Target-oriented Opinion Words Extraction via graph convolutional network

BaoKai Li1,a, YueMing Dai1,*
1 School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu, 214122, China
a lbk2715@163.com *dai_ym_516@jiangnan.edu.cn

Abstract. Target-oriented opinion word extraction (TOWE) is a sequence labelling subtask of aspect-level sentiment analysis (ABSA), which aims to extract corresponding opinion words for a given sentence and opinion target. In view of the existing related work, the structural information of sentences is not fully considered. In this study, we explore the integration of neural network and dependency tree to handle the TOWE and propose a model based on graph convolution network (GCN). The model uses the long short-term memory network (LSTM) to learn the semantic features of sentences. On this basis, the model uses the GCN to model the sentence structure through dependency tree and capture the syntactic dependency relationship between the target word and the opinion word. The experimental results show that GCN can effectively improve the performance of TOWE. The F1 values of the model on the semeval-rest14 and semeval-laptop14 reached 82.60% and 74.45%.

1. Introduction
Sentiment analysis can fully mine the information in comments[1]. Aspect-based Sentiment Analysis (ABSA) is an important research area in sentiment analysis, it provides more detailed information than document-level sentiment analysis or sentence-level sentiment analysis[2]. More specifically, ABSA involves three fundamental subtasks: aspect term extraction, opinion term extraction and aspect-level sentiment classification. Consider the example in Figure 1, in the sentence “The food is great, but the waiter is unfriendly.”, the aspect terms are “food/waiter” and the opinion terms are “great/unfriendly”. The comprehensive semantics of the sentence indicates that the emotional polarity corresponding to the opinion target is positive and negative respectively.
Recently, deep learning methods have made a lot of progress in Opinion Term Extraction. Liu apply a recurrent neural network(RNN) with pre-trained word embeddings and Xu propose a Convolutional Neural Networks(CNN) model with double embeddings to solve this task[3][4]. However, these works ignored the relationship between aspect terms and opinion terms. In view of this, researcher proposed Target-oriented Opinion Words Extraction (TOWE). As shown in Figure 2, the objective of TOWE is to extract the corresponding opinion words evaluating the target from the review.

More recently Fan design an aspect-fused sequence labelling approach and Wu utilize a transfer learning method that leverages latent opinions knowledge from auxiliary datasets to handle the TOWE[5][6]. However, most TOWE-related methods only extract sequence features in the text, which is not enough to characterize the complex semantics of the text. In fact, use graph structure features connect different types word segmentation according to the relationship between words can more comprehensively express the grammatical relationship in the sentence. Therefore, we believe that it is feasible to integrate the information in the dependency tree in the neural network model to improve the performance of TOWE.

In this study, we design a model combining bidirectional long short-term memory network and graph convolution network (Bi-LSTM-GCN). The model uses Bi-LSTM to learn the representation of sentence features, and then uses GCN to operates directly on the dependent syntax tree of the sentence to further enhance embedding. Our model propagates both contextual and dependency information from target words to opinion words. The experimental results show that the model can fully mine the information in the dependency syntax tree and effectively improve the performance of TOWE. Compared with the existing benchmarks, the performance of our model has been significantly improved in semeval-rest14 and semeval-laptop14. Therefore, this study makes a major contribution to research on TOWE.

2. Materials and Methods

2.1. Labeling scheme

We use the BIO tagging scheme on TOWE. Specifically, we tag each word \(w_i\) in the sentence \(s\) as \(y_i \in \{B, I, O\}\) (B: Beginning, I: Inside, O: Others). For example, the sentence “The food is great, but the waiter is unfriendly.” is tagged in \(w_i / y_i\) style as follow:

The/O food/O is/O \{great/B\} ,/O but/O the/O waiter/O is/O unfriendly /O ./O (Given opinion target: food, extract “great” as corresponding opinion word).

2.2. Framework

Our model is divided into four parts: presentation layer, Bi-LSTM layer, GCN layer and SoftMax layer. The framework is shown in Figure 3.
2.3. Presentation layer

Our model uses pre-trained GloVe vectors to complete word embedding[7]. The word embedding vector is generated for each word in the sentence through the embedding lookup table $L \in R^{d \times |V|}$, where $d$ is the embedding dimension and $|V|$ is vocabulary size. The presentation layer maps $s = \{w_1, w_2, ...w_i, ... , w_n\}$ to a vector sequence $\{e_1, e_2, ... , e_i, e_n\}$ as a word representation, where $e_i \in R^d$.

2.4. Encoding Layer

2.4.1. BI-LSTM

The Bi-LSTM coding layer is composed of two parallel LSTM layers, namely the Forward-LSTM and the Backward-LSTM[8]. The neuron structure of the Forward-LSTM is shown in Figure 4.

LSTM is composed of four main elements: memory unit, input gate, forget gate and output gate. In the forward network, each time a new word segmentation feature vector $x_i$ is input, and the state $h_{t-1}$ at the previous time is combined to produce the state $h_t$ at the next time, where $t$ represents the time step. The calculation of the hidden state $h_t$ is shown in the following formula:
\[ i_t = \delta(W_x x_t + W_h h_{t-1} + W_c c_{t-1} + b) \] (1)

\[ f_t = \delta(W_x x_t + W_h h_{t-1} + W_c c_{t-1} + b) \] (2)

\[ z_t = \tanh(W_x x_t + W_h h_{t-1} + b) \] (3)

\[ c_t = f_t c_{t-1} + i_t z_t \] (4)

\[ o_t = \delta(W_x x_t + W_h h_{t-1} + W_c c_t + b) \] (5)

\[ h_t = o_t \tanh(c_t) \] (6)

\[ h_t = [\vec{h}_t \oplus \vec{h}_t] \in \mathbb{R}^{2d_x} \] (7)

\[ H = (h_1, h_2, \ldots, h_n) \in \mathbb{R}^{2d_x \times n} \] (8)

Where \( i \) is input gate, \( f \) is forget gate, \( o \) is output gate, \( b \) is the bias term, and \( W \) is the parameter matrix. The output of Forward-LSTM is denoted as \( \vec{h}_t \) and the output of Backward-LSTM is denoted as \( \vec{h}_t \). Finally, concatenate \( \vec{h}_t \) and \( \vec{h}_t \) to represent the encoded information of the \( t \)-th word segmentation, as shown in formula (7), where \( \oplus \) represents the vector concatenation, and \( d_x \) is the one-way LSTM network dimension. For the input \( S \), the output of this layer is shown in formula (8), and \( H \) is output to the next layer as input.

2.4.2. GCN

As shown in Figure 5, the dependency syntax tree shows the dependency between text segmentation. In the dependency syntax tree, root is the virtual root node, and there is one and only one node that depends on the root node.

![Dependency tree](image)

Figure 5. The dependency tree of “But the staff was so horrible to us.”.

Dependency syntax tree is a topological graph \( G \) with \( n \) nodes. The nodes of the dependency tree are embedded by the context coding representation \( h \). The dependency tree \( G \) of the sentence can be expressed as an \( n \times n \) adjacency matrix \( A \). Specifically, if the node \( i \) is connected to the node \( j \) through the dependent path in \( G \), then \( A_{ij} = 1 \), otherwise \( A_{ij} = 0 \).

GCN can effectively use the dependent path to transform and propagate the information on the path, and update the node embedding by aggregating the propagated information\[9\]. GCN only calculates the first-order neighbourhood of the node when modelling its embedding, and \( k \) consecutive GCN operations can spread information in the first-order neighbourhood. The single-node embedding update process is shown in formula (9):
\[ h_i^{(k+1)} = \phi \left( \sum_{j=1}^{n} c_i^j A_j \left( W(k) h_j^{(k)} + b^{(k)} \right) \right) \]  

Where \( h_j^{(k)} \) is the hidden state representation of the node \( j \) in the \( k \)th GCN layer, \( b^{(k)} \) is the bias term, \( W(k) \) is the parameter matrix, \( c_i^j \) is the normalization constant, \( d_i \) is the degree of the node \( i \) in the graph, \( \phi() \) is the nonlinear activation function, \( h_i^{(k+1)} \) is the final output of the layer node. After GCN completes the node update on the dependency syntax tree of the opinion sentence, the feature embedding that integrates the syntax dependency information is obtained:

\[ H_{GCN} = [h_1, h_2, \ldots, h_n] \]  

In summary, the encoding layer can be understood as an information transfer network. Bi-LSTM makes each word in the sentence contain its contextual information and generates a coded representation of the overall semantics of the sentence \( \tilde{H} \); GCN captures the syntactic dependency information between the target word and the opinion word, ensuring that the sentence structure represented by the dependency tree is effectively encoded as \( H_{GCN} \). Finally, the coding layer concatenates \( \tilde{H} \) and \( H_{GCN} \) to obtain the final representation of each word in the sentence \( r_i \):

\[ r_i = \left[ h_i; h_{GCN} \right] \]  

2.5. SoftMax

In decoding, we can use a sequential representation \( r \) to compute \( p(y | r) \), where \( y = \{y_1, \ldots, y_n\} \) are BIO-label sequence for the sentence and \( y_i \in \{B, I, O\} \). Greedy search is to select the output value with the largest probability at each step. It does not consider the dependencies between tags but runs fast. We use softmax to compute the probability and formulate three types of classification problems independently at each position:

\[ p(y_i | r_i) = \text{softmax} \left( W_i r_i + b_i \right) \]  

Our model uses the negative log likelihood (NLL) as the loss for one sentence:

\[ L(s) = -\sum_{i=1}^{n} \sum_{k=1}^{3} \mathbb{I}(y_i = k) \log p(y_i = k | w_i) \]

3. Result & Discussion

3.1. Datasets and Parameter settings

We evaluated the performance of our model on further annotated Rest14 and Laptop14. The two datasets are reviews in the field of restaurants and laptops. We summarize the statistics of the datasets in Table 1:

| Dataset   | Sentence | Target |
|-----------|----------|--------|
| Rest14    | Rest14-train | 1627   | 2643    |
|           | Rest14-test  | 500    | 865     |
| Laptop14  | Laptop14-train | 1158  | 1634    |
|           | Laptop14-test | 343   | 482     |

We exploited 300-dimensional GloVe vectors for the word embeddings; We learned a 300-dimensional Bi-LSTM embeddings; We used Stanford parser to parse all sentences; We used the
Adam optimizer with learning rate 0.05 for all datasets. We randomly dropped out 20% of neurons per layer. The GCN model was trained for 50 epochs with batch size 16.

3.2. Results and Discussion
We used a rule-based Dependency-rule method[10] and three neural network-based methods BI-LSTM, Target-Concatenated Bi-LSTM (TC-Bi-LSTM)[11], IO-LSTM -Global context (IOG) and LOTN, a total of 5 methods for comparison. The experimental results are shown in Table 2:

| Model                | Rest14 P  | Rest14 R  | Rest14 F1 | Laptop14 P | Laptop14 R | Laptop14 F1 |
|----------------------|-----------|-----------|-----------|------------|------------|-------------|
| Dependency-rule      | 64.57     | 52.72     | 58.04     | 31.57      | 37.14      |
| BI-LSTM              | 58.34     | 61.73     | 59.95     | 64.52      | 61.45      | 62.71       |
| TC-Bi-LSTM           | 67.65     | 67.67     | 67.61     | 62.45      | 60.14      | 61.21       |
| IOG                  | 82.85     | 77.38     | 80.02     | 73.43      | 68.74      | 70.99       |
| LOTN                 | 84.00     | 80.52     | 82.21     | 77.08      | 67.62      | 72.02       |
| BI-LSTM-GCN          | 85.87     | 79.76     | 82.70     | 77.07      | 71.60      | 74.23       |

Experimental results showed that the rule-based method lacked robustness. The performance of Dependency-rule method was the worst among all the methods; BI-LSTM could effectively capture contextual features, but the BI-LSTM method had nothing to do with the target word, resulting in its lower accuracy than the IOG method by about 18%; TC-Bi-LSTM contained target word information in a cascaded manner, but the cascaded target word information would interfere with other targets in the same opinion sentence, resulting in its performance being about 10% lower than IOG; The IOG method effectively encoded the semantic information of the target word into the context. LOTN transferred latent opinion information from external sentiment classification datasets to improve the performance. Both IOG and LOTN had achieved satisfactory results. Compared with the above method, our method achieved the best performance and the F1 values on the two data sets reached 82.7% and 74.23% respectively. We could conclude that BI-LSTM-GCN can better capture the correspondence between targets and opinion words.

4. Conclusions
In this study, we proposed a sequence labeling model based on graph convolutional network. The model combined neural network and dependency syntax tree for convolution operation. The model effectively encoded the overall semantic information and syntactic dependency information of the sentence into the context through the neural network. The experimental results showed that the model in this paper effectively improved the performance of TOWE. In future work, we could utilize this model to improve the performance of downstream sentiment analysis tasks, and we can also combine this task to build models that are easier to explain, such as multi-task learning.

Acknowledgments
This research was supported by funding from the National Natural Science Foundation of China (61572237) and the National Key Research and Development Program(2020YFB1711102).

References
[1] Schuller, B. Mousa, E. D. & Vryniotis, V. (2015). Sentiment analysis and opinion mining. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 5(5), 255-263.
[2] Ma, D. Li, S. Zhang, X. & Wang, H. (2017). Interactive attention networks for aspect-level sentiment classification.
[3] Liu, P., Joty, S., & Meng, H. (2015). Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.

[4] Xu, H., Liu, B., Shu, L., & Yu, P. S. (2018). Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction. Meeting of the Association for Computational Linguistics.

[5] Fan, Z., Wu, Z., Dai, X. Y., Huang, S., & Chen, J. (2019). Target-oriented Opinion Words Extraction with Target-fused Neural Sequence Labeling. Proceedings of the 2019 Conference of the North.

[6] Wu, Z., Zhao, F., Dai, X. Y., Huang, S., & Chen, J. (2020). Latent opinions transfer network for target-oriented opinion words extraction.

[7] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Conference on Empirical Methods in Natural Language Processing.

[8] Hochreiter, Sepp, Schmidhuber, & Jurgen. (1997). Long short-term memory. Neural Computation.

[9] Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., R Gómez-Bombarelli, Hirzel, T., & Aspuru-Guzik, A., et al. (2015). Convolutional networks on graphs for learning molecular fingerprints. MIT Press.

[10] Li Zhuang, Feng Jing, and Xiaoyan Zhu. (2006). Movieview mining and summarization. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pages 43–50.

[11] D Tang, Qin B, Feng X. (2015) Effective LSTMs for Target-Dependent Sentiment Classification[J]. Computer Science.