Improving Aspect Sentiment Quad Prediction by Relational Mask Multi-Head Attention and Template-Order Grouping

Anonymous EMNLP submission

Abstract

Aspect sentiment quad prediction (ASQP) has become a popular task in the field of aspect-based sentiment analysis, which aims to predict four sentiment elements: *aspect category, aspect term, opinion term, sentiment polarity*. Although its great success, existing methods still have shortcomings. First, the sentiment element is only related to the specific words in the input sentence. The existing works predict quads based on the whole input, which adds redundant information. Second, recent methods convert quad prediction into a generative task through a pre-defined templates. Constructing different template orders can improve the performance of the model. However, most methods simply utilize pre-trained language models to select template order groupings without deeply analyzing the relationships between template orders. In this paper, we propose a relational mask multi-head attention and template-order grouping method, which not only reduces the redundant information in the input but also select appropriate template order groupings. Specifically, we construct a trainable relation mask matrix and fuse it into the multi-head attention of the T5 decoder. Then we introduce relation constraint loss to reduce redundant information in the input. In addition, we quantify the effect of one template order's gradient on another template order's loss to determine the template order groupings. Experiments on multiple public datasets demonstrate that our method outperforms state-of-the-art methods.

1 Introduction

ASQP task has received widespread attention in the field of aspect-based sentiment analysis (ABSA). It focuses on extracting four elements of aspect-level sentiment, including (1) aspect category (*ac*) defines the type of the concerned aspect; (2) aspect term (*at*) is the opinion target which is explicitly or implicitly in the given text; (3) opinion term (*ot*) expresses the sentiment towards the aspect; (4) sentiment polarity (*sp*) describes the orientation of the sentiment over an aspect term. If the aspect and opinion terms are implicit in the given text, they are set as *NULL*. For example, the sentence "*The view is spectacular, and the food is great.*" contains two sentiment quadruples (*location general, view, spectacular, positive*) and (*food quality, food, great, positive*).

Existing methods (Zhang et al., 2021a; Hu et al., 2022, 2023; Gou et al., 2023; Bai et al., 2024) gradually use generative methods to handle ASQP task and have achieved good performance. They convert sentiment quads into natural language sentences through pre-defined templates and then train the model using the sequence-to-sequence method. However, the above method still has some issues. First, the sentiment element is only related to specific words in the sentence. For example, "*food quality*" corresponds to "*food*" in the sentence. Existing methods predict quads based on the entire input, which may add redundant information and harm the performance of the model. Second, different template orders can augment quads and improve the performance of the model. Yet, previous methods simply use pre-trained language models to select template order groupings with minimal entropy (Hu et al., 2022) or jensen–shannon divergence (Bai et al., 2024) without deeply analyzing the correlations between the template orders.

In this paper, we propose a relational mask multi-head attention and template-order grouping method to address the above problems. First, we introduce a trainable relation mask matrix and integrate it into the multi-head attention module of the T5 (Raffel et al., 2020) decoder. We construct the corresponding true relation mask matrices and use relation constraint loss to reduce the redundant information of the input sentence. Second, we use different template orders to augment quads and relation mask matrices. We train all template orders together and quantify the impact of gradient updates of one order on the loss of another order to measure the correlation score between template orders. Then we find all groups containing K_g template orders and select the group with the greatest correlation score. In summary, the main contributions of our work are summarized as follows:

- We construct a trainable relation mask matrix and use relation constraint loss to reduce the redundant information in the input sentence. To the best of our knowledge, this work is the first focus on the relationship between input sentences and quads in the ASQP task.
- We propose a template-order grouping method that can select more appropriate template order groups by deeply analyzing the relationship between the orders.
- Experimental results show that our method outperforms other state-of-the-art methods on multiple public datasets.

2 Related Work

2.1 Aspect-base Sentiment Analysis

ABSA has received wide attention in recent years. Early studies focus on predicting a single sentiment element, such as aspect term extraction (Liu et al., 2015; Ma et al., 2019; Xu et al., 2019), aspect category detection (Zhou et al., 2015; Bu et al., 2021), and sentiment polarity classification for a given aspect term (Wang et al., 2016; Huang and Carley, 2018; Sun et al., 2019). Some works further consider the relationship between multiple sentiment elements, including the aspect-opinion pair extraction (Wu et al., 2020; Gao et al., 2021), aspect term-polarity co-extraction (Li et al., 2019; Luo et al., 2019; Chen and Qian, 2020), aspect sentiment triplet extraction (ASTE) (Peng et al., 2020), and ASQP (Zhang et al., 2021a). Among these, ASQP is the most complete and also the most challenging task.

2.2 Aspect Sentiment Quad Prediction

ASQP can reveal a more comprehensive and complete aspect-level sentiment structure. Generative methods have gradually become mainstream because they use the information from label semantics and are highly universal. These methods can mainly be classified as template-based (Hu et al., 2022), structure-based (Mao et al., 2022; Bao et al., 2022, 2023). This paper focuses only on template-based methods. (Hu et al., 2023) propose an uncertainty-aware unlikelihood learning, which boosts original learning and reduces mistakes. Multi-view Prompting (MVP) (Gou et al., 2023) is an element order-based prompt learning method and improves the performance of the model by aggregating multi-view results. Broad-view Soft Prompting (BvSP) (Bai et al., 2024) aggregates multiple templates with a broader view by considering the correlations between different templates. Self-Consistent Reasoning-based Aspect sentiment quadruple Prediction (SCRAP) (Kim et al., 2024) uses the reasoning of large language models to improve the accuracy and interpretability of the model. Since labeled quads are scarce, some studies augment the training samples to solve the high annotation cost problem. (Wang et al., 2023) use quads-to-text generation task to generate the texts and utilize average context inverse document frequency to evaluate the difficulty of augmented samples and balance the difficulty distribution. (Yu et al., 2023) and (Zhang et al., 2024b) use the self-training mechanism to filter out mismatched samples to improve the quality of generated samples. (Zhang et al., 2024a) propose an adaptive data augmentation method to tackle the quad-pattern imbalance and aspect-category imbalance.

The sentiment element in the quads is only associated with the specific words in the input. Most of the above methods utilize the entire input to predict the quads, which adds redundant information. Our approach constructs multiple trainable relation mask matrices and uses relation constraint loss to make the sentiment element focus on related words. Furthermore, we deeply analyze the relationship between template orders to find more suitable template order groupings.

3 Approach

3.1 Task Definition

Given an input sentence $I = \{w_1, w_2, ..., w_N\}$ containing N words, ASQP aims to predict all quads (*at, ot, ac, sp*). In order to better predict implicit aspect terms and opinion terms, we add special markers to the input sentence: "[IA] [IO] I". Following the previous template-based method (Hu et al., 2022), we use special markers to convert the quads into a target sequence: "[AT] *at* [OT] *ot* [AC] *ac* [SP] *sp*". If a sentence contains multiple quads, the target sequences are concatenated with a special marker [SSEP] to obtain the final target sequence.

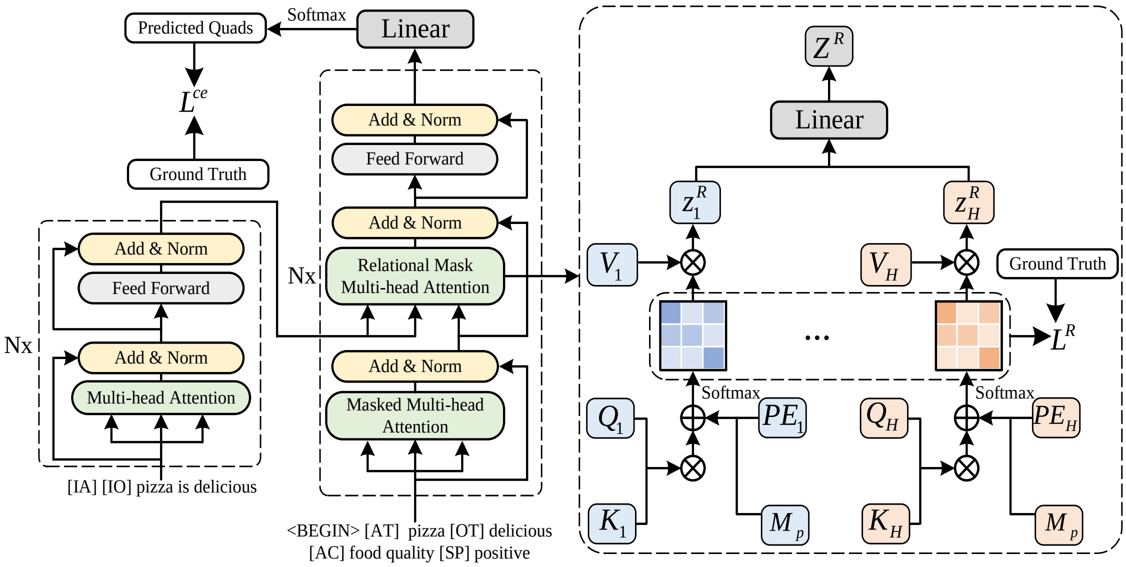


Figure 1: The architecture of relational mask multi-head attention.

3.2 Relational Mask Multi-Head Attention

Existing template-based methods predict quads based on the entire input. The sentiment elements in the quads are only related to specific words in a sentence. In this paper, we propose a relational mask multi-head attention that incorporates a trainable relation mask matrix into the multi-head attention of the T5 decoder in Figure 1. Formally, $X \in \mathbb{R}^{N \times d}$ denotes the feature representations of I and is projected through three matrices $W_Q \in \mathbb{R}^{d \times d_q}$, $W_K \in \mathbb{R}^{d \times d_q}$ and $W_V \in \mathbb{R}^{d \times d_q}$ to obtain Q, K, and V. d and d_q are dimensions. $PE \in \mathbb{R}^{N \times N}$ is the position embedding. The multi-head attention of the T5 model is computed as follows:

$$
\begin{aligned}
Q_h &= XW_Q^h \\
K_h &= XW_K^h \\
V_h &= XW_V^h \\
z_h &= softmax(Q_h K_h^T + PE_h)V_h \\
Z &= concat(z_1, z_2, ..., z_H)W_Z
\end{aligned}
\tag{1}
$$

where z_h is the h-th head. H is the number of heads. $W_Z \in \mathbb{R}^{Hd_q \times d}$ is the parameter matrices. We introduce a trainable relation mask matrix $M_p \in \mathbb{R}^{N \times N}$ and integrate it into the multi-head attention of the T5 decoder. The h-th head is computed as follows:

$$z_h^R = softmax(Q_h K_h^T + PE_h + M_p)V_h \tag{2}$$

Note that M_p is the same in each head. The relational mask multi-head attention is as follows:

$$Z^R = concat(z_1^R, z_2^R, ..., z_H^R)W_Z \tag{3}$$

3.3 Relation Constraint

We introduce a relation constraint to establish the connection between sentiment elements and corresponding words in the input, which can reduce redundant information in the input. First, we construct the real relation mask matrix. For aspect terms, we keep the corresponding aspect terms in the input and mask other words. For opinion terms, we keep the corresponding aspect terms and opinion terms in the input and mask other words. If the aspect terms or opinion terms are implicit, they are mapped with the corresponding special markers. The aspect category and sentiment polarity are consistent with the aspect terms and opinion terms, respectively. [SSEP] does not mask words in the input. For example, the sentence is "*The food is terrible and not worth going again*" and the target sequence is "<BEGIN> [AT] *food* [OT] *terrible* [AC] *food quality* [SP] *negative* [SSEP] [AT] *NULL* [OT] *not worth* [AC] *restaurant general* [SP] *negative*". The true relation mask matrix is shown in Figure 2. For a template order, we compute the true and predicted cross-attention and use the euclidean distance to compute the relation constraint loss:

$$
\begin{aligned}
A_h^p &= softmax(Q_h K_h^T + PE_h + M_p) \\
A_h^g &= softmax(Q_h K_h^T + PE_h + M_g) \\
L_h^R &= ED(A_h^p, A_h^g) \\
L^R &= \frac{1}{H}\sum_{h=1}^{H} L_h^R
\end{aligned}
\tag{4}
$$

where M_g is the true relation mask matrix.

	[IA]	[IO]	The	food	is	terrible	and	not	worth	going	again
<BEGIN>	○	○	○	○	○	○	○	○	○	○	○
[AT] food	●	●	●	○	●	●	●	●	●	●	●
[OT] terrible	●	●	●	○	●	○	●	●	●	●	●
[AC] food quality	●	●	●	○	●	●	●	●	●	●	●
[SP] negative	●	●	●	○	●	○	●	●	●	●	●
[SSEP]	○	○	○	○	○	○	○	○	○	○	○
[AT] NULL	○	●	●	●	●	●	●	●	●	●	●
[OT] not worth	○	●	●	●	●	●	●	○	○	●	●
[AC] restaurant general	○	●	●	●	●	●	●	●	●	●	●
[SP] negative	○	●	●	●	●	●	●	○	○	●	●

○ No mask
● Mask

Figure 2: The true relation mask matrix between the input sentence and the target sequence.

3.4 Template-Order Grouping

Inspired by (Hu et al., 2023), we construct all target sequences with multiple order mapping functions o_i, where $i \in [0, 23]$. Note that we only sort the sentiment quadruple. Formally, θ_s represents the shared parameters and $\{\theta_i = M_p^i | i \in [0, 23]\}$ represents the private parameters corresponding to each template order. We train the model using the sequence-to-sequence method. The encoder-decoder model converts the input sentence into the target sequence $\{y^{o_i}\}$ by o_i. The cross-entropy loss is as follows:

$$L_i^{ce} = -\sum_{t=1}^{N} \log p_{\{\theta_s, \theta_i\}}(y_t^{o_i} | I, y_{<t}^{o_i}) \tag{5}$$

Existing methods simply use pre-trained language models to select template order groupings without deeply analyzing the correlations between the template orders. In this paper, we propose a template order grouping method that can quantify the effect of one template order's gradient on another template order's loss to select the appropriate groupings. For the training batch D^t at time-step t, we define $\theta_{s_i}^{t+1}$ to represent the updated shared parameters after template order i is updated. The formula is as follows:

$$\theta_{s_i}^{t+1} = \theta_s^t - \eta \nabla_{\theta_s^t} L_i(D^t, \theta_s^t, \theta_i^t) \tag{6}$$

where η is the learning rate. $L_i(D^t, \theta_s^t, \theta_i^t)$ denotes the relation constraint loss and cross-entropy loss of template order i. For the same training batch, we can compare the loss of template order j before and after applying the gradient update of template order i. We define an asymmetric measure to evaluate the correlation score between template order i and template order j at time-step t.

$$C_{i \to j}^t = 1 - \frac{L_j(D^t, \theta_{s_i}^{t+1}, \theta_j^t)}{L_j(D^t, \theta_s^t, \theta_j^t)} \tag{7}$$

Notice that a positive value of $C_{i \to j}^t$ denotes that the update of shared parameters is beneficial to template order j, while a negative value of $C_{i \to j}^t$ denotes that the update of template order i will reduce the performance of template order j. Then we can calculate the correlation score over the whole training set.

$$\overline{C}_{i \to j} = \frac{1}{T} \sum_{t=1}^{T} C_{i \to j}^t \tag{8}$$

where T is the number of iterations. For all groups containing K_g template orders, we first calculate the correlation score of each group. For example, for the group consisting of template orders $\{1, 2, 3\}$, the correlation score is as follows:

$$\begin{aligned} \overline{C}_{1,2,3} = & \frac{\overline{C}_{2 \to 1} + \overline{C}_{3 \to 1}}{2} + \frac{\overline{C}_{1 \to 2} + \overline{C}_{3 \to 2}}{2} \\ & + \frac{\overline{C}_{1 \to 3} + \overline{C}_{2 \to 3}}{2} \end{aligned} \tag{9}$$

Then we pick the group with the highest score. Algorithm 1 describes the process of template-order grouping.

3.5 Training Strategy

We train the model by combining relation constraint loss and cross-entropy loss on the selected template-order grouping:

$$L = \frac{1}{K_g} \sum_{k=1}^{K_g} \lambda L^{R_k} + L_k^{ce} \tag{10}$$

where λ controls the impacts of relation constraint, balancing the two learning objectives.

4 Experiment

4.1 Dataset Preparation

We evaluate our method on four tasks. Rest15 and Rest16 datasets are proposed by (Zhang et al.,

Algorithm 1 Procedure of template-order grouping

Input: Training dataset D, N_t is the batch size, θ_s is the shared parameter of all template orders, $\{\theta_i | i \in [0, 23]\}$ is the private parameter of each template order, K_g and N_{K_g} are the number of selected template orders and groups, T is the total number of iterations.

Stage 1: Template order correlation calculation:

```
Let t = 0.
while t < T do
    Randomly select N_t samples D^t from D
    Let i = 0.
    while i < 24 do
        Compute the forward loss of all template
        orders {L_j(D^t, θ_s^t, θ_j^t) | j ∈ [0, 23]}
        Update the θ_s^t and θ_i^t of the i-th template
        order
        Compute the forward loss of all template
        orders {L_j(D^t, θ_{s_i}^{t+1}, θ_j^t) | j ∈ [0, 23]}
        Compute the correlation score C_i^t be-
        tween the i-th template order and all tem-
        plate orders
        i = i + 1
    end while
    Obtain the correlation score matrix C^t by
    connecting {C_j^t | j ∈ [0, 23]}
    t = t + 1
end while
Compute the final correlation score matrix C
by averaging {C^t | t ∈ [0, T]}
```

Stage 2: Template order grouping:

```
Let t = 0.
while t < N_{K_g} do
    Compute the correlation score G_t of the t-th
    group
    t = t + 1
end while
Select the group with the highest score from
{G_t | t ∈ [0, N_{K_g}]}
```

Data	Train		Test		Val	
	#S	#Q	#S	#Q	#S	#Q
Rest15	834	1354	537	795	209	347
Rest16	1264	1989	544	799	316	507
Restaurant	1530	2484	583	916	171	261
Laptop	2934	4172	816	1161	326	440

Table 1: Statistics of the experimental datasets. #S: number of sentences. #Q: number of sentiment quadruple labels.

2021a). They are based on SemEval Shared Challenges (Pontiki et al., 2015, 2016). The annotations of the opinion term and aspect category are derived from (Peng et al., 2020) and (Wan et al., 2020) respectively. Restaurant and Laptop datasets are proposed by (Cai et al., 2021). The Restaurant dataset is constructed based on the SemEval 2016 Restaurant dataset (Pontiki et al., 2016) and its expansion datasets (Fan et al., 2019; Xu et al., 2020). The Laptop dataset is collected from the Amazon platform at the years of 2017 and 2018. Table 1 summarizes the all datasets. In addition, we also conduct experiments on augmented dataset (Zhang et al., 2024b).

4.2 Implementation Details

We adopt T5-base (Raffel et al., 2020) as the pre-trained generative model. During the training, The maximum sequence length, learning rate, and batch size is 200, 1e-4, and 16, respectively. The epochs of the original dataset and augmented dataset are 20 and 10. For the hyper-parameter K_g and λ, the experimental results are in Section 4.6. During the inference, we employ a beam size of 1 and use different templates to generate results. Then we get the final quadruple on the original dataset through the voting mechanism. For the augmented dataset, we use the reranking method (Zhang et al., 2024b) to improve the prediction performance of the model. All the reported results are the average of 5 runs.

4.3 Baselines

We compare our model with the strong baselines. They include both the large language model, i.e. **ChatGPT** (Xu et al., 2023), and the following state-of-the-art methods, namely **Extract-Classify-ACOS** (Cai et al., 2021), **GAS** (Zhang et al., 2021b), **Paraphrase** (Zhang et al., 2021a), **SS, DLO, ILO** (Hu et al., 2022), **MvP** (Gou et al., 2023), **GenDA** (Wang et al., 2023), **ADA** (Zhang et al., 2024a), **ST-Scorer** (Zhang et al., 2024b), and **UGTS** (Su et al., 2025).

4.4 Experiment Results

We compare our method with other state-of-the-art methods on the four datasets and the experimental results in Table 2. SS+Ours and ST-Scorer+Ours represent the experimental results of our method on the original and augmented datasets. As can

Model	Rest15			Rest16			Restaurant			Laptop		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
ChatGPT	29.66	37.86	33.26	36.09	46.93	40.81	29.66	37.86	33.26	36.09	46.93	40.81
Extract-Classify*	35.64	37.25	36.42	38.40	50.93	43.77	38.54	52.96	44.61	45.56	29.48	35.80
GAS*	45.31	46.70	45.98	54.54	57.62	56.04	57.09	57.51	57.30	43.45	43.29	43.37
Paraphrase*	46.16	47.72	46.93	56.63	59.30	57.93	59.85	59.88	59.87	43.44	42.56	43.00
SS*	48.24	48.93	48.58	58.74	60.35	59.53	59.98	58.40	59.18	43.58	42.72	43.15
DLO*	47.08	49.33	48.18	57.92	61.80	59.79	60.02	59.84	59.93	43.40	43.80	43.60
ILO*	47.78	50.38	49.05	57.58	61.17	59.32	58.43	58.95	58.69	44.14	44.56	44.35
MvP*	-	-	51.04	-	-	60.39	-	-	61.54	-	-	43.92
GenDA*	49.74	50.29	50.01	60.08	61.70	60.88	-	-	-	-	-	-
ADA*	49.31	**53.96**	51.53	59.34	62.83	61.03	60.15	61.95	61.04	45.03	44.53	44.78
ST-Scorer*	51.94	52.00	51.97	63.46	64.31	63.88	65.43	61.92	63.63	47.05	45.32	46.17
UGTS*	52.76	52.43	52.59	65.72	64.50	65.10	65.94	63.47	64.68	48.21	**46.39**	**47.28**
SS+Ours	52.28	50.63	51.44	61.31	59.95	60.62	64.91	59.71	62.20	45.83	43.66	44.72
ST-Scorer+Ours	**54.22**	52.69	**53.44**	**66.90**	**66.23**	**66.56**	**66.72**	**63.96**	**65.31**	**48.37**	45.94	47.12

Table 2: Evaluation results compared with baseline methods. The experimental results of baseline methods, marked with *,are obtained from (Hu et al., 2023) and (Su et al., 2025).

Model	Rest15			Rest16			Restaurant			Laptop		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Original Datasets												
SS+Ours	**52.28**	**50.63**	**51.44**	**61.31**	**59.95**	**60.62**	**64.91**	**59.71**	**62.20**	**45.83**	**43.66**	**44.72**
w/o RMMA	50.84	49.61	50.22	60.72	58.97	59.83	63.67	58.51	60.98	44.36	43.04	43.69
w/o RC	50.12	47.26	48.65	59.83	56.77	58.26	61.15	57.32	59.17	43.17	42.06	42.61
w/o TOG	51.06	48.68	49.84	59.96	58.41	59.17	62.88	58.39	60.55	44.13	42.98	43.55
Augmented Datasets												
ST-Scorer+Ours	**54.22**	**52.69**	**53.44**	**66.90**	**66.23**	**66.56**	**66.72**	**63.96**	**65.31**	**48.37**	**45.94**	**47.12**
w/o RMMA	53.77	51.88	52.81	66.23	65.89	66.06	65.48	63.54	64.50	47.24	44.53	45.84
w/o RC	52.39	50.52	51.44	65.12	64.85	64.98	65.17	62.18	63.64	46.71	44.16	45.40

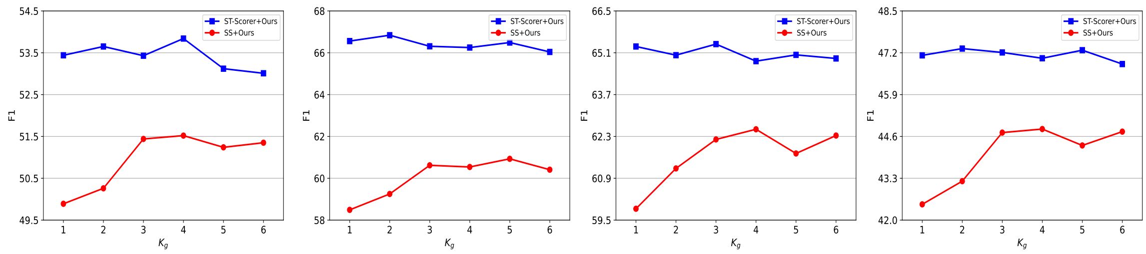
Table 3: Results of ablation on Rest15, Rest16, Restaurant, and Laptop datasets. w/o means deletion operation.

be seen, our method has achieved the best performance on most tasks.

Specifically, we have the following observations: (1) Compared to the pipeline Extract-Classify, end-to-end methods achieve better performance because they can reduce the error propagation problem. (2) Compared with ILO, SS+Ours gains absolute F1-score improvements by 2.39% (4.87% relatively), 1.30% (2.19% relatively), 3.51% (5.89% relatively), and 0.37% (0.88% relatively) in Rest15, Rest16, Restaurant and Laptop datasets, respectively. Similarly, SS+Ours also outperform MvP, DLO, and SS on all datasets. (3) On the augmented datasets, ST-Scorer+Ours outperforms ST-Scorer and UGTS on most datasets. Overall, our method reduces the redundant information in the input and selects more appropriate groups by deeply analyzing the relationship between the template orders. The experimental results verify the effectiveness of the proposed method.

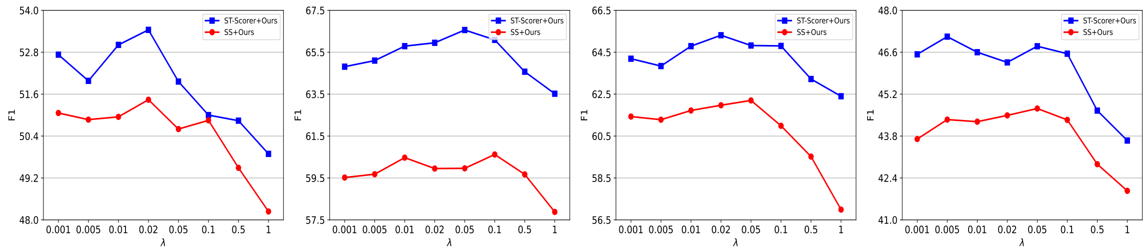
4.5 Ablation Study

To analyze the effect of relational mask multi-head attention (RMMA), relation constraint (RC), and template-order grouping (TOG), we conduct the ablation experiments in Table 3. The experimental results show that adding the trainable relation mask matrix can improve classification accuracy. When we remove the relation constraint loss, the classification accuracy of w/o RC degrades on Rest15, Rest16, Restaurant, and Laptop datasets. It shows that RC is beneficial to improve model performance. Besides, template order grouping can further improve the performance of the model on the original dataset. Although RMMA, RC, and TOG are both beneficial to improve the performance of



(a) Rest15 (b) Rest16 (c) Restaurant (d) Laptop

Figure 3: F1-score under different K_g values on Rest15, Rest16, Restaurant, and Laptop datasets.



(a) Rest15 (b) Rest16 (c) Restaurant (d) Laptop

Figure 4: F1-score under different λ values on Rest15, Rest16, Restaurant, and Laptop datasets.

the model, RC tends to play a more essential role.

K_g	**F1**	**T-Speedup**	**I-Speedup**
1	49.89	1.00x	1.00x
2	50.26	0.50x	0.56x
3	51.44	0.33x	0.35x
4	51.52	0.25x	0.26x
5	51.24	0.20x	0.21x
6	51.35	0.17x	0.18x

Table 4: The F1, training speedup, and inference speedup under different K_g values on the Rest15 dataset in the original dataset.

4.6 Hyperparameter Study

We observe the effect of two hyperparameters: K_g and λ. K_g is the number of selected template orders. λ balances relation constraint loss and cross-entropy loss.

We analyze the effect of the K_g value on the origin and augmented datasets in Figure 3. The range of K_g is 1, 2, 3, 4, 5, 6. It can be seen that increasing the K_g can improve the performance of the model on the original dataset. However, the improvement on the augmented dataset is small or even decreases. The augmented dataset has more training data, and increasing the K_g may cause overfitting. Besides, we also analyze the impact of K_g on training time and inference time in Table 4. As K_g increases, the training and inference time gradually increases. Considering the model performance, training, and inference efficiency, we choose $K_g = 3$ and $K_g = 1$ on the original and augmented datasets.

We investigate the effect of the λ value on the origin and augmented datasets in Figure 4. We vary the λ value with 0.001, 0.005, 0.01, 0.02, 0.05, 0.1, 0.5, and 1 respectively. The F1 increases first and then decreases as λ increases on most tasks. It shows that our method can improve the performance of the model through appropriate parameters.

4.7 Effect of Trainable Relation Mask Matrix

For each attention head, we construct two different ways to observe the effects of the trainable relation mask matrix in Table 5: same trainable relation mask matrix (STRMM) and different trainable relation mask matrices (DTRMM). The experimental results show that DTRMM does not achieve better performance. For example, STRMM obtains a higher F1 score on the Rest15 dataset. Finally, we use the same trainable relation mask matrix in the relational mask multi-head attention module.

4.8 Effect of Correlation Score

The template order grouping is obtained according to the correlation score matrix between different

Model	Rest15			Rest16			Restaurant			Laptop		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
						Original Datasets						
STRMM	**52.28**	**50.63**	**51.44**	**61.31**	59.95	**60.62**	**64.91**	59.71	62.20	**45.83**	43.66	44.72
DTRMM	52.09	49.82	50.93	60.24	**60.57**	60.40	64.76	**60.59**	**62.61**	45.55	**44.26**	**44.90**
						Augmented Datasets						
STRMM	54.22	**52.69**	**53.44**	66.90	66.23	66.56	**66.72**	**63.96**	**65.31**	48.37	45.94	47.12
DTRMM	**54.24**	52.56	53.39	**67.01**	**66.45**	**66.73**	66.07	63.51	64.76	**48.49**	**46.53**	**47.49**

Table 5: Effect of trainable relation mask matrix on Rest15, Rest16, Restaurant, and Laptop datasets.

templates, so how calculating the correlation scores between different templates is very important. In the consecutive steps of model training, the correlation scores between different templates are likely to be similar. We set eight correlation score calculation methods and analyze the performance of the model on the Rest15 dataset. The experimental results demonstrate that the 10-steps is 6.16x faster while achieving more than 99.54% the performance of the 1-step. In addition, First 50%, Middle 50%, and Final 50% will reduce the performance of the model. This result suggests the correlation scores between different templates are constantly changing during the training process. Considering computational cost and model performance, we choose 10-steps on the original and augmented datasets.

Model	F1	Speedup
1-step	51.68	1.00x
5-steps	51.37	3.95x
10-steps	51.44	6.16x
15-steps	49.01	7.93x
20-steps	47.26	9.25x
First 50%	49.13	1.90x
Middle 50%	50.53	1.90x
Final 50%	49.67	1.90x

Table 6: Effect of correlation score on the Rest15 dataset in the original dataset. First 50%, Middle 50%, and Final 50% represent the start, middle and end of training.

4.9 Attention Visualization

For more intuitive understanding our approach, we visualize the attention between the input and target sequences. We train the model using a single template order and visualize the attention of the last layer in Figure 5. For aspect term and opinion term, our method can focus on specific words in the sentence and reduce redundant information. For aspect category and sentiment polarity, our method cannot pay attention to the specified words in the sentence well. For example, "positive" should focus on "nice" and "calm" instead of "The", The observations are similar in other template orders, which are presented in the Appendix A.1.

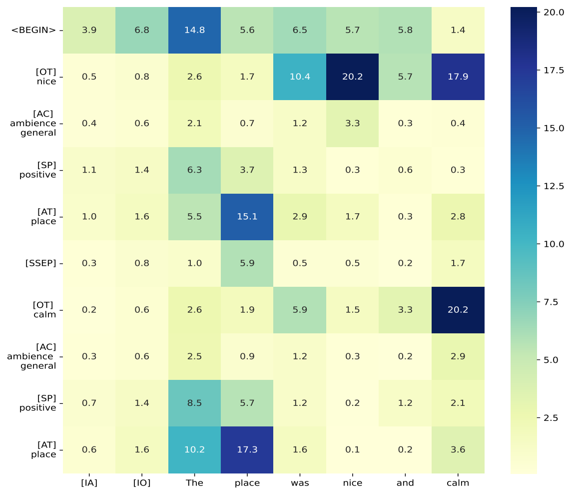


Figure 5: The visualization of attention between input sequence and target sequence. The template order is "[OT] *ot* [AC] *ac* [SP] *sp* [AT] *at*".

5 Conclusion

In this paper, we propose a relational mask multi-head attention and template-order grouping method, which can reduce the redundant information in the sentence and select appropriate template order groupings. First, we introduce a trainable relation mask matrix and use the relation constraint loss to reduce the redundant information in the input sentence. Second, we use different template orders to augment quads and deeply analyze the relationship between different templates to select the template order groupings. Finally, experiments on the original and augmented datasets demonstrate that our method outperforms the state-of-the-art methods.

Limitations

The limitations of our method are as follows:

(1) We use euclidean distance to calculate the distance between the true and predicted cross-attention. There may be other measurement methods that can achieve better results.

(2) Although the template-order grouping method can deeply analyze the relationship between different templates and achieve better performance, it also has a higher computational cost. However, the correlation score matrix between different templates is only calculated once.

References

Yinhao Bai, Yalan Xie, Xiaoyi Liu, Yuhua Zhao, Zhixin Han, Mengting Hu, Hang Gao, and Renhong Cheng. 2024. Bvsp: Broad-view soft prompting for few-shot aspect sentiment quad prediction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8465–8482.

Xiaoyi Bao, Xiaotong Jiang, Zhongqing Wang, Yue Zhang, and Guodong Zhou. 2023. Opinion tree parsing for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7971–7984.

Xiaoyi Bao, Zhongqing Wang, Xiaotong Jiang, Rong Xiao, and Shoushan Li. 2022. Aspect-based sentiment analysis with opinion tree generation. In *IJCAI*, volume 2022, pages 4044–4050.

Jiahao Bu, Lei Ren, Shuang Zheng, Yang Yang, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. Asap: A chinese review dataset towards aspect category sentiment analysis and rating prediction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2069–2079.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 340–350.

Zhuang Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3685–3694.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2509–2518, Minneapolis, Minnesota. Association for Computational Linguistics.

Lei Gao, Yulong Wang, Tongcun Liu, Jingyu Wang, Lei Zhang, and Jianxin Liao. 2021. Question-driven span labeling model for aspect–opinion pair extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12875–12883.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. Mvp: Multi-view prompting improves aspect sentiment tuple prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4380–4397.

Mengting Hu, Yinhao Bai, Yike Wu, Zhen Zhang, Liqi Zhang, Hang Gao, Shiwan Zhao, and Minlie Huang. 2023. Uncertainty-aware unlikelihood learning improves generative aspect sentiment quad prediction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13481–13494.

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022. Improving aspect sentiment quad prediction via template-order data augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7900.

Binxuan Huang and Kathleen M Carley. 2018. Parameterized convolutional neural networks for aspect level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1091–1096.

Jieyong Kim, Ryang Heo, Yongsik Seo, SeongKu Kang, Jinyoung Yeo, and Dongha Lee. 2024. Self-consistent reasoning-based aspect-sentiment quad prediction with extract-then-assign strategy. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7295–7303.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting bert for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1910.00883*.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1433–1443.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. Doer: Dual cross-shared rnn for aspect term-polarity co-extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 591–601.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3538–3547.

Yue Mao, Yi Shen, Jingchao Yang, Xiaoying Zhu, and Longjun Cai. 2022. Seq2path: Generating sentiment tuples as paths of a tree. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2215–2225.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8600–8607.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Guixin Su, Yongcheng Zhang, Tongguan Wang, Mingmin Wu, and Ying Sha. 2025. Unified grid tagging scheme for aspect sentiment quad prediction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3997–4010.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385.

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9122–9129.

An Wang, Junfeng Jiang, Youmi Ma, Ao Liu, and Naoaki Okazaki. 2023. Generative data augmentation for aspect sentiment quad prediction. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (* SEM 2023)*, pages 128–140.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.

Xiancai Xu, Jia-Dong Zhang, Rongchang Xiao, and Lei Xiong. 2023. The limits of chatgpt in extracting aspect-category-opinion-sentiment quadruples: A comparative analysis. *arXiv preprint arXiv:2310.06502*.

Yongxin Yu, Minyi Zhao, and Shuigeng Zhou. 2023. Boosting aspect sentiment quad prediction by data augmentation and self-training. In *2023 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510.

Wenyuan Zhang, Xinghua Zhang, Shiyao Cui, Kun Huang, Xuebin Wang, and Tingwen Liu. 2024a. Adaptive data augmentation for aspect sentiment quad prediction. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11176–11180. IEEE.

Yice Zhang, Jie Zeng, Weiming Hu, Ziyi Wang, Shiwei Chen, and Ruifeng Xu. 2024b. Self-training with pseudo-label scorer for aspect sentiment quad prediction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11862–11875.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2015. Representation learning for aspect category detection in online reviews. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

A Appendix

A.1 Attention Visualization of Other Template Orders

We visualize attention on multiple template orders in Figure 6, Figure 7, Figure 8, Figure 9, and Figure 10. We scale up the original attention value by 100 times for better display. If the sentiment element contains multiple words, we average the attention.

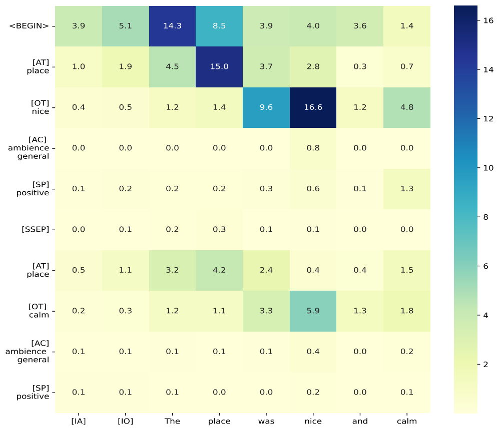


Figure 6: The visualization of attention between input sequence and target sequence. The template order is "[AT] *at* [OT] *ot* [AC] *ac* [SP] *sp*".

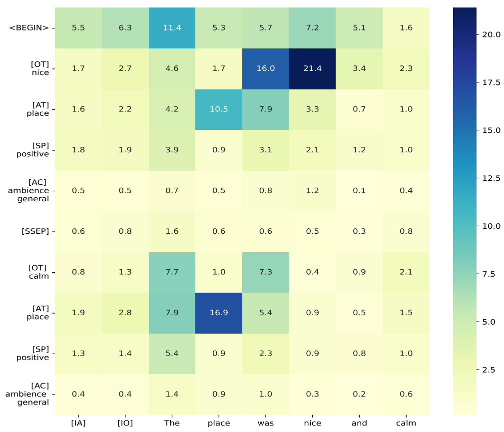


Figure 7: The visualization of attention between input sequence and target sequence. The template order is "[OT] *ot* [AT] *at* [SP] *sp* [AC] *ac*".

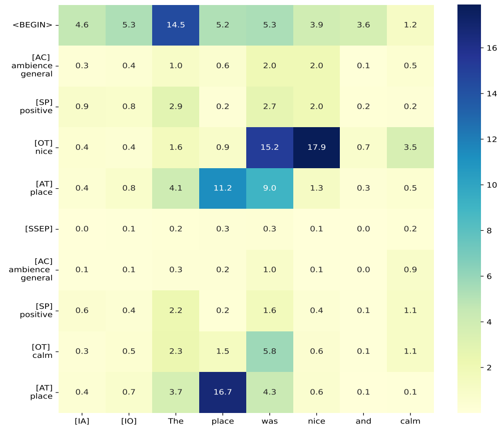


Figure 8: The visualization of attention between input sequence and target sequence. The template order is "[AC] *ac* [SP] *sp* [OT] *ot* [AT] *at*".

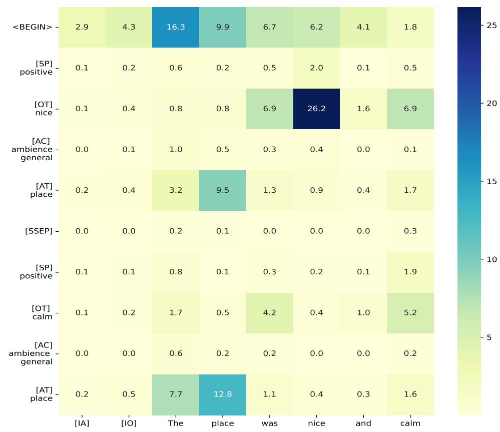


Figure 9: The visualization of attention between input sequence and target sequence. The template order is "[SP] *sp* [OT] *ot* [AC] *ac* [AT] *at*".

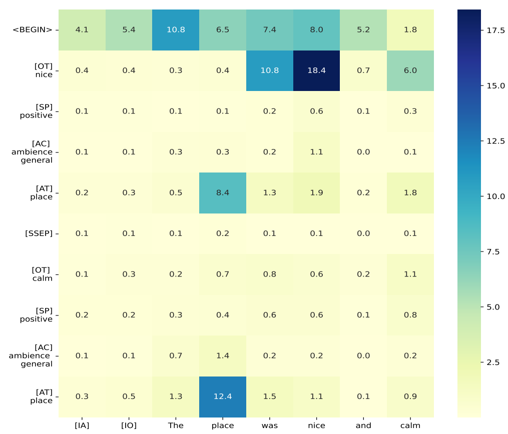


Figure 10: The visualization of attention between input sequence and target sequence. The template order is "[OT] *ot* [SP] *sp* [AC] *ac* [AT] *at*".