T-T: Table Transformer for Tagging-based Aspect Sentiment Triplet Extraction

Kun Peng[1,2], **Chaodong Tong**[1], **Cong Cao**[1], **Hao Peng**[3], **Qian Li**[4], **Guanlin Wu**[5], **Lei Jiang**[1], **Yanbing Liu**[1,2] and **Philip S. Yu**[6]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]School of Cyber Science and Technology, Beihang University
[4]School of Computer Science, Beijing University of Posts and Telecommunications
[5]College of Systems Engineering, National University of Defense Technology
[6]Department of Computer Science, University of Illinois at Chicago
{pengkun, tongchaodong}@iie.ac.cn

Abstract

Aspect sentiment triplet extraction (ASTE) aims to extract triplets composed of aspect terms, opinion terms, and sentiment polarities from given sentences. The table tagging method is a popular approach to addressing this task, which encodes a sentence into a 2-dimensional table, allowing for the tagging of relations between any two words. Previous efforts have focused on designing various downstream relation learning modules to better capture interactions between tokens in the table, revealing that a stronger capability to capture relations can lead to greater improvements in the model. Motivated by this, we attempt to directly utilize transformer layers as downstream relation learning modules. Due to the powerful semantic modeling capability of transformers, it is foreseeable that this will lead to excellent improvement. However, owing to the quadratic relation between the length of the table and the length of the input sentence sequence, using transformers directly faces two challenges: overly long table sequences and unfair local attention interaction. To address these challenges, we propose a novel Table-Transformer (T-T) for the tagging-based ASTE method. Specifically, we introduce a stripe attention mechanism with a loop-shift strategy to tackle these challenges. The former modifies the global attention mechanism to only attend to a 2-dimensional local attention window, while the latter facilitates interaction between different attention windows. Extensive and comprehensive experiments demonstrate that the T-T, as a downstream relation learning module, achieves state-of-the-art performance with lower computational costs.

1 Introduction

Aspect sentiment triplet extraction (ASTE) remains a crucial research direction in the era of large language models (LLMs) [Wang *et al.*, 2023; Zhang *et al.*, 2024], widely used for fine-grained opinion mining from user reviews, social media news, and other types of text [Zhang *et al.*, 2022a]. ASTE task aims to extract corresponding aspect terms, opinion terms, and sentiment polarities from a given sentence. The sentiment polarity here is classified into three categories: {Positive, Neutral, Negative}. For example, as shown above in Figure 1, given the sentence "*The screen of the phone is smaller, but overall it's good.*", it has two sentiment triplets: (*screen*, *smaller*, Negative) and (*phone*, *good*, Positive), where *screen* and *phone* are the aspect terms, *smaller* and *good* are the corresponding opinion terms, Negative and Positive are the corresponding sentiment polarities.

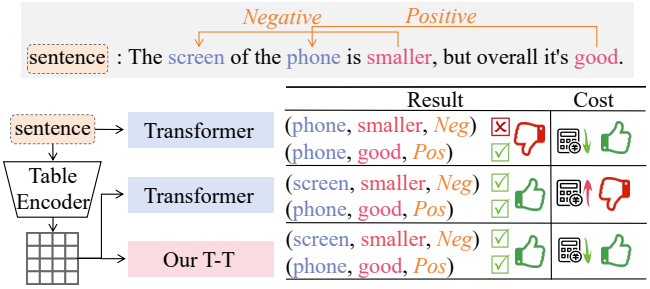


Figure 1: A toy example of three different ASTE methods. Our T-T model achieves commendable results in both performance and cost.

Recently, a variety of techniques have been proposed for addressing the ASTE task, including span-based methods [Xu *et al.*, 2021; Chen *et al.*, 2021; Li *et al.*, 2023b; Zhao *et al.*, 2024], generative methods [Gao *et al.*, 2022; Gou *et al.*, 2023; Xianlong *et al.*, 2023; Mukherjee *et al.*, 2023; Zou *et al.*, 2024], and tagging methods, among which the tagging method is particularly attractive. Sequence tagging methods [Peng *et al.*, 2020; Xu *et al.*, 2020; Xianlong *et al.*, 2023] use the BIESO scheme[1] to tag aspect and opinion terms in sentences, but these methods fail to fully capture the relations between individual words. Another more competitive tagging approach is the table tagging methods [Wu *et al.*, 2020; Chen *et al.*, 2022; Zhang *et al.*, 2022b; Liang *et al.*, 2023b; Sun *et al.*, 2024]. Given a sentence, it

[1]BIESO means "begin, inside, end, single, other", respectively.

begins with encoding using a transformer-based pre-trained language model. Then, the obtained representations are encoded into a relation table. In this relation table, the horizontal axis represents aspect terms, the vertical axis represents opinion terms, and each cell (e.g. $cell_{ij}$) in the table denotes the relation between the i-th and j-th words in the sentence. With this approach, sentiment triplets can be easily annotated within the table, and the relations between any two words can be fully captured.

One of the keys to the table tagging methods lies in designing different downstream relation encoding modules to help the model learn stronger relations representations. Previous methods [Wu *et al.*, 2020; Jing *et al.*, 2021; Chen *et al.*, 2022; Zhang *et al.*, 2022b; Liang *et al.*, 2023a; Peng *et al.*, 2024b; Peng *et al.*, 2024a] have designed various downstream relation encoding modules, such as using Multi-Layer Perception (MLP) [Jing *et al.*, 2021], Graph Convolutional Network (GCN) [Chen *et al.*, 2022], and Convolutional Neural Network (CNN) [Zhang *et al.*, 2022b]. These works demonstrate that a stronger relation encoding module can yield greater overall benefits for the model. This motivates us to seek even more powerful relation encoding modules.

In this work, we attempt to directly utilize transformer layers [Vaswani *et al.*, 2017] as our downstream relation encoding module. Ideally, the use of self-attention mechanisms in transformers enables the model to capture dependencies across the entire table sequences. This capability holds the promise of significantly enhancing the model's performance. However, in reality, two formidable challenges impede the realization of this technique. **Challenge 1: Overly long table sequences.** It is known that, if assuming the input sentence sequence length is n, the time and space complexity of the transformer layer in the multi-head self-attention module is $O(n^2)$. However, when dealing with table sequences, where the length of the table itself is n^2, the computational complexity of the attention mechanism in the transformer layer escalates to $O(n^4)$. This is unacceptable with limited computational resources. **Challenge 2: Unfair local attention interaction.** Long sequences encourage us to use local attention mechanisms, but in reality, the correlation between tokens is not confined to local areas. When each token only focuses on its immediate periphery, capturing information from interactions with distant tokens becomes challenging.

To address the aforementioned challenges, we propose a novel Table-Transformer (T-T) for ASTE's table tagging method. As shown in Figure 1, when directly utilizing transformer-based sequence labeling models to process sentences, although cost-effective, their performance suffers due to their inability to fully capture word relations. However, when encoding sentences into table sequences first and then using transformer layers as the relation encoder can yield good results, but the costs are intolerable due to the challenge of overly long sequences. Our approach, tailored for table sequence inputs, introduces unique enhancements to the original transformer layers, enabling the model to achieve strong performance while maintaining relatively low computational costs. Specifically, to address the **overly long table sequences** challenge, we propose a stripe attention mechanism. It enhances the original global self-attention mechanism by restricting it to operate only within a fixed-size window range (assumed to be a constant k). This modification reduces the original time-space complexity from $O(n^4)$ to $O(k^2n^2)$. For the **unfair local attention interaction** challenge, we devise a novel loop-shift strategy that effectively facilitates information interaction between different windows.

Our contributions can be summarized as follows:

1) We observe the two-dimensional nature of table sequences and based on this insight, we have designed an enhanced transformer layer called T-T. It can effectively capture local information within table sequences without relying on external knowledge or task-specific designs.

2) We introduce two novel techniques aimed at effectively addressing the two challenges encountered by the original transformer when handling table sequences.

3) Experimental results show that our approach achieves state-of-the-art performance within acceptable costs.

2 Related Work

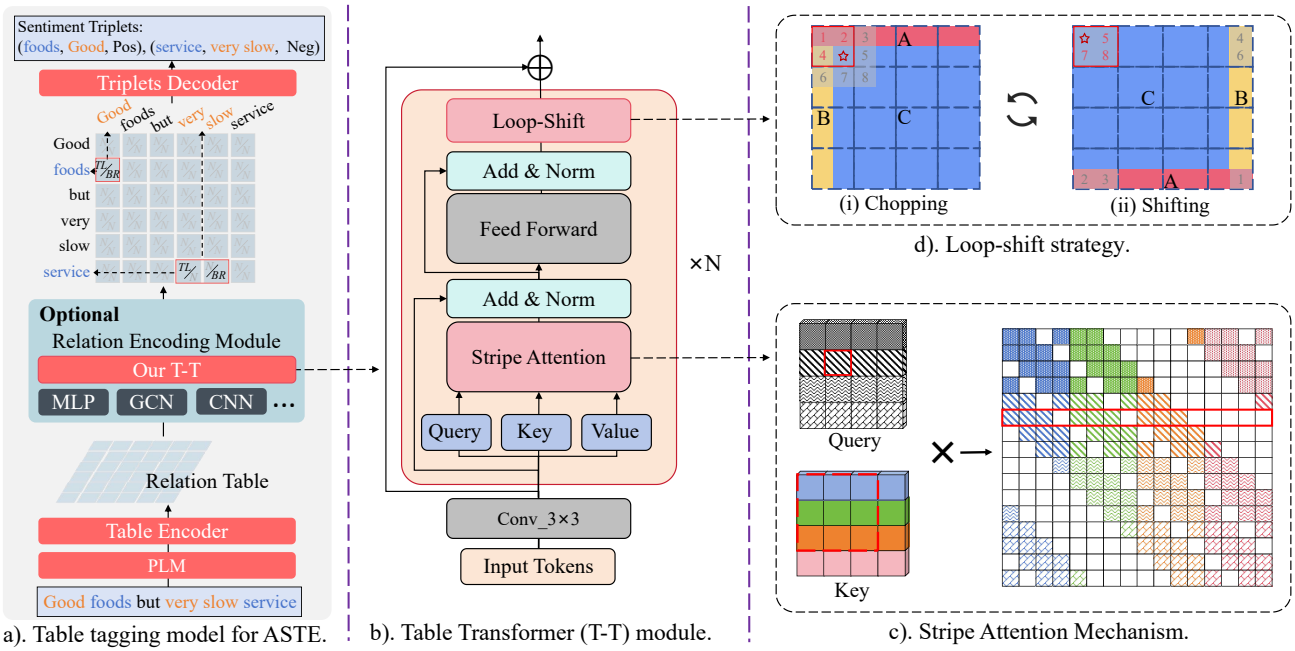
Aspect Sentiment Triplet Extraction.

To meet the demand for a more detailed exploration of the opinion contained within the text, [Peng *et al.*, 2020] first proposed and addressed the ASTE task in a pipeline manner. Subsequently, more diversified techniques have been proposed. MRC methods [Mao *et al.*, 2021; Zhai *et al.*, 2022; Zou *et al.*, 2024] treated the ASTE task as a form of machine reading comprehension task. Generative methods [Zhang *et al.*, 2021; Gao *et al.*, 2022; Gou *et al.*, 2023; Mukherjee *et al.*, 2023] treated the ASTE task as an index generation task. Span-based methods [Xu *et al.*, 2021; Li *et al.*, 2023b; Zhao *et al.*, 2024] extracted all possible spans and considered the interplay of information at the span level.

Table tagging, as another vibrant research direction, was initially proposed by GTS [Wu *et al.*, 2020] to annotate sentiment triplets in a 2D table. Subsequent extensive work, despite variations in the tagging schema, primarily focused on how to capture sufficient relation information for the table. EMC-GCN [Chen *et al.*, 2022] utilized GCN to incorporate rich syntactic information. BDTF [Zhang *et al.*, 2022b] employed CNN to fully grasp word boundaries. STAGE [Liang *et al.*, 2023a] and SimSTAR [Li *et al.*, 2023a] integrated the learning of span-level information into the relation encoder. MiniConGTS [Sun *et al.*, 2024] proposed a new table tagging scheme based on contrastive learning. In conclusion, the design of a more powerful downstream relation encoder stands as a pivotal focus in the table tagging methods.

Efficient Transformer.

The transformer model architecture [Vaswani *et al.*, 2017] has become an indispensable tool in modern deep learning research due to its effectiveness in the field of NLP [Tay *et al.*, 2022]. However, the self-attention mechanism and stacked design logic of the transformer layers result in high computational resource requirements. A large amount of previous work has focused on addressing this issue. Some efforts attempted to reduce computational costs by modifying the scope of self-attention calculations. These strategies include sliding windows [Beltagy *et al.*, 2020; Zaheer *et al.*, 2020] or different attention mechanisms at various layers [Zhang *et*



a). Table tagging model for ASTE. b). Table Transformer (T-T) module. d). Loop-shift strategy. c). Stripe Attention Mechanism.

Figure 2: a) shows the architecture of our table tagging model for ASTE. One of the core components is the configurable relation encoding module. *TL* and *BR* donate the top-left and bottom-right vertex cells of the sentiment region, respectively. b) shows the architecture of our proposed T-T module. In sub-figure c), the left half shows the query matrix and key matrix divided into 4^2 blocks, while the right half represents the attention map, where the colored blocks indicate the dot product computations. In sub-figure d), the final outputs of different layers shift between state (i) and state (ii). For a boundary token (marked with a red star), it attends to tokens 1, 2, and 4 in the output state (i), and to tokens 5, 7, and 8 in the output state (ii).

al., 2023]. Other efforts have aimed at optimizing the computational efficiency of the self-attention matrix, such as applying a low-rank matrix [Wang *et al.*, 2020], a kernel function [Katharopoulos *et al.*, 2020], or a focused attention module [Han *et al.*, 2023].

3 Preliminary

Task Definition.

Given a sentence $X = \{x_1, x_2, ..., x_n\}$ of length n, the aim of ASTE is to extract all sentiment triplets $T = \{(a_1, o_1, s_1), (a_2, o_2, s_2), ..., (a_{|T|}, o_{|T|}, s_{|T|})\}$ from X. Here, a and o respectively represent the aspect term and opinion term, which are spans within X, while s denotes the corresponding sentiment polarity in {*Positive, Neutral, Negative*}.

Tagging Scheme.

We choose the boundary-based tagging scheme [Zhang *et al.*, 2022b; Ning *et al.*, 2023] in our work. For sentence X, it is mapped onto a two-dimensional table of size $n \times n$, where the vertical and horizontal axes represent aspect terms and opinion terms, respectively. For an sentiment triplet (a_i, o_i, s_i) in X, assuming that a_i and o_i are located at positions $[x, y]$ and $[m, n]$ ($x \leq y, m \leq n$). They can form a rectangular region in the table, with coordinates (x, m) as the top-left corner and (y, n) as the bottom-right corner. We label the cells on the boundary of this region as *TL* (top-left vertex) and *BR* (bottom-right vertex), while all other cells are tagged as *None*. When the sentiment elements are single words, the *TL* and *BR* positions coincide. Subsequently, we label the sentiment polarity of this rectangular region according to s_i.

Generalised Attention Mechanism.

For an input sequence $C = \{c_1, c_2, ..., c_n\} \in \mathbb{R}^{n \times d}$, the generalised attention mechanism of Transformer [Vaswani *et al.*, 2017] treats the attention calculation over the entire sequence as a directed graph $G = \{C, A\}$, where the adjacency matrix $A \in \mathbb{R}^{n \times n}$ is used to describe directed edges. For any $a_{ij} \in A$ ($a_{ij} = 1$ or 0), if $a_{ij} = 1$, it indicates that the query c_i attends to the key c_j. Let $N(c_i)$ denote all neighbors of c_i (including itself), the attention calculation for c_i is defined as:

$$ATTN(c_i) = \sigma(Q(c_i)K(N(c_i))^T)V(N(c_i)), \quad (1)$$

where $Q(\cdot), K(\cdot) : \mathbb{R}^d \to \mathbb{R}^m$ and $V(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ are the query, key, and value functions, respectively. σ is a softmax function.

4 Method

Our framework is shown in Figure 2, with the relation encoding module configured as our T-T[2].

4.1 Table Encoding

As illustrated in Figure 2 a), sentence X is first inputted into a pre-trained language model (PLM) for encoding, yielding the last hidden layer's representation $H = \{h_1, h_2, ..., h_n\}$.

[2]Codes are available at https://github.com/KunPunCN/T-T

Subsequently, we extract the features of aspects and opinions from $H \in \mathbb{R}^{n\times d}$ using two separate linear layers:

$$h_i^a = Linear_a(h_i), \quad h_j^o = Linear_o(h_j), \tag{2}$$

where $h_i^a \in \mathbb{R}^d$ and $h_j^o \in \mathbb{R}^d$ denote the aspect and opinion representation, respectively. To ensure that the table captures sufficient relation information, which is beneficial for downstream tasks, we followed the previous works [Chen *et al.*, 2022], employing Biaffine Attention [Dozat and Manning, 2017] to capture the relations distribution between words. This process is formalized as:

$$r_{ij} = \boldsymbol{W_1}(h^a{}_i \oplus (h^o)_j) \oplus h^a{}_i{}^T \boldsymbol{W_2} h^o{}_j \oplus Pooling(h_{[i,j]}), \tag{3}$$

where $\oplus$ denotes the concatenation operation. The first two terms represent the biaffine attention process, where $\boldsymbol{W_1} \in \mathbb{R}^{d\times 2d}$ and $\boldsymbol{W_2} \in \mathbb{R}^{d\times\sqrt{d}\times d}$ are learnable parameters. The last term $Pooling(h_{[i,j]})$ represents the maxpooling operation for the sentence from token h_i to h_j, which aims to extract the span-level information. $r_{ij} \in \mathbb{R}^{2d+\sqrt{d}}$ is the table representation. Finally, a dense linear layer $Linear_d$: $\mathbb{R}^{2d+\sqrt{d}} \rightarrow \mathbb{R}^{d'}$ is used to compress r_{ij}'s dimension to d': $r'_{ij} = Linear_d(r_{ij})$. Therefore, the final representation of the relation table is denoted as $R \in \mathbb{R}^{n\times n\times d'}$.

4.2 Relation Encoding with T-T module

The initialized relation table needs additional refinement of the relations distribution through the proposed T-T module. As depicted in Figure 2 b), we first use a simple convolutional layer with a kernel size of 3 to capture spatial correlations, and then feed the obtained representation $R^0 \in \mathbb{R}^{n\times n\times d'}$ into the T-T module. The T-T improves upon the original transformer architecture by replacing the full attention mechanism with an enhanced stripe attention mechanism and adding a novel loop-shift strategy at the end of each layer.

Stripe Attention Mechanism.

Here we propose a novel stripe attention mechanism to address the overly long table sequence challenge. As described in the Preliminary section, the generalised attention mechanism treats the attention calculation as a directed graph $G = \{C, A\}$. In the original transformer layer, a full attention mechanism is used, which means A is the full ones matrix and $N(c_i) = C$ in Formula (1). When the input is a sequence $C \in \mathbb{R}^{n\times d'}$, each token needs to interact with all other tokens and store all computation results, leading to a spatiotemporal complexity of $O(n^2)$. The derivation reveals that the spatiotemporal complexity of the full attention layer scales quadratically with the length of the input sequence. When our input is the flattened table sequence $R^0 \in \mathbb{R}^{n\times n\times d'}$, the overall spatiotemporal complexity becomes $O(n^4)$. This is intolerable for limited computational resources.

To maintain good performance while reducing the spatiotemporal complexity, the proposed stripe attention mechanism partitions the table sequence R^0 into multiple smaller blocks, treating them as the smallest computational units. Assuming each block has a width of b and a shape of $\mathbb{R}^{b\times b\times d'}$ (where n is padded beforehand to a multiple of b), this results in a total of l^2 (we define $l = \frac{n}{b}$) blocks: $O = \{o_1, o_2, ..., o_{l^2}\} \in \mathbb{R}^{l^2\times d'}$. It is worth noting that we apply a cyclic shift operation on this sequence, which means that for a block o_i in this sequence, its left and right neighbors in the attention map are $o_{i\pm1} = o_{(i\pm1+l^2)\%l^2}$, and its upper and lower neighbors are $o_{i\pm l} = o_{(i\pm l+l^2)\%l^2}$, where % is used for the remainder calculation. Similar to Formula 1, for each block o_i, it's attention calculation is defined as:

$$ATTN(o_i) = \sigma(Q(o_i)K(N(o_i))^T)V(N(o_i)). \tag{4}$$

Per Formula 4, the query matrix and the key matrix require a dot product operation to generate the attention map.

As shown in the left half of Figure 2 c), the query and key matrices are also partitioned into multiple blocks. According to Formula 4, a dot product operation is needed for the query matrix and the key matrix to generate the attention map. In the full attention mechanism, each query block needs to undergo a dot product operation with all key blocks, resulting in a complexity of $O(n^4)$. To enhance attention performance while maintaining lower complexity, in our stripe attention mechanism, each query block only selectively attends to its neighboring blocks in the key matrix. As indicated by the highlighted red boundary box in Figure 2 c), for a query block, we define its neighbors in the 2D table space as all key blocks within a square area centered around it. Assuming a window width of $w(w \leq l)$ centered around each block for attention, then the neighbors of block o_i can be defined as:

$$N(o_i) = \begin{Bmatrix} o_{i-\lfloor\frac{w}{2}\rfloor l-\lfloor\frac{w}{2}\rfloor}, & \cdots & o_{i-\lfloor\frac{w}{2}\rfloor l+\lfloor\frac{w}{2}\rfloor}, \\ o_{i-(\lfloor\frac{w}{2}\rfloor-1)l-\lfloor\frac{w}{2}\rfloor}, & \cdots & o_{i-(\lfloor\frac{w}{2}\rfloor-1)l+\lfloor\frac{w}{2}\rfloor}, \\ \vdots & \vdots & \vdots \\ o_{i+\lfloor\frac{w}{2}\rfloor l-\lfloor\frac{w}{2}\rfloor}, & \cdots & o_{i+\lfloor\frac{w}{2}\rfloor l+\lfloor\frac{w}{2}\rfloor} \end{Bmatrix}, \tag{5}$$

where w is odd. The number of $N(x_i)$ is $|N(x_i)| = w^2$, which means that for each block (total of n^2 tokens), it attends to all blocks within a square area of w^2, encompassing a total of $w^2 \times b^2$ tokens. The overall complexity is $O(n^2 \times w^2 b^2) = O(w^2 b^2 n^2)$. In addition, when $w = l$, $N(x_i)$ will include all blocks, and stripe attention will degenerate into full attention.

An example of the stripe attention map is shown in the right half of Figure 2 (c), where $n = 4b$ and $w = 3$. Compared to local attention, stripe attention allows tokens to attend to a broader range of relevant positions in the 2D table space, thereby enhancing learning capability, while only marginally increasing complexity by a constant factor.

Loop-shift Strategy.

Here we propose an innovative loop-shift strategy to address the unfair local attention interaction challenge. While the stripe attention mechanism achieves a balance between attention scope and computational load, partitioning the table sequence into fixed block widths prohibits interaction between different blocks. This presents an unfair scenario for tokens situated at block edges, as their crucial neighbors may reside in different blocks. Therefore, inspired by [Liu *et al.*, 2021], we introduce a novel loop-shift operation to the table sequence between different T-T layers. For the output of any given layer, which has a shape of $n \times n$, before feeding

it into the next layer, we first transform the positional distribution of the table tokens. As illustrated in Figure 2 d), we perform a two-step shift operation on the table with a width of $\frac{b}{2}$. Firstly, we chop the tokens from the top of the table, forming a shape of $\frac{b}{2} \times n$, and move them to the bottom of the table. Secondly, we chop the tokens on the left side of the table, forming a shape of $n \times \frac{b}{2}$, and move them to the right side of the table. This operation effectively cyclically shifts the entire table to the upper-left direction by a distance of $\frac{b}{2}$. Similarly, for the next layer's output, we need to cyclically shift the entire table in the opposite direction, by a distance of $\frac{b}{2}$ towards the bottom-right direction, to restore the table's distribution. Therefore, the shift operation occurs in pairs, implying that the number of T-T layers in our model is even.

After passing through N layers of the T-T module, we obtain the last layer's representation $R^N \in \mathbb{R}^{n \times n \times d'}$. Then a weighted residual connection is used to derive the final representation: $R = \boldsymbol{W_3} R^N + (1 - \boldsymbol{W_3}) R^0$.

4.3 Triplet Decoding

During the decoding stage, we employ two separate full linear layers to predict the coordinates of all *TL* and *BR*, respectively. The predicted results for each position are p_{ij}^{TL} and p_{ij}^{BR}, respectively. Given the labels y_{ij}^{TL} and y_{ij}^{BR}, the corresponding loss is:

$$\mathcal{L}_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} L_{CE}(p_{ij}^{TL}, y_{ij}^{TL}) + L_{CE}(p_{ij}^{BR}, y_{ij}^{BR}), \quad (6)$$

where L_{CE} is the cross entropy loss function. Subsequently, for any pair of *TL* and *BR*, they form a candidate rectangle when the *TL* is positioned to the upper left or coincides with the *BR*. This rectangle is represented as:

$$r_{abcd} = r_{ab} \oplus r_{cd} \oplus pooling(r_{ab}, ..., r_{cd}), \quad (7)$$

where r_{ab} and r_{cd} respectively represent the token at *TL* and *BR*, $r_{ab}, ..., r_{cd}$ denote all tokens within this rectangle. Then, a sentiment classifier $Linear_s$: $\mathbb{R}^{3d} \rightarrow \mathbb{R}^4$ is utilized to predict the sentiment polarity of this region as one of { *Pos, Neu, Neg, Invalid*}: $p_k^s = Linear_s(r_{abcd})$. Since each *TL/BR* may form candidate rectangles with multiple *BR/TL*, the label *Invalid* is additionally used to assess the validity of the candidate region. When the prediction p_k^s is *Invalid*, the region should be dropped. When the prediction p_k^s falls within { *Pos, Neu, Neg*}, the corresponding sentiment triplet can be extracted based on the *TL*, *BR* coordinates, and sentiment polarity of the region. The loss of this process is:

$$\mathcal{L}_2 = \sum_{k=1}^{m} L_{CE}(p_k^s, y_k^s), \quad (8)$$

where m represents the number of candidate rectangles, p_k^s and y_k^s are the prediction and label, respectively, for the k-th candidate rectangles. The overall loss is: $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$.

5 Experiments

5.1 Experimental Settings

Datasets and Baselines.

We conduct our experiments on four benchmark datasets [Peng *et al.*, 2020; Xu *et al.*, 2020], which are originally derived from the SemEval challenge [Pontiki *et al.*, 2014; Pontiki *et al.*, 2015; Pontiki *et al.*, 2016]. The detailed statistics are provided in Table 1. We categorize the comparison baselines into five types: seq tagging, MRC-based, generative, span-based, and table tagging.

Dataset	14res		14lap		15res		16res	
	#S	#T	#S	#T	#S	#T	#S	#T
train	1,266	2,338	906	1460	605	1,013	857	1,394
dev	310	577	219	346	148	249	210	339
test	492	994	328	543	322	485	326	514

Table 1: Statistics of datasets, where #S and #T represent the number of sentences and triplets, respectively.

Implementation.

The proposed model contains a table encoder and a relation encoder, with hidden state dimensions of $d = 768$ and $d' = 1024$, respectively. We initialize the table encoder with the BERT-base-uncased [Devlin *et al.*, 2019] version. As a relation encoder, our T-T module consists of two transformer layers and utilizes the parameters from the last two layers of the BERT-large version for initialization. The block width b is 7 for 14res, 15res, 16res and 5 for 14lap. The window width w is 3. During training, we use the AdamW optimizer with an initial learning rate of 3e-5 for all layers. The model is trained for 15 epochs on RTX 3090 GPUs, with a batch size of 4. In each epoch, we evaluate the training model on the development set and save the best one. We use the sentence-level F1 score as the evaluation metric, which means that a sentence is considered a true positive only when all triples within it are correctly extracted. All the reported results are the average of five runs with different random seeds.

5.2 Main Results

As shown in Table 2, we have the following observations: Our model surpassed the best baseline, D2E2S, with improvements of 0.66%, 0.02%, 2.75%, and 0.1% F1 scores on 14res, 14lap, 15res, and 16res, respectively, resulting in an overall average improvement of 0.89% F1 score. While D2E2S introduced additional tools to provide syntactic dependency information and developed task-specific unique modules, our model solely utilized a streamlined architecture to effectively capture local information within 2D table sequences.

Our model significantly outperformed all table tagging methods. Compared to the best table-based model, MiniConGTS, our model surpassed it by 0.2%, 0.07%, 3.46%, and 0.01% F1 scores on 14res, 14lap, 15res, and 16res, respectively, under the F1 scores, with an overall average improvement of 0.93%. This demonstrates that our T-T module exhibits stronger relation encoding capabilities compared to previous relation encoding modules.

Compared to initializing the T-T parameters using the last two layers of BERT, "Random init." indicates random initialization of the T-T, which resulted in an average decrease of 0.70% F1 score in overall model performance. We attribute this to the insufficient training data size to optimize the parameters of the two transformer layers, resulting in model underfitting. On the other hand, initializing with the last two lay-

Model	14res			14lap			15res			16res			Avg-F1
	P.	R.	F1	P.	R.	F1	P.	R.	F1	P.	R.	F1	
Seq tagging													
Peng-two-stage [Peng *et al.*, 2020]	43.24	63.66	51.46	37.38	50.38	42.87	48.07	57.51	52.32	46.96	64.24	54.21	50.22
TAGS‡ [Xianlong *et al.*, 2023]	74.92	73.81	74.36	64.69	61.89	63.26	69.55	65.25	67.33	75.40	72.48	74.17	69.78
MRC-based													
COM-MRC [Zhai *et al.*, 2022]	75.46	68.91	72.01	62.35	58.16	60.17	68.35	61.24	64.53	71.55	71.59	71.57	67.07
Triple-MRC [Zou *et al.*, 2024]	/	/	72.45	/	/	60.72	/	/	62.86	/	/	68.65	66.17
Generative													
GPT3.5‡ [Ouyang *et al.*, 2022]	60.77	54.22	57.31	59.13	49.27	53.76	58.24	48.23	52.77	59.33	56.02	57.60	55.36
LEGO [Gao *et al.*, 2022]	/	/	73.7	/	/	62.2	/	/	64.4	/	/	69.9	67.55
MvP [Gou *et al.*, 2023]	/	/	74.05	/	/	63.33	/	/	65.89	/	/	73.48	69.19
CONTRASTE [Mukherjee *et al.*, 2023]	73.6	74.4	74.0	64.2	**61.7**	62.9	65.3	**66.7**	66.1	72.2	76.3	74.2	69.3
Span-based													
Span-ASTE [Xu *et al.*, 2021]	72.89	70.89	71.85	63.44	55.84	59.38	62.18	64.45	63.27	69.45	71.17	70.26	66.19
D2E2S [Zhao *et al.*, 2024]	75.92	74.36	75.13	67.38	60.31	63.65	70.09	62.11	65.86	**77.97**	71.77	74.74	69.84
Table tagging													
GTS-BERT† [Wu *et al.*, 2020]	68.09	69.54	68.81	59.40	51.94	55.42	59.28	57.93	58.60	68.32	66.86	67.58	62.60
EMC-GCN [Chen *et al.*, 2022]	71.21	72.39	71.78	61.70	56.26	58.81	61.54	62.47	61.93	65.62	71.30	68.33	65.21
BDTF [Zhang *et al.*, 2022b]	75.53	73.24	74.35	68.94	55.97	61.74	68.76	63.71	66.12	71.44	73.13	72.27	68.62
STAGE-3D [Liang *et al.*, 2023a]	**78.58**	69.58	73.76	**71.98**	53.86	61.58	**73.63**	57.90	64.79	76.67	70.12	73.24	68.34
SimSTAR [Li *et al.*, 2023a]	76.23	71.63	73.86	66.46	58.23	62.07	71.71	59.59	65.09	72.07	74.12	73.06	68.52
MiniConGTS [Sun *et al.*, 2024]	76.10	**75.08**	75.59	66.82	60.68	63.61	66.50	63.86	65.15	75.52	74.14	74.83	69.80
T-T (Ours)	77.06	74.56	**75.79**	67.50	60.25	**63.67**	72.15	65.40	**68.61**	75.05	**74.63**	**74.84**	**70.73**
-w/ Random init.	76.24	73.44	74.81	66.61	60.19	63.24	71.17	65.39	68.16	74.60	73.19	73.89	70.03

Table 2: Main results on 4 datasets. †, ‡ denote that results are obtained from [Chen *et al.*, 2022] and conducted by us, other results are from the original papers. The best results are in **bold**, while the second best are underlined.

Ablation Settings	#Param.	#Training Cost	#Interfere Cost	14res F1	14lap F1	15res F1	16res F1
Full Model	184.3M	5.93 /ms	1.44 /ms	75.79	63.67	68.61	74.84
-w/o Loop-shift	184.3M	5.53 /ms	1.33 /ms	73.63	62.35	66.38	72.73
-w/o Stripe Attention (SA)	184.3M	19.14 /ms	4.40 /ms	75.81	63.89	68.73	74.80
-w/ Normal layers	184.3M	19.03 /ms	4.32 /ms	75.74	63.81	68.52	74.68
-w/o T-T Relation Encoder	159.1M	4.74 /ms	1.12 /ms	71.42	60.28	64.67	70.51

Table 3: Ablation results on 4 datasets. "-w/o" means without. "/ms" donates the average computation time per sentence.

ers of pre-trained BERT incorporates certain semantic knowledge, thereby improving the model's effectiveness.

5.3 Ablation Study and Computational Cost

We conduct an ablation experiment to validate the proposed module's improvements in model performance and costs. As shown in Table 3, since loop-shift and Stripe Attention (SA) are parameter-independent mechanisms, removing them will not alter the model's parameters.

When the loop-shift strategy is removed, the training and inference costs remain largely unchanged. However, the performance decreased by 2.16%, 1.32%, 2.23%, and 2.11% F1 scores on the four datasets, demonstrating the crucial importance of information interaction across different blocks for enhancing the overall model performance. When SA is removed, the model degrades to full attention while retaining the Loop-shift strategy, resulting in performance similar to that of directly using normal layers. After removing SA, we observe a slight improvement in model performance, but there is a significant deterioration in both training and inference costs, which are 13.21/ms and 2.96/ms, respectively. This demonstrates that the Stripe Attention mechanism not only greatly improves attention computation efficiency but also effectively attends to relevant tokens. When the T-T relation encoder module is removed, the model's performance significantly deteriorates, with respective drops of 4.37%, 3.39%, 3.94%, and 4.33%. This demonstrates the critical importance of designing a stronger relation encoding module.

Model	14res	14lap	15res	16res	Avg-F1
GTS†	74.63	66.46	67.52	74.20	70.70
EMC-GCN†	76.33	67.94	67.26	74.15	71.42
STAGE†	77.87	69.70	70.60	79.98	74.54
MiniConGTS	79.60	73.23	73.87	76.29	75.75
T-T	**79.97**	**73.40**	**74.35**	**80.11**	**76.96**

Table 4: Results on the AOPE task. † donates that results are derived from [Liang *et al.*, 2023a]. The best results are in **bold**, while the second best are underlined.

5.4 Auxiliary Experiment on Subtask

To further investigate the effectiveness of T-T, we conduct an auxiliary experiment on the Aspect Opinion Pair Extraction

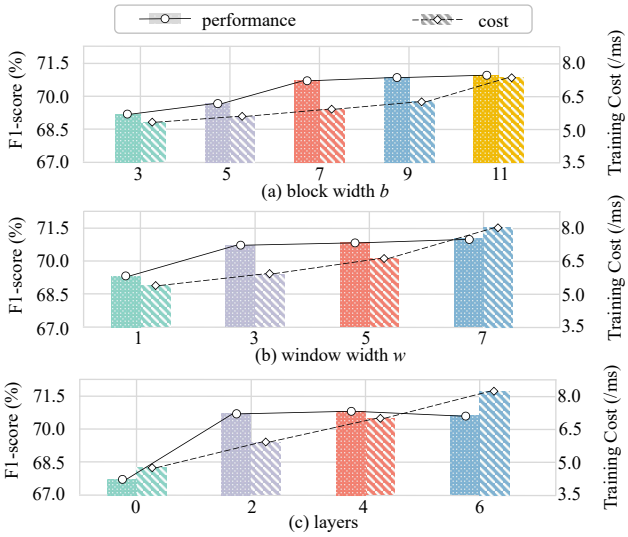


Figure 3: The sensitivity of different hyperparameters.

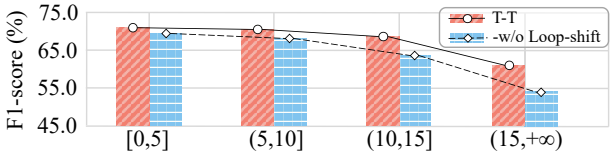


Figure 4: F1 scores for different aspect-opinion word distances on the test set. The sample counts in different distance intervals are 2442, 449, 107, and 52, respectively.

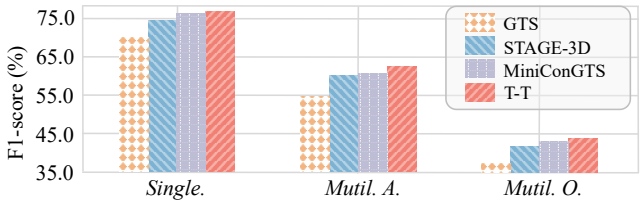


Figure 5: Performance of different word spans. *Single.* denotes triplets with single-word aspects and opinions. *Multi. A./Multi. O.* denote triplets with multiple-word aspects/opinions.

(AOPE) task, which aims to extract all aspect-opinion pairs from a sentence. By modifying the category width of the sentiment classifier $Linear_s$ from 4 ($\{$*Pos, Neu, Neg, invalid*$\}$) to 2 ($\{$*Valid, invalid*$\}$), our model can directly address these tasks without any additional modifications. As depicted in Table 4, we chose the table tagging methods for comparison to demonstrate the specific improvements of our model. The results demonstrate that our T-T achieves comprehensive improvements across four datasets, with an average F1 score increase of 0.96% compared to the best baseline, MiniConGTS. This further validates the important role of T-T in capturing and matching token relationships within the table sequences.

5.5 Hyperparameter Analysis

To investigate the impact of different hyperparameter settings in the T-T module on model performance, we conduct an additional experiment. The results, depicted in Figure 3, demonstrate the effects when varying (a) the block width b, (b) the window width w, and (c) the T-T layer count while keeping other settings consistent with the main experiment.

In Figure 3 (a), as the block width b increases, the training time cost exhibits a nearly quadratic growth, consistent with the $O(w^2b^2n^2)$ complexity of the stripe attention mechanism as stated earlier. Additionally, when the block width exceeds 7, there is minimal improvement in the model's performance. We attribute this to the observation range ($b \times w = 3 \times 7$) approaching saturation, effectively attending to all important tokens. The findings drawn from Figure 3 (b) are consistent with Figure 3 (a) as the training time cost exhibits a quadratic increase with the window width w. Additionally, when the window width exceeds 3, there is minimal performance improvement. In Figure 3 (c), as the number of T-T layers increases, the computational cost shows linear growth. When the number of layers exceeds 2, there is a slight decline in model performance, which we attribute to potential underfitting caused by an excessive number of parameters.

5.6 Further Analysis on Loop-shift Strategy

The loop-shift strategy is employed to address the challenge of **unfair local attention interaction**. We conducted an additional experiment to validate this strategy further. As shown in Figure 4, removing the loop-shift strategy results in a significant performance drop of the model on long-distance pairs. This demonstrates the effectiveness of the loop-shift strategy, as removing it prevents tokens at the boundaries of attention windows from fully capturing valuable information, resulting in significant performance degradation.

5.7 Performance of Different Word Spans

We also compare the performance of T-T with other table tagging methods across different word spans, including single-word (*Single.*), multi-word aspect (*Mutil. A.*), and multi-word opinion (*Mutil. O.*). The results are shown in Figure 5. Our model outperforms previous methods in all settings, with the improvement being more pronounced in the multi-word setting. This highlights the ability of T-T to effectively capture word boundary information.

6 Conclusion

In this paper, we propose a novel Table-Transformer (T-T) approach, which uses enhanced transformer layers as the relation encoding module for the table-based ASTE task. To address the challenges posed by overly long table sequences and hard local attention interaction, we respectively propose our stripe attention mechanism and the loop-shift strategy. The former reduces attention computation costs by focusing on local tokens within the 2D table space, while the latter facilitates interaction between attention windows through loop-shift operations. Extensive experiments on four datasets demonstrate the effectiveness of our T-T over the best baselines. In future work, we aim to adapt this method to a broader range of information extraction tasks, such as relation extraction and event extraction.

Acknowledgments

This research is supported by the National Key R&D Program of China (No. 2023YFC3303800), NSFC through grants 62322202 and 62441612, "Pioneer and Leading Goose R&D Program of Zhejiang" through grant 2025C02044, National Key Laboratory under grant 241-HF-D07-01, and Hebei Natural Science Foundation through grant F2024210008.

References

[Beltagy *et al.*, 2020] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[Chen *et al.*, 2021] Zhexue Chen, Hong Huang, Bang Liu, Xuanhua Shi, and Hai Jin. Semantic and syntactic enhanced aspect sentiment triplet extraction. In *Findings of ACL-IJCNLP 2021*, pages 1474–1483, Online, August 2021.

[Chen *et al.*, 2022] Hao Chen, Zepeng Zhai, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In *ACL 2022*, pages 2974–2985, 2022.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *NAACL 2019*, pages 4171–4186, Minneapolis, Minnesota, June 2019.

[Dozat and Manning, 2017] Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*, 2017.

[Gao *et al.*, 2022] Tianhao Gao, Jun Fang, Hanyu Liu, Zhiyuan Liu, Chao Liu, Pengzhang Liu, Yongjun Bao, and Weipeng Yan. LEGO-ABSA: A prompt-based task assemblable unified generative framework for multi-task aspect-based sentiment analysis. In *ICCL*, pages 7002–7012, Gyeongju, Republic of Korea, October 2022.

[Gou *et al.*, 2023] Zhibin Gou, Qingyan Guo, and Yujiu Yang. MvP: Multi-view prompting improves aspect sentiment tuple prediction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *ACL*, pages 4380–4397, Toronto, Canada, July 2023.

[Han *et al.*, 2023] Dongchen Han, Xuran Pan, Yizeng Han, Shiji Song, and Gao Huang. Flatten transformer: Vision transformer using focused linear attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5961–5971, 2023.

[Jing *et al.*, 2021] Hongjiang Jing, Zuchao Li, Hai Zhao, and Shu Jiang. Seeking common but distinguishing difference, a joint aspect-based sentiment analysis model. In *EMNLP*, 2021.

[Katharopoulos *et al.*, 2020] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[Li *et al.*, 2023a] Dongxu Li, Zhihao Yang, Yuquan Lan, Yunqi Zhang, Hui Zhao, and Gang Zhao. Simple approach for aspect sentiment triplet extraction using span-based segment tagging and dual extractors. In *SIGIR*, SIGIR '23, page 2374–2378, New York, NY, USA, 2023. Association for Computing Machinery.

[Li *et al.*, 2023b] Pan Li, Ping Li, and Kai Zhang. Dual-channel span for aspect sentiment triplet extraction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *EMNLP*, pages 248–261, Singapore, December 2023. Association for Computational Linguistics.

[Liang *et al.*, 2023a] Shuo Liang, Wei Wei, Xian ling Mao, Yuanyuan Fu, Rui Fang, and Dangyang Chen. Stage: Span tagging and greedy inference scheme for aspect sentiment triplet extraction. In *AAAI 2023*, 2023.

[Liang *et al.*, 2023b] Shuo Liang, Wei Wei, Xian-Ling Mao, Yuanyuan Fu, Rui Fang, and Dangyang Chen. Stage: span tagging and greedy inference scheme for aspect sentiment triplet extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 13174–13182, 2023.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV 2021*, pages 9992–10002. IEEE, 2021.

[Mao *et al.*, 2021] Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. A joint training dual-mrc framework for aspect based sentiment analysis. In *AAAI 2021*, pages 13543–13551, 2021.

[Mukherjee *et al.*, 2023] Rajdeep Mukherjee, Nithish Kannen, Saurabh Pandey, and Pawan Goyal. CONTRASTE: Supervised contrastive pre-training with aspect-based prompts for aspect sentiment triplet extraction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of EMNLP 2023*, pages 12065–12080, Singapore, December 2023. Association for Computational Linguistics.

[Ning *et al.*, 2023] Jinzhong Ning, Zhihao Yang, Yuanyuan Sun, Zhizheng Wang, and Hongfei Lin. OD-RTE: A one-stage object detection framework for relational triple extraction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *ACL*, pages 11120–11135, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Ouyang *et al.*, 2022] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[Peng *et al.*, 2020] Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *AAAI*, pages 8600–8607, 2020.

[Peng *et al.*, 2024a] Kun Peng, Lei Jiang, Qian Li, Haoran Li, Xiaoyan Yu, Li Sun, Shuo Sun, Yanxian Bi, and Hao

Peng. Table-filling via mean teacher for cross-domain aspect sentiment triplet extraction. In *CIKM 2024*, pages 1888–1898, 2024.

[Peng *et al.*, 2024b] Kun Peng, Lei Jiang, Hao Peng, Rui Liu, Zhengtao Yu, Jiaqian Ren, Zhifeng Hao, and Philip S Yu. Prompt based tri-channel graph convolution neural network for aspect sentiment triplet extraction. In *SDM 2024*, pages 145–153, 2024.

[Pontiki *et al.*, 2014] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35. Association for Computational Linguistics, August 2014.

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

[Pontiki *et al.*, 2016] 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. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, June 2016.

[Sun *et al.*, 2024] Qiao Sun, Liujia Yang, Minghao Ma, Nanyang Ye, and Qinying Gu. MiniConGTS: A near ultimate minimalist contrastive grid tagging scheme for aspect sentiment triplet extraction. In *EMNLP*, pages 2817–2834, 2024.

[Tay *et al.*, 2022] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2020] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[Wang *et al.*, 2023] Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. Is chatgpt a good sentiment analyzer? a preliminary study. *ArXiv*, abs/2304.04339, 2023.

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

[Xianlong *et al.*, 2023] Luo Xianlong, Meng Yang, and Yihao Wang. Tagging-assisted generation model with encoder and decoder supervision for aspect sentiment triplet extraction. In *EMNLP*, pages 2078–2093, 2023.

[Xu *et al.*, 2020] Lu Xu, Hao Li, Wei Lu, and Lidong Bing. Position-aware tagging for aspect sentiment triplet extraction. In *EMNLP*, pages 2339–2349, 2020.

[Xu *et al.*, 2021] Lu Xu, Yew Ken Chia, and Lidong Bing. Learning span-level interactions for aspect sentiment triplet extraction. In *ACL-IJCNLP*, pages 4755–4766, 2021.

[Zaheer *et al.*, 2020] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020.

[Zhai *et al.*, 2022] Zepeng Zhai, Hao Chen, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. COM-MRC: A COntext-masked machine reading comprehension framework for aspect sentiment triplet extraction. In *EMNLP 2022*, 2022.

[Zhang *et al.*, 2021] Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. Towards generative aspect-based sentiment analysis. In *ACL-IJCNLP*, pages 504–510, 2021.

[Zhang *et al.*, 2022a] Wenxuan Zhang, Xin Li, Yang Deng, and et al. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Zhang *et al.*, 2022b] Yice Zhang, Yifan Yang, Yihui Li, and et al. Boundary-driven table-filling for aspect sentiment triplet extraction. In *EMNLP*, pages 6485–6498, 2022.

[Zhang *et al.*, 2023] Qingru Zhang, Dhananjay Ram, Cole Hawkins, Sheng Zha, and Tuo Zhao. Efficient long-range transformers: You need to attend more, but not necessarily at every layer. In *Findings of EMNLP 2023*, pages 2775–2786, December 2023.

[Zhang *et al.*, 2024] Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. Sentiment analysis in the era of large language models: A reality check. In *Findings of NAACL 2024*, pages 3881–3906, June 2024.

[Zhao *et al.*, 2024] Xiaowei Zhao, Yong Zhou, and Xiujuan Xu. Dual encoder: Exploiting the potential of syntactic and semantic for aspect sentiment triplet extraction. In *CoNLL*, 2024.

[Zou *et al.*, 2024] Wang Zou, Wubo Zhang, Wenhuan Wu, and Zhuoyan Tian. A multi-task shared cascade learning for aspect sentiment triplet extraction using BERT-MRC. *Cogn. Comput.*, 16(4):1554–1571, 2024.