BERT4GCN: Using BERT Intermediate Layers to Augment GCN for Aspect-based Sentiment Classification

Zeguan Xiao[1], Jiarun Wu[1], Qingliang Chen[1,2]*, Congjian Deng[3]
[1]Department of Computer Science, Jinan University, Guangzhou 510632, China
[2]Guangzhou Xuanyuan Research Institute Company, Ltd., Guangzhou 510006, China
[3]Guangzhou Yunqu Information Technology Company, Ltd., Guangzhou 510665, China
xiaozg@stu2018.jnu.edu.cn, benwu@stu2019.edu.cn
tpchen@jnu.edu.cn, dcj@yunqu-info.com

Abstract

Graph-based Aspect-based Sentiment Classification (ABSC) approaches have yielded state-of-the-art results, expecially when equipped with contextual word embedding from pre-training language models (PLMs). However, they ignore sequential features of the context and have not yet made the best of PLMs. In this paper, we propose a novel model, BERT4GCN, which integrates the grammatical sequential features from the PLM of BERT, and the syntactic knowledge from dependency graphs. BERT4GCN utilizes outputs from intermediate layers of BERT and positional information between words to augment GCN (Graph Convolutional Network) to better encode the dependency graphs for the downstream classification. Experimental results demonstrate that the proposed BERT4GCN outperforms all state-of-the-art baselines, justifying that augmenting GCN with the grammatical features from intermediate layers of BERT can significantly empower ABSC models.

1 Introduction

Aspect-based sentiment classification (ABSC), a fine-grained sentiment classification task in the field of sentiment analysis, aims at identifying the sentiment polarities of aspects explicitly given in sentences. For example, given the sentence *"The price is reasonable although the service is poor."*, ABSC needs to correctly assign a positive polarity to *price* and a negative one to *service*.

Intuitively, matching aspects with their corresponding opinion expressions is the core of ABSC task. Some of previous deep learning approaches (Tang et al., 2016; Ma et al., 2017; Huang et al., 2018; Song et al., 2019) use various types of attention mechanisms to model the relationship between aspects and opinion expressions in an implicit way. However, these attention-based models do not take good advantage of the syntactic information of sentences, such as the dependency graph, to better associate the aspects with their sentimental polarities, thus leading to poor performance.

In order to better pair aspects and their corresponding opinion expressions, the syntactic features must be taken into account. Recent work such as (Huang and Carley, 2019; Sun et al., 2019; Zhang et al., 2019; Wang et al., 2020; Tang et al., 2020) employs either Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) or Graph Attention Networks (GATs) (Velickovic et al., 2018) on the dependency graph, in which words in the sentence are nodes in the graph and grammatical relations are edge labels. The dependency graph they use can be obtained from either ordinary dependency tree parser or reshaping it using heuristic rules. These models can achieve promisingly better performance with this additional grammatical features, especially when they incorporate the contextual word embedding, such as BERT (Devlin et al., 2019).

Meanwhile, BERTology researchers have investigated what linguistic knowledge can be learned from unlabeled data by language models (Clark et al., 2019; Hewitt and Liang, 2019; Hewitt and Manning, 2019; Jawahar et al., 2019). And it has been shown that BERT captures a rich hierarchy of linguistic information, with surface features in lower layers, syntactic features in middle layers and semantic features in higher layers (Jawahar et al., 2019).

Inspired by Jawahar et al. (2019), we propose a novel framework of BERT4GCN to make the best of the rich hierarchy of linguistic information from BERT in this paper. Specifically, we firstly encode the context with a BiLSTM (Bidirectional Long Short-Term Memory) to capture the contextual information regarding word orders. Then, we use the hidden states of BiLSTM to initiate node representations and employ multi-layer GCN on

[*] Corresponding author (Qingliang Chen).

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 9193–9200
November 7–11, 2021.

the dependency graph. Next, we combine the node representations of each layer of GCN with the hidden states of some intermediate layer of BERT, i.e., for the neighborhood aggregation of GCN. In this way, BERT4GCN can fuse grammatical sequential features with graph-based representations. Besides, we further prune and add the edge of the dependency graph based on self-attention weights in the Transformer encoder (Vaswani et al., 2017) of BERT to deal with parsing errors and make dependency graph better suit ABSC task. In addition, we develop a method which incorporates relative positional embedding in node representations to make GCN position-aware.

Our contributions in this paper are summarized as follows:

- We propose a BERT4GCN model, which naturally incorporates grammatical sequential features and syntactic knowledge from intermediate layers in BERT to augment GCN and thus can produce better encodings for the downstream ABSC task. BERT4GCN can make the best of the hidden linguistic knowledge in BERT without learning from scratch on other sources.
- The experiments have been conducted on SemEval 2014 Task 4 (Pontiki et al., 2014) and Twitter (Dong et al., 2014) datasets, with promising results showing that BERT4GCN achieves new state-of-the-art performance across these prestigious benchmarks.

2 Related Work

Modeling the connection between aspect terms and opinion expressions is the core of the ABSC task. State-of-the-art models combine GNNs (Graph Neural Networks) with dependency graphs whose syntactic information is very helpful. Some have stacked several GNN layers to propagate sentiment from opinion expressions to aspect terms (Huang and Carley, 2019; Sun et al., 2019; Zhang et al., 2019). Some have tried to convert the original dependency graph to the aspect-oriented one and encode dependency relations (Wang et al., 2020). More recently, a dual structure model has been proposed in (Tang et al., 2020), jointly taking the sequential features and dependency graph knowledge together. Our BERT4GCN model has a similar structure to that in (Tang et al., 2020), but ours utilizes the grammatical sequential features directly from intermediate layers in BERT, instead of learning from scratch on other sources.

3 BERT4GCN

3.1 Word Embedding and BiLSTM

Given $s = [w_1, w_2, ..., w_i, ..., w_{i+m-1}, ..., w_n]$ as a sentence of n words and a substring $a = [w_i, ..., w_{i+m-1}]$ representing aspect terms, we first map each word to a low-dimensional word vector. For each word w_i, we get a vector $x_i \in \mathbb{R}^{d_e}$ where d_e is the dimensionality of the word embedding. After that, we employ a BiLSTM to word embeddings to produce hidden state vectors $\mathbf{H} = [h_1, h_2, ..., h_n]$, where each $h_t \in \mathbb{R}^{2d_h}$ represents the hidden state at time step t from bidirectional LSTM, and d_h is the dimensionality of a hidden state output by an unidirectional LSTM. $\mathbf{H}$ will fuse with BERT hidden states to produce the input of the first GCN layer.

3.2 The Usage of BERT

We consider BERT as a grammatical knowledge encoder that encodes input text features in hidden states and self-attention weights. The input of BERT model is formulated as [CLS]s[SEP]a[SEP], where s is the sentence and a is the aspect term.

3.2.1 Hidden States as Augmented Features

BERT captures a rich hierarchy of linguistic information, spreading over the Transformer encoder blocks (Clark et al., 2019; Jawahar et al., 2019). Jawahar et al. (2019) also show that neighbouring layers of BERT learn similar linguistic knowledge. Therefore we select layers uniformly from BERT as the source of augmented features. BERT4GCN utilizes the hidden states of the 1st, 5th, 9th and 12th layers of BERT-Base model as augmented features[1]. We define the hidden states of all BERT layers as $\mathbf{B} = [\mathbf{H}_1^B, \mathbf{H}_2^B, ..., \mathbf{H}_{12}^B]$, where each $\mathbf{H}_i^B \in \mathbb{R}^{n \times d_{BERT}}$, and d_{BERT} is the dimensionality of the hidden state. When a word is split into multiple sub-words, we just use the hidden state corresponding to its first sub-word. Then we can get the augmented features $\mathbf{G}$ for GCN as:

$$\mathbf{G} = [\mathbf{H}_1^B, \mathbf{H}_5^B, \mathbf{H}_9^B, \mathbf{H}_{12}^B] \qquad (1)$$

3.2.2 Supplementing Dependency Graph

Self-attention mechanism captures long-distance dependencies between words. Therefore, we apply self-attention weights of BERT to supplement

[1] We also try other layers to build BERT4GCN$_{2,7,12}$ and BERT4GCN$_{1,4,7,10,12}$ (the numbers in subscript mean layers used), but the model in the paper is the best in experimental results on average.

dependency graphs to deal with parsing errors and make dependency graphs better suit the ABSC task. After getting dependency graphs from an ordinary parser, we substitute the unidirectional edges to bidirectional ones.

Next, we obtain attention weight tensors $\mathbf{A}^{att} = \left[\mathbf{W}_1^{att}, \mathbf{W}_5^{att}, \mathbf{W}_9^{att}, \mathbf{W}_{12}^{att}\right]$, where each $\mathbf{W}_i^{att} \in \mathbb{R}^{h \times n \times n}$ is the self-attention weight tensor of i-th layer Transformer encoder of BERT, and h is the number of attention heads. Then, we average them over the head dimension $\bar{\mathbf{A}}_l^{att} = \frac{1}{h}\sum_{i=1}^{h} \mathbf{A}_{l,i}^{att}$, where $\mathbf{A}_l^{att}$ is the l-th element in $\mathbf{A}^{att}$, and $l \in \{1, 2, 3, 4\}$. Finally, we prune or add directed edges between words if the attention weight is larger or smaller than some thresholds. So the supplemented dependency graph for the l-th layer of GCN is formulated as follows:

$$\mathbf{A}_{l,i,j}^{sup} = \begin{cases} 1 & \alpha \leq \bar{\mathbf{A}}_{l,i,j}^{att} \\ \mathbf{A}_{i,j} & \beta < \bar{\mathbf{A}}_{l,i,j}^{att} < \alpha \\ 0 & \bar{\mathbf{A}}_{l,i,j}^{att} \leq \beta \end{cases} \tag{2}$$

where $\mathbf{A}$ is the adjacency matrix of the original dependency graph, $\mathbf{A}_l^{sup}$ is the adjacency matrix of the supplemented dependency graph and $\mathbf{A}_{l,i,j}^{sup}$ is the element on the i-th row and j-th column of $\mathbf{A}_l^{sup}$, α and β are hyperparameters.

3.3 GCN over Supplemented Dependency Graph

We then apply GCN over the supplemented dependency graph in every layer, whose input is a fusion of BERT hidden states and output of the previous layer as follows:

$$\mathbf{R}_1 = \mathrm{ReLU}\left(\mathbf{G}_1\mathbf{W}_1\right) + \mathbf{H} \tag{3}$$

$$\mathbf{R}_k = \mathrm{ReLU}\left(\mathbf{G}_k\mathbf{W}_k\right) + \mathbf{O}_{k-1} \tag{4}$$

$$\mathbf{O}_{l,i} = \mathrm{ReLU}\left(\frac{1}{d_i}\sum_{j=1}^{n} \mathbf{A}_{l,i,j}^{sup}\mathbf{W}^l\mathbf{R}_{l,j} + \mathbf{b}^l\right) \tag{5}$$

where $k \in \{2, 3, 4\}, l \in \{1, 2, 3, 4\}$, $\mathbf{G}_1$ and $\mathbf{G}_k \in \mathbb{R}^{n \times d_{BERT}}$ is the first and k-th element in $\mathbf{G}$, respectively. $\mathbf{R}_1$ and $\mathbf{R}_k$ are the node representations which fuse the hidden states of BERT with the hidden states of BiLSTM or the output of the preceding GCN layer, before feeding to the first and k-th GCN layer. $\mathbf{O}_{l,i}$ is output of l-th layer in GCN. d_i is the outdegree of node i. The weight $\mathbf{W}_1$,$\mathbf{W}_k$, $\mathbf{W}^l$ and bias $\mathbf{b}^l$ are trainable parameters.

3.4 Taking Relative Positions

The GCN aggregates neighbouring nodes representations in an averaging way and ignores the relative linear positions in the original context. To address this issue, we learn a set of relative position embeddings $\mathbf{P} \in \mathbb{R}^{(2w+1) \times d_p}$ to encode positional information, where w is the positional window hyperparameter. Before aggregating neighbouring node representation, we add the relative position embedding to the node representation, formalizing the relative linear position to the current word as follows:

$$\mathbf{O}_{l,i} = \mathrm{ReLU}\left(\frac{1}{d_i}\sum_{j=1}^{n} \mathbf{A}_{l,i,j}^{sup}\mathbf{W}^l\mathbf{R}_{l,j}^{p} + \mathbf{b}^l\right) \tag{6}$$

$$\mathbf{R}_{l,j}^{p} = \mathbf{R}_{l,j} + \mathbf{P}_{clip(j-i,w)} \tag{7}$$

$$clip\left(x, w\right) = max\left(-w, min(w, x)\right) \tag{8}$$

where $\mathbf{R}_{l,j}^{p}$ is the positional node representation, and $clip$ function returns the embedding index.

3.5 The Training

Having obtained the representation of words after the last GCN layer, we average the representation of the current aspect terms as the final feature for classification:

$$h_a = \frac{1}{m}\sum_{p=i}^{i+m-1} \mathbf{O}_{4,p} \tag{9}$$

It is then fed into a fully-connected layer, followed by a softmax layer to yield a probability distribution $\mathbf{p}$ over polarity space:

$$\mathbf{p} = softmax\left(h_a\mathbf{W}_c + \mathbf{b}_c\right) \tag{10}$$

where $\mathbf{W}_c$ and $\mathbf{b}_c$ are trainable weights and biases, respectively.

The proposed BERT4GCN is optimized by the gradient descent algorithm with the cross entropy loss and L_2-regularization:

$$\mathrm{Loss} = -\sum_{(x,y)\in\mathbb{D}} \mathbf{lnp}_y + \lambda\|\Theta\|_2 \tag{11}$$

where $\mathbb{D}$ denotes the training dataset, y is the ground-truth label, $\mathbf{p}_y$ is the y-th element of $\mathbf{p}$, Θ represents all trainable parameters, and λ is the coefficient of the regularization term.

4 Experiments

4.1 Datasets and Experimental Setup

We have evaluated our model on three widely used datasets, including Laptop and Restaurant datasets

Category	Model	Twitter		Laptop		Restaurant	
		Acc.	Macro-F1	Acc.	Macro-F1	Acc.	Macro-F1
Others	LSTM	67.69	65.61	67.04	58.26	75.63	61.33
	BERT-SPC	72.88	71.78	76.19	70.84	83.74	74.77
	RoBERTa-MLP	72.76	71.73	81.11	77.11	**86.79**	**79.76**
Attention	MemNet	68.41	65.63	68.35	61.61	77.68	64.99
	IAN	70.33	68.12	68.24	59.49	75.78	61.12
	AOA	69.61	67.07	69.61	62.84	77.12	63.73
	AEN-BERT	71.13	70.00	75.64	69.04	80.77	68.87
Graph	CDT	71.93	69.96	72.38	67.16	79.90	68.83
	BERT+GCN	72.46	71.68	73.89	67.62	82.44	72.33
	ASGCN	69.28	66.63	70.55	64.26	78.65	67.03
	ASGCN-BERT	72.20	70.82	76.71	71.61	81.74	71.00
	RGAT	68.24	66.29	71.47	65.29	79.77	69.43
	RGAT-BERT	72.62	71.34	75.67	70.53	83.23	74.26
	DGEDT-BERT	72.79	71.35	76.03	70.70	83.85	75.18
Ours	BERT4GCN	**74.73**	**73.76**	**77.49**	**73.01**	**84.75**	**77.11**
	w/o pos.	74.48	73.22	77.24	72.82	84.41	76.29
	w/o att.	74.15	72.90	77.12	72.46	83.91	75.28
	w/o pos. & w/o att.	74.36	73.31	77.23	72.58	84.09	75.93
	RoBERTa4GCN	**74.75**	**74.00**	**81.80**	**78.16**	86.23	78.61

Table 1: Comparisons of BERT4GCN with various baselines. The w/o pos. indicates the one without using relative position module and w/o att. is without using supplemented dependency graph. Accuracy (Acc.) and Marco-F1 are used for metrics. And we report the average of 10-fold experimental results (%).

from SemEval 2014 Task 4 (Pontiki et al., 2014) and Twitter datasets from Dong et al. (2014). And we compare the proposed BERT4GCN with a series of baselines and state-of-the-art models, including LSTM, BERT-SPC (Song et al., 2019), RoBERTa-MLP (Dai et al., 2021), MemNet (Tang et al., 2016), IAN (Ma et al., 2017), AOA (Huang et al., 2018), AEN-BERT (Song et al., 2019), CDT (Sun et al., 2019), BERT+GCN, ASGCN and ASGCN-BERT (Zhang et al., 2019), RGAT and RGAT-BERT (Wang et al., 2020), and DGEDT-BERT (Tang et al., 2020). We divide them into three categories. The category **Others** includes models that are general and task agnostic ones. The LSTM acts as a baseline for models without using PLM, while BERT-SPC and RoBERTa-MLP are baselines for models with the corresponding PLMs. The categories **Attention** and **Graph** are models mainly based on attention mechanism and graph neural networks, respectively. The details of datasets and experimental setup can be found in Appendix A.1.

4.2 Experimental Results

4.2.1 Overall Results

We now present the comparisons of performance of BERT4GCN with other models in terms of classification accuracy and Macro-F1 on Table 1. From the table, we can observe that our BERT4GCN outperforms all other BERT-based models across all three datasets, justifying that augmenting GCN with the grammatical features from intermediate layers of BERT can empower ABSC models.

4.2.2 Results Analysis

BERT-SPC vs. RoBERTa-MLP. RoBERTa-MLP significantly outperforms BERT-SPC on Laptop and Restaurant datasets, while has similar results to BERT-SPC on Twitter dataset. The same pattern is also observed in the comparison of BERT4GCN and RoBERTa4GCN. One possible reason for this phenomenon is that the corpora which the two PLMs were pre-trained on are far different from Twitter dataset. Therefore, RoBERTa's superiority is not shown on the Twitter dataset, which we call the **out-domain dataset**.

BERT-SPC vs. BERT-based models. From the table, we also see that BERT-SPC can parallel BERT-based models, indicating that model architecture engineering only has a marginal effect when using BERT.

BERT-SPC vs. BERT4GCN. Observed from the experimental results, the improved performance of BERT4GCN over BERT-SPC on Twitter dataset is higher than the other datasets. A similar pattern also appears in the comparison of RoBERTa4GCN and RoBERTa-MLP, where RoBERTa4GCN is

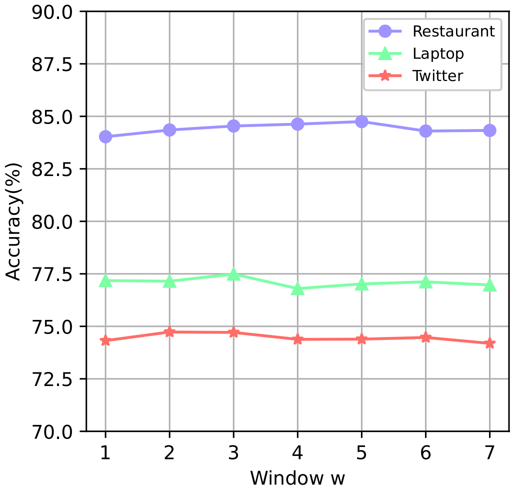

Figure 1: Effect of the size of relative position window w on Twitter (bottom), Laptop (mid) and Restaurant (top) datasets.

just comparable to RoBERTa-MLP on Laptop and Restaurant datasets. For this phenomenon, we have two conjectures: (1) BERT4GCN framework is more flexible to handle the **out-domain dataset**; (2) When PLM is strong enough, with the current model framework, heavy model architecture engineering is unnecessary on the **in-domain dataset**. These two conjectures need to be further explored in future work.

4.2.3 Ablation Study

To examine the effectiveness of different modules in BERT4GCN, we have carried out ablation studies as shown in Table 1. As presented in Table 1, the full model of BERT4GCN has the best performance. And we observe that only fusing grammatical sequential features with GCN (i.e., w/o pos. & w/o att.) can still achieve state-of-the-art results. With the supplemented the dependency graphs (i.e., w/o pos.), the performance increases just slightly.

It is notable that adding the relative position module alone (w/o att) produces a negative effect, and the power of the relative position module can only be revealed when combined together with the supplemented dependency graphs. We conjecture that the supplemented dependency graphs prune some edges which connect nearby aspect terms to words that are irrelevant to sentiment classification, thus reducing the noise.

4.2.4 Effect of Relative Position Window

We have also investigated the effect of the relative position window w on BERT4GCN across three datasets. As shown in Figure 1, Twitter and Laptop datasets prefer a smaller window, while Restaurant dataset prefers a bigger one.

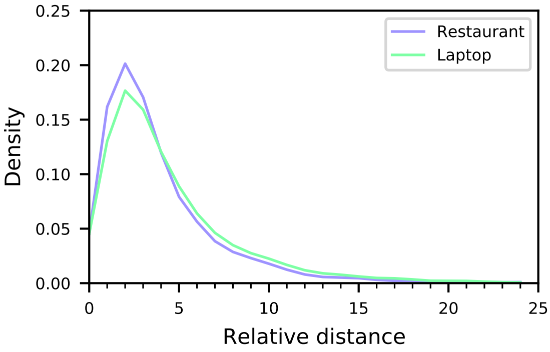

Figure 2: Distributions of the distances between aspect and opinion terms on Laptop and Restaurant datasets.

We calculate the relative distances between aspect and opinion terms of Laptop and Restaurant datasets with annotated aspect-opinion pairs by Fan et al. (2019), and visualize the distributions with kernel density estimation in Figure 2. Although the two distributions are very similar, the optimal window size for the two datasets is not the same. Therefore we hypothesize that the preference of window size is also influenced by the dataset domain. The long-tailed distributions imply that we need to carefully set the window size for the trade-off between the benefits and losses of position biases.

5 Conclusion

In this paper, we propose a BERT4GCN model which integrates the grammatical sequential features from BERT along with the syntactic knowledge from dependency graphs. The proposed model utilizes intermediate layers of BERT, which contain rich and helpful linguistic knowledge, to augment GCN, and furthermore, incorporates relative positional information of words to be position-aware. Finally, experimental results show that our model achieves new state-of-the-art performance on prestigious benchmarks.

Acknowledgements

This research is supported by Qinghai Provincial Science and Technology Research Program (grant No.2021-QY-206), Guangdong Provincial Science and Technology Research Program (grant No.2020A050515014), and National Natural Science Foundation of China (grant No. 62071201).

References

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. 2021. Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1816–1829.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. 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)*, pages 49–54, Baltimore, Maryland. Association for Computational Linguistics.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2509–2518, Minneapolis, Minnesota. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Binxuan Huang and Kathleen Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5469–5477, Hong Kong, China. Association for Computational Linguistics.

Binxuan Huang, Yanglan Ou, and Kathleen M Carley. 2018. Aspect level sentiment classification with attention-over-attention neural networks. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 197–206. Springer.

Lianzhe Huang, Xin Sun, Sujian Li, Linhao Zhang, and Houfeng Wang. 2020. Syntax-aware graph attention network for aspect-level sentiment classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 799–810.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4068–4074. ijcai.org.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. 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), pages 5679–5688, Hong Kong, China. Association for Computational Linguistics.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224, Austin, Texas. Association for Computational Linguistics.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6578–6588, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4568–4578, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Datasets and Experimental Setup

We have evaluated our BERT4GCN model on three widely used datasets, including Laptop and Restaurant datasets from SemEval 2014 Task 4 and Twitter datasets. The statistics of these three datasets are listed in Table 2. Just like other research work, we remove samples with conflicting polarities.

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

Table 2: Statistics of datasets

We use 10-fold cross validation to evaluate the model performance. Specifically, we evaluate models at the end of epochs and save the checkpoint that achieves highest accuracy in the validation set. Finally, we test the models on the test set and report the average of 10-fold experimental results.

A.2 Implementation Details

We choose 300-dimensional Glove vectors for the word embeddings. During training, the learning rate of BERT and other modules linearly warms up from 0 to 0.00002 and 0.001, respectively, then linearly decreases to 0.

In this paper, we name our model as BERT4GCN and use the bert-base-uncased model as the source of grammatical sequential features, but it is easy to extend it to other transformer-based pre-training language models[2].

The training of BERT4GCN model has been run on a Nvidia RTX 3090 that requires about 14GB of GPU memory. The PyTorch implementation of BERT[3] is used in the experiments. The dimensionality of unidirectional LSTM hidden state and positional embedding is set to 300. We set α to 0.01 and β to 0.25, respectively. The position window w is set to 2, 3, 5 on Twitter, Laptop and Restaurant datasets, respectively. And the batch size is set to 32, with Adam as the optimizer. As for the regularization, dropout is applied to BERT and GCN output with the rate of 0.8. And we use Spacy[4] toolkit to generate dependency trees.

For comparisons of baseline models, we use their official implementations or the implementation in ABSA-PyTorch [5], configured with recommended hyperparameters. For BERT+GCN model, we build the graph in the same way as Huang et al. (2020).

[2] For RoBERTa4GCN, we use roberta-base model.

[3] https://github.com/huggingface/transformers

[4] https://spacy.io/

[5] https://github.com/songyouwei/ABSA-PyTorch