Aspect-Based Sentiment Analysis Based on Multi-Channel and Dynamic Weight

Jin Cheng* and Huixiong Yi

Department of Computer Science and Technology, University of Science and Technology of China, Hefei, China
*Corresponding Author's Email: cheng0l0jin@gmail.com

Abstract. With the wide use of graph convolution network (GCN) in various NLP tasks, researchers introduce GCN to aspect-based sentiment analysis. They utilize GCN over dependency tree to capture the syntactic information and obtain the final representation by modeling the interaction between the semantic and syntactic information. However, due to the unreliability of the construction of dependency tree, these interaction-based models may suffer from the side effect of unreliable syntactic information. To tackle the problem, we propose a multi-channel aspect-specific attention mechanism to ease the possible side effect by retaining and exploiting the original semantic information, which is not affected by the syntactic information. We also propose a dynamic weighted concatenation mechanism to sufficiently utilize the multi-channel information by dynamically assigning weights to the multi-channel information. The extensive experiments on five benchmark datasets indicate the effectiveness of the proposed mechanisms.

1. Introduction

Aspect-based sentiment analysis (ABSA) is a fine-grained type of sentiment analysis. The aim of ABSA is to predict the sentiment polarity of the given text toward the given aspect.

Early works are mostly based on neural networks and attention mechanism [1][2]. Although these works achieve promising performance, the attention mechanism may be insufficient to capture the syntactical relation between the context words and the given aspect [3].

With the wide use of GCN in NLP tasks [4], some researchers try to introduce GCN over dependency tree to the ABSA task. Zhang et al. [3] combine GCN with Bi-LSTM to capture the contextual information and dependency information of the given text. Tang et al. [5] utilize Bi-GCN and Transformer to model the flat and graph information. These works obtain the final representation by capturing the interaction between the semantic and syntactic information and achieve state-of-the-art performance. However due to the unreliability of the construction of dependency tree, the syntactic information may become noise in the interactive process.

Considering this, we propose a multi-channel aspect-specific attention mechanism to alleviate the possible side effect of dependency tree. The multi-channel mechanism includes three channels to sufficiently utilize the semantic and syntactic information, i.e., the semantic channel, the syntactic channel, and the semantic-syntactic channel. The semantic channel is only relevant to the original semantic information, which is not affected by the syntactic information. In this way, the possible side effect caused by the unreliability of the construction of dependency tree is alleviated.

To properly exploit the multi-channel information, we also propose a dynamic weighted concatenation mechanism. The idea is based on the observation that different datasets have different linguistic features and different texts in the same dataset are also various in morphology and grammar.
Therefore the model may prefer the semantic or syntactic information for the specific text. The dynamic weighted concatenation mechanism is to model this preference by dynamically assigning weights to the multi-channel information, where the weights are closely related to the specific input.

Based on these two aforementioned mechanisms, we propose a novel sentiment analysis model, namely Aspect Sentiment based on Multi-Channel and Dynamic Concatenation (ASMCDC).

2. ASMCDC

In this section, we will elaborate the ASMCDC model for ABSA. The architecture of ASMCDC is illustrated in Figure 1. The model can mainly be divided into three parts: 1) the basic model, which is the basic network for extracting the semantic and syntactic information of the input sequence; 2) the multi-channel aspect-specific attention (MCASA), which obtains three types of aspect-specific information, the semantic information, the syntactic information and the semantic-syntactic information, respectively; 3) the dynamic weighted concatenation (DWC), which aggregates the three types of information by dynamically assigning weights to the information.

![Figure 1. The architecture of ASMCDC.](image)

2.1. Basic Model

The basic model is the same as the corresponding part of ASGCN [3]. Given a \( n \)-word sentence \( c = \{w_1, w_2, ..., w_{a+1}, ..., w_{a+m}, ..., w_n\} \) with a \( m \)-word aspect \( a = \{w_{a+1}, w_{a+2}, ..., w_{a+m}\} \), we first embed each word into a low-dimensional vector space with GloVe embedding [6]. Then a Bi-LSTM network is used to capture the contextual information of the input:

\[
H^c = \{h^c_1, h^c_2, ..., h^c_n\} = \text{BiLSTM}(\{x_1, x_2, ..., x_n\}),
\]

where \( x_i \in \mathbb{R}^{d_e} \) and \( h^c_i \in \mathbb{R}^{2d_h} \) are the corresponding embedding and hidden representation of \( w_i \).

Following Bi-LSTM is an \( L \)-layer GCN to model the syntactic information of the dependency tree. And the normalization factor and position-aware transformation are adopted as below:

\[
\hat{h}_i^l = \sum_{j=1}^{n} A_{ij} W_l h_{j}^{l-1}, h_i^l = \text{ReLU}(\hat{h}_i^l (d_i + 1)^{-1} + b_i), h_i^l = F(\hat{h}_i^l),
\]

where \( h_i^l \in \mathbb{R}^{2d_h} \) is the output of the \( l \)-th GCN layer for the \( i \)-th token, \( d_i = \sum_{j=1}^{n} A_{ij} \) is the degree of the \( i \)-th token in the dependency tree and \( W_l, b_l \) are trainable parameters. \( F() \) is a position weights assigning function:
\[ q_i = \begin{cases} 
1 - (a + 1 - i)n^{-1} & 1 \leq i < a + 1 \\
0 & a + 1 \leq i \leq a + m \\
1 - (i - a - m)n^{-1} & a + m < i \leq n, 
\end{cases} \]  
(3)

\[ F(h^i_l) = q_i h^i_l, \]  
(4)

where \( q_i \in \mathbb{R} \) is the position-aware weight of the \( i \)-th token. The final syntactic-aware output of the \( L \)-layer GCN is \( H^L = \{ h^1_L, h^2_L, ..., h^d_{L+1}, ..., h^d_{L+m}, h^d_{L+n} \} \). For more details, please refer to [3].

2.2. Multi-Channel Aspect-Specific Attention

In recent works for ABSA, most studies focus on incorporating the semantic information and syntactic information by utilizing the dependency tree of the input in various ways [3][7]. However, due to the unreliability of the construction of dependency tree, these methods may not always be effective, or even be harmful for the task in some situations since the syntactic information may become noise. Based on this fact, we propose a novel multi-channel aspect-specific attention mechanism to fully utilize the semantic and syntactic information with three channels and ease the possible side effect caused by the unreliability of the construction of dependency tree by retaining and exploiting the original semantic information simultaneously.

Considering the aspect-oriented feature in ABSA, an aspect-specific attention mechanism is adopted. In the semantic-syntactic channel of the model, the attention weights and final representation are calculated as follows:

\[ \hat{a}^{ss}_k = \sum_{i=a+1}^{a+m} h^e_i h^T_l, \quad \alpha^{ss}_k = \frac{\exp(\hat{a}^{ss}_k)}{\sum_{i=1}^{n} \exp(\hat{a}^{ss}_i)}, \quad h^{ss} = \sum_{i=1}^{n} \alpha^{ss}_i h^e_i, \]  
(5)

where \( ss \) means semantic-syntactic, \( \alpha^{ss}_k \) is the aspect-oriented attention weight of the \( k \)-th word in the semantic-syntactic channel, and \( h^{ss} \in \mathbb{R}^{2dh} \) is the final representation that represents the semantic-syntactic information.

To fully utilize the semantic and syntactic information and ease the possible side effect caused by the unreliability of the construction of dependency tree, we propose a multi-channel mechanism as below:

\[ h^{se} = ASA(H^c, H^L), \quad h^{se} = ASA(H^c, H^c), \quad h^{sy} = ASA(H^l, H^l), \]  
(6)

where \( se \) and \( sy \) mean semantic and syntactic respectively, ASA denotes the aspect-specific attention as described in Equation (5). \( h^{se} \in \mathbb{R}^{2dh} \), \( h^{se} \in \mathbb{R}^{2dh} \) and \( h^{sy} \in \mathbb{R}^{2dh} \) are the final representations of the three channels that represent the semantic-syntactic information, semantic information and syntactic information, respectively.

By this means, we obtain three types of information, i.e., the semantic information, which is only related to the semantic representation \( H^c \); the syntactic information, which is obtained only using the syntactic representation \( H^l \); and the semantic-syntactic information that is related to both \( H^c \) and \( H^l \).

2.3. Dynamic Weighted Concatenation

There is an obvious fact that different datasets have different linguistic features and different texts in the same dataset are diverse in morphology and grammar. Therefore the model may prefer the semantic or the syntactic information for the specific text. Considering this, we propose a novel dynamic weighted concatenation mechanism to model this preference by dynamically assigning weights to the three types of information mentioned in the last section.

Firstly, the three types of information are concatenated as a full representation of the input, followed by a linear transformation and sigmoid function to obtain the corresponding dynamic weights:

\[ \beta = \sigma([h^{se}; h^{se}; h^{sy}] W), \]  
(7)

where \( W \in \mathbb{R}^{6dh \times 3} \) is a trainable parameter matrix and \( \beta \in \mathbb{R}^3 \) represents the three corresponding weights, closely related to the input text.
Then the final representation $h_f \in \mathbb{R}^{2d_h}$ of the input for ABSA is obtained by combining the three types of information according to their corresponding weights:

$$h_f = \beta_1 h^{ss} + \beta_2 h^{se} + \beta_3 h^{sy},$$

(8)

where $\beta_i (i \in \{1, 2, 3\})$ represents the $i$-th element of $\beta$.

In this way, the model can dynamically assign weights to the semantic, syntactic and semantic-syntactic information according to the specific input. A proper final representation is obtained with the dynamic weights.

Finally, a linear transformation and a softmax function are used to calculate the probability distribution $p \in \mathbb{R}^{d_p}$ of the sentiment polarity as below:

$$p = \text{softmax}(h_f W_p + b_p),$$

(9)

where $d_p$ is the number of the sentiment polarity, and $W_p \in \mathbb{R}^{2d_h \times d_p}$, $b_p \in \mathbb{R}^{d_p}$ are trainable parameters.

Table 1. Details of datasets

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

2.4. Loss Function

The model is trained by Adam algorithm with the cross-entropy loss and $L_2$-regularization as follows:

$$L = - \sum_{(c,y_p) \in D} \log (p_{yp}) + \lambda \| \theta \|_2,$$

(10)

Where $D$ denotes the training dataset, $(c,y_p)$ is a training sample and the corresponding ground truth label, and $p_{yp}$ represents the $y_p$-th element of $p$. $\theta$ and $\lambda$ represent all trainable parameters and the coefficient of the $L_2$-regularization term, respectively.

3. Experiments

3.1. Datasets

The experiments are conducted on five widely used datasets, including Twitter which is originally built by [8], and other four datasets (Lap14, Rest14, Rest15, Rest16) respectively from SemEval 2014 task 4, SemEval 2015 task 12 and SemEval 2016 task 5. The detailed statistics of the datasets are listed in Table 1.

3.2. Experiment Setup

For all of our experiments, we follow [3] to use 300-dimensional pretrained GloVe vectors [6] as the initialization of word embeddings. All model parameters are initialized with uniform distribution. The dimensionality of hidden state vectors is 300. Adam is used as the optimizer with a learning rate in $[0.0010, 0.0011, ..., 0.0019]$ according to the performance of the model. The coefficient of $L_2$-regularization is $10^{-5}$ and the batch size is set to 32. The number of GCN layers is 2, which is the
same as [3]. The experimental results are obtained by averaging the performance of 3 runs with random initialization, where the performance is measured by Accuracy and Macro-Averaged F1.

3.3. Results
To confirm the effectiveness of our proposed model, we compare the proposed ASMCDC model with some baselines and state-of-the-art models. As shown in Table 2, ASMCDC achieves the best Macro-Averaged F1 scores among all the models on all five datasets except Lap14 dataset, indicating the effectiveness of ASMCDC. For Lap14 dataset, ASMCDC is slightly worse than ASGCN, with about 0.6 F1 score lower. We suspect the reason is that Lap14 dataset has an extreme preference for the syntactic information, which is hard for the dynamic weighted concatenation mechanism to learn. This conjecture may be confirmed by the big gap between the performance of ASCNN and ASGCN. Comparing ASMCDC with ASGCN, which two have the same basic model, ASMCDC surpasses 2.44 F1 score averagely on the four datasets with nearly the same number of parameters. This confirms the effectiveness of the proposed mechanisms.

| Model        | Twitter | Lap14 | Rest14 | Rest15 | Rest16 |
|--------------|---------|-------|--------|--------|--------|
|              | Acc.    | F1.   | Acc.   | F1.    | Acc.   | F1.    |
| SVM†         | 63.40   | 63.30 | 70.49  | -      | 80.16  | -      |
| LSTM†        | 69.56   | 67.70 | 69.28  | 63.09  | 78.13  | 67.47  |
| AOA†         | 72.30   | 72.20 | 72.62  | 67.52  | 79.97  | 70.42  |
| TNet-LF†     | 72.98   | 71.43 | 74.61  | 70.14  | 80.42  | 71.03  |
| ASCNN†       | 71.05   | 69.45 | 72.62  | 66.72  | 81.73  | 73.10  |
| ASGCN†       | 72.15   | 70.40 | 75.55  | 71.05  | 80.77  | 72.02  |
| ASMCDC       | **73.84**| **72.43**| 74.45  | 70.47  | **82.53**| **74.40**|
replace the original semantic and syntactic information completely. However, it is hard to guarantee the correctness of the assumption due to the unreliability of the construction of dependency tree.

5. Conclusion

We point out that the recent interaction based methods may suffer from the side effect of the unreliability of the construction of dependency tree. Therefore, we propose a multi-channel aspect-specific attention mechanism and a dynamic weighted concatenation mechanism to tackle this problem. Based on the two mechanisms, we propose a novel model ASMCDC. The results compared with some baselines and state-of-the-art models indicate the effectiveness of the proposed ASMCDC model. In the future, we will combine ensemble learning or model space [19][20] with our work.

6. References

[1] Huang B, Ou Y and Carley K 2018 Aspect level sentiment classification with attention-over-attention neural networks Social, Cultural, and Behavioral Modeling pp 197-206
[2] Li L, Liu Y and Zhou A 2018 Hierarchical attention based position-aware network for aspect-level sentiment analysis Proceedings of the 22nd Conference on Computational Natural Language Learning pp 181-9
[3] Zhang C, Li Q and Song D 2019 Aspect-based sentiment classification with aspect-specific graph convolutional networks Proceedings of the 2019 Conference on EMNLP-IJCNLP
[4] Bastings J, Titov I, Aziz W, Marcheggiani D et al. 2017 Graph convolutional encoders for syntax-aware neural machine translation Proceedings of the 2017 Conference on EMNLP
[5] Tang H, Ji D, Li C and Zhou Q 2020 Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification Proceedings of the 58th Annual Meeting of ACL
[6] Pennington J, Socher R and Manning C 2014 GloVe: global vectors for word representation Proceedings of the 2014 Conference on EMNLP pp 1532-43
[7] Wang K, Shen W, Yang Y et al. 2020 Relational graph attention network for aspect-based sentiment analysis Proceedings of the 58th Annual Meeting of ACL pp 3229-38
[8] Dong L, Wei F, Tan, C et al. 2014 Adaptive recursive neural network for target-dependent twitter sentiment classification Proceedings of the 52nd Annual Meeting of ACL pp 49-54
[9] Hu M and Liu B 2004 Mining and summarizing customer reviews Proceedings of the 10-th ACM SIGKDD pp 168-77
[10] Tang F, Peter T, Pedro A and Chen H 2015 The benefits of modelling slack variables in SVMs Neural Computation pp 954-81
[11] Yu L, Chen H, Wang S and Lai K 2009 Evolving least squares support vector machines for stock market trend mining IEEE Transactions Evolutionary Computation pp 87-102
[12] Chen H, Peter T and Yao X 2009 Probabilistic classification vector machines IEEE Transactions on Neural Networks pp 901-14
[13] Lyu S, Tian X, Li Y et al. 2019 Multi-class probabilistic classification vector machines IEEE Transactions on Neural Networks and Learning Systems vol 31 pp 3906-19
[14] Zhao X, Xiao F, Zhong H et al. 2020 Condition aware and revise transformer for question answering In Proceedings of the World Wide Web Conference (WWW'20) 2020
[15] Lyu S and Liu J 2020 Hybrid framework of convolution and recurrent neural networks for text classification Proc. of IEEE Int. Conf. on Knowledge Graph pp 313-20
[16] Lyu S, Cheng J, Wu X et al. 2020 Auxiliary learning for relation extraction IEEE Transactions on Emerging Topics in Computational Intelligence doi: 10.1109/TETCI.2020.3040444
[17] Tang D, Qin B, Feng X et al. 2016 Effective LSTMs for target-dependent sentiment classification Proceedings of COLING 2016 pp 3298-307
[18] Wang J, Yu L, Lai K et al. 2019 Investigating dynamic routing in tree-structured LSTM for sentiment analysis Proceedings of the 2019 Conference on EMNLP-IJCNLP pp 3432-37
[19] Chen H, Peter T, Ali R and Yao X 2014 Learning in the model space for cognitive fault diagnosis IEEE Transactions on Neural Networks and Learning Systems pp 124-36
[20] Gong Z, Chen H, Yuan B and Yao X 2017 Multiobjective learning in the model space for time series classification IEEE Transactions on Cybernetics