Dependency Guided Graph Convolutional Network for Aspect-Based Sentiment Analysis

1st Peng Shi
Department of Computer Science and Technology
National Key Laboratory for Novel Software Technology at Nanjing University
Nanjing, China
spwannasing@gmail.com

2nd Sheng Bi
Department of Computer Science and Technology
National Key Laboratory for Novel Software Technology at Nanjing University
Nanjing, China
benson.bi.cs@gmail.com

***Abstract*—Aspect-based sentiment analysis(ABSA) is a task to identify the sentiment polarity of the given aspect. Usually, attention mechanisms, Convolution Neural Networks(CNN) and Recurrent Neural Network(RNN) are applied to this field. These models can learn the semantic information of sentences but ignore the syntactic information. Dependency tree can help improve this. But in the previous model, it is not treated seriously. In this paper, we propose a model named Dependency Guided Graph Convolutional Network(*DGGCN*) to tackle the shortcomings of previous models. The model uses different types of dependency information to guide the operation of graph convolutional network and employ a filter layer to enhance the important parts while filtering out the irrelevant parts of the sentence. Experiments on four benchmark datasets demonstrate our model's effectiveness, which significantly outperforms previous state-of-the-art baselines.**

***Index Terms*—Aspect-based sentiment analysis, GCN, dependency tree, attention**

I. Introduction

Aspect-based (or aspect-level) sentiment analysis is a refined sentiment classification task aiming at identifying the sentiment polarity (e.g., positive, negative, neutral) of the given aspect words. It is an important subtask in natural language processing and has attracted much attention. As shown in Table I, there is an example of aspected-based sentiment analysis. Aspect-based sentiment analysis is a challenging task because modeling semantic relatedness between the aspect and its relevant contexts is difficult [1]. Therefore, how to capture the relevant syntactical constraints and long-range word dependencies should be taken into consideration.

The initial approach was to manually refine the features [2]. But the manual method is not only slow but also poor in performance. Then the neural network methods are widely used because of its efficiency. Recurrent Neural Networks(RNN) is the most commonly-used technique for this task (see [3], [4], [5], [6] for details). But these models have the weakness of pooling functions [1]. To solve this problem, Zhang et al. [4] proposed a sentence level neural model using two gated neural networks. And Wang et al. [7] adopted attention mechanism, which is first proposed in machine translation [8]. Tang et al. [9] adopted memory network into attention mechanism to explicitly capture the importance of each context word when

TABLE I: An example of a aspect-based sentiment analysis. There is a sentence "Not only was the **food** outstanding, but the little **'perks'** were great." There are two aspects *food* and *perks*, whose sentiment polarities are both positive.

sentence	Not only was the **food** outstanding, but the little **'perks'** were great.	
Aspect	food	perks
Sentiment polarity	positive	positive

inferring the sentiment polarity of the given aspect. These attention-based neural network methods regard the aspect as a query of the calculation, so they can learn the association between aspect and contexts directly. However, the above attention models simply use a single vector to represent the aspect. These models fail to make further improvements on the performance of aspect sentiment classification because the poor capacity of a single vector. Based on this and in order to solve the problem of neglecting the contexts of aspects and the distance feature between context words and the given aspect in the same time, dependency tree is employed ([10], [11], [12]). Dependency tree is mainly used to model the syntactic relations between the aspect and its context words. In this way, the model can take into consideration the grammatical dependency structure of the entire sentence. Zhang et al. [11] employ a convolutional opertion on this type of dependency tree for aspect-based sentiment analysis. The model maps the dependency tree naively to a adjacency matrix and runs graph convolutional network on it to obtain the contextual representation of the sentence.

However, the above models have some drawbacks more or less. First, most of these models for aspect-based sentiment analysis employ attention mechanism to get a vector of a sentence by summing the weighted word embeddings. Although this can capture the semantic information conveniently and has been verified its effectiveness in many previous works, they may lose some important relevant information because of the capability of a single vector and capture some noise irrelevant to the given aspect. They can not attend to the important words and filter out the irrelevant parts of the sentence effectively. Second, their utilization of the dependency tree is too naive and simple. They just map the dependency tree to an adjacency

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

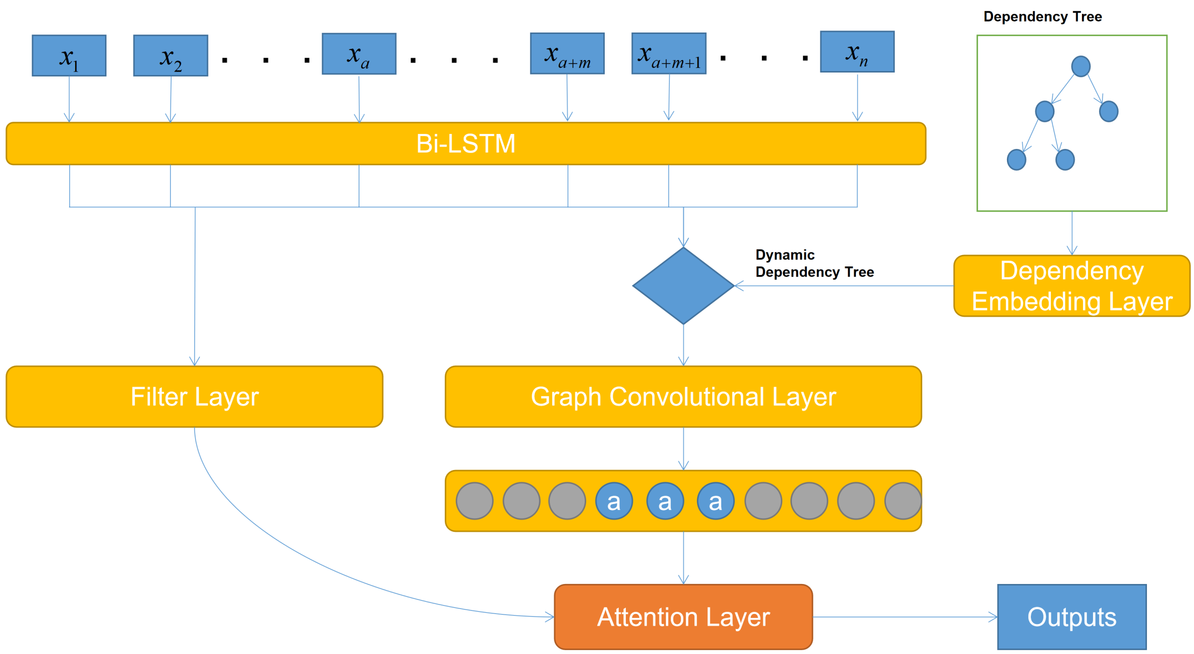

Fig. 1: An overview of our proposed model.

matrix. The item of the adjacency matrix is set to 1 if the corresponding two nodes are adjacent in the dependency tree, otherwise the item is set to 0. In this way, they can not distinguish between two different dependency relations. These models treat these different types of relations the same. Consequently, it limits a further improvement of the results.

In this paper, we propose a novel model named Dependency Guided Graph Convolutional Network(*DGGCN*) to tackle the shortcomings of previous models. Experiments on three benchmarking datasets show that DGGCN effectively addresses both limitations of the current aspect-based sentiment classification models, and outperforms a range of state-of-the-art models.

First, we propose the dependency embedding to construct a dynamic dependency tree. The dynamic dependency tree can model the different syntactic dependency relations between two words, which can help the aspect words consider the relevant context words more. And the embedding mechanism maps different dependency relations to different vectors, which can help distinguish different types of dependency relations and more consideration is given to the important dependency relations during the process of Convolution operations.

In our proposed model DGGCN, whose overview structure is shown in Figure 1, a Bidirectional LSTM is used to capture the contextual representation of the whole sentence. Then, a dynamic dependency tree is constructed by our proposed dependency embedding. This dynamic dependency tree guides the convolutional operations. To get refined text representation, we run graph convolutional network over this dynamic dependency tree, after which we obtain the aspect representation by a mask layer. Next, we run our proposed filter layer on the original text representation to filter out the irrelevant words and take more consideration into the important words. We make a dot production between the aspect representation and enhanced text representation to get a similarity score. We use this score to weight the representation of the text and then sum it. Finally, we feed it to a full connection layer to get the prediction results.

Our contributions are as follows:

- We propose to make deep use of dependency tree, i.e., different types of dependency relations, instead of exploiting it in a naive way.
- We first employ dependency embedding to construct dynamic dependency tree for aspect-based sentiment analysis, which can identify different types of relations between two words so that further improve the performance.
- We propose a filter layer to help the model attend relevant words and introduce less noise of the irrelevant words.
- Extensive experiment results verify that the effectiveness of our model in exploiting different types of dependency relations and demonstrate the importance of filtering out the irrelevant parts of the sentences.

II. Related Work

Aspect-based sentiment analysis (ABSA) can be considered as a fine-grained sentiment analysis task that aims at identifying the sentiment polarity of a sentence expressed

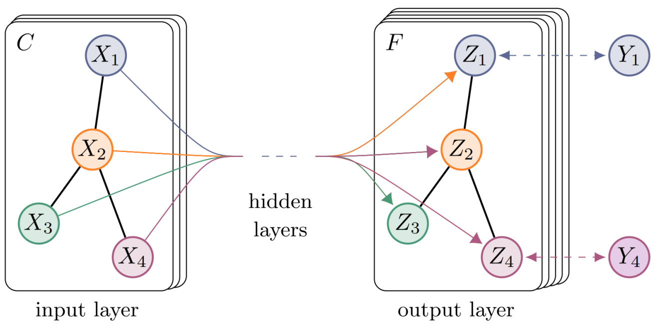

Fig. 2: An example of GCN.

towards a specific aspect [13]. The traditional methods for aspect-based sentiment analysis mostly rely on rich manual features and syntactic dependencies ([14], [15], [16]), which are labor-intensive and can't model the relatedness between the aspect and context words. Jiang et. al [2] first emphasize the importance of targets according to their manual evaluation results that 40% of sentiment classification errors are caused by not considering aspects. Since then, research on ABSA has focused on how to take the aspect information into account better, some related works are described below.

A. Neural Network for Aspect-based Sentiment Analysis

In recent years, aspect-based sentiment analysis has attracted more and more attention with the development of deep learning and has achieved some significant results. Aspect-based sentiment analysis is more difficult than traditional sentiment classification because it needs to model the semantic relatedness of the given aspect with its surrounding context words.

Dong et al. [3] proposed an Adaptive Recursive Neural Network(AdaRNN) for target-dependent twitter classification. The model uses context and syntactic structure to calculate the sentiments of the given aspect words. Tang et al. [5] proposed a model based on LSTM because of its capability of capturing semantic relations between the aspect and its context words. Zhang et al. [4] proposed a sentence level neural model using two gated neural networks. Wang et al. [7] proposed an attention-based LSTM method with target embedding which enforces the model to attend to the important part of a sentecne. Tay et al. [17] designed a Dyadic Memory Network (DyMemNN) that models dyadic interations between aspect and context.

B. Graph Convolutional Networks

In recent years, Graph Convolution Networks (see [18], [19], [20], [21] for details, an example shown in Figure 2) have attracted more and more attention in the field of artificial intelligence and have made some contributions in the field of natural language processing (such as [22]). Marcheggiani and Titov [23] claimed that GCN could be considered as a complement to LSTM, and proposed a GCN-based model for semantic role labeling. Guo et al. [24] proposed to adopt multi-head mechanism [25] into Graph Convolution Networks to automatically learns how to selectively attend to the relevant sub-structures useful for the relation extraction task. Ding et al. [26] use graph convolutional network to handle millions of documents for multi-hop reasoning questions in the HotpotQA fullwiki dataset, which is a dataset of machine reading comprehension. In this paper, we first propose the dependency embedding to construct the dynamic dependency tree. We employ graph convolutional network to capture their hidden relations on our proposed dynamic weight dependency tree.

III. Proposed Model

Figure 1 gives an overview of DGGCN. We will introduce our proposed methods in details in this section.

A. Embedding Layer

Given a n-word sentence $\{w_1, ..., w_{a+1}, ..., w_{a+m}, ..., w_n\}$, the aspect words' span is from *a+1* to *a+m*. The Embedding Layer embeds each token into a low-dimensional real-valued vector space [27] with embedding matrix $\mathbf{E} \in \mathbb{R}^{|V| \times d_e}$, where $|V|$ is the size of vocabulary and d_e is the dimension of the word embeddings. In this way, we can get the inputs of the following layers $\mathbf{X} = \{x_1, x_2..., x_n\} \in \mathbb{R}^{n \times d_e}$.

B. Encoding Layer

However, only using vanilla word embedding can not catch contextual information. Here, we employ the Bidirectional LSTM (see [28] for details) to process the input (i.e., X) to obtain the hidden contextual representation $\mathbf{H} = \{h_0, h_1, ..., h_{a+1}, ..., h_{a+m}, ..., h_n\} \in \mathbb{R}^{n \times 2d_h}$, where $h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}]$ represents the hidden state of the bidirectional LSTM at the i-th time step. Formally, $\overrightarrow{h_i}, \overleftarrow{h_i}$ are calculated as follows:

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(x_i, \overrightarrow{h_{i-1}}) \tag{1}$$
$$\overleftarrow{h_i} = \overleftarrow{LSTM}(x_i, \overleftarrow{h_{i+1}}) \tag{2}$$

C. Dynamic Dependency Tree

Attention-based models can impose some syntactical constraints on attention weights, but the effect of syntactical structure was not fully exploited. The common dependency tree can exploit more but not enough. In order to solve the previous models' problem that the previous dependency tree can not distinguish between different types of relations, we adopt a dynamic dependency tree into our model and it will guide the graph convolutional operation. In this part, we will introduce how to construct relation-specific adjacent matrix and how to obtain dynamic dependency tree through the dependency embedding and relation-specific adjacency matrix.

1) Construct Dependency Tree: The dependency tree is used to express the dependency relations between words in a sentence. Specifically, it analyzes and identifies the grammatical components such as *subject*, *predicate*, *object*, *attribute*, *adverbial* and *complement* in a sentence, each node of which is a word.

TABLE II: Model comparison results

	TWITTER		REST14		REST15		REST16	
Model	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
SVM	63.40	63.30	80.16	N/A	N/A	N/A	N/A	N/A
LSTM	69.56	67.70	78.13	67.47	77.37	55.17	86.80	63.88
ASGCN	72.15	70.40	**80.77**	**72.02**	**79.89**	**61.89**	**88.99**	**67.48**
MemNet	71.48	69.90	79.61	69.64	77.31	58.28	85.44	65.99
IAN	**72.50**	**70.81**	79.26	70.09	78.54	52.65	84.74	55.21
AOA	72.30	70.20	79.97	70.42	78.17	57.02	87.50	66.21
DGGCN(our proposed model)	**74.27**	**72.79**	**81.99**	**74.11**	**80.20**	**63.95**	**87.88**	**70.32**

A adjacency matrix of the dependency tree $\mathbf{A} \in \mathbb{R}^{n \times n}$ is obtained by spacy toolkit, where n is the length of the sentence. If $A_{ij} = 0$, it means there is no dependency between them. Otherwise, we map each type of dependency to a specific integer, e.g., *nsubj* to 1 and *conj* to 2.

2) Dependency Embedding Layer: In previous works, they treat different types of dependency relations between two words the same. Intuitively, different types of children of a node in a dependency tree play different roles, e.g., *nsubj* is different from *conj*.

Here we first propose Dependency Embedding in this field as far as we know. We build an embedding matrix $\mathbf{E_{dep}} \in \mathbb{R}^{k \times d_{dep}}$, where k is the number of different types of dependency relations and d_{dep} means the dimension of the Dependency Embedding. Following the novel dependency embedding layer, we add a full connection layer. Then we obtain a new adjacency matrix $A' \in \mathbb{R}^{n \times n}$

D. Graph Convolution Layer

To model the dependency relations of each part of the sentence with the new adjacency matrix A', we employ Graph Convolution Network(GCN) [22]. Formally, consider an undirected graph $\mathbf{G} = (V, E)$, where V and E are set of nodes and edges respectively. The goal of GCN is to learn a function $f(\cdot, \cdot)$ over the graph $\mathbf{G}$, which takes feature descriptions $\mathbf{H}^k \in \mathbb{R}^{n \times d}$ and corresponding adjacency matrix $\mathbf{A'} \in \mathbb{R}^{n \times n}$ as inputs, and outputs the updated node features $\mathbf{H}^{k+1} \in \mathbb{R}^{n \times d}$. Each GCN layer can be computed as equation 3, especially $\mathbf{H}_0 = \mathbf{H}$.

$$\mathbf{H}^{k+1} = f(\mathbf{H}^k, \mathbf{A}') \tag{3}$$

In our proposed method, the function $f(\cdot, \cdot)$ consist of ReLU [29] and matrix multiplication as equation 4, where $W^k \in \mathbb{R}^{d \times d}$ denotes learnable weight matrix of the k-th GCN layer and $ReLu(\cdot)$ is a rectifier linear unit activation function.

$$\mathbf{H}^{k+1} = ReLU(\mathbf{A'}\mathbf{H}^k W^k) \tag{4}$$

E. Filter Layer&Re-allocate weight

Intuitively, not each part has the same effect on the judgment of sentiment polarity. The more relevant this part is, the more helpful it is to identify the sentiment polarity of the aspect.

So we decide to introduce a weight matrix $\mathbf{W} \in \mathbb{R}^n$, whose item is from 0 to 1, to measure the impact of each word in a sentence of length n. We construct it as follows:

$$H' = [H, H_a, P] \tag{5}$$
$$\hat{H}' = W_t tanh(W_{H'} H' + b_{H'}) \tag{6}$$
$$\mathbf{W} = \sigma(\hat{H}') \tag{7}$$

where H_a is the aspect embedding obtained by averaging the hidden states of the bidirectional LSTM running on the aspect word embedding $\{h_{a+1}, h_{a+2}, ..., h_{a+m}\}$, $W_{H'} \in \mathbb{R}^{d_{hid} \times (2d_h + d_{h_a} + d_p)}$, $W_t \in \mathbb{R}^{1 \times d_{hid}}$, and $b_{H'} \in \mathbb{R}^{d_{hid}}$ are learnable weight matrix, d_{hid} is a hyper-parameter, $P \in \mathbb{R}^n$ is the position embedding computed as equation 8.

$$P_i = \begin{cases} 0, & a+1 \leq i \leq a+m \\ min(|i-(a+1)|, |i-(a+m)|), & Otherwise \end{cases} \tag{8}$$

Then, the re-weighted word embedding is calculated as equation 9, where the $\odot$ operation means elementwise multiplication, $W_f \in \mathbb{R}^{2d_h \times (2d_h + d_{h_a} + d_p)}$ is a learnable weight matrix.

$$H_{filter} = W_f(\mathbf{W} \odot H') \tag{9}$$

What is more, we add residual connection to further improve the performance as follows, where λ is a hyperparameter:

$$H_{final} = \lambda H_{filter} + (1 - \lambda) H \tag{10}$$

F. Attention Layer

Based on previous layers, a refined text representation through l layers of GCN H^l and a reweighted text representation obtained by filter layer H_{final} are produced. Here, we

TABLE III: DGGCN ablation experiments' results

	TWITTER		REST14		REST15		REST16	
Model	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
DGGCN	**74.27**	**72.79**	**81.99**	**74.11**	**80.31**	**63.95**	**88.25**	**70.32**
DGGCN w/o Filter Layer	72.97	70.89	81.88	73.66	79.80	61.59	88.09	68.40
DGGCN w/o Dependency Embedding	72.11	70.47	81.90	73.71	80.20	61.91	87.82	69.66
DGGCN w/o both	72.10	70.22	81.39	72.95	79.70	61.33	87.25	66.79

adopt attention mechanism to summarize all relevant parts in the sentence given the aspect as follows:

$$mask_i = \begin{cases} 1, & a+1 \leq i \leq a+m \\ 0, & Otherwise \end{cases} \quad (11)$$

$$\hat{H}^l = mask \odot H^l \quad (12)$$

$$S = \hat{H}^l H^T_{final} \quad (13)$$

$$\alpha = Softmax(S) \quad (14)$$

$$Softmax(S) = \frac{e^{S_i}}{\sum_{i=1}^{n} e^{S_i}} \quad (15)$$

$$H_{result} = \sum_{i=1}^{n} \alpha_i H_{final_i} \quad (16)$$

First we need to get the $mask \in \mathbb{R}^n$, where $mask_i = 0$ if the i-th word is not one of the aspect words and $mask_i = 1$ if the i-th word is one of the aspect words. Then we can get the aspect representation $\hat{H}_l$ by elementwise multiplication between $mask$ and H_l. After we get $\hat{H}_l$, we can calculate the similarity score between $\hat{H}_l$ and H_{final} by dot production. Then we use the softmax of similarity score as the weights to sum H_{final} to get H_{result}, which used to calculate the final outputs.

G. Loss and Training

Having obtained the sentence's final state $H_{result} \in \mathbb{R}^{2d_h}$, it needs to be fed into a fully-connected layer to yield a probability distribution $\mathbf{P} \in \mathbb{R}^{d_{polarity}}$ over polarity decision space as follows:

$$\mathbf{P} = W_{result} H_{result} \quad (17)$$

where $W_{results} \in \mathbb{R}^{d_{polarity} \times 2d_h}$ is a learnable matrix.

The model is trained by minimizeing the cross entropy using back propagation algorithm, the loss function is defined as follows:

$$Loss = -\sum_{i=1}^{d_{polarity}} y_i log \mathbf{P}_i \quad (18)$$

where $d_{polarity}$ is the number of the sentiment polarity, $y \in \mathbb{R}^{d_{polarity}}$ is a one-hot vector of the ground truth.

IV. Experiment

A. Datasets and Evaluation metrics

Our experiments are conducted on four datasets: one (TWITTER) is originally built by Dong et al [3] containing twitter posts, while the other three (REST14, REST15, REST16) are respectively from SemEval 2014 task 4 [13], SemEval 2015 [30] task 12 and SemEval 2016 task 5 [31].

Following the previous work (see [11], [32] for details), we choose macro-F1 score to evaluate the performance of our model. The accurate score is also reported to assist the analysis.

B. Experiment setting

For all our experiments, we use 300-dimensional pre-trained GloVe vectors [33] to initialize word embeddings, and the out-of-vocabulary (OOV) words are replaced with *UNK* token initialized with uniform distribution $U(-0.1, 0.1)$. The dimensionality of the LSTM's hidden state vector is set to 300. The dimensionality of the hidden state of the Filter layer is also set to 300. We use the Adam [34] optimization method to minimize the cross-entropy loss over the training data. For the hyper-parameters of the Adam optimizer, we set the batch size to 64, the learning rate $\alpha = 0.001$, momentum parameters$\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The dimensionality of dynamic dependency embedding is set to 1. What is more,the number of GCN layers is set to 2 and the number of filter layers is set to 1.

Following the previous work such as [11], the experimental results are obtained by averaging 3 runs with the random initialization, where Accuracy and Macro-Averaged F1 are adopted as the evaluation metrics.

C. Models for comparison

In order to comprehensively evaluate our model, namely, DGGCN, we compare it with a range of baselines and state-of-the-art models, as listed below:

- SVM [35] is the model which has won SemEval 2014 task 4 with conventional feature extraction methods.
- LSTM [5] uses the last hidden state vector of LSTM to predict sentiment polarity.
- ASGCN [11] is the model using dependency tree too, but its usage is naive compared to ours and it can not filter out the irrelevant parts of the sentecne.

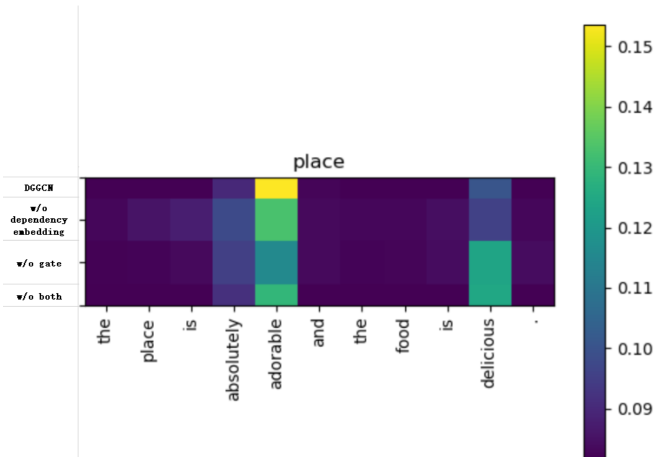

Fig. 3: A case of the attention scores of four models.

- MemNet [9] considers contexts as external memories and benefits from a multi-hop architecture.
- IAN [32] interactively models the relationships between aspects and their contexts.
- AOA [36] borrows the idea of attention-over-attention from the field of machine translation.

D. Results

As shown in Table II, our proposed model DGGCN consistently outperforms all compared models on all datasets. The results demonstrate the effectiveness of DGGCN and drawbacks of previous models. Formally, it makes an improvement of 2.4% accuracy rate and 2.79% on TWITTER dataset compared with IAN [32], which performs best at this dataset among the previous models. For the other dataset, ASGCN [11] performs best among the previous models. And our model DGGCN yields a significant performance gain compared with it, i.e., about an improvement of 2.3% of macro-F1 on these datasets compared with ASGCN [11].

E. Ablation study

In order to verify the contribution of the most innovative components of our model ,i.e., dependency embedding and filter layer, we design three simplified model to conduct the ablation study:

- DGGCN w/o Dependency Embedding removes the dependency embedding,instead we just use the naive adjacency matrix of the dependency tree as previous works [11].
- DGGCN w/o Filter Layer removes the Filter layer module, which can not filter out the irrelevant words given aspect words.
- DGGCN w/o both removes both dependency embedding and filter layer.

As we can see from Table III, both the dependency embedding and filter layer make a great contribution to the final performance. The complete model has a significant lead in macro-F1 scores on each data set.

DGGCN w/o Dependency Embedding suffers a reduction of 2.32% macro-F1 at most. This is because a naive adjacency matrix cannot tell the difference between different dependency relations. The complete model automatically learn the difference between different dependency relations in the process of training so that makes a further improvement. DGGCN w/o Filter Layer suffers a reduction of 1.92% macro-F1 at most. This is because the irrelevant parts of a sentence have negative effects on the results. The complete model makes an improvement by reducing the weights of useless words and take more consideration into the important words.

F. Case Study

To further understand our model, we will show case study of some examples in the test set to verify its performance. We visualize the attention scores of the layer before the prediction layer. Three ablation models, i.e., DGGCN without filter layer, DGGCN without dependency embedding, and DGGCN without both, are compared with the complete DGGCN model, whose results are shown in Figure 3.

In the example shown in Figure 3, there is a sentence “the place is absolutely adorable and the food is delicious.”. There are two aspects “place” and “food”, and the attention scores of each word over the “place” have been visualized. The most important parts for determining the sentiment polarity of the given aspect is recognizing the corresponding adjective in the sentence, i.e., “adorable” instead of “delicious”. When considering the aspect “food”, the DGGCN puts most attention on “adorable” and puts least attention on “delicious”. The other models don’t recognize this situation and have a worse performance. The attention may attend to some wrong aspect’s context words when there are more than one aspect in one sentence. The dynamic embedding and filter layer in DGGCN can help release the problem.

G. Investigation on the Dependency Embedding

To Investigate the effectiveness of our proposed Dependency Embedding, which helps to construct the dynamic dependency tree, we project the dependency embedding into one-dimension vector space and do a softmax calculation on it. Then we visualize the final result as shown in Figure 4.

In the section one, i.e., Introduction, we have an example “Not only was the food outstanding,but the little ‘perks’ were great”. ‘outstanding’, which is the adjectival modifier(amod), is much more important than ‘the’, which is a determiner(det), when determining the sentiment polarity of the given aspect ‘food’. This situation can be shown in Figure 4. The brighter the color, the greater the effect of this dependency relation. This result is corresponding to our knowledge.

V. Conclution

In this paper, we propose a novel model named Dependency Guided Graph Convolutional Network (DGGCN) for aspect-based sentiment analysis and have verified its effectiveness. We first propose a novel dependency embedding and filter layer in this field to tackle with the drawbacks of previous

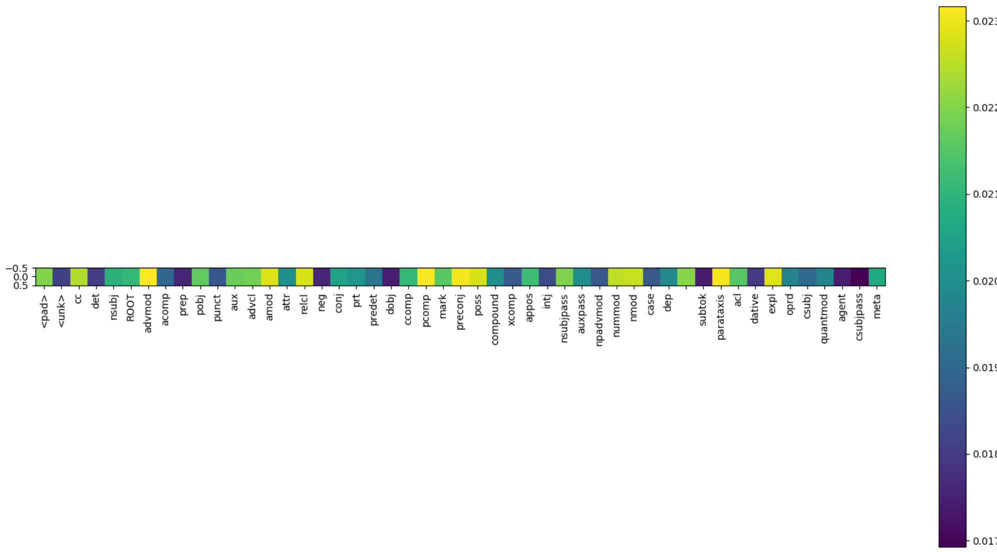

Fig. 4: Visualization of the dynamic embedding(projection)

models. The dependency embedding can help to construct a dynamic dependency tree, which is used to refine the text representation and it can make full use of different types of dependency relations. The filter layer is used to filter out the irrelevant parts of a sentence and attend to the important words. Extensive experiments on four datasets have demonstrated our proposed model's effectiveness.

This study may be further improved by making deeper use of different types of dependency relations. What is more, previos models seldom consider the relations between two aspects in one sentence and this is worthy to conduct more study.

VI. Acknowledgement

This paper is supported by the Natural Science Foundation of China (Grant No. U1811462), the Collaborative Innovation Center of Novel Software Technology and Industrialization at Nanjing University.

References

[1] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[2] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 151–160. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002492

[3] D. Li, F. Wei, C. Tan, D. Tang, and X. Ke, "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.

[4] M. Zhang, Y. Zhang, and D.-T. Vo, "Gated neural networks for targeted sentiment analysis," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[5] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective lstms for target-dependent sentiment classification," *Computer Science*, 2015.

[6] S. Ruder, P. Ghaffari, and J. G. Breslin, "A hierarchical model of reviews for aspect-based sentiment analysis," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 999–1005. [Online]. Available: https://www.aclweb.org/anthology/D16-1103

[7] 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*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 606–615. [Online]. Available: https://www.aclweb.org/anthology/D16-1058

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[9] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network."

[10] Z. Ouyang and J. Su, "Dependency parsing and attention network for aspect-level sentiment classification," in *Natural Language Processing and Chinese Computing*, M. Zhang, V. Ng, D. Zhao, S. Li, and H. Zan, Eds. Cham: Springer International Publishing, 2018, pp. 391–403.

[11] C. Zhang, Q. Li, and D. Song, "Aspect-based sentiment classification with aspect-specific graph convolutional networks."

[12] G. Castellucci, S. Filice, D. Croce, and R. Basili, "UNITOR: Aspect based sentiment analysis with structured learning," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 761–767. [Online]. Available: https://www.aclweb.org/anthology/S14-2135

[13] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 27–35. [Online]. Available: https://www.aclweb.org/anthology/S14-2004

[14] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in *Proceedings of the 2Nd International Conference on Knowledge Capture*, ser. K-CAP '03. New York, NY, USA: ACM, 2003, pp. 70–77. [Online]. Available: http://doi.acm.org/10.1145/945645.945658

[15] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, ser. EMNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 79–86. [Online]. Available: https://doi.org/10.3115/1118693.1118704

[16] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 417–424. [Online]. Available: https://doi.org/10.3115/1073083.1073153

[17] Y. Tay, L. A. Tuan, and S. C. Hui, "Dyadic memory networks for aspect-based sentiment analysis," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: ACM, 2017, pp. 107–116. [Online]. Available: http://doi.acm.org/10.1145/3132847.3132936

[18] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[19] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[20] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[21] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks."

[23] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," *arXiv preprint arXiv:1703.04826*, 2017.

[24] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," *arXiv preprint arXiv:1906.07510*, 2019.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[26] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, "Cognitive graph for multi-hop reading comprehension at scale," *arXiv preprint arXiv:1905.05460*, 2019.

[27] E. J. Kandola, T. Hofmann, T. Poggio, and J. Shawe-Taylor, "A neural probabilistic language model," *Studies in Fuzziness and Soft Computing*, vol. 194, pp. 137–186, 2006.

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

[29] G. E. Hinton, "Rectified linear units improve restricted boltzmann machines vinod nair," 2010.

[30] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 486–495. [Online]. Available: https://www.aclweb.org/anthology/S15-2082

[31] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit, "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 19–30. [Online]. Available: https://www.aclweb.org/anthology/S16-1002

[32] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification."

[33] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization."

[35] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "Nrc-canada-2014: Detecting aspects and sentiment in customer reviews," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 437–442.

[36] B. Huang, Y. Ou, and K. M. Carley, "Aspect level sentiment classification with attention-over-attention neural networks."