Target-Based Attention Model for Aspect-Level Sentiment Analysis

Wei Chen[1], Wenxin Yu[1](✉), Zhiqiang Zhang[1], Yunye Zhang[1], Kepeng Xu[1], Fengwei Zhang[1], Yibo Fan[2], Gang He[3], and Zhuo Yang[4]

[1] Southwest University of Science and Technology, Mianyang, China
18681268340@163.com, yuwenxin@swust.edu.cn, star_yuwenxin27@163.com
[2] State Key Laboratory of ASIC and System, Fudan University, Shanghai, China
[3] Xidian University, Xi'an, China
[4] Guangdong University of Technology, Guangzhou, China

Abstract. Aspect-level sentiment classification, which aims to determine the sentiment polarity of the specific target word or phrase of a sentence, is a crucial task in natural language processing (NLP). Previous works have proposed various attention methods to capture the important part of the context for the desired target. However, these methods have less interaction between aspects and contexts and can not accurately quantify the importance of context words with the information of aspect. To address these issues, we firstly proposed a novel target-based attention model (TBAM) for aspect-level sentiment analysis, which employs an attention mechanism between the position-aware context representation matrix. TBAM can generate more accurate attention scores between aspects and contexts at the word level in a joint way, and generate more discriminative features for classification. Experimental results show that our model achieves a state-of-the-art performance on three public datasets compared to other architectures.

Keywords: Natural language processing · Sentiment analysis · LSTM · Attention mechanism

1 Introduction

Aspect based sentiment analysis aims to determine the sentiment polarity (negative, neutral, or positive) of the specific target or aspect of a sentence. For example, given a sentence "Dreadful food but the service was good" the sentiment polarity of aspect "food" is negative while the polarity of aspect "service" is positive. There are some different sentiment polarities in the same sentence for different aspects.

In recent years, neural networks have played an increasingly important role in the field of aspect-level sentiment analysis. In these methods, aspect words are usually regarded with equal importance across the context words, the aspect information is not fully considered into the context in the neural network model.

T. Gedeon et al. (Eds.): ICONIP 2019, LNCS 11955, pp. 259–269, 2019.
https://doi.org/10.1007/978-3-030-36718-3_22

Later, some neural attention mechanisms have applied to this task, although performance has improved through these methods, according to our empirical study, there are still some common problems shared by these previous models. For example, given the sentence "The local sweet food is delicious, but the service is dreadful.", the aspect words are "local sweet food". Suppose these words become one-dimensional vectors after word embedding, and the specific value of each word is '1', '3' and '5'. If we directly average these vectors with the same importance in previous methods, so the average result is '3', which is same to the value of the word 'sweet'. In other words, the semantic information of "local sweet food" is similar to the word "sweet", obviously, it is not correct and the key word for this sentence is 'food'. They only consider unilateral information and do not consider the impact of each context word on the aspect word.

To solve the above issues, in this paper, a target-based attention model (TBAM) is induced for aspect-level sentiment analysis. TBAM can better detect the most important textual information in the given aspect of a sentence. Aspect information plays a key role in its sentence. In our datasets, the number of words containing more than one word in the aspect account for approximately 25%, 38% and 70% respectively. The previous processing methods treated the word vectors average as a word vector, and often ignore the important relationship between the aspect words, TBAM uses the attention information to assign the corresponding weights to each aspect word, which compute the attention scores between content and aspect at the word level in a joint way, and effectively quantify the representations. Moreover, TBAM can observe the position information between the context and aspect at the sentence level, which is important for capturing the key words. The model is evaluated on three datasets: Restaurant and Laptop are from SemEval 2014 [12], and the third one is Twitter dataset. The experimental results show that TBAM can effectively predict the polarity of the given aspect sentence and reach the highest level.

The remainder of this paper is structured as follows: Sect. 2 discusses the overview of related work, Sect. 3 gives a detailed description of TBAM, Sect. 4 presents extensive experiments to justify the effectiveness of TBAM, and Sect. 5 provides some conclusions and the future direction.

2 Related Work

Aspect-level sentiment analysis is designed to determine the sentiment polarity of the sentence for a given aspect or target. Traditional approaches to solve the problem are to manually design set of features and most of them focus on building sentiment classifiers with feature [10,11]. However, the results highly depend on the quality of these features and the feature engineering is labor intensive.

Recursive neural networks (RecNNs) were firstly introduced into this field by Dong et al. [1], and their proposed algorithm can adaptively propagate the sentiment of contexts to the aspect, but they often make mistakes in the face of some grammatical errors that are common in practice. Later, the recurrent neural networks (RNNs) have been demonstrated to be more effective for the tasks of sentence sequence [13,14]. Tang et al. [2] introduced the TD-LSTM approach which learns the feature representation from the leftmost and rightmost sides of the sentence. Vo and Zhang [3] used neural pooling functions to extract features from word embeddings.

Most of the neural network models often suffer from the semantic mismatching problems. Hence, attention mechanism has been successfully applied to the aspect-level sentiment analysis. Wang et al. [9] proposed the attention based LSTM with aspect embedding (ATAE-LSTM), which firstly applied attention mechanism to aspect level sentiment analysis by simply concatenating the aspect vector into the sentence hidden representations and achieving a good performance. Tang et al. [5] developed a deep memory network based on a multi-hop attention mechanism (Mem-Net), which introduced the position information into the hidden layer. Ma et al. [4] proposed an interactive attention mechanism (IAN), which interactively learns attentions from the aspect and context, their approach is similar ours. However, the semantic information between context and aspect and the relation between aspects in the same sentence are not well exploited, they only consider unilateral information and do not consider the impact of each context word on the aspect word. For example, given the sentence "The sweet food is delicious, but the service is dreadful.", the aspect information is "sweet food", it's easy to know the word "good" is more important to the aspect information than the word "dreadful" according to the position information. In addition, based on our Linguistic habit, we know the word good" is more to describe the word "food" than the word "sweet", however, in the previous work, they only qualitatively described the importance and did not materialize it.

Compared with the above models, TBAM captures the attention scores to assign the corresponding weights for each aspect word and can also observe the location information between the aspect and context, which is the first work to explore the aspect-level interactions.

3 Model Overview

Our target-based attention modeling framework consists of four components: Input Embedding Layer, Contextual Layer, Target-Based Attention Layer and Output Layer. As the Fig. 1 has shown, TBAM takes a sentence $w = [w_1, w_2, ..., w_n]$ and an aspect $t = [t_1, t_2, ..., t_n]$ as input, and the goal of its process is to predict the sentiment polarity of the sentence over the aspect.

3.1 Input Embedding Layer

The input embeddings layer contains two components: context word embeddings and aspect word embeddings. Given a sentence $w = [w_1, w_2, ...w_i..., w_n]$ and an aspect $t = [t_i, t_{i+1}, ..., t_{i+m-1}]$ where n is the sentence length and m is the aspect length. They would be mapped from two one-hot matrixes

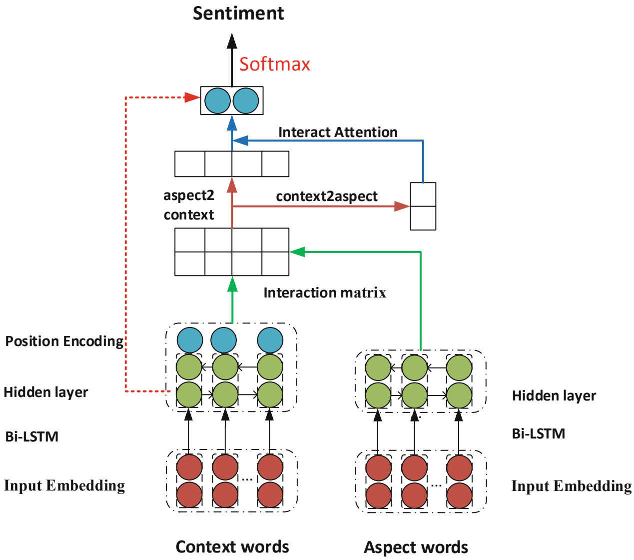

Fig. 1. The framework of target-based attention model.

into two matrixes M^{V*d} make up with vector $v_1; v_2; ...; v_n] \in R^{n*d}$ and vector $[v_i; v_{i+1}; ...; v_{i+m-1}] \in R^{m*d}$ from both context and aspect where d is the dimension of word embedding and V is the vocabulary size.

3.2 Contextual Layer

The contextual layer with Bi-LSTM architecture is used for learning the more abstract representation of sentence and aspect, which can obtain richer semantic information from both ends of the sentence and avoid vanishing-gradient and over-fitting in the same time.

The output $h = [\overrightarrow{h}, \overleftarrow{h}]$ of Bi-LSTM encoder is the concatenation of forward hidden state $h = [\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_n}]$ and backward hidden state $h = [\overleftarrow{h_1}, \overleftarrow{h_2}, ..., \overleftarrow{h_n}]$ where n is the length of the sentence.

In addition, considering that the context words with closer distance to an aspect may have higher influence to the aspect, and the hidden layer neurons have the same weight in one sentence, which is not flexible enough for predicting respective sentiments of these aspects. We utilize the position encoding mechanism to simulate the observation. Based upon this understanding, the position encoding is defined as follows:

$$dis(i) = \begin{cases} i - m_0, & i < m_0 \\ i - m_0 - m, & n \geq i > m_0 + m \\ 0, & m_0 + m \geq i \geq m_0 \end{cases} \tag{1}$$

$$l_i = 1 - \frac{|dis(i)|}{n} \tag{2}$$

where i is the current word index and m_0 is the first word index of the aspect, $dis(i)$ indicates the relative distance of the i-th word. m is the length of the aspect and n is the length of the context. The l_i is used to as the weight of the hidden layer neuron to measure the relative position between each word and the first aspect word. As a result, there are new hidden layer neurons $\overline{v} = [v_1, v_2, ..., v_i, ..., v_n] \in R^{2d}$.

$$v_i = l_i \cdot \overline{v_i} \tag{3}$$

where the $\overline{v_i}$ is the i-th original hidden layer neuron.

3.3 Target-Based Attention Layer

Attention mechanism is a common way to capture the interactions between the aspect and context words. When getting the hidden semantic representations of the context and the aspect, the dot product is used for obtain the pairwise matching matrix. Unlike previous works on introducing complex architectures or many untrainable hyperparameters into models, our operational mechanism is much simpler, but better than the advanced systems. The matching matrix as follow:

$$M(i,j) = h_s(i) \cdot h_t(j)^T \tag{4}$$

where $M(i,j)$ is the similarity between the i-th word in the sentence and the j-th word in aspect. The attention mechanism is used to automatically determine the importance of each word, rather than using simple heuristics (such as summation or averaging) to focus these words' attention on the final attention. The attention mechanism consists of two parts. Firstly, with the portrait normalization, we obtain the attention score of each aspect word to the context. Then, with the portrait normalization, we can also obtain the attention score of each context word to the aspect, which is inspired by the AOA module in question answering [15]. The attention score is calculated as:

$$\alpha_{i,j} = \frac{exp(M_{i,j})}{\sum_i exp(M_{i,j})} \tag{5}$$

$$\beta_{i,j} = \frac{exp(M_{i,j})}{\sum_i exp(M_{i,j})} \tag{6}$$

In order to consider the differences between the aspect words, and make full use of the information of the aspect term, the attention score β is used to them as weights. We interact two attention α and β scores and get the final attention weight $\overline{\alpha}$. h is the representation of the entire sentence, which obtains sentence representation r. And the interacting formula is as follows:

$$\overline{\alpha} = \varphi(\alpha_{i,j} \odot \beta_{i,j}) \tag{7}$$

$$r = h^T \cdot \overline{\alpha} \tag{8}$$

where the function represents the matrix summing by row, the range of values of i is $0 \leq j \leq n-1$ and j is $0 \leq j \leq m-1$. The $\odot$ means the matrix is multiplied by the element and $r \in R^{2d}$.

3.4 Output Layer

At last, we get the final classification feature vector r, and feed it into a linear layer, the length of whose output equals to the number of class. Finally, we add a softmax layer to compute the probability distribution for judging the sentiment polarities as positive, negative or neutral:

$$y = softmax(W \cdot r + b) \tag{9}$$

where W and b are the weight matrix and bias respectively, y is the estimated probability. TBAM is trained to use end-to-end back propagation and minimize the cross-entropy loss with L_2 regularization. p is the distribution for sentence, $\widehat{p}$ is the predicted sentiment distribution, the loss objective is defined as follows:

$$loss = -\sum_{i}\sum_{j} p_i^j \log \widehat{p}_i^j + \lambda \parallel \theta \parallel^2 \tag{10}$$

where i stands for the index of the sentence, j stands for the category index, and there are three categories, λ indicates L_2 regularization parameter, θ is the parameter of setting weights in Bi-LSTM networks and liner classifier.

4 Experiment

4.1 Dataset and Settings

Our algorithm is evaluated on SemEval 2014 dataset (PontiKi et al. 2014), which includes the review data for Restaurant and Laptop and the other one is Twitter. They were widely used in previous work. The first two reviews are labeled with four sentiment polarities: positive, neutral, negative and conflict, we remove conflict category as the number of conflict samples is very small and make the dataset extremely unbalanced. Table 1 shows the training and test sample numbers in each sentiment polarity. Taking into account the differences between aspect words, the number of aspect words of each dataset is listed in Table 2.

Table 1. Details of the experimental datasets.

Dataset	Positive	Neutral	Negative
Restaurant-train	2164	637	807
Restaurant-test	728	196	196
Laptop-train	994	464	870
Laptop-test	341	169	128
Twitter-train	1561	3217	1560
Twitter-test	173	346	173

Table 2. Statistics on the number of aspect words.

Dataset	Num(word) = 1	Num(word) > 1
Restaurant	0.7447	0.2553
Laptop	0.6110	0.3840
Twitter	0.2999	0.7001

The pre-trained 300-dimensional Glove word vectors are initialized for our experiments. The dimensions of hidden layer are set to 300. Model training through 128 samples in each batch, Set the learning rate to 0.001, dropout rate to 0.5 and L_2 regularization to 0.001. The model runs on the ubuntu16.04 system, operating environment are pytorch and NVIDIA GTX 1080ti.

4.2 Results and Discussions

To verify the validity of our model, we compare it to several baseline methods. In the experiments, the classification accuracy and macro-F1 score are used as the evaluation metrics. The performances of these baselines are cited from their original papers (Table 3).

Table 3. Classification results of different methods on three datasets. The result of '*' are retrieved from Li et al. [18], '–' means this result is not available, the '[]' stand for their original papers index.

Method	Laptop (%)		Restaurant (%)		Twitter (%)	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
Feature-SVM [15]	70.49	–	80.16	–	63.4	63.3
TD-LSTM [4]	68.13	68.43	75.63	66.73	70.8	69
ATAE-LSTM [2]	68.7	62.45	77.2	64.95	–	–
IAN [13]	72.1	67.48	78.6	67.9	–	–
MemNet [7]	72.37	64.09	80.32	65.83	68.5	66.91
RAM [16]	74.49	71.35	80.23	70.8	69.36	67.3
TBAM	**74.66**	**71.39**	**80.99**	**71.59**	**73.12**	**71.24**

The performance fluctuates with different random initialization, which is a common issue in training neural networks, so the model is run 5 times, and report the best average performance. We can observe that TBAM achieves the best performance among all these methods, it learns the attention weights between content and aspect at the word level in a joint way, which is helpful for aspect sentiment analysis. This indicates that exploiting the clues of target and position effectively can improve the performances.

In addition, The TD-LSTM model, which gets the worst performance, the main reason is the semantics of the divided sentences are already incomplete. Further, the LSTM based model ATAE-LSTM and IAN perform better than TD-LSTM on three datasets. One main reason maybe the introduction of an attention mechanism that can make the models notice the important parts of the sentence for a given aspect. This result also shows that introducing aspect clues only by splitting the sentence according to position of aspect is not enough.

4.3 Effects of Position Encoding

Obviously, the closer the word is to aspect, the higher weight the word would be assigned. In this experiment, we use absolute values to measure the distance between them, which is most effective. In order to verify the validity of the position encoding, we conducted two sets of comparative experiments, which were tested on two public datasets.

Table 4. Effects of position encoding on three datasets

Models	Laptop (%)		Twitter (%)	
	Acc	Macro-F1	Acc	Macro-F1
No-Position	72.57	68.19	72.39	70.86
TBAM	**74.66**	**71.39**	**73.12**	**71.24**

In Table 4, we report the performance of the two models. The No-Position represents the model that removes location information in the TBAM. After inducing the position embeddings, the performance has an increase of about 3% and 1% on two datasets, which indicates that exploiting the position clues effectively can improve the performance of TBAM in this task.

4.4 Effects of Attention Interaction

In TBAM, with the portrait normalization, we obtain the attention score β of each context word to the aspect, which is used to the aspect words as weights. Then we interact two attention α and β scores and get the final attention. In order to verify the validity of the method of attention interaction, we test the following three sets of experiments and the results are shown in Table 5.

No-Inter-mean refers to directly calculate attention α by average, in other words, it only calculates the attention weights of the aspect words for the context words. Similarly, No-Inter-max refers to calculate attention by maximum pooling. Inter-mean refers to calculate two attentions and interact them, but removing the influence of location information. We can be observed that Inter-mean performs mostly better than other models, which verifies that interacting two attentions weights according to context words for the aspect is ineffective

Table 5. Effects of information interaction on three datasets.

Models	Laptop (%)		Restaurant (%)		Twitter (%)	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
No-Inter-mean	71.92	67.13	80.61	71.27	71.38	69.12
No-Inter-max	71.64	**69.27**	**80.64**	71.32	71.56	69.94
Inter-mean	**72.57**	68.19	80.44	**72.11**	**72.39**	**70.86**

in this task. We also found that the accuracy on Restaurant is lower than No-Inter-max. On the one hand, because the proportion of more than two words in aspect is relatively small, on the other hand, it is mostly about food in Restaurant, which is relatively easy to find the main semantic information of the aspect information.

4.5 Case Study

In order to better understand TBAM, the sample of the restaurant dataset is extracted and visualized the weight of the context words in the sentence in Fig. 2. The aspect of context a is "food" while the aspect of context b is "service". The color depth represents the level of weight $\overline{\alpha}$. The sentence is "**good food but the service was dreadful!**", when the current aspect term is food, obviously, its neighboring words such as "good" should play a great role for judging sentiment polarity of **food**. For aspect term **service**, it is obvious that the word "dreadful" is more important to express the aspect term than the word "**service**. In addition, given the sentence "The sweet food is good, but the service is dreadful.", based on our Linguistic rules, we know the word "good" is more to describe the word food" than the word "sweet", TBAM captures the attention information to assign the corresponding weights for each aspect word and help it find out the sentiment.

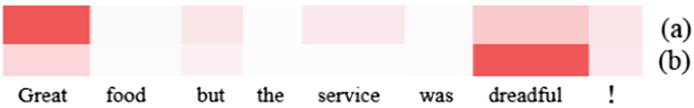

Fig. 2. Visualization of specific context aspect.

5 Conclusion

In this paper, we proposed the target-based attention model (TBAM) for aspect-level sentiment analysis. The main proposal is to compute the attention scores between contents and aspects at the word level in a joint way, and effectively quantify the representations. Moreover, the position encoding is induced into the model which makes TBAM more robust to against irrelevant information and achieves the best results on SemEval 2014 Datasets and Twitter.

Before the popularity of deep learning, the predecessors summarized a lot of knowledge bases, which could be a huge treasure in the field of natural language processing. Now many scholars are trying to use these knowledge bases. In the future, we would like to explore how to fuse linguistic rules into the neural network models.

Acknowledgement. This research was supported by 2018GZ0517, 2019YFS0146, 2019YFS0155, which supported by Sichuan Provincial Science and Technology Department, 2018KF003 Supported by State Key Laboratory of ASIC & System. No. 61907009 Supported by National Natural Science Foundation of China, No. 2018A030313802 Supported by Natural Science Foundation of Guangdong Province, No. 2017B010110007 and 2017B010110015 Supported by Science and Technology Planning Project of Guangdong Province.

References

1. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent Twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers). ACL, Baltimore (2014)
2. Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification. In: Proceedings of COLING (2016)
3. Vo, D.-T., Zhang, Y.: Target-dependent Twitter sentiment classificaiton with rich automatic features. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, pp. 1347–1353 (2015)
4. Ma, D., Li, S., Zhang, X., Wang, H.: Interactive attention networks for aspect-level sentiment classification. In: Proceedings of 26th IJCAI, Melbourne, Australia, pp. 4068–4074 (2017)
5. Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. In: Proceedings of EMNLP (2016)
6. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent Twitter sentiment classification. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers). ACL, Baltimore (2014)
7. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: Detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation, pp. 437–442 (2014)
8. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 452–461 (2017)
9. Wang, Y., Huang, M., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: EMNLP, pp. 606–615 (2016)
10. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent twitter sentiment classification. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 151–160 (2011)

11. Wagner, J., et al.: DCU: aspect-based polarity classification for semeva task 4 (2014)
12. Pontiki, M., et al.: Semeval 2016 task 5: aspect based sentiment analysis. In: ProWorkshop on Semantic Evaluation (SemEval-2016), pp. 19–30 (2016)
13. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, vol. 333, pp. 2267–2273 (2015)
14. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015)
15. Cui, Y., Chen, Z., Wei, S., Wang, S., Liu, T., Hu, G.: Attention-over-attention neural networks for reading comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 593–602 (2017)