Syntactic-Guided Semantic Attention Matrix Enhancement for Aspect Sentiment Triplet Extraction

Anonymous ACL submission

Abstract

Aspect Sentiment Triplet Extraction (ASTE) is a fine-grained task in Aspect-Based Sentiment Analysis (ABSA), aiming to extract triplets of aspect terms, their corresponding opinion terms, and sentiment orientation. Recent studies have employed semantic and syntactic features to enhance extraction performance. However, as the extraction of both syntactic and semantic information is not task-specific, such information does not target aspect-opinion pairs within sentences, thereby potentially introducing noise unrelated to the task. To address this issue, we propose a *Syntactic-Guided Semantic Attention Matrix Enhancement* (SG-SAME) framework. First, we use a multi-view masking mechanism and a biaffine-based gating module to derive a task-aligned shared matrix from the existing semantic and syntactic matrices. Second, the shared matrix is injected into the attention score matrix of LLM layers to enhance the model's representation learning. Finally, we propose a chain-of-thought (CoT) framework with negative candidate injection to steer the model toward autonomous reasoning and error self-rectification. Experimental results on multiple benchmark datasets demonstrate that our method outperforms existing state-of-the-art approaches in ASTE tasks.

1 Introduction

Aspect Sentiment Triplet Extraction (ASTE) is an emerging fine-grained information extraction task that aims to extract aspect terms, opinion terms, and their corresponding sentiment polarities as triplets from review texts. This enables a more comprehensive understanding of users' emotional expression intents (Peng et al., 2020). ASTE is widely applied in fine-grained opinion mining across various text types, including user reviews and social media posts (Zhang et al., 2023), and categorizes

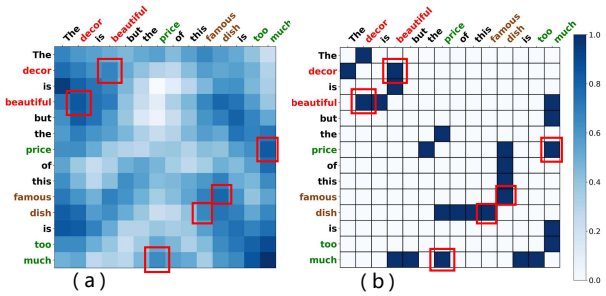

Figure 1: Visualization heatmaps of the semantic matrix (a) and the symmetrized raw syntactic matrix (b). Red boxes highlight the associations between aspects and opinions, where tokens in the same font color belong to the same sentiment triplet.

sentiment polarity into three classes: Positive, Neutral, and Negative. In this task, accurately associating aspect terms with opinion terms is the cornerstone of extracting sentiment triplets. Since Aspect-Opinion (A-O) pairs enable the model to capture fine-grained triplet relationships more effectively, reinforcing their mutual attention intensity within the attention mechanism remains a pivotal challenge for enhancing ASTE performance.

Limitations of Semantic Attention Mechanisms In many ASTE and other aspect-based sentiment analysis (ABSA) tasks, a prevailing approach is to leverage attention mechanisms to capture semantic information. However, attention mechanisms are often vulnerable to the interference of prominent sentiment words. Figure 1 (a) shows the semantic attention heatmap for: *"The decor is beautiful, but the price of this famous dish is just a bit too much."* It can be observed that, in addition to the crucial aspect–opinion (A-O) pairs, numerous task-irrelevant high-weight regions are also present. This phenomenon arises because high-frequency words with strong sentiment polarity consistently dominate the optimization of hidden states, thereby yielding a larger vector magnitude

Code will be available.

within the embedding space. During the forward propagation of the attention mechanism, the large vector magnitude erroneously propagates spurious attention to other irrelevant tokens through dot-product and softmax operations (Gao et al., 2019; Oyama et al., 2023). For instance, *dish* may falsely assign excessive weight to *beautiful*, which easily leads to the generation of erroneous triplets. This indicates that relying solely on semantic encoders is insufficient to effectively handle this task (Chen et al., 2022a; Zhang et al., 2022). Therefore, effectively optimizing the semantic attention matrix—specifically by strengthening task-relevant A-O connections—remains a critical challenge.

Limitations of Syntactic Dependency Representations Many studies employ dependency parsers to obtain syntactic dependency representations, aiming to compensate for the limitations of semantic encoders (Li et al., 2023b; Zhang et al., 2022; Zhao et al., 2024a; Liu et al., 2025). However, such approaches heavily depend on the generated dependency trees, which may introduce task-irrelevant connections, as syntactic parsers are designed to capture linguistic structures rather than task-specific sentiment features. As illustrated in Figure 1(b). According to the Universal Dependencies (UD) v2 standard (Nivre et al., 2020), coordination is handled by linking all subsequent conjuncts to the first one via the $conj$ relation, thereby establishing a task-irrelevant connection between *beautiful* and *much*. Simultaneously, under the UD principle of prioritizing content words, a task-irrelevant connects is established between the nouns *price* and *dish* in the phrase *"price of this famous dish"*. Integrating a syntactic matrix with such connections will inevitably introduce noise, as comprehensively analyzed in Appendix E. Therefore, how to distill the syntactic matrix to focus exclusively on core A-O associations is the key to boosting ASTE performance.

Contemporary applications of LLMs in sentiment analysis include directly injecting syntactic information into the embedding layer (Liu et al., 2025), leveraging LLMs as judges (Bodke et al., 2025), and employing ensemble voting mechanisms (Sheng et al., 2025). However, these methods generally exhibit insufficient task adaptation and limited instruction-following capabilities for this task. Therefore, exploring advanced prompting strategies, such as Chain-of-Thought (CoT), offers a promising avenue to effectively alleviate these shortcomings and enhance LLM performance on this task.

As stated above, both semantic attention and syntactic dependency representations exhibit inherent limitations. We observe that target aspect-opinion (A-O) pairs tend to consistently co-occur in both the syntactic and semantic matrices, whereas task-irrelevant links in one view are typically absent in the other (refer to Appendix G for detailed explanation). Therefore, we construct the shared matrix by referencing the intersection of these two views to serve as highly task-relevant connections.

Specifically, a novel **S**yntactic-**G**uided **S**emantic **A**ttention **M**atrix **E**nhancement (SG-SAME) framework is proposed. We first derive semantic and syntactic matrices using attention mechanisms and external parsers, respectively. Subsequently, a multi-view masking mechanism and a biaffine-based gating module are employed to extract shared components from these views, constructing a shared matrix. This matrix is then injected into the hidden representations of the LLM to facilitate syntactic-guided representation learning. Finally, we design a chain-of-thought (CoT) framework with negative candidate injection, leveraging a four-step reasoning process to enhance the model's capability for discriminating and rectifying complex sentiment relations.

To summarize, our main contributions include the following:

- We propose a shared matrix acquisition strategy, which uses a multi-view masking mechanism and a biaffine-based adaptive gating module to extract task-salient shared aspect-opinion pair matrices from existing semantic and syntactic matrices.

- We propose injecting the shared task-salient matrix into LLMs by adding it to the original attention score matrix of LLM layers, thereby achieving a precise enhancement of the model's hidden representations.

- We design a CoT framework incorporating negative candidate injection, which steers the model to perform self-correction via a four-step reasoning process.

- Extensive experiments on four benchmark datasets demonstrate that our model consistently achieves state-of-the-art performance.

2 Related Work

PLM-based Methods for ASTE As a key fine-grained ABSA subtask (Pontiki et al., 2014a), ASTE was initiated by Peng et al. (2020) via a two-stage pipeline. Span-ASTE (Xu et al., 2021) classifies all potential text spans, establishing span-based methods as the mainstream paradigm among PLM-based frameworks. To further utilize syntactic information within these frameworks, Chen et al. (2022b) inject distance into dependency trees, Li et al. (2023b) filter noisy spans via syntactic relations, and Zhang et al. (2022) constrain syntax-semantics distance using KL divergence. Meanwhile, Zhao et al. (2024a) integrate syntactic structures into grid tags. Although these methods successfully inject syntactic connections into semantic information via mechanisms such as Graph Convolutional Networks (GCNs), they largely overlook the deep, intrinsic connections between syntax and semantics.

LLM-based Methods for ASTE In LLM-based sentiment analysis, existing methods primarily focus on feature enhancement and post-processing. For instance, Liu et al. (2025) inject dependency tree representations into embedding layers via GCN, while Bodke et al. (2025) deploy GPT as a critic to filter preliminary tuples. Meanwhile, Sheng et al. (2025) enhance robustness via multilingual data conversion and multi-vote mechanisms. However, LLM architecture research dedicated exclusively to ASTE remains scarce, mostly limited to isolated prompt exploration or local component optimization. Therefore, exploring a more comprehensive LLM reasoning workflow presents an effective path to fully unleash the potential of LLMs on this task.

3 Method

In this section, the overall architecture of our proposed model SG-SAME is shown in Figure 2.

3.1 Task Definition

Given an input sentence $S = \{w_1, w_2, \ldots, w_n\}$, the objective of Aspect Sentiment Triplet Extraction (ASTE) is to identify a set of triplets $\mathcal{T} = \{(a_i, o_i, y_i)\}_{i=1}^{m}$, where $m \geq 0$ denotes the total number of triplets in S. Here, $a_i = (s_a, e_a)$ and $o_i = (s_o, e_o)$ denote the continuous text spans of the aspect and opinion terms, respectively, where $1 \leq s \leq e \leq n$. The term $y_i \in \{\text{POS}, \text{NEU}, \text{NEG}\}$ represents the sentiment polarity expressed by the opinion o_i toward the aspect a_i.

3.2 Syntactic and Semantic Matrix

Syntactic Matrix To extract the foundational syntactic layout, we employ CoreNLP (Manning et al., 2014) to parse each sentence into a directed dependency graph. This graph is mathematically characterized by a directed adjacency matrix $\tilde{\mathbf{A}}_{\text{syn}} \in \mathbb{R}^{B\times n\times n}$, where the unweighted, asymmetric connection patterns from a governor token w_i to a dependent token w_j are defined as:

$$\tilde{\mathbf{A}}_{\text{syn},ij} = \begin{cases} 1, & \text{if } w_i \xrightarrow{\text{dep}} w_j, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here, $\tilde{\mathbf{A}}_{\text{syn},ij}$ denotes the directed dependency edge from the governor token w_i to the dependent token w_j.

Semantic Matrix Given initial hidden representations $\mathbf{H} \in \mathbb{R}^{B\times n\times d}$ encoded by Llama-3.2-1B, we apply a multi-head self-attention mechanism to construct the raw semantic matrix $\tilde{\mathbf{A}}_{\text{sem}} \in \mathbb{R}^{B\times n\times n}$:

$$\tilde{\mathbf{A}}_{\text{sem}} = \text{Softmax}\left(\frac{(\mathbf{H}\mathbf{W}_Q)(\mathbf{H}\mathbf{W}_K)^\top}{\sqrt{d_k}}\right) \tag{2}$$

To balance weight scale and eliminate extreme attention values, we perform logarithmic scaling followed by min-max normalization:

$$\mathbf{A}_{\text{sem},ij} = \frac{\log\left(\tilde{\mathbf{A}}_{\text{sem},ij} + \varepsilon\right) - s_{\min}}{s_{\max} - s_{\min} + \delta} \tag{3}$$

where $s_{\min}$ and $s_{\max}$ are the minimum and maximum values in $\log(\tilde{\mathbf{A}}_{\text{sem}} + \varepsilon)$, respectively. ε is a smoothing factor, and δ is a tiny constant avoiding division by zero, scaling $\mathbf{A}_{\text{sem},ij}$ into $[0, 1]$.

3.3 Shared Matrix

This subsection aims to construct a shared matrix $\mathbf{A}_{sha}$ by identifying the intersection of syntactic and semantic views. This matrix captures high-confidence associations where the two views reach a consensus, thereby extracting core connections that are highly relevant to the ASTE task.

Syntactic Gating and Symmetrization The original syntactic matrix is a directional 0-1 matrix and is not aligned with the semantic matrix.

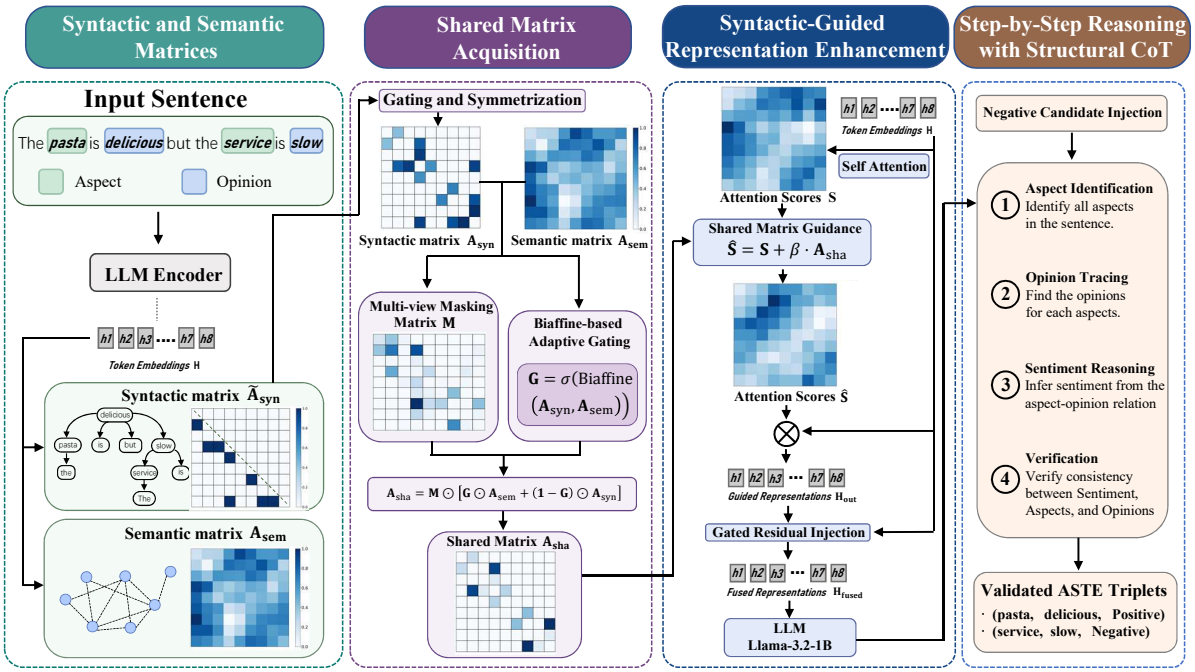

Figure 2: The overall framework of SG-SAME. The framework of our proposed model is divided into four distinct sections from left to right, where all illustrated matrices represent the heatmap visualizations of our actual step-by-step computational processes. Here, β denotes the coefficient that controls the guidance intensity of the shared matrix.

We observe that aspects tokens and opinions tokens exhibit denser connections in the dependency tree, meaning their average connection strength naturally reflects node importance. Therefore, We introduce a learnable weight to dynamically scale the original syntactic matrix, forcing the model to focus on vital connections and align with semantic features.

We obtain the average connection strength of each node by performing row-wise mean pooling, and then apply an adaptive mapping to produce node-level gating coefficients for the strength of the edge. Finally, we symmetrize and obtain the bidirectional connections.

$$g_i = \sigma\left(w_g \cdot \frac{1}{L}\sum_{j=1}^{L} \tilde{\mathbf{A}}_{\text{syn},ij} + b_g\right) \tag{4}$$

$$\hat{\mathbf{A}}_{\text{syn},i:} = \tilde{\mathbf{A}}_{\text{syn},i:} \cdot g_i \tag{5}$$

$$\mathbf{A}_{\text{syn}} = \hat{\mathbf{A}}_{\text{syn}} + \hat{\mathbf{A}}_{\text{syn}}^{\top} \tag{6}$$

where i and j denote the head and tail nodes in a directed dependency link, respectively, and i : represents all tail nodes connected from the head i.

Multi-view Masking Inspired by the Laplace kernel (Williams, 2003), we define an exponential multi-view mask $\mathbf{M} \in \mathbb{R}^{B\times n\times n}$ to quantify the connection consistency between the matrix $\mathbf{A}_{\text{syn}}$ and $\mathbf{A}_{\text{sem}}$. For any token pair (i, j), the mask is calculated as:

$$\mathbf{M}_{ij} = \exp\left(-\frac{\left|\mathbf{A}_{\text{syn},ij} - \mathbf{A}_{\text{sem},ij}\right|}{\tau}\right) \tag{7}$$

where τ is a learnable temperature scaling parameter, initialized to 0.1, which is constrained to be positive during training via a lower-bound clipping operation (i.e., $\tau \leftarrow \max(\tau, 10^{-4})$). This mask ensures that only connections corroborated by both perspectives are retained with high confidence.

Biaffine-based Adaptive Gating To simultaneously account for the global connections and mutual interactions of both views during fusion, we introduce a biaffine gating mechanism to adaptively regulate the contribution of each view. The gating matrix $\mathbf{G}$ is computed as:

$$\begin{aligned}\mathbf{G} = \sigma\Big(&\mathbf{A}_{\text{syn}}^{\top}\mathbf{W}_{\text{bi}}\mathbf{A}_{\text{sem}} \\ &+ \mathbf{U}_{\text{bi}}[\mathbf{A}_{\text{syn}}; \mathbf{A}_{\text{sem}}] + \mathbf{b}_{\text{bi}}\Big)\end{aligned} \tag{8}$$

where $\mathbf{W}_{\text{bi}}$ and $\mathbf{U}_{\text{bi}}$ are learnable weight matrices, $\mathbf{b}_{\text{bi}}$ is a bias vector, and $[\mathbf{A}_{\text{syn}}; \mathbf{A}_{\text{sem}}] \in \mathbb{R}^{B\times n\times n\times 2}$ denotes the concatenation along the

last dimension. Then the final shared matrix $\mathbf{A}_{\text{sha}}$ is synthesized by applying the multi-view mask to the gated fusion of the two views:

$$\mathbf{A}_{\text{sha}} = \mathbf{M} \odot \left[\mathbf{G} \odot \mathbf{A}_{\text{sem}} + (\mathbf{1} - \mathbf{G}) \odot \mathbf{A}_{\text{syn}}\right] \quad (9)$$

where $\odot$ denotes the Hadamard product.

3.4 Syntactic-Guided Representation Alignment

We integrate the shared matrix as a relational bias to reshape the attention score distribution and obtain the representations. Subsequently, the refined representations $\mathbf{H}_{\text{out}}$ are injected into the original embeddings.

Adaptive Representation Learning We first compute the semantic attention matrix for the original embeddings $\mathbf{H}$. Subsequently, to dynamically evaluate the guidance intensity of the shared matrix for each token acting as the query, we compute the contribution coefficient β via a linear projection over its representation $\mathbf{h}_i \in \mathbb{R}^D$.

$$\mathbf{S} = \frac{(\mathbf{H}\mathbf{W}_Q)(\mathbf{H}\mathbf{W}_K)^\top}{\sqrt{d_k}} \quad (10)$$

$$\beta_i = \sigma\left(\mathbf{w}_\beta^\top \mathbf{h}_i + b_\beta\right) \quad (11)$$

The guided representations $\mathbf{H}_{\text{out}}$ are obtained via a weighted summation of the value vectors based on the updated attention distribution:

$$\hat{\mathbf{S}}_{ij} = \mathbf{S}_{ij} + \beta_i \cdot \mathbf{A}_{\text{sha},ij} \quad (12)$$

$$\mathbf{H}_{\text{out}} = \text{softmax}(\hat{\mathbf{S}})\mathbf{H}\mathbf{W}_V \quad (13)$$

Gated Residual Injection We facilitate the interaction between the original LLM embeddings $\mathbf{H}$ and $\mathbf{H}_{\text{out}}$, subsequently generating an adaptive gating weight to modulate the fusion of them.

$$\boldsymbol{\lambda} = \sigma\left(\mathbf{W}_\lambda[\mathbf{H}; \mathbf{H}_{\text{out}}] + \mathbf{b}_\lambda\right) \quad (14)$$

$$\mathbf{H}_{\text{mix}} = \boldsymbol{\lambda} \odot \mathbf{H}_{\text{out}} + (\mathbf{1} - \boldsymbol{\lambda}) \odot \mathbf{H} \quad (15)$$

where $[\cdot;\cdot]$ denotes the feature-level concatenation. To preserve the LLM's pre-trained linguistic proficiency while ensuring training stability, $\mathbf{H}_{\text{mix}}$ is isolated as a residual increment, which is directly injected into the original hidden states to derive the final augmented representations $\mathbf{H}_{\text{fused}}$:

$$\mathbf{H}_{\text{fused}} = \mathbf{H} + \alpha\left[\text{LayerNorm}(\mathbf{H}_{\text{mix}})\right] \quad (16)$$

where α is a learnable intensity parameter, scaling the contribution of the shared connections. Finally, we perform an in-place update by replacing the original embedding with the fused representations.

3.5 Step-by-Step Reasoning with CoT

To address the limitation of lacking an explicit reasoning process and circumvent the cumbersome workflow of multi-turn model invocations, we propose a Chain-of-Thought (CoT) framework. Recognizing that the vanilla form of CoT, which directly allows the LLM to indiscriminately output the reasoning process, may decrease the performance (Zhang et al., 2024), we decompose the reasoning process into four steps with predefined output objectives for each. The prompts used for CoT data generation are detailed in Appendix I.

- **Step 1 (Aspect Identification):** The model first identifies explicit aspect terms $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ mentioned in the sentence.
- **Step 2 (Opinion Tracing):** For each identified aspect $a_i \in \mathcal{A}$, the model traces its specific corresponding opinion expressions $\mathcal{O}_{=}\{o_{i,1}, o_{i,2}, \ldots\}$. This stage focuses on establishing potential aspect-opinion pairs $(a_i, o_{i,j})$. It is worth noting that a single aspect may correspond to multiple opinions.
- **Step 3 (Sentiment Reasoning):** For each established pair $(a_i, o_{i,j})$, the model infers the sentiment polarity $S \in \{\text{positive}, \text{neutral}, \text{negative}\}$ by analyzing the semantic orientation of the opinion phrase within the given context.
- **Step 4 (Verification and Filtering):** The final step summarizes the reasoning from the previous steps and performs a reverse verification on the candidate triplets $(a_i, o_{i,j}, s_{i,j})$. It specifically verifies the presence of the sentiment,as well as the target aspect and specific opinion source corresponding to that sentiment, concluding with the triplet itself paired with a "yes" or "no" decision.

3.6 Contrastive Chain-of-Thought Tuning Data

Since standard CoT data derived from gold triplets always results in a "yes" decision during verification, the model may overfit during the fine-tuning process. To counter this, we introduce Contrastive CoT Tuning Data constructed through rigorous rule-based mechanisms. In addition, we ensure that the number of positive and negative samples is balanced.

These negative candidates are designed to simulate common model failures via three mechanisms:

- **Wrong Pairing:** An aspect a_i is paired with an opinion o_j from a different triplet in the same sentence, testing the model's ability to identify precise modification relations.

- **Sentiment Reversal:** A valid aspect-opinion pair is assigned an incorrect sentiment label (e.g., swapping *positive* with *negative*), forcing the model to re-examine the context.

- **Irrelevant Phrase:** An aspect is paired with a non-opinion span randomly sampled from the sentence, ensuring the model distinguishes between emotional and neutral descriptions.

During Step 4, the model is trained to identify negative candidates as invalid and discard them while providing explicit justifications. Consequently, the final `[Result]` block is supervised to yield only validated gold triplets, forcing the model to develop self-correction capabilities. We present the detailed generation process and prompts in Appendix I.

Training and Optimization We perform **full-parameter fine-tuning** on the Llama-3.2-1B backbone. To integrate guided representations, we perform an in-place replacement of the LLM's original hidden states of the input sentence with our augmented representations $\mathbf{H}_{\text{fused}}$, while keeping the prompt's representations $\mathbf{H}_{\text{prompt}}$ unchanged. The model is optimized using a causal language modeling objective to minimize the negative log-likelihood of the target sequence Y (comprising reasoning paths and gold triplets):

$$\mathcal{L} = -\sum_{t=1}^{|Y|} \log P(y_t \mid y_{<t}, [\mathbf{H}_{\text{fused}}; \mathbf{H}_{\text{prompt}}]) \tag{17}$$

where $[\cdot;\cdot]$ denotes the sequence-level concatenation of the augmented sentence and prompt embeddings.

4 Experiments

4.1 Datasets

To validate the effectiveness of our proposed SG-SAME framework, we conduct experiments on the ASTE-Data-V2 dataset, which covers the restaurant and laptop domains and includes four benchmark sub-datasets. These sub-datasets are derived from three ABSA SemEval Challenges (Pontiki et al., 2014b, 2015; Álvarez-López et al., 2016; Peng et al., 2020) annotated the opinion terms following the method proposed by Fan et al. (2019). Subsequently, Xu et al. (2020) remove conflicting sentiment triplets and release the final version of ASTE-Data-V2, which has been widely adopted. The detailed information of this dataset can be found in **Appendix A**.

4.2 Baselines

We compare the proposed model with several leading baseline methods, including large language models (LLMs) and PLM-based approaches utilizing state-of-the-art span classification methods.

For completeness, detailed descriptions of the baselines are provided in **Appendix B**.

4.3 Implementation

We conducted full-parameter fine-tuning on the Llama-3.2-1B-Instruct model using the SFTTrainer from the Hugging Face TRL library. All experiments were executed on NVIDIA A100 GPUs (80GB). For optimization, we adopted the AdamW optimizer with a learning rate of 5×10^{-6}, a weight decay of 0.01, and a linear warmup ratio of 0.1. The maximum training duration was set to 6 epochs with a per-device batch size of 2 and gradient accumulation steps of 4, yielding an effective batch size of 8. To closely monitor the training trajectory, the model was evaluated at an interval of 50 optimization steps. Based on the validation F1-score, an early stopping mechanism was employed with a patience of 3 epochs to mitigate overfitting and improve efficiency. α is initialized to 0.1. Detailed parameter settings for the ablation experiments are provided in Appendix D. To ensure reproducibility and eliminate statistical variance, all reported macro metrics in Table 1 are averaged over 3 independent runs using different random seeds.

5 Results and Analysis

5.1 Main Results

The main experimental results are reported in Table 1.

We compare our approach with previous studies using F1 score as the primary evaluation metric on four benchmark datasets (i.e., `REST14`, `LAP14`, `REST15`, `REST16`), and report the results in **Table 1**. The proposed SG-SAME model achieves state-of-the-art results on all datasets compared to existing methods. It outperforms the SOTA results

Model	14lap			14res			15res			16res		
	P	R	F	P	R	F	P	R	F	P	R	F
PLM-Based Span Classifier												
Span-ASTE (Xu et al., 2021)	63.44	55.84	59.38	72.89	70.89	71.85	62.18	64.45	63.27	69.45	71.17	70.26
S^3E^2 (Chen et al., 2021)†	59.43	46.23	52.01	69.08	64.55	66.74	61.06	56.44	58.66	71.08	63.13	66.87
EMC-GCN(Chen et al., 2022b)†	61.70	56.26	58.81	71.21	72.39	71.78	61.54	62.47	61.93	65.62	71.30	68.33
SimSTAR (Li et al., 2023a)	66.46	58.23	62.07	76.23	71.63	73.86	71.71	59.59	65.09	72.02	74.12	73.06
STAGE-3D (Liang et al., 2023)	71.98	53.86	61.58	78.58	69.58	73.76	73.63	57.90	64.79	76.67	70.12	73.24
Pairing (Yang et al., 2023)	–	–	61.68	–	–	72.53	–	–	62.78	–	–	71.38
EPISA (Naglik and Lango, 2023)	66.98	60.55	63.56	75.29	72.56	73.89	66.44	64.74	65.54	71.12	72.45	71.77
Dual-Span (Li et al., 2023b)†	67.14	62.13	64.49	77.01	74.00	75.47	67.97	66.34	67.13	73.56	73.48	73.49
D2E2S (Zhao et al., 2024b)†	67.38	60.31	63.65	75.92	74.36	75.13	70.09	62.11	65.03	73.45	70.12	71.73
ASTE-Transformer (Naglik and Lango, 2024)	65.56	60.36	62.83	74.51	76.05	75.27	67.94	67.91	67.89	74.96	74.27	74.61
MiniConGTS (Sun et al., 2024)	66.82	60.68	63.61	76.10	75.08	75.59	66.50	63.86	65.15	75.52	74.14	74.83
BTF-CCL (Li et al., 2025)	70.66	57.31	63.29	80.41	71.83	75.88	71.66	64.12	67.68	72.56	75.10	73.80
LLM-based framework												
GPT-3.5-turbo zero-shot	26.21	40.69	31.88	39.21	56.17	46.18	31.21	52.75	39.21	35.28	59.64	44.34
GPT-3.5-turbo few-shots	39.79	50.09	44.35	51.51	65.19	57.55	43.34	63.09	51.39	51.12	71.01	59.45
GPT-4o zero-shot	17.81	22.55	19.90	32.99	38.13	35.37	27.85	37.73	32.05	32.17	43.00	36.80
GPT-4o few-shots	38.23	48.61	42.80	54.11	66.20	59.55	45.57	60.41	51.95	52.90	71.01	60.63
Instruct-ABSA (Scaria et al., 2024)	–	–	61.86	–	–	71.17	–	–	60.63	–	–	70.72
PASTEL (Bodke et al., 2025)	67.83	62.61	65.22	81.95	79.81	80.87	72.54	71.18	71.86	80.89	77.43	79.12
SG-SAME (DeBERTa-v3-large) *	71.13	66.32	68.64	77.20	79.31	78.24	72.53	68.60	70.51	79.03	76.43	77.71
SG-SAME	71.80	67.81	**69.75**	80.19	84.71	**82.39**	72.13	74.01	**73.06**	79.29	80.58	**79.93**

Table 1: Experimental results comparison on four benchmark datasets. The best results according to F1-score are bolded, and the second-best results are underlined. For LLM-based methods, both PASTEL (Bodke et al., 2025) and our S3P are fully fine-tuned on the Llama-3.2-1B backbone. The symbol † indicates that the model is a syntax-based method. The symbol (*) denotes our framework integrated with the DeBERTa backbone.

Model	14lap	14res	15res	16res
Ours	**69.75**	**82.39**	**73.06**	**79.93**
Shared Matrix Module				
w/o Matrix Guidance	66.12	78.35	70.82	76.15
w/o Multi-view Masking	67.84	80.12	72.25	76.95
Biaffine → Avg	68.91	81.45	71.88	79.10
Shared → Syntactic	66.58	79.84	71.45	76.42
COT				
w/o Contrastive Data	64.37	77.25	69.08	74.91
w/o COT	66.84	79.61	70.95	77.39
w/o Reason	68.52	80.79	72.33	79.06

Table 2: F1 scores of ablation results of our proposed framework on four benchmark datasets. "w/o" denotes the removal of a specific component. The arrow "→" denotes the replacement of a component.

Model	Tuning	14lap	14res	15res	16res
Pre-trained Language Models (PLMs)					
BERT-Large-Uncased	Full	64.25	76.11	66.42	75.47
RoBERTa-Large†	Full	67.62	77.02	68.51	76.68
Large Language Models (LLMs)					
Qwen3-8B-Instruct	LoRA	65.95	75.75	67.09	73.81
Llama-3.1-8B-Instruct	LoRA	66.31	76.02	66.28	74.19
SG-SAME (Llama-3.2-1B)	**Full**	**69.75**	**82.39**	**73.06**	**79.93**

Table 3: F1 scores of performance comparison across different backbones (PLMs vs. LLMs) and tuning paradigms. † denotes that detailed analysis and extended evaluations for this model are provided in Appendix C.

by +4.53%, +1.52%, +1.2%, and +0.81% in F1 score, respectively.

Compared to syntax-based baselines (S^3E^2 (Chen et al., 2021), EMC-GCN (Chen et al., 2022b), Dual-Span (Li et al., 2023b), D2E2S (Zhao et al., 2024b)) and state-of-the-art PLM-based span methods, our model achieves the best results, outperforming the top PLM baseline by 5.51% in average F1-score. Our framework also yields competitive results when deployed on DeBERTa-v3-large. These prove that our model successfully unleashes the potential of LLMs and seamlessly adapts to PLMs.

In addition, our performance achieves the best results among LLM-based models, outperforming PASTEL (Bodke et al., 2025), which is also based on the Llama-1B model. Compared with it, we find that our model's recall achieves an average improvement of 4.02% across the four datasets, exceeding the 2.02% improvement in F1-score. This indicates that our proposed framework, which utilizes shared matrix guidance and performs four-step reasoning, can extract more comprehensive results and reduce the omission of triplets.

5.2 Effect of Different Modules

To verify the effectiveness of each component in our framework, we conduct ablation experiments. The results are summarized in Table 2.

Effect of Shared Matrix Module This section provides an analysis of the effectiveness of the Shared Matrix Module. (1) Removing the entire

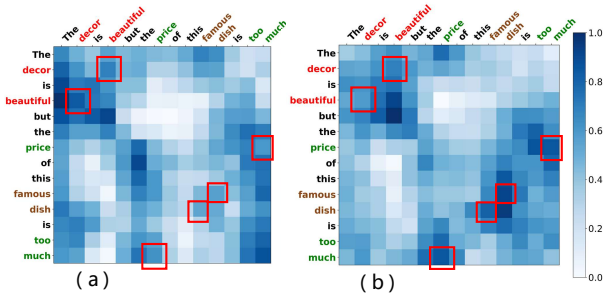

Figure 3: Visualization of attention scores between aspects and opinions (highlighted in red boxes). Subfigure (a) illustrates the vanilla attention scores $\mathbf{S}$ from Eq. (10), while (b) shows the attention scores $\hat{\mathbf{S}}$ as per Eq. (12). For better contrast, both matrices are Min-Max normalized (3).

matrix guidance (*w/o Matrix Guidance*) causes the sharpest performance drop ($\Delta F1 \approx 2.24 \sim 4.04\%$), proving that a shared matrix effectively enhances task-relevant attention. (2) The decline in *w/o Multi-View Masking* demonstrates that the semantic-syntactic shared matrix is crucial for steering attention toward task-specific objectives. (3) The superiority of Biaffine over Average pooling (*Biaffine* $\rightarrow$ *Avg*) suggests that the cross-space interaction is non-linear and requires higher-order transformation to capture fine-grained Aspect-Opinion (A-O) associations. (4) Replacing the Shared Matrix with the raw syntactic tree (*Shared* $\rightarrow$ *Syntactic*) drops the F1 score by 2.71 on average, confirming that raw dependency trees introduce noisy connections. As illustrated in Figure 3, the attention weights for Aspect-Opinion (A-O) connections (highlighted in the red box) are significantly intensified. Driven by the guidance of our Shared Matrix, this empirical evidence validates that the integration of syntactic information effectively reinforces task-specific A-O dependencies, thereby forcing the attention mechanism to focus more sharply on task-relevant connections.

Effect of CoT This section evaluates individual components within our CoT strategy. (1) The performance of *w/o CoT* declines by 4.88 in F1 score on average, demonstrating that decomposing the complex ASTE task into a sequential "extraction-filtering-verification" chain is more effective than direct prediction. (2) Interestingly, *w/o Contrastive Data* yields worse results on the 14res dataset, proving that without negative candidates, the reasoning chain suffers from hallucinated associations. (3) The performance gap in *w/o Reason* confirms that generating justifications guides the model to enhance interpretability and perform self-correction during output generation.

5.3 Impact of Varied Backbones and Tuning Paradigms

To investigate different backbones and tuning paradigms, we compare SG-SAME across various architectures, including PLMs (BERT, RoBERTa) and PEFT-based Qwen3-8B. All backbones are integrated into our framework to ensure a fair evaluation.

As shown in Table 3, SG-SAME (Llama-3.2-1B with full-parameter fine-tuning) consistently achieves superior F1 scores across all four datasets, outperforming the larger Qwen3-8B-Instruct model tuned with LoRA. This aligns with Instruct-ABSA (Scaria et al., 2024), indicating that full-parameter fine-tuning enables better adaptation of internal representations to task-specific inter-token relations. Additionally, our LLM-based framework consistently outperforms PLM baselines, achieving average F1 improvements of 5.72% and 3.83% relative to BERT and RoBERTa. respectively. This confirms that the reasoning capabilities of the LLM architecture provide a critical edge in capturing complex sentiment-aspect dependencies.

6 Conclusion

This paper addresses the issue of potential noise inherent in syntactic and semantic features for the ASTE task. First, we derive semantic and syntactic matrices using attention mechanisms and external parsers, respectively. Then, a multi-view masking mechanism together with a biaffine-based gating module is employed to derive a task-aligned shared matrix from the existing semantic and syntactic matrices. This matrix is subsequently injected into the hidden representations of a large language model (LLM) to enable syntax-guided representation learning. Finally, we design a CoT framework incorporating negative candidate injection, which leverages a four-step reasoning process to enhance the model's ability to discriminate and rectify complex sentiment relations. Extensive experiments on multiple benchmark datasets demonstrate that our SG-SAME model consistently outperforms state-of-the-art baselines.

Limitations

Although our method proposes an approach to guide the attention matrix within LLM layers, it does not completely eliminate task-irrelevant noise inherent in semantic or syntactic features, which may interfere with performance. While improving performance, our method also introduces additional syntactic and shared matrices, which incur extra inference time and bring acceptable computational overhead. Furthermore, our method has only been validated on the ASTE task and has not been extended to other aspect-based sentiment analysis tasks. We leave the exploration of these issues to future work.

References

Tamara Álvarez-López, Jonathan Juncal-Martínez, Milagros Fernández Gavilanes, Enrique Costa-Montenegro, and Francisco Javier González-Castaño. 2016. GTI at semeval-2016 task 5: SVM and CRF for aspect detection and unsupervised aspect-based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 306–311. The Association for Computer Linguistics.

Aaditya Bodke, Avinoor Singh Kohli, Hemant Subhash Pardeshi, and Prathamesh Bhosale. 2025. PASTEL : Polarity-aware sentiment triplet extraction with llm-as-a-judge. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 25523–25533. Association for Computational Linguistics.

Hao Chen, Zepeng Zhai, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022a. Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2974–2985. Association for Computational Linguistics.

Hao Chen, Zepeng Zhai, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022b. Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2974–2985. Association for Computational Linguistics.

Zhexue Chen, Hong Huang, Bang Liu, Xuanhua Shi, and Hai Jin. 2021. Semantic and syntactic enhanced aspect sentiment triplet extraction. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1474–1483. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2509–2518. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dongxu Li, Zhihao Yang, Yuquan Lan, Yunqi Zhang, Hui Zhao, and Gang Zhao. 2023a. Simple approach for aspect sentiment triplet extraction using span-based segment tagging and dual extractors. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2374–2378. ACM.

Pan Li, Ping Li, and Kai Zhang. 2023b. Dual-channel span for aspect sentiment triplet extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 248–261. Association for Computational Linguistics.

Qingling Li, Wushao Wen, and Jinghui Qin. 2025. Boundary-driven table-filling with cross-granularity contrastive learning for aspect sentiment triplet extraction. *CoRR*, abs/2502.01942.

Shuo Liang, Wei Wei, Xian-Ling Mao, Yuanyuan Fu, Rui Fang, and Dangyang Chen. 2023. STAGE: span tagging and greedy inference scheme for aspect sentiment triplet extraction. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 13174–13182. AAAI Press.

Xinjing Liu, Pengyue Lin, Xinyu Tu, Wenqi Jia, Chen Jiang, and Ruifan Li. 2025. SDG-MLLM: injecting structured dialogue graphs into MLLM for multimodal conversational aspect-based sentiment analysis. In *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pages 14114–14121. ACM.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language

processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60. The Association for Computer Linguistics.

Iwo Naglik and Mateusz Lango. 2023. Exploiting phrase interrelations in span-level neural approaches for aspect sentiment triplet extraction. In *Advances in Knowledge Discovery and Data Mining - 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25-28, 2023, Proceedings, Part IV*, volume 13938 of *Lecture Notes in Computer Science*, pages 222–233. Springer.

Iwo Naglik and Mateusz Lango. 2024. Astetransformer: Modelling dependencies in aspect-sentiment triplet extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 2324–2339. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis M. Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4034–4043. European Language Resources Association.

Momose Oyama, Sho Yokoi, and Hidetoshi Shimodaira. 2023. Norm of word embedding encodes information gain. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2108–2130. Association for Computational Linguistics.

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 *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8600–8607. AAAI Press.

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@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 486–495. The Association for Computer Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014a. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 27–35. The Association for Computer Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014b. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 27–35. The Association for Computer Linguistics.

Kevin Scaria, Himanshu Gupta, Siddharth Goyal, Saurabh Arjun Sawant, Swaroop Mishra, and Chitta Baral. 2024. Instructabsa: Instruction learning for aspect based sentiment analysis. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Short Papers, NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 720–736. Association for Computational Linguistics.

Dongming Sheng, Kexin Han, Hao Li, Yan Zhang, Yucheng Huang, Jun Lang, and Wenqiang Liu. 2025. Test-time code-switching for cross-lingual aspect sentiment triplet extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 5041–5053. Association for Computational Linguistics.

Qiao Sun, Liujia Yang, Minghao Ma, Nanyang Ye, and Qinying Gu. 2024. Minicongts: A near ultimate minimalist contrastive grid tagging scheme for aspect sentiment triplet extraction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 2817–2834. Association for Computational Linguistics.

Christopher K. I Williams. 2003. Learning with kernels: Support vector machines, regularization, optimization, and beyond. *Publications of the American Statistical Association*, 98(462):1.

Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning span-level interactions for aspect sentiment triplet extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4755–4766. Association for Computational Linguistics.

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 2020, Online, November 16-20, 2020*, pages

2339–2349. Association for Computational Linguistics.

Fan Yang, Mian Zhang, Gongzhen Hu, and Xiabing Zhou. 2023. A pairing enhancement approach for aspect sentiment triplet extraction. In *Knowledge Science, Engineering and Management - 16th International Conference, KSEM 2023, Guangzhou, China, August 16-18, 2023, Proceedings, Part III*, volume 14119 of *Lecture Notes in Computer Science*, pages 399–410. Springer.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2023. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Trans. Knowl. Data Eng.*, 35(11):11019–11038.

Zheng Zhang, Zili Zhou, and Yanna Wang. 2022. SSEGCN: syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4916–4925. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2024. Multimodal chain-of-thought reasoning in language models. *Trans. Mach. Learn. Res.*, 2024.

Xiaowei Zhao, Yong Zhou, and Xiujuan Xu. 2024a. Dual encoder: Exploiting the potential of syntactic and semantic for aspect sentiment triplet extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 5401–5413. ELRA and ICCL.

Xiaowei Zhao, Yong Zhou, and Xiujuan Xu. 2024b. Dual encoder: Exploiting the potential of syntactic and semantic for aspect sentiment triplet extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 5401–5413. ELRA and ICCL.

Dataset	Split	NEU	POS	NEG	#S	#T
14LAP	Train	126	817	517	906	1460
	Dev	36	169	141	219	346
	Test	63	364	116	328	543
14RES	Train	166	1692	480	1266	2338
	Dev	54	404	119	310	577
	Test	66	773	155	492	994
15RES	Train	25	783	205	605	1013
	Dev	11	185	53	148	249
	Test	25	317	143	322	485
16RES	Train	50	1015	329	857	1394
	Dev	11	252	76	210	339
	Test	29	407	78	326	514

Table 4: Statistics for the ASTE-Data-V2 dataset.

A Dataset Statistics

We present the detailed information of the datasets used in 4. The abbreviations NEU, POS, and NEG represent the number of triplets with neutral, positive, and negative sentiment polarities, respectively. In addition, #S and #T indicate the total number of sentences and the total number of aspect sentiment triplets contained in each dataset split. These statistics provide an overview of the dataset distribution and the relative proportion of different sentiment categories across training, development, and test sets.

B Additional baseline description

We provide a detailed account of the vital baselines, which include syntax-based models, LLM-based models, and span-based PLM models that have achieved promising performance:

- S^3E^2 (Chen et al., 2021) The model jointly extracts triplets by consolidating sequential contextual semantics via an LSTM, concurrently with syntactic dependencies, semantic relatedness, and positional relationships using a Graph Neural Network (GNN).

- EMC-GCN (Chen et al., 2022b) The model jointly extracts triplets by constructing a Multi-Channel Graph Convolutional Network (Multi-Channel GCN) integrated with diverse linguistic features (e.g., part-of-speech, dependency syntax, and positional distance), complemented by a refining strategy that leverages implicit intermediate extraction results for word-pair representation refinement.

- Dual-Span (Li et al., 2023b) The framework implements a dual-channel span generation mechanism that integrates both syntactic dependencies and part-of-speech attributes. Concurrently, a Relational Graph Attention Network (RGAT) and a gating module are employed to fuse these features for joint triplet extraction, effectively slashing the number of candidate spans by nearly a half.

- D2E2S (Zhao et al., 2024b) The model extracts triplets by employing a dual-channel encoder (where BERT and an enhanced LSTM capture semantic and syntactic information, respectively) combined with a Heterogeneous Feature Interaction Module (HFIM). To better differentiate the two streams, a KL divergence similarity separation mechanism is enforced to segregate the semantic and syntactic views.

- ASTE-Transformer (Naglik and Lango, 2024) The model achieves joint triplet extraction through a three-stage progressive Transformer pipeline, which sequentially integrates language modeling, self-attention-driven span-pair matching, and globally interactive decision-making. To bootstrap the framework, a pre-training scheme leveraging pseudo-labels from sentiment classification data is further introduced.

- BTF-CCL (Li et al., 2025) enhance performance by combining boundary-driven table filling with cross-granularity contrastive learning to align sentence- and word-level semantics.

- PASTEL (Bodke et al., 2025) The framework achieves joint triplet extraction by decomposing the task into structured subtasks. Specifically, fine-tuned LLMs are first deployed to independently extract aspect targets and polarity-aware opinion phrases, from which a comprehensive candidate pool is populated via a Cartesian product. Finally, GPT-4o is employed as an LLM-as-a-Judge to cross-verify and prune the candidate triplets.

C Computational Efficiency

To comprehensively evaluate the computational efficiency of our proposed method, we present a detailed comparison of resource overhead and F1-scores across different models on the 14Lap dataset in Table 5. All experiments are independently conducted three times to report the average results,

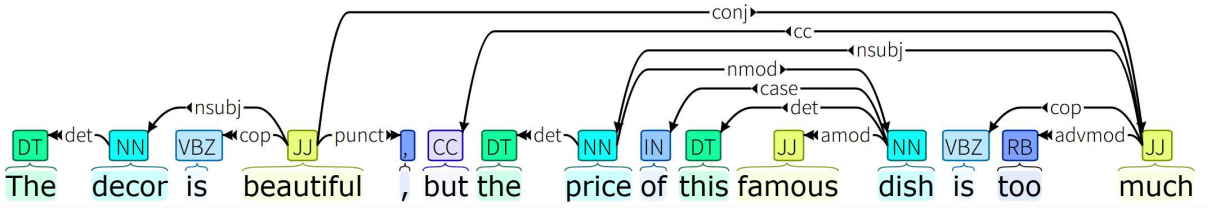

Figure 4: The syntactic dependency graph of the example used in our work.

Table 5: Computational efficiency and comparison of F1-scores for different methods on the 14Lap dataset.

Model	Trainable Params (M)	Peak Memory (GB)	Time / Epoch (s)	Total Training Time (min)	F1 (%)
LLM-Based					
SG-SAME (ours)	1567.32	49.67	1013.94	112.58	69.75
SG-SAME (w/o COT)	1567.32	44.90	362.43	34.49	66.84
Qwen3-8B-Instruct (LoRA)	1348.78	54.45	962.48	86.01	64.33
PLM-Based					
DeBERTa-v3-large	1772.72	34.47	65.83	78.04	68.64

and the training time per epoch is calculated as the average duration across all training epochs. The hyperparameter configurations of our models can be found in Appendix D.

The complete SG-SAME (Ours) achieves the highest F1-score of 69.75%, outperforming its variant without CoT (SG-SAME w/o CoT) by 2.91%. However, since the CoT reasoning paths significantly increase the input sequence length—which scales up the computational workload during forward and backward propagation—the total training time of the complete model (112.58 min) is 2.26 times that of the non-CoT variant, accompanied by a slight increase in peak memory. This indicates that the performance gain yielded by CoT comes at the cost of training time.

Qwen3-8B-Instruct (LoRA) possesses a massive total parameter size of 9537.32M. Although its trainable parameters (1348.78M) are reduced by 13.94% compared to our LLaMA-1B full fine-tuning—tuning only 14.13% of its total parameters—its peak memory usage (54.45 GB) remains the highest among all models. This is primarily attributed to the massive activation memory inherently generated by the larger model backbone during training.

Compared with Qwen3-8B-Instruct (LoRA), our SG-SAME (w/o CoT) demonstrates substantial lightweight advantages in both total training time (34.49 min vs. 86.01 min) and peak memory (44.90 GB vs. 54.45 GB), while simultaneously delivering a 2.51% improvement in F1-score (66.84% vs. 64.33%). This strongly demonstrates that rather than blindly scaling up the language model size, explicitly injecting structural priors enables a smaller-scale LLM to achieve superior task adaptation.

We observe that while DeBERTa-v3-large possesses a volume of trainable parameters (1772.72M) comparable to our model, it exhibits a significantly lower peak memory consumption (34.47 GB). This phenomenon can be primarily attributed to the structural characteristics of the bi-directional encoder-based PLM, which features a smaller vocabulary size and avoids the massive dynamic activation memory overhead inherently triggered by causal autoregressive LLMs during training.

D Hyperparameter Configurations

To ensure the reproducibility of all experiments in this study, we present the detailed hyperparameter configurations for all baseline models evaluated in the ablation studies (Section 3) in Table 6.

For the LLM-based methods, such as Qwen3-8B-Instruct and LLaMA-3.1-8B-Instruct, we employ the LoRA fine-tuning strategy, incorporating an early stopping mechanism to mitigate overfitting.

For the PLM-based approaches, including DeBERTa-v3-large, RoBERTa-large, and BERT-large-uncased, we perform full-parameter fine-tuning. Critical hyperparameters, such as the learning rate and warmup ratio, are carefully tuned according to the specific requirements of each task. All models are evaluated under identical data splits to guarantee a fair and rigorous comparison.

Table 6: Detailed hyperparameter configurations for the evaluation models in ablation studies.

Hyperparameter	**Qwen3-8B**	**Llama-3.1-8B**	**DeBERTa-v3-large**	**RoBERTa-large**	**BERT-large-uncased**
Tuning Strategy	LoRA	LoRA	Full	Full	Full
Batch Size	1	1	8	32	16
Gradient Accumulation	2	2	1	1	1
Max Epochs	8	8	70	70	80
Learning Rate	5e-5	5e-5	1e-5	3e-5	5e-5
Weight Decay	0.01	0.01	0.01	0.01	0.01
LR Warmup Ratio	-	-	0.2	0.1	0.1
LoRA Rank (r)	16	16	-	-	-
LoRA Alpha (α)	32	32	-	-	-
LoRA Dropout	0.02	0.02	-	-	-
Eval Steps	100	100	-	-	-
Early Stopping Patience	10	10	-	-	-

Dataset	$\mathbf{A}_{syn}$ **(%)**	$\mathbf{A}_{sem}$ **(%)**	$\mathbf{A}_{sha}$ **(%)**
14lap	12.62	8.12	31.84
14res	11.16	10.13	36.45
15res	10.36	6.46	29.19
16res	15.65	8.95	34.35
Average	**12.45**	**8.42**	**32.96**

Table 7: The proportion of Aspect-Opinion (A-O) connection weights.

E Detailed Analysis of Syntactic Dependency Patterns

Consider the following example: *"The decor is beautiful, but the price of this famous dish is too much."* As illustrated in Figure 4, the dependency parse tree of this example is presented, with their respective connection types labeled. According to the Universal Dependencies (UD) v2 standard (Nivre et al., 2020), coordination is analyzed by using the $conj$ relation to connect all non-first conjuncts to the first one. This convention creates an explicit dependency edge: *Beautiful* $\xrightarrow{conj}$ *much*. Furthermore, since *Price* serves as the nominal subject of *much*, the edge *price* $\xrightarrow{nsubj}$ *much* is established. These connections form a 2-hop path between *beautiful* and *price*. Beyond the entanglement across clauses, syntactic noise also manifests within local phrase structures. The preposition *of* in the phrase *price of this famous dish* serves as a *case* marker and is excluded from the dependency core under UD rules. Consequently, a direct nmod relation is established between the nouns, denoted as *price* $\xrightarrow{nmod}$ *dish*. Since the adjective *famous* is attached to its head noun *dish* via an amod relation, a 2-hop path is formed : *price* $\xrightarrow{nmod}$ *dish* $\xrightarrow{amod}$ *famous*.

Consequently, when these task-irrelevant two-hop connections are injected into a GCN with more than two layers or processed by attention mechanisms, it leads to aggregation between the two task-irrelevant tokens or inflates the attention scores between them, thereby detrimental to the model's performance.

F Component Analysis of the Shared Matrix Module

We calculate the Aspect-Opinion (A-O) connection weights for the syntactic, semantic, and shared matrices derived from Eqs. 2,5, and 9, respectively. To ensure a fair comparison, Min-Max normalization is applied to all three matrices. The A-O index sets are determined based on the original datasets, and the corresponding values are extracted from the matrices to compute the final weight proportions. As shown in Table 7, the raw syntactic and semantic matrices assign limited attention to A-O pairs with averages of 12.45% and 8.42% respectively, while our proposed shared matrix yields a higher density reaching up to 32.96%. This increase substantiates that our shared matrix module effectively concentrates on task-relevant connections, providing superior inputs for the subsequent CoT process via the attention mechanism.

G Definition and Verification of the Shared Matrix

In this section, we explicitly define the concepts of *co-occurrence* and *absence* mentioned in Section 1. Since the semantic attention matrix $\mathbf{A}_{\text{sem}}$ is a dense matrix where all tokens have connec-

Table 8: Case study comparing triplet extraction results on representative instances from the test sets. Correctly extracted triplets are in gold triplets, while incorrect or hallucinated elements are highlighted in red with underlines. None denotes empty extractions.

Category	Text and Extracted Triplets
Case 1: Laptop 14, ID: 1004293:144	
Text	First off, I really do like my MBP; once used to the OS, it is pretty easy to get around, and the overall build is great; e.g., the keyboard is one of the best to type on.
Gold Triplets	(OS, like, POSITIVE); (OS, easy, POSITIVE); (overall build, great, POSITIVE); (keyboard, best, POSITIVE)
BTF-CCL	(MBP, like, POSITIVE); (OS, easy, POSITIVE); (overall build, great, POSITIVE); (keyboard, best to type on, POSITIVE)
SG-SAME (w/o CoT)	(OS, like, POSITIVE); (OS, easy, POSITIVE); (overall build, great, POSITIVE); (keyboard, best, POSITIVE); (MBP, easy, POSITIVE)
SG-SAME (Ours)	(OS, like, POSITIVE); (OS, easy, POSITIVE); (overall build, great, POSITIVE); (keyboard, best, POSITIVE)
Case 2: Restaurant 14, ID: 1004293:439	
Text	The homage to India is most evident in the delectable roti canai appetizer, a fried pancake served with pungent curry dipping sauce, while the mango chicken offers a surprisingly sophisticated, fresh take on sweet-and-sour.
Gold Triplets	(roti canai appetizer, delectable, POSITIVE); (mango chicken, surprisingly sophisticated, POSITIVE); (mango chicken, fresh, POSITIVE)
BTF-CCL	(roti canai, delectable, POSITIVE); (curry dipping sauce, pungent, POSITIVE); (mango chicken, fresh, POSITIVE)
SG-SAME (w/o CoT)	(roti canai appetizer, delectable, POSITIVE); (mango chicken, surprisingly sophisticated, POSITIVE); (mango chicken, fresh, POSITIVE); (curry dipping sauce, pungent, POSITIVE)
SG-SAME (Ours)	(roti canai appetizer, delectable, POSITIVE); (mango chicken, surprisingly sophisticated, POSITIVE); (mango chicken, fresh, POSITIVE)
Case 3: Restaurant 15, ID: 1004293:232	
Text	The crust has a great bite and a good chew, the sauce is light with a nice acidity to it, and the salt from the cheese is great, really heightening the flavor of all the other components.
Gold Triplets	(crust, great, POSITIVE); (crust, good, POSITIVE); (sauce, light, POSITIVE); (cheese, great, POSITIVE)
BTF-CCL	(crust, great, POSITIVE); None; (sauce, light, POSITIVE); (cheese, great, POSITIVE)
SG-SAME (w/o CoT)	(crust, great, POSITIVE); (crust, good, POSITIVE); (sauce, light, POSITIVE); (cheese, great, POSITIVE); (sauce, nice, POSITIVE)
SG-SAME (Ours)	(crust, great, POSITIVE); (crust, good, POSITIVE); (sauce, light, POSITIVE); (cheese, great, POSITIVE)

tions, and the syntactic dependency tree $\mathbf{A}_{\text{syn}}$ is a sparse 0-1 matrix, we define these two concepts as follows:

- **Definition of Co-occurrence**: For an authentic Aspect-Opinion (A-O) pair, co-occurrence means it simultaneously exists in both views. Specifically, in the syntactic dependency tree, they have a direct connection (such as *nsubj* or *amod*), meaning $\mathbf{A}_{\text{syn},ij} = 1$. Concurrently, in the semantic attention matrix, their attention weight $\mathbf{A}_{\text{sem},ij}$ is significantly high and stands out above the background noise due to the real sentiment modification relationship. The intersection of this syntactic connection and high semantic attention weight is defined as *co-occurrence*.

- **Definition of Absence**: The absence of task-irrelevant noise means that the connection fails to be validated by both views at the same time. This typically includes two scenarios:

  1. *Syntactic connection absent in semantics*: Certain token pairs are connected in the syntactic dependency tree ($\mathbf{A}_{\text{syn},ij} = 1$) (e.g., the *beautiful–much* connection) purely due to parsing rules. However, they lack substantive sentiment relevance, resulting in very low weights in the semantic attention matrix ($\mathbf{A}_{\text{sem},ij} \to 0$). This near-zero attention weight is defined as being *absent* from the semantic view.

  2. *Semantic connection absent in syntax*: Interfered by prominent sentiment words, certain task-irrelevant token pairs may falsely receive high weights in the semantic attention matrix. Nevertheless, they share no dependency relation in

the external syntactic dependency tree ($\mathbf{A}_{\text{syn},ij} = 0$). This lack of grammatical grounding is defined as being *absent* from the syntactic view.

By applying these definitions, our framework can accurately extract the co-occurring authentic A-O pairs.

H Case Study

To intuitively evaluate the practical performance of our proposed method, we present a detailed qualitative analysis in Table 8. Specifically, we select three representative test instances across different domains from the Laptop14, Restaurant14, and Restaurant15 datasets, respectively. The evaluation models for comparison include our complete framework (SG-SAME), the ablation variant without the reasoning chain (SG-SAME w/o CoT), and the state-of-the-art open-source baseline method BTF-CCL (Li et al., 2025). Through this comprehensive comparison, we derive several critical architectural insights:

In Case 1 (from Laptop14), the sentence contains dense explicit targets (*MBP*, *OS*, *overall build*, *keyboard*). The traditional discriminative PLM, BTF-CCL, suffers from local sentiment mismatching, erroneously binding the opinion of *OS* to "*MBP*", and extracts a corrupted opinion span "*best to type on*" due to loose token-level boundaries. Concurrently, the non-CoT variant also generates a hallucinated connection "(*MBP*, *easy*, *POSITIVE*)". In contrast, our complete SG-SAME (Ours) perfectly matches all gold triplets, validating that our multi-step joint reasoning strategy maintains high fidelity in isolating dense spans and complex syntactic dependencies.

In Case 2 (from Restaurant14), when encountering long-distance syntactic structures, BTF-CCL exhibits a typical short-span truncation defect inherent to PLMs, shrinking the compound attribute "*roti canai appetizer*" into "*roti canai*". More crucially, the text contains an objective descriptive token "*pungent*"; both BTF-CCL and the w/o CoT variant, lacking logical reasoning depth, falsely interpret it as a sentiment opinion and deliver redundant extractions. Benefiting from the explicit Contrastive CoT that enforces multi-step rationalization and rebuttal verification, our full model successfully filters out such deceptive elements at the Step-4 phase, showcasing superior anti-noise capabilities.

Case 3 (from Restaurant15) poses a severe challenge characterized by a multi-opinion structure (*great bite* and *good chew* for *crust*) and densely parallel targets. Under this layout, BTF-CCL fails to capture the second triplet (denoted as "*None*"), whereas the w/o CoT variant triggers over-extraction, isolating "*nice*" from the phrase "*nice acidity*" to formulate a spurious triplet. Distinctly, our full SG-SAME achieves zero-error conformity with the ground-truth. This strongly demonstrates that explicitly incorporating structural priors alongside contrastive learning effectively restrains boundary bleeding and eliminates false-positive misclassifications in fine-grained sentiment extraction tasks.

I Generation of Chain-of-Thought Tuning Data

We present the detailed procedures and the corresponding prompt templates for both the Chain-of-Thought (CoT) data generation phase and the subsequent training and inference stages. DeepSeek-v4-Pro is utilized to construct a four-step joint reasoning dataset encompassing both positive and negative samples. Considering that different models adapt to varying prompt lengths, we employ a three-shot setting during the data generation phase. In contrast, for the training and inference phases, a one-shot setting—featuring a single exemplar that incorporates negative samples—is utilized for in-context learning.

I.1 Process and Prompts for CoT Data Generation

Initially, the ground-truth triplets (gold triplets) of each sentence are recovered from the original annotations to serve as the benchmark. For positive samples, all ground-truth triplets of a sentence are retained, and the model is guided to generate a complete four-step Chain-of-Thought (CoT) reasoning process.

For negative samples, the construction follows an asymmetric negative sampling strategy based on a 'mixture of true and false' paradigm: on the basis of fully retaining the ground-truth triplets, one to two items of perturbative noise (incorrect triplets) are explicitly introduced according to the rules detailed in Appendix A. At the input stage, the large language model is simultaneously provided with the ground-truth triplets, the incorrect triplets, and their corresponding perturbation rules, thereby in-

ducing the model to generate highly robust rebuttal evidence regarding the causes of errors. During the final verification phase of the four-step reasoning (Step 4), the model executes an instance-level, one-by-one identification over all candidate triplets, precisely rejecting the incorrect triplets with negative labels, while the ground-truth triplets are aggregated into the final structured JSON output. This scheme aims to establish a rigorous discriminative contrastive learning environment, effectively suppressing the model's inclination toward negative bias and false negative misclassifications in fine-grained sentiment analysis, thereby significantly enhancing its generalization performance.

To avoid redundancy, we define the reusable prompt segments as standardized components:

Task Description (`[Task_Desc]`)

You are an aspect-sentiment triplet extraction expert. I will give you a sentence. Extract all valid triplets and verify them jointly in 4 steps. Be faithful to the sentence and do not fabricate evidence. Each step must contain both Result and Reason.

We provide the following three negative sampling rules for reference

Negative Sampling Rules (`[Neg_Rules]`)

`wrong_pair` → Build the negative triplet by taking the aspect from one gold triplet and combining it with the opinion from another gold triplet in the same sentence.

`wrong_sentiment` → Keep a real aspect-opinion pair but replace the gold sentiment polarity with an incorrect polarity.

`irrelevant_phrase` → Use a phrase that appears in the sentence, but this phrase is not the true opinion expression for the target aspect.

The following are the reference labels provided during data generation, which correspond to the triplets that should be constructed in the first three steps. The positive samples are explicitly labeled, and the corresponding construction rules for negative samples are provided to assist the model in articulating its reasoning. These negative samples will be identified and discarded during the step-4 verification phase.

Reference Labels (`[Ref_Label]`)

The triplets that need to be constructed in the first three steps:

```
(Aspect1, Opinion1, Sentiment1)
(Aspect2, Opinion2, Sentiment2)
(Aspect3, Opinion3, Sentiment3)
```

The genuinely existing sample :

```
(Aspect1, Opinion1, Sentiment1)
(Aspect2, Opinion2, Sentiment2)
```

The synthesized negative sample :

```
(Aspect3, Opinion3, Sentiment3)
```

The rule applied to construct this negative sample :

```
(wrong_pair / wrong_sentiment / irrelevant_phrase) [Neg_Rules]
```

Next, we present the global prompt for data generation, utilizing three few-shot exemplars for reference.

Reference Labels ([Ref_Label])

[Task_Desc]

Step 1 (Aspect Identification): identify all explicit aspects.

Step 2 (Opinion Tracing): find the true opinion phrase for each aspect.

Step 3 (Sentiment Reasoning): infer sentiment from the validated aspect-opinion relation.

Step 4 (Verification): List the triplets and verify whether the sentiment polarity matches the opinion and aspect. Mark YES if true, otherwise NO.

Output exactly this format:

```
[Reasoning]

Step 1 (Aspect Identification):
Result:
Reason:

Step 2 (Opinion Tracing):
Result:
Reason:

Step 3 (Sentiment Reasoning):
Result:
Reason:

Step 4 (Verification):
(aspect, opinion, sentiment)
YES/NO
(aspect, opinion, sentiment)
YES/NO
Reason:

[Result]
(aspect, opinion, sentiment)
```

Here are examples to help you better understand the task:

[Example1]

[Example2]

[Example3]

[Neg_Rules]

[Ref_Label]

I.2 Prompt for Training and Inference

During the training and inference phases, the prompt structure remains consistent with the data generation stage, with the exceptions of [Neg_Rules] and [Ref_Label]. Specifically, since our objective shifts from data generation to the training and inference processes, we remove the correct triplets used as a reference to prevent data leakage, and require the model to generate intermediate Chain-of-Thought triplets and final verification results purely based on the input text. We present the examples utilized during the training and inference processes here:

Example

Sentence: "The decor is beautiful, but the price is too much."

[Reasoning]
Step 1 (Aspect Identification):
Result: ["decor", "price"]
Reason: Both are explicit aspect entities in the text.

Step 2 (Opinion Tracing):
Result: ["decor" → "beautiful", "price" → "too much", "decor" → "but"]
Reason: "beautiful" and "too much" are modifiers; "but" is a structural contrast.

Step 3 (Sentiment Reasoning):
Result: ["beautiful" → positive, "too much" → negative, "but" → none]
Reason: Evaluation words carry sentiment polarities; conjunctions do not.

Step 4 (Verification):
(decor, beautiful, positive) YES
(price, too much, negative) YES
(decor, but, none) NO
Reason: "but" is a conjunction, not an opinion.

[Result]
(decor, beautiful, positive)
(price, too much, negative)

Now analyze the following sentence.