***Accuracy(%)***	***Macro-F1(%)***	***Accuracy(%)***	***Macro-F1(%)***
w/o aspect-attention	75.43	73.94	76.48	72.49	83.03	75.07
w/o dependency subtree	73.69	72.32	76.01	72.03	80.26	69.03
w/o cross network	74.85	73.34	76.17	71.47	82.76	74.76
ASAC-DT	76.16	74.47	77.59	74.01	83.39	75.81

Removing the aspect-level attention score (w/o aspect-attention) from the model reduced the accuracy of the model by 0.73%, 1.11% and 0.36% in the three datasets Twitter, Lap14 and Rest14. At the same time, the corresponding dependent subtree structure (w/o dependency subtree) in the model is removed, and the accuracy decreases by 2.47%, 1.58% and 3.13%.

Because the aspect-attention is in the Att-GCN between the integration and the dependent subtree, when the dependent subtree structure is removed, the corresponding aspect-attention will also be removed. From the results of the data, it can be seen that aspect-attention plays an important role in the model and strengthens the correlation with aspect words.

Relying on the subtree structure, the influence of noise data is also reduced in the acquisition of syntactic information. The corresponding accuracy of the ablation of the interaction network (w/o cross network) on the three data sets decreased by 1.31%, 1.42% and 0.63, respectively. It shows that the interactive network also plays a certain role in the integration of sentence information.

V. CONCLUSION

The proposed ASAC-DT model is a GCN model based on attention mechanism and dependency tree. The interaction network is used in the model for aspect-level sentiment analysis tasks. According to the particularity of the graph convolutional neural network, the corresponding attention mechanism score and the syntactic dependency information generated by the dependency tree are integrated into it. The dependency relationship obtained through the attention mechanism and the syntactic information provided by the dependency tree enable the model to more accurately capture the connection between aspect words and opinion words. Among them, the semantic information and syntactic information of the sentence have been effectively transmitted, which makes the model have a more effective text sentiment classification ability. The experimental results show that the experimental model in this paper is superior to the current baseline model and has achieved good results on various public datasets.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (61902116).

REFERENCES

[1] Wang, Y. , et al. "Attention-based LSTM for Aspect-level Sentiment Classification." Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing 2016.

[2] Peng, C. , et al. "Recurrent Attention Network on Memory for Aspect Sentiment Analysis." Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing 2017.

[3] Sun, K. , et al. "Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree." 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.

[4] Kipf, T. N. , and M. Welling . "Semi-Supervised Classification with Graph Convolutional Networks." (2016).

[5] Sun, K. , et al. "Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree." 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.

[6] Kiritchenko, S. , et al. "NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews." Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) 2014.

[7] Li, Haihui, et al. "Co-attention networks for aspect-level sentiment analysis." CCF International Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2019.

[8] Wang, J. , et al. "Aspect sentiment classification with both word-level and clause-level attention networks." Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18 2018.

[9] Fan, F. , Y. Feng , and D. Zhao . "Multi-grained Attention Network for Aspect-Level Sentiment Classification." Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing 2018.

[10] Li, D. , et al. "Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification." Meeting of the Association for Computational Linguistics 2014.

[11] Nguyen, Thien Hai, and Kiyoaki Shirai. "Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis." Proceedings of the 2015 conference on empirical methods in natural language processing. 2015.

[12] Sun, K. , et al. "Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree." 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.

[13] Dai, Anan , et al. "Learning from word semantics to sentence syntax by graph convolutional networks for aspect-based sentiment analysis." International Journal of Data Science and Analytics 14.1(2022):17-26.

[14] Wang, K. , et al. "Relational Graph Attention Network for Aspect-based Sentiment Analysis." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics 2020.

[15] Chen, Zhaowei , et al. "Aspect-Based Sentiment Analysis Using Graph Convolutional Networks andCo-attention Mechanism." Springer, Cham (2021).

[16] Pang, Shiguan , et al. "Highway-Based Local Graph Convolution Network for Aspect Based Sentiment Analysis." CCF International Conference on Natural Language Processing and Chinese Computing Springer, Cham, 2021.

[17] Phan, H. T. , N. T. Nguyen , and D. Hwang . "Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis." Information Sciences 589(2022):416-439.

[18] Vaswani, A. , et al. "Attention Is All You Need." arXiv arXiv, 2017.

[19] Wang, R. , et al. "Deep & Cross Network for Ad Click Predictions." ADKDD'17 ACM, 2017.

[20] Pontiki, M., Papageorgiou, H., Galanis, D., Androutsopoulos, I., Pavlopoulos,J., Manandhar, S.: Semeval-2014 task 4: Aspect based sentiment analysis. In:SemEval, vol. 2014, p. 27 (2014)

[21] Pennington, J. , R. Socher , and C. Manning . "Glove: Global Vectors for Word Representation." Conference on Empirical Methods in Natural Language Processing 2014.

[22] Kingma, D. , and J. Ba . "Adam: A Method for Stochastic Optimization." Computer Science (2014).

[23] Zhang, M. , and T. Qian . "Convolution over Hierarchical Syntactic and Lexical Graphs for Aspect Level Sentiment Analysis." Empirical Methods in Natural Language Processing Association for Computational Linguistics, 2020.

[24] Chen, C. , Z. Teng , and Y. Zhang . "Inducing Target-Specific Latent Structures for Aspect Sentiment Classification." Empirical Methods in Natural Language Processing Association for Computational Linguistics, 2020.

[25] Tang, H. , et al. "Dependency Graph Enhanced Dual-transformer Structure for Aspect-based Sentiment Classification." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics 2020.