Graph Convolutional Networks for Target-oriented Opinion Words Extraction with Adversarial Training

Yan Zhou[1,2], Wei Jiang[3], Po Song[1,2], Yipeng Su[1,*], Tao Guo[1], Jizhong Han[1], Songlin Hu[1,2]
[1]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*
[3]*Department of Energy Internet, State Grid Corporation of China*
zhouyan@iie.ac.cn,wei-jiang@sgcc.com.cn,{songpo,suyipeng,guotao,hanjizhong,husonglin}@iie.ac.cn

***Abstract*—The task of Target-oriented Opinion Words Extraction aims to extract the corresponding opinion words for a given opinion target from the sentence. Recently, the methods based on recurrent neural networks have shown promising results for this task. However, these approaches only considered the sequential information of the sentences and ignored the syntactic structure. In this paper, we propose a novel graph convolutional network with adversarial training to extract the opinion words. We present a graph convolutional network based on dependency tree to learn the syntactic representation of the input. Besides, we train our model with the mixture of original examples and adversarial examples, which can improve the robustness of the model. We conduct experiments on four benchmarking datasets and the results illustrate that our proposed model consistently outperforms the state-of-the-art methods.**

I. INTRODUCTION

Sentiment analysis, also known as opinion mining [1], [2], is an important area in natural language processing (NLP). Traditionally, researchers mainly study sentiment analysis at sentence level or document level, which aims at detecting the overall opinion of the sentence or document. However, different targets or aspects in the same review are usually associated with different sentiments. Different from sentence level and document level sentiment analysis, Aspect Based Sentiment Analysis (ABSA) focuses on detecting the opinions of specific targets in the text, which is more fine-grained.

Target-oriented Opinion Words Extraction (TOWE) [3] is a new subtask of ABSA. This task aims at identifying the corresponding opinion words of the given targets from the input sentence. Opinion targets are the words or phrases representing features or entities towards which users show attitude. Opinion words refer to the terms carrying subjective emotion. For example, in the review "*The food is well prepared and the service is impeccable.*", this task need to identify the opinion words associated with the target "food" is "well prepared" and the opinion word associated with the target "service" is "impeccable". The results of this task can be considered to be an extractive pair-wise opinion summarization, or be further used in downstream applications such as aspect sentiment classification.

There are many researches focused on the co-extraction of the opinion targets and opinion words. Wang et al. [4] integrated recursive neural networks and conditional random fields into a unified framework for aspect and opinion terms co-extraction. A coupled multi-layer attention based neural network was introduced to jointly detect opinion targets and opinion words [5]. Li and Lam [6] designed neural memory operations for handling the extraction of aspects and opinions via memory interactions. In these works, the extracted targets and opinion words are separated and the correspondence is not obtained.

In the literature, there are only a few works about extracting opinion targets and opinion words as pairs. Hu and Liu [7] regarded the nearest adjective of the target as the opinion words. Zhuang et al. [8] used keyword list and dependency relation templates together to mine explicit target-opinion pairs. To reduce the engineering effort, the neural network based method has been proposed to learning the input features automatically. Fan et al. [3] encoded opinion target information by an Inward-Outward LSTM for different targets to extract the corresponding opinion words.

Though great improvements have been achieved by neural network based methods, there still exist some limitations. Firstly, these methods mostly rely on the sequential representation of the sentence and ignore the syntactic dependency structure, which would be beneficial to link the opinion words to the targets. When an opinion word is separated away from its target, it is hard to capture the dependencies between them in a word sequence. Generally, a dependency tree can help to shorten the distance between them. For instance, consider the dependency tree presented in Figure 1, the target "cracking calamari salad" is much closer to the opinion words "crispy" and "lightly dressed" in the dependency graph than in the word sequence. Secondly, it has been shown that neural networks tend to be locally unstable and even tiny perturbations to the inputs can lead to incorrect decisions [9].

In this paper, we propose a novel model based on graph convolutional networks and adversarial training (GCNAT) to extract the opinion words, which can overcome the limitations mentioned above. Specifically, we first use BiLSTM [10] to capture the contextual information between successive words. To integrate the syntactic information, we employ graph convolutional networks (GCN) [11], [12] to model the structure of a sentence through its dependency tree. Finally, we adopt adversarial training (AT) [13] by adding small perturbations to input word embeddings to enhance the robustness of our

* Corresponding author

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

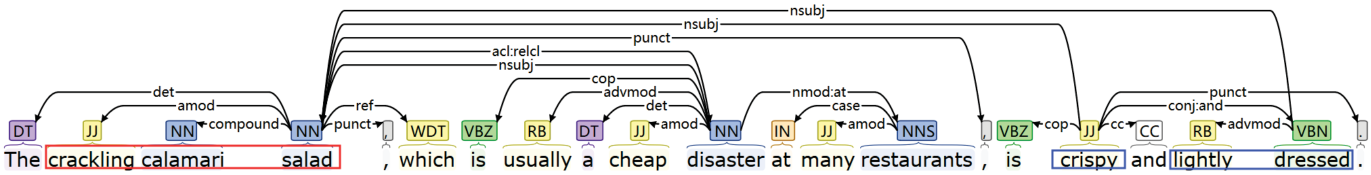

Fig. 1. An example of dependency parsing tree produced by Stanford CoreNLP. The opinion target is "cracking calamari salad" and the corresponding opinion words are "crispy" and "lightly dressed".

model.

The main contributions of this paper are as follows:

- This paper proposes to integrate the syntactic structure of a sentence in TOWE task, and show that GCN is effective for this purpose. To the best of our knowledge, this is the first investigation in this direction.
- We create adversarial samples to conduct the adversarial training, which can enhance the generalization and robustness of our model.
- We conduct experiments on four benchmark datasets and the results show that our method outperforms the state-of-the-art approaches.

II. Related work

A. Target-oriented Opinion Words Extraction

There are a lot of researches about the extraction of opinion target, which is closely related to the task of TOWE. The early studies for opinion target extraction include rule-based approaches [7], [14] and feature-engineering-based approaches [15], [16]. Recently, deep-learning-based approaches have been proposed for this task. Liu et al. [17] proposed a model based on recurrent neural networks (RNNs) and word embeddings to extract opinion targets. Li et al. [18] introduced a framework for aspect term extraction by exploiting the opinion summary and the aspect detection history. A span-based model has been proposed to detect the targets and predict the sentiment of them jointly [19]. Several studies also extract the targets and opinion words simultaneously. These methods conduct the two subtasks into a multi-task learning architecture to extract them jointly, which have achieved great progress on both subtasks [4]–[6]. But the extracted targets and opinion words are separated and lack correspondences between them.

There are only a few works focusing on extracting opinion pairs. Early approaches designed some rules to extract the correpsonding opinion words of the targets [7], [8] . Recently the neural network based models have been proposed. Fan et al. [3] proposed an Inward-Outward LSTM to well encode the opinion target information into context, which achieved promising results. However, these methods neglect the dependency structure of the sentence which is capable of shortening the distance between the targets and opinion words. Besides, small scale perturbations to the input may lead to incorrect decisions.

B. Graph Convolutional Networks

Recently, GCN has attracted a growing attention and has been applied in many NLP tasks. Marcheggiani and Titov [20] proposed a GCN-based model for semantic role labeling. GCN was adopted to learn syntactic contextual representations of each node for event extraction [21], [22]. Zhang et al. [23] encoded the dependency structure over the input sentence with efficient graph convolution operations, then extracts entity-centric representations for relation extraction. The graph-based models has also been sucessfully applied in the task of aspect-based sentiment classification [24]–[27].

C. Adversarial Training

The concept of adversarial training was originally introduced in image classification tasks to improve the robustness of the model by injecting malicious perturbations into input images [9], [13]. Miyato et al. [28] firstly extended adversarial training to the text domain by adding perturbations to the word embeddings. As a a regularization method, AT is further explored in various NLP tasks such as relation extraction [29], part-of-speech tagging [30], jointly extracting entities and relations [31] and neural machine translation [32].

III. Our approach

A. Task Formulation

Given a sentence consisting of n words $X = \{w_1, w_2, \ldots, w_n\}$ and an opinion target ot in the sentence, the TOWE task is to extract the corresponding opinion words of ot. In this paper, this task is formulated as a sequence labeling problem with $\{B, I, O\}$ tagging schema, where B, I, and O denote the beginning of, inside and outside of the corresponding opinion words, respectively. Thus the output of the model is a sequence of tags $Y = \{t_1, t_2, \ldots, t_n\}$, where $t_i \in C$ and $C = \{B, I, O\}$.

For instance, given a sentence "*The food is well prepared and the service is impeccable .*", the tagging sequences for the targets "food" and "service" are as follows:

1. *The/O food/O is/O well/B prepared/I and/O the/O service/O is/O impeccable/O ./O* (Given opinion target "**food**", extract "**well prepared**" as corresponding opinion words).

2. *The/O food/O is/O well/O prepared/O and/O the/O service/O is/O impeccable/B ./O* (Given opinion target "**service**", extract "**impeccable**" as corresponding opinion word).

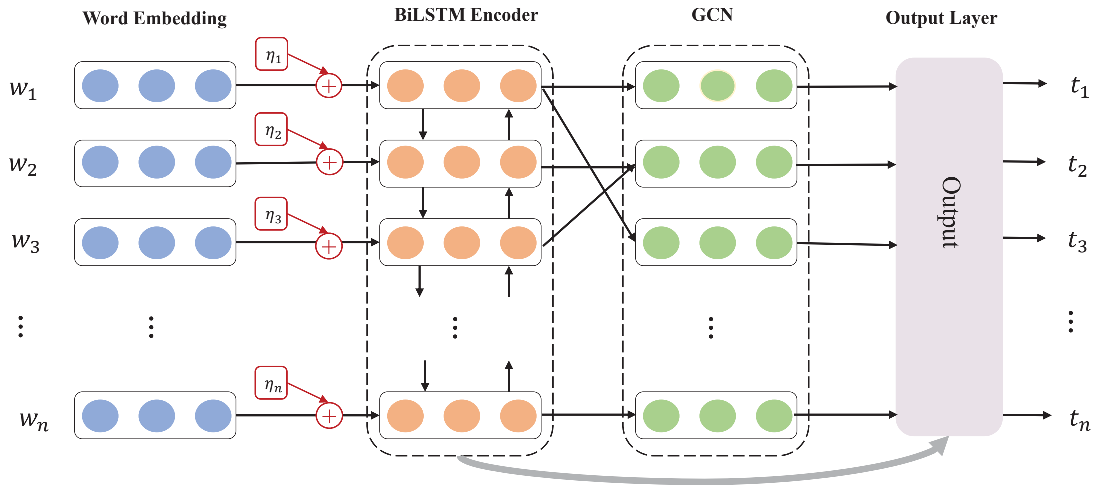

Fig. 2. The architecture of GCNAT.

B. Overview

The architecture of our approach is shown in Figure 2. The input sentence is encoded by a BiLSTM to learn the contextual information of the words. Then we adopt a GCN over the dependency tree to compute the syntactic representation of the sentence. After that, we integrate the outputs from the BiLSTM and GCN to predict the label of each word. During the training processing, we create adversarial samples to conduct the adversarial training, which can improve the robustness and generalization of our model.

C. Word Representation and BiLSTM Encoder

For each word w_i in the input sentence, we create its representation x_i by concatenating its word embedding x_i^w and target mark embedding x_i^m:

$$x_i = [x_i^w; x_i^m] \tag{1}$$

The word embedding is obtained by looking up a pre-trained word embedding matrix Glove [33]. The target mark embedding is created from the binary mark feature. If the word is the opinion target, the value of mark feature is 1; otherwise the value of mark feature is 0.

We employ BiLSTM on the top of the embeddings to capture the local sequential context of each word. One LSTM takes the embeddings $\{x_1, x_2, \ldots, x_n\}$ as inputs and returns a sequence of hidden states $\left\{\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_n}\right\}$. In this way, we can get the left context information of each word. However, the right context is also important, thus we adopt another LSTM to learn the hidden states $\left\{\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots, \overleftarrow{h_n}\right\}$, which contains the right context. We concatenate the hidden states of the two directions together to get the final sequential representation $\{h_1, h_2, \ldots, h_i, \ldots, h_n\}$, where $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$.

D. Graph Convolution over Dependency Trees

The opinion targets and opinion words may be far away from each other in a sequence and the sequential network is low efficiency in capturing such long-range dependencies. An intuitive way to alleviate this phenomenon is to use the dependency tree to shorten the distance between them. Since the dependency trees have graph-like structures, we adopt a graph convolutional network to learn the syntactically relevant information.

When the dependency tree of a sentence is generated, we construct a graph based on the tree. We define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the nodes and edges, respectively. $\mathcal{V}$ consists of n nodes corresponding to the input tokens $\{w_1, w_2, \ldots, w_n\}$. Then we define the edges of the graph based on the dependency parsing tree. If there is a directed syntactic arc from token w_i to token w_j, then the edge (w_i, w_j) belongs to $\mathcal{E}$ and the edge type is $K(w_i, w_j) = dep$. To allow the information flow from the opposite direction, edge (w_j, w_i) is also included in $\mathcal{E}$ and its edge type is $K(w_j, w_i) = rev$. Following the previous work [20], we also add self-loop (w_i, w_i) to the edge set $\mathcal{E}$ and the edge type is $K(w_i, w_i) = loop$. For instance, for the dependency tree in Figure 1, there are four edges in the subgraph containing only two words "crispy" and "salad": the dependency edge $(\text{“}crispy\text{”}, \text{“}salad\text{”})$ with type dep, the inverse dependency edge $(\text{“}salad\text{”}, \text{“}crispy\text{”})$ with type rev and two self-loops $(\text{“}crispy\text{”}, \text{“}crispy\text{”})$ and $(\text{“}salad\text{”}, \text{“}salad\text{”})$ with type $loop$.

After the graph is constructed, we employ multi-layer Graph Convolutional Networks over the graph to obtain syntactic representation. In the k-th layer of the GCNs, we compute the graph convolution vector $h_v^{(k+1)}$ for node $v \in \mathcal{V}$ as follows:

$$h_v^{(k+1)} = ReLU\Big(\sum_{u \in \mathcal{N}(v)} (W_{K(u,v)}^{(k)} h_u^{(k)} + b_{K(u,v)}^{(k)})\Big) \tag{2}$$

where $K(u, v)$ denotes the type of edge (u, v), $\mathcal{N}(v)$ includes v and the neighbors of it; $W_{K(u,v)}^{(k)}$ and $b_{K(u,v)}^{(k)}$ are the weight matrix and bias for edge type $K(u, v)$.

As not all words are equally important to the TOWE task, it is not appropriate to weight the neighbors uniformly. Moreover, there may exist some noises in the dependency parsing tree. In order to address these issues, we calculate an important weight $g^k_{(u,v)}$ for each edge (u, v) [21] :

$$g^k_{(u,v)} = \sigma(h^{(k)}_u \hat{W}^{(k)}_{K(u,v)} + \hat{b}^{(k)}_{K(u,v)}) \tag{3}$$

where σ is the sigmoid function, $\hat{W}^{(k)}_{K(u,v)}$ and $\hat{b}^{(k)}_{K(u,v)}$ denote the weight matrix and bias, respectively. Combining with the edge weight, we formulate the final computation of *k*-th layer GCN as follows:

$$h^{(k+1)}_v = ReLU(\sum_{u\in\mathcal{N}(v)} g^k_{(u,v)}(W^{(k)}_{K(u,v)} h^{(k)}_u + b^{(k)}_{K(u,v)})) \tag{4}$$

We stack m layers of GCN on the top of the BiLSTM layer to implement the syntactic convolution. Note that we utilize the hidden states computed by the BiLSTM as inputs to the first layer of GCN. And we use $\bar{H} = \{\bar{h}_1, \bar{h}_2, \ldots, \bar{h}_n\}$ to denote the vectors computed by the top layer of GCN.

E. Output Layer

The graph convolution can learn a hidden representation from the local graph context of each node. But the number of GCN layer limits the ability to capture the local sequential information. Therefore, we concatenate the representations from the BiLSTM and GCN together to predict the tag of each word. Moreover, since we formulate the TOWE task as a sequence tagging problem, the tag of each word is conditioned on its previous tag. For instance, the tag "O" can not be followed by tag "I" in the $\{B, I, O\}$ tagging schema. To model such dependencies, we also incorporate the tag embedding of the previous word for the prediction. Thus, the final representation s_i to predict the label of i-th word is as follows:

$$s_i = [h_i; \bar{h}_i; o_{(i-1)}] \tag{5}$$

where $o_{(i-1)}$ is the tag embedding directly converted from the tag of the $(i-1)$-th word. Then we adopt softmax to calculate the probability distribution y_i of the i-th word:

$$y_i = softmax(W_y s_i + b_y) \tag{6}$$

We adopt the cross-entropy as the loss function, and the formula is as follows:

$$\mathcal{L}_{ow}(\boldsymbol{X}; \theta) = -\sum_{(x,t)\in D} \sum_{i=1}^{L_x} g_i log(y_i) \tag{7}$$

where $\boldsymbol{X}$ denotes the word embeddings, θ represents all the model parameters, x and t represent the sentence and the opinion target in the sentence, D denotes the set of pair (x, t), L_x is the length of sentence x, g_i is a one-hot vector representing the gold label of the *i*-th word.

F. Adversarial Training

$Adversarial\ Training$ (AT) has been employed as a regularization method, which uses both the clean and adversarial examples to improve the generalization and robustness of the model. The adversarial examples are generated by add noises to the inputs. To be specific, we create an adversarial example by adding the worst-case perturbations η_{adv} to the original word embeddings. The worst-case perturbation η_{adv} is the one that maximize the loss function as follows:

$$\eta_{adv} = \arg\max_{\|\eta\|\leqslant\epsilon} \mathcal{L}_{ow}(\boldsymbol{X} + \eta; \hat{\theta}) \tag{8}$$

where $\hat{\theta}$ is the value of the current model parameters. However, the calculation of η_{adv} is intractable in neural networks. Following the previous work of Goodfellow et al. [13], we approximate the value of η_{adv} by linearizing it as follows:

$$\eta_{adv} = \frac{\epsilon g}{\|g\|}, \quad \text{where} \quad g = \nabla_{\boldsymbol{X}} \mathcal{L}_{ow}(\boldsymbol{X}; \hat{\theta}) \tag{9}$$

where ϵ is a hyperparameter.

At the training step, we generate the adversarial example by $\boldsymbol{X}_{adv} = \boldsymbol{X} + \eta_{adv}$. Then we train the model by the mixture of the original examples and adversarial examples. Therefore the final loss function is :

$$\mathcal{L}_{AT} = \mathcal{L}_{ow}(\boldsymbol{X}; \theta) + \mathcal{L}_{ow}(\boldsymbol{X}_{adv}; \theta) \tag{10}$$

where $\mathcal{L}_{ow}(\boldsymbol{X}; \theta)$ and $\mathcal{L}_{ow}(\boldsymbol{X}_{adv}; \theta)$ represent the cross-entroy loss on the original examples and adversarial examples, respectively.

IV. Experiments

A. Datasets

Dataset	Training		Testing	
	sentences	targets	sentences	targets
14res	1627	2643	500	865
14lap	1158	1634	343	482
15res	754	1076	325	436
16res	1079	1512	329	457

TABLE I
Statistics of datasets.

To evaluate the performance of our approach, we conduct experiments on four benchmark datasets. The four datasets are built by Fan et al. [3] based on the SemEval challenge 2014 Task 4, SemEval Challenge 2015 task 12 and SemEval Challenge 2016 task 5 [34]–[36]. The sentences containing the pairs of the targets and opinion words are preserved. The suffixes "res" and "lap" represent "restaurant" and "laptop", respectively. The statistics of the four datasets is presented in Table I: 14res and 14lap are from SemEval 2014, 15res and 16res are from SemEval 2015 and SemEval 2016. The sentences in 14res, 15res and 16res consist of the reviews from the restaurant domain. And 14lap is comprised by the reviews from laptop domain.

B. Experimental settings

The dimension of the word embeddings, mark embeddings and tag embeddings are 300, 20 and 20, respectively. We adopt 2-layers BiLSTM in our model and the number of the hidden units for each BiLSTM layer is 100. We use Stanford Parser [37] to generate the dependency tree of the inputs. And we adopt 2-layers GCN in our experiments and the dimension of

Model	14res			14lap			15res			16res		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Distance-rule	58.39	43.59	49.92	50.13	33.86	40.42	54.12	39.96	45.97	61.90	44.57	51.83
Dependency-rule	64.57	52.72	58.04	45.09	31.57	37.14	65.49	48.88	55.98	76.03	56.19	64.62
ME-LSTM	60.95	75.88	67.60	55.86	64.90	60.04	57.88	71.44	63.95	67.21	75.66	71.19
TC-BiLSTM	67.65	67.67	67.61	62.45	60.14	61.21	66.06	60.16	62.94	73.46	72.88	73.10
ME-BiLSTM	**83.69**	79.18	81.37	73.50	69.14	71.25	74.11	73.22	73.67	82.85	81.75	82.30
IOG	82.85	77.38	80.02	73.24	69.63	71.35	76.06	70.71	73.25	85.25	78.51	81.69
GCNAT	83.38	**82.27**	**82.82**	**75.43**	**73.09**	**74.24**	**75.74**	**76.59**	**76.17**	**85.43**	**82.90**	**84.14**

TABLE II
COMPARISON RESULTS WITH BASELINES.

Model	14res			14lap			15res			16res		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
GCNAT	83.38	**82.27**	**82.82**	**75.43**	73.09	**74.24**	75.74	**76.59**	**76.17**	85.43	82.90	**84.14**
-GCN	82.42	81.73	82.08	73.51	71.60	72.54	**76.03**	75.09	75.56	**85.78**	81.48	83.58
-AT	**83.73**	80.37	82.01	73.90	**73.33**	73.62	74.76	76.11	75.43	85.02	**83.27**	84.13
-TEMD	83.51	81.30	82.39	74.80	72.84	73.80	74.61	75.53	75.07	84.49	82.67	83.56
-LSTM	71.95	70.66	71.30	61.12	67.80	64.29	67.18	64.71	65.92	77.62	74.86	76.21

TABLE III
ABLATION STUDY OF GCNAT.

the hidden states is 100. We apply dropout over the embedding layer and the dropout rate is set to be 0.5. We use RMSprop as the optimizer with learning rate 0.003 and set the batch size as 32. We randomly sample 20% reviews in the training set as validation set. We test our model on testing dataset and report the average results over five runs. The value of ϵ is set to 0.01. We use the Precision (P), Recall(R) and F1 score as metrics in our experiments.

C. Baselines

In order to comprehensively evaluate our method, we compare our model with both the rule-based methods and the neural network based methods:

- **Distance-rule** [7]: Obtain the nearest adjective as the opinion words associated to the target.
- **Dependency-rule** [8]: Employ the POS tagging results and dependency parsing tree to generate the rule templates and then use these templates to extract opinion pairs.
- **ME-LSTM/ME-BiLSTM** [17]: Use LSTM/BiLSTM to learn the representation of the words and predict their labels by a softmax layer. Different from Liu et al., the input embeddings are the concatenation of the word embeddings and the target mark embeddings.
- **TC-BiLSTM** [38]: The model is similar with ME-BiLSTM model, except that the word embeddings and the target embeddings are concatenated as inputs. The target embeddings are obtained by averaging all the embeddings of the words in the target.
- **IOG** [3]: This approach employs an Inward-Outward LSTM to pass target information to the left context and the right context of the target respectively. Then the left, right and global context are combined to predict the opinion words.

D. Main Results

The comparison results with the baselines are presented in Table II. From the results we observe that our model GCNAT achieves significant improvements over all the baselines in F1 score. Specifically, GCNAT outperforms the state-of-the-art method by 1.45%, 2.89%, 2.5%, and 1.84% on 14res, 14lap, 15res and 16res, respectively.

All the neural network based methods perform better than the rule based methods. The performance of ME-LSTM and TC-BiLSTM is worse than other neural network based methods. This demonstrates that the effective learning of target information is very important for TOWE task. In the approach of ME-LSTM, the words on the left of the target are unable to use the target information. And the TC-BiLSTM directly concatenate the target vector to every word vector, which makes the model difficult to explicitly learn the target information. Since IOG and ME-BiLSTM can learn target-specific representations for each words, they obtain better results than ME-LSTM and TC-BiLSTM. Comparing the F1 score of ME-BiLSTM and IOG, we find that mark embedding is an effective way to indicate the target information. And simply combing the mark embedding with BiLSTM can achieve comparative or even better results than the complicated model IOG which employs three types of LSTM to learn the target-specific representations. Though ME-BiLSTM achieves better results, the performance of it is still lower than our model GCNAT. This demonstrates that the syntax structure of the sentence is useful for TOWE task and adversarial training can enhance the generalization of our model.

E. Ablation Study

To further investigate the effect of each component in the model of GCNAT, we conduct a set of ablation experiments as shown in Table III. The first column indicates the modification of our model: *-GCN* removes the graph convolutional networks; *-AT* removes the adversarial training; *-TEMD* removes

Sentence	Output of ME-BiLSTM	Output of IOG	Output of GCNAT
1. I am pleased with the fast log on , speedy wifi connection and the long battery life ($>$ 6 hrs)	pleased, fast ✓	pleased, fast ✓	pleased, fast ✓
2. I am pleased with the fast log on , speedy wifi connection and the long battery life ($>$ 6 hrs)	speedy ✗	speedy ✗	pleased, speedy ✓
3. I am pleased with the fast log on , speedy wifi connection and the long battery life ($>$ 6 hrs)	pleased ✗	long ✗	pleased, long ✓
4. Only suggestion is that you skip the dessert , it was overpriced and fell short on taste	NONE ✗	overpriced, fell, short ✗	overpriced ✓
5. Only suggestion is that you skip the dessert , it was overpriced and fell short on taste	short ✓	short ✓	short ✓

TABLE IV
EXAMPLES OF THE EXTRACTED RESULTS. THE BLUE WORDS IN THE SENTENCE ARE THE TARGETS AND THE RED WORDS ARE THE CORRESPONDING OPINION WORDS OF IT.

the tag embeddings in the output layer of the model; *-LSTM* removes the BiLSTM encoder layer from the model.

From the results, we observe that the integrated model performs better than the others. And the F1 score drops considerably, when GCN is removed from our model. This indicates that the syntactic information is useful for the extraction of corresponding opinion words. The performance of the GCNAT becomes poor, when it is only trained on the original examples. This is because the adversarial examples can improve the robustness and generalization of the model. Since the TOWE task is formulated as a sequence tagging problem, there exist dependencies between the successive tags of the words. From the line of *-TEMD*, we can see that the performance will decline when the model does not leverage such dependency. From the last line, we find that the performance degrades significantly when the BiLSTM encoder is removed from the model. The local information is very important for TOWE task, since it need to detect the opinion terms which are consecutive words. The number of GCN layer limits the ability of capturing such local sequential information. The recurrent neural network is excellent at learning the local semantic information of the words.

F. Effect of GCN Layers and Scaling Parameter ϵ

We study the effect of GCN layers on the datasets 14res and 14lap as shown in Figure 3. We vary the number of GCN layers from 1 to 5 and report the results of the model. GCNAT achieves the best performance when the number of layers is 1 on 14res and 2 on 14lap. And the performance of the model will not increase as the layers of GCN are added. We consider this is because there may be some noise in the dependency tree and too many layers of GCN can lead to noise accumulation.

We also investigate of effect of different scaling parameter ϵ on the datasets of 14res and 14lap. We evaluate the value of ϵ in the set $\{0.001, 0.01, 0.1, 1, 10\}$ and the results are illustrated in Figure 4. It can be observed that GCNAT obtain the best F1 score when the value of ϵ is 0.1 on both datasets. When the value of ϵ is greater than 0.1, the performance of the prediction will decline. The reason is probably that the semantics of the inputs may be changed by greater noise.

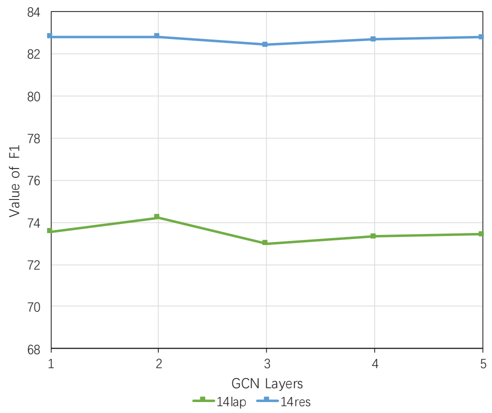

Fig. 3. The effect of GCN layers.

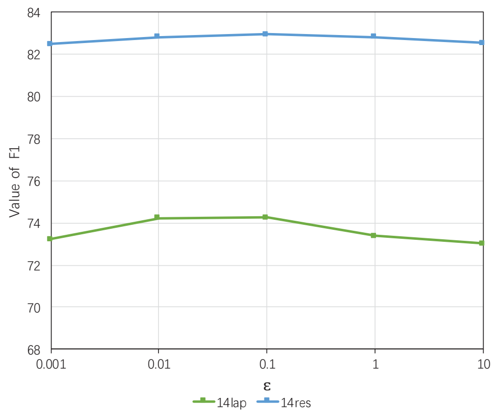

Fig. 4. The effect of scaling parameter ϵ.

G. Case study

To demonstrate the effectiveness of our approach, we pick a few review examples from the test datasets and show the extraction results in Table IV. The first column is the input sentence. The blue words in the sentence are the targets and the red words are the corresponding opinion words of it. The other three columns present the prediction results from ME-BiLSTM, IOG and GCNAT, respectively. In the first and last sentences, as the opinion words lies close to the targets, all the three models can extract the opinion words correctly. In other cases, ME-BiLSTM and IOG fail to identify certain opinion words related to the target. For example, both of them do not detect the opinion word "pleased" to the target "wifi connection" in the second sentence. And in the fourth example, IOG extracts irrelevant opinion words (i.e. "fell" and "short"), while ME-BiLSTM fails to extract any opinion words. And our approach is able to extract the opinion words from all the sentences. This shows that our model, which can utilize the syntactic information, is better at detecting the corresponding opinion words.

V. Conclusion

In this paper, we propose a novel graph convolutional network with adversarial training (GCNAT) to detect target-oriented opinion words. We introduce to use GCN over the dependency graph to learn the representation of the words, which can propagate the information across the tree and capture the syntactic relations between words. To improve the generalization and robustness of our proposed model, we adopt the mixture of the clean examples and adversarial examples to train the model. The experimental results on four public datasets prove the effectiveness of our proposed method.

References

[1] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[2] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[3] Zhifang Fan, Zhen Wu, Xinyu Dai, Shujian Huang, and Jiajun Chen. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *NAACL*, pages 2509–2518, 2019.

[4] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive neural conditional random fields for aspect-based sentiment analysis. In *EMNLP*, pages 616–626, 2016.

[5] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*, 2017.

[6] Xin Li and Wai Lam. Deep multi-task learning for aspect term extraction with memory interaction. In *EMNLP*, pages 2886–2892, 2017.

[7] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[8] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *CIKM*, pages 43–50. ACM, 2006.

[9] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[12] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *Proceedings of ICLR*, 2016.

[13] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[14] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27, 2011.

[15] Wei Jin, Hung Hay Ho, and Rohini K Srihari. A novel lexicalized hmm-based learning framework for web opinion mining. In *ICML*, pages 465–472. Citeseer, 2009.

[16] Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. Structure-aware review mining and summarization. In *Proceedings of the 23rd international conference on computational linguistics*, pages 653–661. Association for Computational Linguistics, 2010.

[17] Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*, pages 1433–1443, 2015.

[18] 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.

[19] Yan Zhou, Longtao Huang, Tao Guo, Jizhong Han, and Songlin Hu. A span-based joint model for opinion target extraction and target sentiment classification. In *IJCAI*, pages 5485–5491. AAAI Press, 2019.

[20] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. In *EMNLP*, pages 1506–1515, 2017.

[21] Xiao Liu, Zhunchen Luo, and Heyan Huang. Jointly multiple events extraction via attention-based graph information aggregation. *EMNLP*, 2018.

[22] Thien Huu Nguyen and Ralph Grishman. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*, 2018.

[23] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*, pages 2205–2215, 2018.

[24] Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. Aspect-level sentiment analysis via convolution over dependency tree. In *EMNLP*, pages 5683–5692, 2019.

[25] Chen Zhang, Qiuchi Li, and Dawei Song. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *EMNLP*, pages 4560–4570, 2019.

[26] Binxuan Huang and Kathleen M Carley. Syntax-aware aspect level sentiment classification with graph attention networks. In *EMNLP*, pages 5472–5480, 2019.

[27] Junjie Chen, Hongxu Hou, Yatu Ji, and Jing Gao. Graph convolutional networks with structural attention model for aspect based sentiment analysis. In *IJCNN*, pages 1–7. IEEE, 2019.

[28] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *ICLR*, 2017.

[29] Yi Wu, David Bamman, and Stuart Russell. Adversarial training for relation extraction. In *EMNLP*, pages 1778–1783, 2017.

[30] Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. Robust multilingual part-of-speech tagging via adversarial training. In *NAACL*, pages 976–986, 2018.

[31] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Adversarial training for multi-context joint entity and relation extraction. In *EMNLP*, pages 2830–2836, 2018.

[32] Motoki Sano, Jun Suzuki, and Shun Kiyono. Effective adversarial regularization for neural machine translation. In *ACL*, pages 204–210, 2019.

[33] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[34] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *SemEval 2014*, pages 27–35, 2014.

[35] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval 2015*, pages 486–495, 2015.

[36] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *SemEval 2016*, pages 19–30, 2016.

[37] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60, 2014.

[38] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective lstms for target-dependent sentiment classification. In *COLING*, pages 3298–3307, 2016.