# Context Iterative Learning for Aspect-Level Sentiment Classification

Wenting Yu(✉), Xiaoye Wang, Peng Yang, Yingyuan Xiao, and Jinsong Wang

Tianjin University of Technology, Tianjin, China
ywthelium@163.com, jiaoliu456@163.com, 29139475@qq.com,
{yyxiao,jswang}@tjut.edu.cn

**Abstract.** Aspect-based sentiment analysis is to predict the sentiment polarity of different aspects of a sentence. Many irrelevant words are mistaken for opinion words in long sentences. According to extensive research, irrelevant words are far removed from the central words. This paper proposes a solution: First, we design the Context Iterative Learning network (CILN). Context attention module (CAM) is proposed, which employs Context Features Dynamic Mask (CDM) to cover words far from the center word and Context Features Dynamic Weighted (CDW) to reduce the weight of words far away. The calculation of CAM is done alternately to reduce the influence of distant irrelevant words. Finally, the obtained feature sequences are linked with the global sentence sequence. The Accuracy and Macro-F1 indicators obtained from the experiments based on benchmark datasets demonstrate the efficacy of the proposed method.

**Keywords:** Aspect-based sentiment analysis · Feature extraction · Distribution reduction

## 1 Introduction

Aspect-based sentiment analysis (ABSA) is a text classification task, which divides the sentiment polarity of content into positive, neutral and negative [2].

The attention mechanism is now a crucial model in solving sentiment analysis tasks [9]. However, the attention mechanism does not always accurately predict aspect polarity [1]. Attention mechanisms can neither capture position information between words nor learn the relationship between sequence information and words in a sentence. Existing ABSA models mainly enhance aspect representation learning, such as MetNet [6]. MetNet may learn disturbing information together. This paper proposes a CAM module built on the CDM/CDW block and multi-head attention. A more accurate aspect-context feature representation is extracted through multiple iterations of CAM by reducing the influence of irrelevant words on sentiment prediction.

We propose the Context Iterative Learning Network (CILN). It is inspired by MemNet [8], AEN-Bert [7] and LCF-Bert [11]. First, we enter the context


C. Strauss et al. (Eds.): DEXA 2022, LNCS 13426, pp. 196–202, 2022.
https://doi.org/10.1007/978-3-031-12423-5_15

sequences and aspect terms sequences so that both sequences can traverse multiple CAMs at the same time. The CDM and CDW modules are used alternately in each CAM. The CDM module is used in the first CAM module. The CDW module is used in the following CAM module. After the multi-layer CAM, the new aspect-context sequence merge with the global sequence.

The main contributions are as follows: We design the CILN, which extracts contextual features by iteration and enhances the attention of aspect terms. The CAM module is designed, and the CDM and CDW modules are used in combination with the multi-head attention mechanism to reduce the influence of irrelevant words on aspect prediction.

## 2 Related Work

Deep learning is primarily used for sentiment analysis now. AOA [5] is an attention-over-attention neural network for aspect-oriented sentiment classification. LCF-BERT [11] is an aspect-based sentiment classification mechanism based on Multi-head Self-Attention (MHSA)-local context focus (LCF). Zhang [12] uses a graph convolutional network to extract sentence features, and uses graph convolution to investigate the influence of the dependency tree. MET-Net [6] designs a hierarchical structure that iteratively enhances the representation of aspects and contexts.

## 3 Context Attention Modules (CAM)

CAM is illustrated in Fig. 1. The input sequence is divided into two data streams. In the first data stream, context sequences are passed through Intra-multi-headed attention mechanism (Intra-MHA), position-wise feed-forward networks (PFFN) and CDM in turn. In the second data stream, context sequences and aspect terms sequences are passed through Inter-multi-headed attention mechanism (Inter-MHA) and PFFN in turn. And $\oplus$ denotes the multi-headed attention mechanism(MHA) that connects two data streams information. CAM is performed iteratively, and the CDM/CDW in each CAM is performed alternately. The purpose of alternate execution is to avoid extracting a single context feature.

### 3.1 Intra-Multi-Headed Attention Mechanism (Intra-MHA) and Inter-Multi-Headed Attention Mechanism (Inter-MHA)

Inter-MHA [7] is a multi-headed attention calculation that takes into account context and aspect terms. The context sequences and the aspect sequences are learned together to solve the long dependency problem. The formula for Inter-MHA is as follows: $h_j = \text{Multihead-Attention}\left(v_{c_i}, v_{a_j}\right)$ [7]. Where $v_{a_j}$ is the aspect sequence vector, $v_{c_i}$ is the context sequence vector.

Intra-MHA [7] learns important features from different heads and can selectively emphasize the sentence's relatively important features. Intra-MHA is expressed as: $h_i = \text{Multihead-Attention}\left(v_{c_i}, v_{c_i}\right)$ [7].

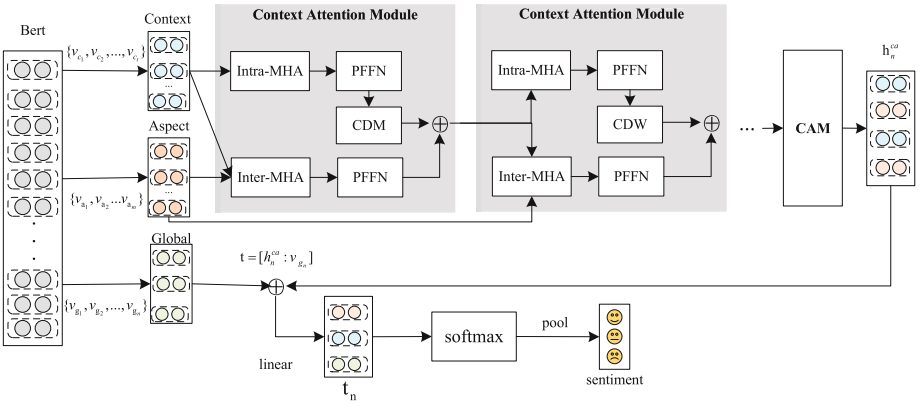


**Fig. 1.** Structure of Context Iterative Learning Network (CILN)

### 3.2 Context Features Dynamic Mask/Context Features Dynamic Weighted (CDM/CDW)

In CAM, the semantic relative distance(SRD) determines the CDM's concealment range and the range of dynamic weight reduction. The SRD is the word distance between the context word tokens and the specific aspect terms.

CDM [11] uses the specific aspect terms as the center and the SRD as the radius to calculate the next attention mechanism for words that are within the SRD distance, and irrelevant words that are masked. The input local context matrix is $V^l$. CDM based on certain SRD threshold $\alpha$ is expressed as: $V_i^m = \begin{cases} O, SRD > \alpha \\ E, SRD \leq \alpha \end{cases}$. Where O represents zero vector, and E represents one vector. $m$ represents CDM. The mask matrix is multiplied with the local context matrix output in the last step: $V^M = [V_1^m, V_2^m, \ldots, V_i^m] \cdot V^l$.

CDW [11] takes the aspect terms as the center and SRD as the radius to reduce the weight of words outside the SRD distance. The input context matrix is $V^l$. CDW based on certain SRD threshold $\alpha$ is expressed as: $V_i^w = \begin{cases} E - \frac{SRD_i - \alpha}{N} \cdot E, SRD > \alpha \\ E, SRD \leq \alpha \end{cases}$, $V^W = [V_1^w, V_2^w, \ldots, V_i^w] \cdot V^l$. Where $SRD_i$ the i-th SRD distance, $N$ is the length of the sentence. $w$ represents CDM.

### 3.3 Position-Wise Feed-Forward Networks (PFFN) and Aspect-Context Representation Output

PFFN transforms the information from the previous step and provides rich feature representations. PFFN is made up of two layers of Feed Forward Neural Networks (FFNNs). The input of PFFN is expressed as $s_c$. PFFN can be expressed as: $PFFN_c = \text{Relu}\left(W_{c1} \times s_c + b_{c1}\right) W_{c2} + b_{c2}$. Where $W_{c1}$ and $W_{c2}$ are trainable weights of two FFNNs. $b_{c1}$ and $b_{c2}$ are learnable biases of two FFNNs.

$V_i^{W/M}$ is the context vector processed by the CDM/CDW, and $P_c$ is the aspect terms vector after the PFFN. The specific aspect-context is expressed as: $h_i^{ca} =$ Multihead-Attention $\left(V_i^{W/M}, P_c\right)$. The MHA here also has its own independent parameters. We input the obtained $h_i^{ca}$ into the next CAM.

## 4 Context Iterative Learning Network (CILN)

We send the comment sentence into Bert to convert the words into vectors (The context sequence is $V_c = \{v_{c_l}, v_{c_2}, \ldots, v_{c_t}\}$. The aspect term sequence is $V_a = \{v_{a_1}, v_{a_2}, \ldots, v_{c_m}\}$. The global sequence is $V_g = \{v_{g_1}, v_{g_2}, v_{g_3}, \ldots, v_{g_n}\}$.) in Fig. 1. Then the converted context sequences and aspect terms sequences are fed into the CAM. After several iterations the aspect-context sequence will be obtained. The representation is expressed jointly with the global sequences ($\oplus$ indicates a connection operation), and finally the resulting final representation is classified into sentiment polarities. 3 represents three kinds of sentiment polorities.

### 4.1 Pooling Layer and Training

We connect CAM's output with the global sequences as $t = [h_n^{ca}, V_g]$. Where $h_n^{ca}$ is the aspect-context representation after several CAM Iterations. Finally, we input the final representation into the softmax layer for sentiment classification. The softmax classification can be expressed as: $Y = \text{softmax}(t) = \frac{\exp f(t)}{\sum_{x=1}^{3} \exp f(t)}$, $f(t) = W_s \times t + b_s$. Where $W_s$ and $b_s$ are learnable weights and biases.

The objective optimization function of this paper is the cross-entropy loss with $L_2$ regularization, and the function is defined as: $\mathrm{L}(\theta) = -\sum_{i=1}^{3} \widehat{y_x} \log y_x + \lambda \sum_{\theta \in \Theta} \theta^2$. Where $y_x$ is the one-hot vector. $\lambda$ is the parameter of $L_2$ regularization, and $\theta$ is the parameter set of the model in this paper.

## 5 Experiment

### 5.1 Datasets and Experimental Settings

To better evaluate the model in this paper. We use three benchmark datasets: SemEval2014 Task 4 (14Rest and 14Lap) and ACL Twitter dataset (Twitter) [4]. The datasets have been adopted by the models proposed by the majority of researchers and are the most frequently used datasets in ABSA.

Most of the hyperparameters follow the common hyperparameter settings for sentiment analysis tasks. The learning rate is set to $2 \times 10^{-5}$, and the hidden dimensions and the embedding dimensions are set to 768. The dropout rate is set to 0.1, the $L_2$ regularization is set to $1 \times 10^{-5}$, and the batch size is set to 16. A total of 12 epochs were trained. The performance of the model is evaluated by using accuracy and macro F1 indicators.

**Table 1.** Experimental results (%). This article uses "–" to indicate unrecorded experimental results. All experimental results are the results of rerunning on our equipment.

| Model | Laptop | | Restaurant | | Twitter | |
|---|---|---|---|---|---|---|
| | ACC | F1 | ACC | F1 | ACC | F1 |
| MemNet | 67.08 | 59.12 | 78.04 | 65.63 | 70.24 | 67.78 |
| RAM | 66.73 | 57.43 | 75.18 | 57.48 | 67.34 | 63.76 |
| AOA | 63.17 | 49.43 | 73.12 | 53.17 | 65.61 | 61.47 |
| Aen-Bert | 78.06 | 74.93 | 80.45 | 69.35 | 72.54 | 71.05 |
| LCF-Bert | 79.00 | 74.60 | 83.93 | 74.68 | 73.55 | 72.65 |
| MCRF-SA | 75.43 | 71.78 | 80.71 | 70.28 | – | – |
| MetNet | 76.18 | 71.83 | 79.11 | 67.84 | 66.76 | 63.52 |
| BiGCN | 74.92 | 71.76 | 79.37 | 68.56 | 73.55 | 71.79 |
| Our | **79.78** | **76.44** | **84.91** | **78.87** | **75.43** | **74.14** |
| −w/o CAM | 78.68 | 73.82 | 83.48 | 74.49 | 72.69 | 71.48 |
| +1 CAM | 79.78 | 75.01 | 84.29 | 77.36 | 72.11 | 70.01 |
| +2 CAM | **79.78** | **76.44** | **84.91** | **78.87** | **75.43** | **74.14** |
| +3 CAM | 78.68 | 74.94 | 84.20 | 77.55 | 74.13 | 73.34 |
| +4 CAM | 78.53 | 75.31 | 83.93 | 76.44 | 72.83 | 72.17 |
| +5 CAM | 78.53 | 74.50 | 84.02 | 75.84 | 72.98 | 72.10 |

### 5.2 Baseline and Result

To comprehensively evaluate our method, this paper compares the proposed method with the model baselines: MemNet (2016) [8], RAM (2017) [3], AOA (2018) [5], Aen-Bert (2019) [7], LCF-BERT (2019) [11], MCRF-SA (2020) [10], MetNet (2020) [6], BiGCN (2020) [13].

The results that our model with 2 layers of CAM outperforms all baselines in Table 1. Twitter's performance is not as good as that of other datasets. Because Twitter has irregular grammatical expressions and many misspellings, which leads to poor performance on Twitter compared to the other two datasets. The accuracy of our model is 12.70% higher than MemNet on the laptop dataset, 6.87% on the Restaurant, and 5.17% on the Twitter. LCF-BERT is the second best performing. The accuracy of our model on the Laptop, Restaurant, and Twitter increased by 0.78%, 0.98%, and 1.88%. We believe that our model outperforms LCF because LCF only uses CDW/CDM once. Whereas we iteratively use CAM and alternate CDM/CDW for each CAM, enriching context feature extraction and resulting in higher ACC and F1 scores.

To explore the application effect of CAM in this model, ablation experiments are carried out on the basis of the best CAM superimposing two layers, including CAM resection. "−w/o" stands for delete a module. The experimental results clearly show that CAM ablation will affect the performance, which shows CAM is helpful to improve the ABSA.

As shown in Table 1, different numbers (from +1 to +5) of CAM layers are tried. The results of 2 layers are the best. First of all, the effect of CAM increases as the number of layers increases. When the number of layers is increased after the model effect has been brought to the best number of layers, the effect gradually decreases and unstable results appear. The model proposed in this paper only models the context feature layer directly related to the specific word in each CAM. Thereby increasing the number of CAM layers can improve ABSA performance. Adding more layers, model is overfitting and the result decreases.

## 6 Conclusion

We propose the CILN to improve the impact of irrelevant words on ABSA. To obtain a better representation, we employ a hierarchical structure CAM to iteratively learn aspects and contexts. The results demonstrate that CILN is useful for ABSA.

**Acknowledgements.** This work is supported by "Tianjin Project + Team" Key Training Project under Grant No. XC202022.

## References

1. Bahdanau, D., Cho, K.H., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of ICLR (2015)
2. Barbieri, F., Camacho-Collados, J., Anke, L.E., Neves, L.: TweetEval: unified benchmark and comparative evaluation for tweet classification. In: Proceedings of EMNLP Findings (2020)
3. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: Proceedings of EMNLP, pp. 452–461 (2017)
4. 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 ACL, pp. 49–54 (2014)
5. Huang, B., Ou, Y., Carley, K.M.: Aspect level sentiment classification with attention-over-attention neural networks. In: Thomson, R., Dancy, C., Hyder, A., Bisgin, H. (eds.) SBP-BRiMS 2018. LNCS, vol. 10899, pp. 197–206. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93372-6_22
6. Jiang, B., Hou, J., Zhou, W., Yang, C., Wang, S., Pang, L.: METNet: a mutual enhanced transformation network for aspect-based sentiment analysis. In: Proceedings of COLING, pp. 162–172 (2020)
7. Song, Y., Wang, J., Jiang, T., Liu, Z., Rao, Y.: Targeted sentiment classification with attentional encoder network. In: Tetko, I.V., Kůrková, V., Karpov, P., Theis, F. (eds.) ICANN 2019. LNCS, vol. 11730, pp. 93–103. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30490-4_9
8. Tang, D., Qin, B., Liu, T.: Aspect level sentiment classification with deep memory network. In: Proceedings of EMNLP, pp. 214–224 (2016)
9. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NeurIPS (2017)
10. Xu, L., Bing, L., Lu, W., Huang, F.: Aspect sentiment classification with aspect-specific opinion spans. In: Proceedings of EMNLP, pp. 3561–3567 (2020)

11. Zeng, B., Yang, H., Xu, R., Zhou, W., Han, X.: LCF: a local context focus mechanism for aspect-based sentiment classification. Appl. Sci. **9**, 3389 (2019)
12. Zhang, C., Li, Q., Song, D.: Aspect-based sentiment classification with aspect-specific graph convolutional networks. In: EMNLP/IJCNLP, no. 1 (2019)
13. Zhang, M., Qian, T.: Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In: Proceedings of EMNLP, pp. 3540–3549 (2020)