A SENTIMENT AND SYNTACTIC-AWARE GRAPH CONVOLUTIONAL NETWORK FOR ASPECT-LEVEL SENTIMENT CLASSIFICATION

Yuxin Yang[a] *Xia Sun*[a,*] *Qiang Lu*[a] *Richard Sutcliffe*[a,b,*] *Jun Feng*[a]

[a] School of Information Science and Technology, Northwest University, Xi'an, China
[b] School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK

ABSTRACT

Aspect-level sentiment classification (ASC) is a significant problem in fine-grained sentiment analysis, which automatically predicts the sentiment polarity of a given aspect in a sentence. Dependency tree-based graph convolutional networks have been widely studied for their ability to effectively capture the dependencies of aspect words with other words. However, constructing more accurate syntactic trees by introducing external knowledge has limited improvement on ungrammatical informal texts and has led to over-parameterization of the model. To alleviate this problem, we propose a sentiment and syntactic-aware graph convolutional network (SaS-GCN) that combines syntactic and sentiment relations. We use an attention mechanism and the Sparsemax activation function to construct a sparse sentiment-dependent graph. Compared with existing methods that use LSTM or CNN to obtain semantics from text directly, this graph, combined with a GCN, contains more semantic features. Moreover, we redesign the network structure of GCN, calling it EN-GCN, to make it sensitive to node dimensional features and hence to have a strong feature mining ability. The experimental results indicate that our model outperforms state-of-the-art methods. In particular, when evaluated on the Rest15 and Rest16 datasets, the F1 scores of the proposed lightweight model are 4.15% and 3.77% better than BERT respectively.

Index Terms— Aspect-level sentiment classification (ASC), Graph Convolutional Network (GCN), Sparsemax activation function

1. INTRODUCTION

Aspect-level sentiment classification (ASC) aims to classify the sentiment polarity (e.g. positive, neural, or negative) of given aspect word(s) in a sentence, which is a fine-grained sentiment analysis task. For example, consider this restaurant review: "*the service is excellent , the decor fancy and unintelligible.*" Given the sentence, the aspect word "*service*" and the aspect word "*decor*", our goal is to be able to infer that the sender has a positive feeling about service and a negative feeling about decor. In contrast to classifying complete sentences for sentiment polarity, ASC is able to extract the sentiment expressed relative to specific aspect, so that the information obtained is more accurate and specific.

The difficulty for ASC is to be able to correctly attend to the relevant words in the sentence that depict the aspect when identifying the sentiment polarity of that aspect in the sentence. With the development of deep learning models, some studies [1, 2, 3, 4, 5] aimed to automatically fuse information about aspect words and other words in sentences by building attention-based neural networks, and they have all achieved good results. Later, more researchers found significant syntactic relationships between aspect words and sentiment expression words, such as the previous example "*the service is excellent*", where "*excellent*" is the descriptor of "*service*".

Thus, the first application of Graph Convolutional Networks (GCN) combined with syntactic dependency trees to the ASC task obtained exciting results [6]. Subsequently more and more work [7, 8, 9, 10, 11, 12, 13] began to investigate modeling the relationship between aspect words and sentiment expression words by constructing more advanced syntactic trees, and these have achieved some degree of improvement. However, on the one hand, these works use a more complex syntax tree construction approach, but with limited improvement and increased model complexity. On the other hand, since there are many informal texts in online comments, such as the previous example "*the decor fancy and unintelligible.*", "*fancy*" and " *unintelligible*" are sentiment expressions of "*decor*", but there is no a linking verb between them, so paying attention to syntactic relations alone is not enough to classify them properly.

In this paper, a Sentiment and Syntactic-Aware Graph Convolutional Network (SaS-GCN) model is thus proposed to jointly explore the sentiment and syntactic associations between words in a sentence. Where the sparse sentiment dependency graph (SenG) is constructed by an attention mechanism and later sparsified to discard noise, while attending to the relationships between arbitrary words in a sentence, even if the sentence has no obvious syntactic structure. In addition, we enhance the relational feature extraction capability of the GCN by combining the GCN layer and the linear layer to redesign the network structure, and by adding activation functions at the end of the neurons that can keep the direction constant. The main contributions of our work can be summarized as follows:

- We propose SaS-GCN to mine both syntactic and sentiment-semantic relations of sentences by graphs combined with GCN, where sentiment dependent graphs are automatically constructed and task-specific, focus on all words in a sentence, but subsequently remove noisy words by graph sparsification to obtain the same sparse graph as the syntactic tree.

- We redesign the GCN layer to attend more to its own node information and to combine nonlinear activation functions that preserve as much as possible the features extracted by the GCN.

- Experimental results on five benchmark datasets show that the model achieves the best performance on most of the metrics, proving the effectiveness of our model.

* Corresponding author.
E-mail addresses: raindy@nwu.edu.cn (X. Sun), rsutcl@nwu.edu.cn, rsutcl@essex.ac.uk (R. Sutcliffe).

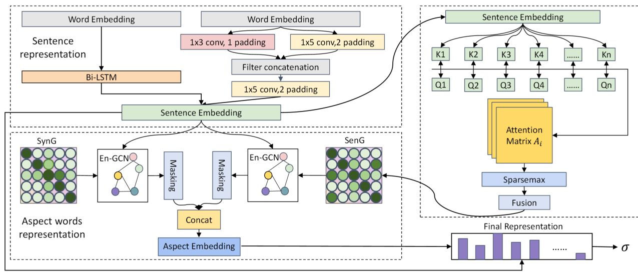

Fig. 1: Overall architecture of SaS-GCN.

2. PROPOSED MODEL

In this section, we present our proposed Sentiment and Syntax-Aware Graph Convolutional Network (SaS-GCN) (Figure 1). The model takes pre-trained word embeddings as input and the sentence representation phase uses BiLSTM and CNN together to extract features. The aspect word representation phase applies Enhanced-GCN (En-GCN) over the syntactic dependency graph (SynG) and the sparse sentiment dependency graph (SenG) respectively to obtain a representation containing both syntactic and sentiment dependencies. Finally, we use retrieval-based attention weights [6] to obtain the final representation before the classification layer.

2.1. Sentence representation phase

First, we use a BiLSTM and a 1d-CNN to mine the sequence information and local semantic information of the sentences, respectively. Then we obtain the final hidden state vector $H^{LSTM} = \left\{h_1^{LSTM}, h_2^{LSTM}, ..., h_n^{LSTM}\right\}$, extracted by the Bi-LSTM, and the vector $H^{CNN} = \left\{h_1^{CNN}, h_2^{CNN}, ..., h_n^{CNN}\right\}$, extracted by the CNN, where $h_t^{LSTM} \in \mathbb{R}^{2d_e}$ and $h_t^{CNN} \in \mathbb{R}^{2d_e}$ represent the final hidden state vector of the t-th word in the sequence calculated by each of them. Finally, we fuse H^{LSTM} and H^{CNN} as the sentence representation for $s = \{w_1, w_2, ..., w_n\}$.

2.2. Aspect words representation phase

Obtaining aspects word representations with rich relational information is critical for ASC tasks. Based on the previously obtained sentence representations, we first construct the syntactic tree-based SynG and the attention mechanism-based SenG separately, and then extract the aspect-oriented relationships in the two graphs using En-GCN. Last, we apply a masking layer to obtain the aspect word representations.Fist, we use a syntactic dependency tree constructed by spaCy[1] to attain an adjacency matrix i.e. SynG.

2.2.1. Sentiment dependency graph

SenG is a dynamic graph constructed for a specific task by the attention mechanism. It constructs sentiment associations different from syntactic dependencies, which complements the model's ability to handle sentences with obscure syntactic relationships.

We use the multi-headed self-attention mechanism to construct SenG. In the SenG, a self-attention layer is first used to compute a matrix $A_{sen} \in \mathbb{R}^{n\times n}$, where n is the sentence sequence length, containing the weights of sentiment relations between words in a sentence. Then we integrate the results of the parallel computation of multiple heads to obtain a dynamic sentiment dependent graph, which can be formulated as:

$$A_{sen} = \frac{\sum_{i=1}^{head} softmax\left(\frac{QW^Q \times \left(KW^K\right)^T}{\sqrt{d}}\right)}{head} \tag{1}$$

where both $Q \in \mathbb{R}^{n\times 2d_e}$ and $K \in \mathbb{R}^{n\times 2d_e}$ are vectors obtained by inheriting the sentence representation s, and $head$ is a settable hyperparameter, which was set to 8 in this study.

In general, the attention mechanism uses softmax as the activation function [14], which is a probability function that maps all input values to the range (0, 1). This causes us to weight the relationship between any two words in the sentence during graph construction, which introduces noise and adds unnecessary computational overhead. Therefore, we replace Eq. (1) by $A_i = sparsemax\,(Q, K)$, where sparsemax [15] is a transformation that depends on Q and K. The sparsemax transformation is defined as:

$$sparsemax(z) = \underset{\alpha \in \triangle^J}{\arg\min} \|\alpha - z\|^2 \tag{2}$$

where $\triangle^J = \left\{\alpha \in \mathbb{R}^J \mid \alpha \geq 0, \sum_j \alpha_j = 1\right\}$. The essence is to project the weights computed by the attention mechanism into Euclidean space, as these projections tend to hit the boundaries of the space, obtaining a sparser probability distribution. This substitution allows the attention mechanism to focus on only some of the associations between words, while other relations are assigned a weight of zero. This enhances the correct sentiment association feature and reduces the noise caused.

2.2.2. Enhanced-GCN

Initially, Graph Convolutional Networks (GCN) were designed to process irregular graph-structured data through a deep neural network paradigm, where the processing transfers information from the graph space to the embedding space and propagates the training error through a target loss function [16]. Moreover, as the number of GCN layers increases, the embedding information of each node v_i contains the global information of the associated nodes.

GCN obtains local semantic features between adjacent nodes, but we observe that the feature information of the nodes themselves (the word embedding dimension) is not well attended to. Therefore, we are inspired by SE-Net [17] to enhance GCN by adding linear layers to improve the performance. Figure 2 shows the resulting network architecture which we call Enhanced-GCN (En-GCN). En-GCN makes the node aware of context by the following equation:

$$\tilde{h}_i^l = \sum_{j=1}^{n} A_{ij} W^l g_j^{l-1} \tag{3}$$

where $\tilde{h}_i^l \in \mathbb{R}^{2d_e}$is the hidden layer representation of the l-th layer, the i-th node ($i \in [1, ..., n]$), A is the adjacency matrix, g_j^{l-1} is the hidden layer representation of the (l-1)-th layer, and the j-th node,W^l is a linear transformation weight. The set of nodes at this point is denoted $\tilde{H} \in \mathbb{R}^{1\times n\times 2d_e}$.

Then, we explicitly model the dependencies of the word embedding dimensions in three steps to enhance the feature learning capability of the GCN so that the model can gain more feature information. **The first step** uses an average pooling operation to encode the entire sentence association information on an embedding in a

[1]We use spaCy toolkit: https://spacy.io/.

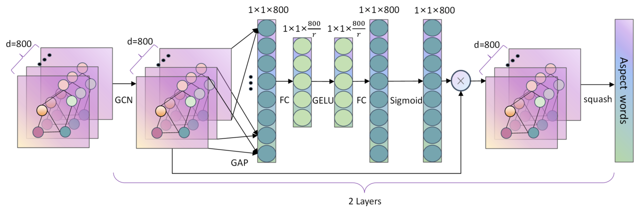

Fig. 2: Overall architecture of En-GCN.

global feature. Formally, an intermediate statistic $z \in \mathbb{R}^{2d_e}$ is calculated by shrinking $\tilde{H}$ through its spatial (sentence) dimensions $1 \times n$, so that the d-th element ($d \in [1, ..., 2d_e]$) of z is calculated by:

$$z_d = \frac{1}{1 \times n} \sum_{i=1}^{n} \tilde{h}_d\left(i\right) \tag{4}$$

Second, a gating mechanism in the form of a sigmoid is used to learn the nonlinear relationships among the embeddings and obtain the attention weights for each embedding:

$$p = Sigmoid(W_2 Gelu(W_1 z)) \tag{5}$$

where $p \in \mathbb{R}^{2d_e}, W_1 \in \mathbb{R}^{\frac{2d_e}{r} \times 2d_e}, W_2 \in \mathbb{R}^{2d_e \times \frac{2d_e}{r}}$, r is a configurable parameter, to reduce the complexity of the model and increase its generalization ability by first reducing the dimensionality through a layer of full connectivity, using the $Gelu$ function activation, and then restoring the original dimensionality through a layer of full connectivity. **Third,** the weights of each embedding learned are multiplied by the original features $\tilde{h}_d$ to obtain the updated state $\tilde{\tilde{h}}_i^l$ of $\tilde{h}_i^l$:

$$\tilde{\tilde{h}}_i^l = \sum_{d=1}^{2d_e} \tilde{h}_d \times p_d \tag{6}$$

where $\tilde{h}_d$ is the originally global feature of the d-th dimension and p_d is the feature weight of the d-th dimension.

Next, the weight-assigned features are fed to the activation function to obtain the output of this layer of GCN. Instead of using $Relu$ as its activation function, as in the classical GCN, we use the $squash$ function that preserves directional information. This function can be more sensitive to the relational information features extracted because of its properties, with the following equation:

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{7}$$

where s_j is the input vector and v_j is the output vector. In this function, the modal lengths of the input and output vectors are monotonic functions, and the larger the modal length of the input vector the larger the modal length of the output vector, which fully preserves the feature information obtained by the En-GCN. The hidden layer after passing the l-th layer of En-GCN is represented as follows:

$$h_i^l = squash\left(\tilde{\tilde{h}}_i^l + b^l\right) \tag{8}$$

where $i \in [1, ..., n]$, $h^L = \left\{h_1^l, h_2^l, ...h_n^l\right\}$ is the updated set of nodes. Then we mask all the words except for the aspect words that do not make any changes in the zero-masked embedding $h^{L'} = \left\{0, ..., h_{a_1}^{l'}, ..., h_{a_m}^{l'}, ...0\right\}$.

Dataset		Pos.	Neu.	Neg.
Twitter	Train	1561	3127	1560
	Test	173	346	173
Lap14	Train	994	464	870
	Test	341	169	128
Rest14	Train	2164	637	807
	Test	728	196	128
Rest15	Train	912	36	256
	Test	326	34	182
Rest16	Train	1240	69	439
	Test	469	30	117

Table 1: Dataset statistics

2.3. Final representation and classification layer

Based on the sentence representation and aspect word representations, we retrieve the important features that are semantically related to aspect words, and set the retrieval-based attention weights [6] for each context word.

3. EXPERIMENTS

3.1. Datasets

We validated our method on five publicly available datasets (Table 1) [18, 19, 20]. Among the Lap14, Rest14, Rest15 and Rest16 datasets are review texts about laptops and restaurants published in SemEval 2014 Task 4, SemEval 2015 Task 12 and SemEval 2016 Task 5, respectively. As the SemEval task has progressed over the years, the amount of informal review text within the datasets has increased, and concern has been expressed about this [21]. Finally, for reviews where there are n aspects, we treat them as n instances in the sample set.

3.2. Baselines

We compare our model with a range of baselines.

- **GCN-free:** ATAE-LSTM [1], RAM [22], AF-LSTM [23].
- **GCN-related:** ASGCN [6], BiGCN [7], InterGCN [9], R-GAT [10], KumaGCN [8], DualGCN [13].
- **Pre-trained:** BERT [24] is an excellent pre-trained language representation model and achieves good results on many text classification tasks. We followed previous work [9] in applying it to the ASC task by fine-tuning it, and using the results as a baseline in our experiments.
- **Ours:** SaS-GCN is our proposed model.

3.3. Results and analysis

Results for all the methods described above are shown in Table 2. The table clearly shows that our proposed method outperforms other GCN-related models in general, and especially on Rest15 and Rest16 which contain informal texts as discussed earlier. Compared with the best results of GCN-related models, our model has 4.34% higher F1 scores on Rest15 and 4.74% higher F1 scores on Rest16, and even compared with the most popular pre-trained model, BERT, our model has 4.15% and 3.77% higher F1 scores on these two datasets, respectively. This indicates that our construction of sparse sentiment dependency graphs can effectively capture the sentiment-semantic dependencies between aspect words and other words to

	Model	Twitter		Lap14		Rest14		Rest15		Rest16	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GCN-free	ATAE-LSTM	69.65	67.40	69.14	63.18	77.32	66.57	75.43	56.34	83.25	63.85
	RAM	69.36	67.30	74.49	71.35	80.23	70.80	79.30	60.49	85.58	65.76
	AF-LSTM	69.21	68.24	69.67	63.49	77.46	65.18	76.12	56.29	85.61	66.15
GCN-related	ASGCN	72.15	70.40	75.55	71.05	80.77	72.02	79.89	61.89	88.99	67.48
	BiGCN	74.16	73.35	74.59	71.84	81.97	73.48	81.16	64.79	88.96	70.84
	InterGCN			77.86	74.32	82.23	74.01	81.76	65.67	89.77	73.05
	R-GAT	75.57	73.82	77.42	73.76	83.30	76.08				
	KumaGCN	72.45	70.77	76.12	72.42	81.43	73.64	80.69	65.99	89.39	73.19
	DualGCN	75.92	74.29	78.48	74.74	84.27	78.08	82.96	68.65	90.33	75.25
Pre-trained	BERT	75.28	74.11	77.59	72.38	84.11	76.68	**83.48**	66.18	90.10	74.16
Ours	SaS-GCN	**75.93**	**74.59**	**78.53**	**74.76**	**84.75**	**78.35**	83.39	**70.33**	**90.74**	**77.93**

Table 2: Comparison of proposed model with existing benchmarks (%). Best results for each data set are shown in bold.

Model	Twitter		Lap14		Rest14		Rest15		Rest16	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
BASE	72.22	$70.67_{\uparrow 0.00}$	75.96	$71.53_{\uparrow 0.00}$	81.03	$72.68_{\uparrow 0.00}$	80.13	$62.03_{\uparrow 0.00}$	89.01	$67.74_{\uparrow 0.00}$
SynG+SenG	74.13	$72.78_{\uparrow 2.11}$	77.27	$73.66_{\uparrow 2.13}$	83.06	$75.89_{\uparrow 3.21}$	81.69	$68.36_{\uparrow 6.33}$	89.31	$75.32_{\uparrow 7.56}$
SynG+En-GCN	74.50	$72.90_{\uparrow 2.23}$	77.12	$73.63_{\uparrow 2.10}$	83.45	$75.76_{\uparrow 3.08}$	81.28	$67.58_{\uparrow 5.55}$	89.95	$74.69_{\uparrow 6.95}$
SaS-GCN	**75.93**	$\mathbf{74.59}_{\uparrow 3.92}$	**78.53**	$\mathbf{74.76}_{\uparrow 3.23}$	**84.75**	$\mathbf{78.35}_{\uparrow 5.67}$	**83.39**	$\mathbf{70.33}_{\uparrow 8.30}$	**90.74**	$\mathbf{77.93}_{\uparrow 10.19}$

Table 3: Ablation study results (%). The upward arrows are used to indicate the improvement of each model over the BASE model in terms of F1 scores for the respective data sets.

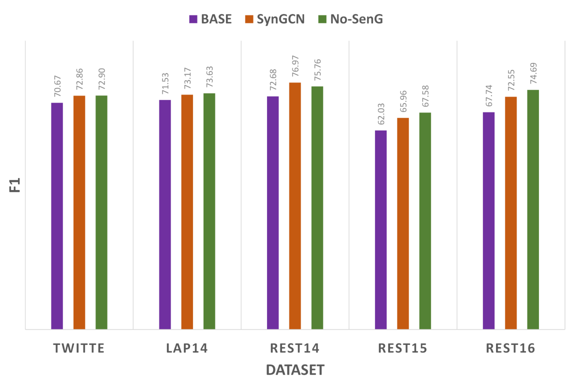

Fig. 3: Histograms of F1 values for the three models BASE, SynGCN [13] and SynG+En-GCN on the six data sets.

complement the classification ability of informal text. Also, for the other three datasets our model can also achieve comparable performance to BERT but with much fewer parameters, and is higher than GCN-related models.

3.4. Ablation Study

In this section, to verify the validity of each component of our model, we carry out an ablation experiment. Baseline model BASE applies the initial graph convolutional network (GCN) to mine the syntactic dependency graph (SynG). The SynG+SenG model builds the sentiment dependency graph (SenG) on the basis of SynG to integrate into the sentiment dependency features. SynG+En-GCN model uses the enhanced-GCN (En-GCN) to mine SynG while ignoring sentiment dependencies in the sentences. SaS-GCN is the most complete model we have proposed. Table 3 shows the performance of all these models. We can see that SynG+SenG improves more, and the F1 scores even improve by 6.33% and 7.56% on Rest15 and Rest16 datasets, respectively, which proves that our constructed SenG can effectively obtain the relationship between words even if the sentences lack syntactic relationships.

As for SynG+En-GCN, there is also a certain improvement, indicating that our network structure En-GCN does enhance the feature extraction ability of the original GCN. In Figure 3, we cite the model SynGCN installed in DualGCN [13], and compare it with the BASE model and the SynG+En-GCN model. In the results SynG+En-GCN outperforms SynGCN on four of the datasets without increasing the training burden, indicating that our proposed En-GCN works well and can mine more feature information in simple graphs, which also illustrates that feature information focusing on the node dimension is necessary.

4. CONCLUSION

In this paper, SaS-GCN also extracts sentiment associations in addition to aspect-specific syntactic dependencies, complementing the ability to correctly classify reviews with weak syntactic relationships. Additionally we propose Enhanced-GCN (En-GCN) that enhances the information extraction ability of its own nodes by reconstructing the network structure. Our results on multiple datasets validate the performance of our model and reveal that it consumes less computational resources while achieving identical performance.

5. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 61877050, and by the 2022 Yulin Science and Technology Plan Project "Research on Learning Behavior Analysis and Emotion Recognition Technology in Large-scale Online Education" under Grant No. CXY-2022-177.

6. REFERENCES

[1] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606–615.

[2] Feifan Fan, Yansong Feng, and Dongyan Zhao, "Multi-grained attention network for aspect-level sentiment classification," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 3433–3442.

[3] Xin Li, Lidong Bing, Wai Lam, and Bei Shi, "Transformation networks for target-oriented sentiment classification," *arXiv preprint arXiv:1805.01086*, 2018.

[4] Shuqin Gu, Lipeng Zhang, Yuexian Hou, and Yin Song, "A position-aware bidirectional attention network for aspect-level sentiment analysis," in *Proceedings of the 27th international conference on computational linguistics*, 2018, pp. 774–784.

[5] Shiliang Zheng and Rui Xia, "Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention," *arXiv preprint arXiv:1802.00892*, 2018.

[6] Chen Zhang, Qiuchi Li, and Dawei Song, "Aspect-based sentiment classification with aspect-specific graph convolutional networks," *arXiv preprint arXiv:1909.03477*, 2019.

[7] Mi Zhang and Tieyun Qian, "Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 3540–3549.

[8] Chenhua Chen, Zhiyang Teng, and Yue Zhang, "Inducing target-specific latent structures for aspect sentiment classification," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5596–5607.

[9] Bin Liang, Rongdi Yin, Lin Gui, Jiachen Du, and Ruifeng Xu, "Jointly learning aspect-focused and inter-aspect relations with graph convolutional networks for aspect sentiment analysis," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 150–161.

[10] Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang, "Relational graph attention network for aspect-based sentiment analysis," *arXiv preprint arXiv:2004.12362*, 2020.

[11] Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong 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)*, 2019, pp. 5679–5688.

[12] Binxuan Huang and Kathleen M Carley, "Syntax-aware aspect level sentiment classification with graph attention networks," *arXiv preprint arXiv:1909.02606*, 2019.

[13] Ruifan Li, Hao Chen, Fangxiang Feng, Zhanyu Ma, Xiaojie Wang, and Eduard Hovy, "Dual graph convolutional networks for aspect-based sentiment analysis," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 6319–6329.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[15] Andre Martins and Ramon Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *International conference on machine learning*. PMLR, 2016, pp. 1614–1623.

[16] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[17] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[18] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke 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)*, 2014, pp. 49–54.

[19] Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos, "Semeval-2015 task 12: Aspect based sentiment analysis," in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015, pp. 486–495.

[20] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al., "Semeval-2016 task 5: Aspect based sentiment analysis," in *International workshop on semantic evaluation*, 2016, pp. 19–30.

[21] Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang, "Aspect term extraction with history attention and selective transformation," *arXiv preprint arXiv:1805.00760*, 2018.

[22] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 452–461.

[23] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui, "Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis," in *Proceedings of the AAAI conference on artificial intelligence*, 2018, vol. 32.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.