$\mathcal{S}^2$IT: Stepwise Syntax Integration Tuning for Large Language Models in Aspect Sentiment Quad Prediction

Bingfeng Chen[1,2], Chenjie Qiu[1], Yifeng Xie[1], Boyan Xu[1*], Ruichu Cai[1,3], Zhifeng Hao[1,4]
[1]School of Computer Science, Guangdong University of Technology
[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
[3]Peng Cheng Laboratory
[4]College of Science, Shantou University
chenbf@gdut.edu.cn
{yausankit, evfxie, hpakyim, cairuichu}@gmail.com
haozhifeng@stu.edu.cn

Abstract

Aspect Sentiment Quad Prediction (ASQP) has seen significant advancements, largely driven by the powerful semantic understanding and generative capabilities of large language models (LLMs). However, while syntactic structure information has been proven effective in previous extractive paradigms, it remains underutilized in the generative paradigm of LLMs due to their limited reasoning capabilities. In this paper, we propose $\mathcal{S}^2$IT, a novel Stepwise Syntax Integration Tuning framework that progressively integrates syntactic structure knowledge into LLMs through a multi-step tuning process. The training process is divided into three steps. $\mathcal{S}^2$IT decomposes the quadruple generation task into two stages: 1) Global Syntax-guided Extraction and 2) Local Syntax-guided Classification, integrating both global and local syntactic structure information. Finally, *Fine-grained Structural Tuning* enhances the model's understanding of syntactic structures through the prediction of element links and node classification. Experiments demonstrate that $\mathcal{S}^2$IT significantly improves state-of-the-art performance across multiple datasets. Our implementation will be open-sourced at https://github.com/DMIRLAB-Group/S2IT.

1 Introduction

Aspect Sentiment Quad Prediction (ASQP) focuses on predicting tuples of sentiment-related elements from a given text (Zhang et al., 2022). These elements consist of four components central to ASQP: aspect term (a), aspect category (c), opinion term (o), and sentiment polarity (s). For example, in the sentence "*I really love sushi!*", the corresponding elements are "*sushi*" (aspect term), "*food quality*" (aspect category), "*love*" (opinion term), and "*positive*" (sentiment polarity). Although ASQP is fundamentally an extractive task, the ability to generalize and generate responses has made fine-tuning models like T5 (Raffel et al., 2019) a mainstream approach, as demonstrated by Zhang et al. (2021).

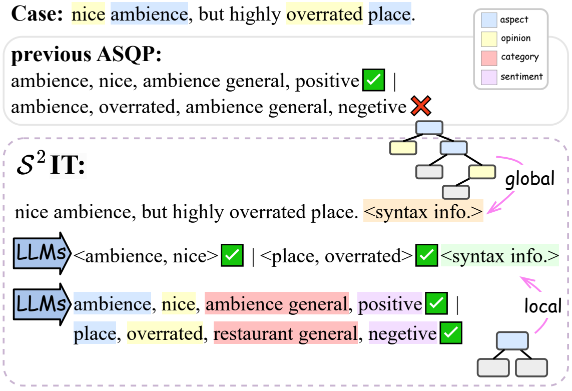
Case: nice ambience, but highly overrated place.
aspect
opinion
category
sentiment
previous ASQP:
ambience, nice, ambience general, positive ✓ |
ambience, overrated, ambience general, negetive ✗
$\mathcal{S}^2$IT:
global
nice ambience, but highly overrated place. <syntax info.>
LLMs <ambience, nice> ✓ | <place, overrated> ✓ <syntax info.>
local
LLMs ambience, nice, ambience general, positive ✓ |
place, overrated, restaurant general, negetive ✓

Figure 1: An example of the ASQP task. The most notable feature that distinguishes $\mathcal{S}^2$IT from previous work is that it decomposes the quadruples and injects syntactic information from different perspectives step by step into the large language model.

Recent approaches to improving language models' performance have incorporated syntactic structure knowledge, typically divided into encoder-only and encoder-decoder. For instance, encoder-only approaches (Liang et al., 2022; Li et al., 2023b; Chen et al., 2024) enhance the connections between elements at the representation level through Graph Neural Networks (GNNs). For encoder-decoder approaches, a key challenge is how to add syntactic structure knowledge to language models. Li et al. (2023a) and Yu et al. (2023) incorporates GNNs into the encoder layer of T5 and BART. These approaches are limited by their reliance on GNN integration, which does not extend well to decoder-only language models.

In this paper, we propose $\mathcal{S}^2$IT, which introduces a novel Stepwise Syntax Integration Tuning framework. Specifically, we decompose the complex

[*] Corresponding author, hpakyim@gmail.com

Findings of the Association for Computational Linguistics: NAACL 2025, pages 6814–6821
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

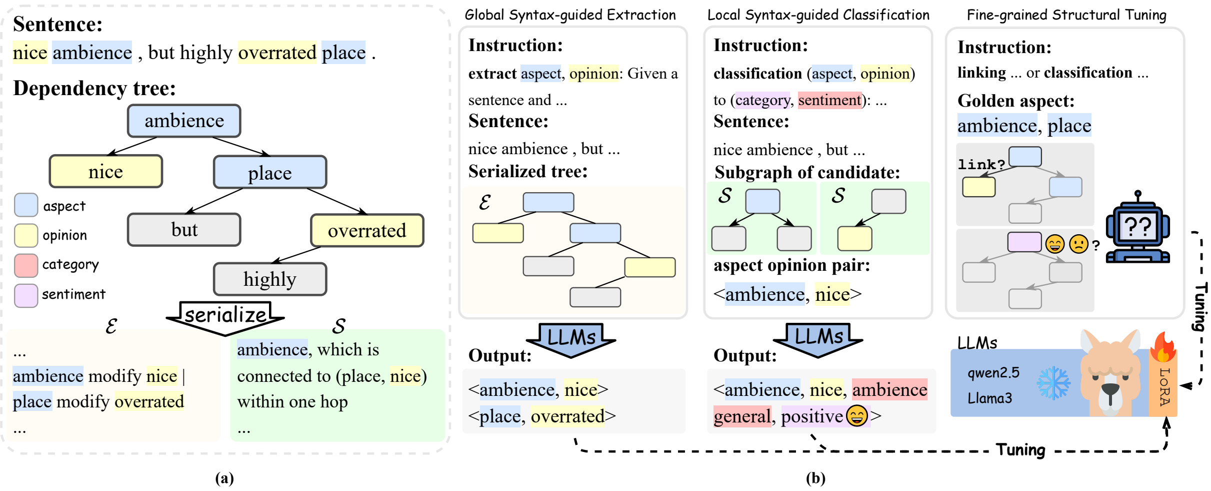

Figure 2: $\mathcal{S}^2$IT framework illustration. (a) shows how we serialize the dependency tree into natural language. (b) shows the main part of our framework and two structure instruction tuning tasks.

task of generating sentiment quadruples into two stages: *Global Syntax-guided Extraction* and *Local Syntax-guided Classification*. First, *Global Syntax-guided Extraction* incorporates global syntactic knowledge to identify aspects and opinions. Second, *Local Syntax-guided Classification* classifies opinions into sentiment and category by utilizing the local syntactic relationships between aspects and opinions. As shown in Figure 1, the incorporation of syntactic information effectively helps $\mathcal{S}^2$IT mitigate the influence of multi-aspect terms.

We also introduce *Fine-grained Structural Tuning* to improve LLMs' ability to understand and use structural knowledge for reasoning. Our main contributions are as follows: (1) We propose $\mathcal{S}^2$IT, a novel Stepwise Syntax Integration Tuning framework that progressively integrates syntactic structure knowledge into large language models (LLMs) through a multi-step tuning process. (2) $\mathcal{S}^2$IT effectively learns sentiment element relationships by leveraging syntactic structure at different granularities and local levels through three stages: Global Syntax-guided Extraction, Local Syntax-guided Classification, and Fine-grained Structural Tuning. (3) Experiments demonstrate that $\mathcal{S}^2$IT achieves the state-of-the-art on the restaurant and laptop datasets.

2 Methodology

We first formulate the task of ASQP (§2.1). As shown in Figure 2, we then present our $\mathcal{S}^2$IT, which was enhanced with the syntax information. It includes the extraction task (§2.2), classification task (§2.3), and structure instruction tuning (§2.4).

2.1 Problem Formulation

For a given sentence $x = \{x_1, x_2, ..., x_n\}$, ASQP aims to predict all the aspect-level sentiment quadruples $\{(a_j, o_j, c_j, s_j)\}_{j=1}^{N}$, which represent the **a**spect term, **o**pinion term, **c**ategory term and **s**entiment polarity, respectively. The aspect category c belongs to a predefined set of categories, and the sentiment polarity s falls into one of three categories: negative, positive, and neutral.

2.2 Global Syntax-guided Extraction

Although promising, current methods fall short of fully leveraging the additional syntactic features available. Our objective is to extract potential aspect-opinion pairs $\mathcal{P} = \{\langle a_j, o_j \rangle\}_{j=1}^{N}$ from sentences, We can introduce syntactic information to help the model discover relationships between the elements.

As an example, we consider the injection of modifier relationships between elements into LLMs. We utilize the template $\mathcal{E}$ to delineate the type of relation found in the dependency tree. Assumed that x_i serves as the head node for x_h, we document the details of this syntactic structure as follows:

$$\mathcal{E}_{i \in \{1,...,n\}} = \{x_i\} \text{ modify } \{x_h\} \qquad (1)$$

Afterward, we concatenate all pairs according to the order of their occurrence within the sentence to create a natural language description of the global structure. This description is subsequently appended to the original sentence, enhancing the syntactic context. Furthermore, we implement instruction tuning using the instruction $\mathcal{I}_1$ and the output $\mathcal{O}_1$. The process is as follows:

$$\mathcal{O}_{1,j\in\{1,\dots,N\}} = \text{aspect}\colon \{a_j\}, \text{opinion}\colon \{o_j\} \tag{2}$$

The objective of supervised fine-tuning is to minimize the loss defined as:

$$\min_{M^*} \mathcal{L}(M^*(\mathcal{I}_1,\ x,\ \mathcal{E}_1|\dots|\mathcal{E}_n, \mathcal{O}_{1,1}|\dots|\mathcal{O}_{1,N})) \tag{3}$$

2.3 Local Syntax-guided Classification

Focusing not only on the global syntactic structure, which has been shown to enhance performance (Cheng et al., 2023; Xie et al., 2023), but also paying attention to the local syntactic structure is crucial (Liang et al., 2022; Chen et al., 2024). After obtaining the correct pairs $\mathcal{P}$, treating these pairs as central nodes in the dependency tree, we extract neighboring words as subgraphs. For example, the description of a subgraph $\mathcal{S}$ centered on a_j, including at most one-hop neighbor details, is:

$$\begin{aligned}\mathcal{S}^{(1)}_{a_{j\in\{1,\dots,N\}}} =& \{a_j\} \text{ is connected with} \\ & \{v \mid \mathcal{A}_{a_j v} = 1\} \text{ within one hop.}\end{aligned} \tag{4}$$

where $\mathcal{A}$ represents the adjacency matrix and $\{v \mid \mathcal{A}_{a_j v} = 1\}$ denotes all one-hop neighbors of a_j in the dependency tree. The formula also holds true for o_j and $\mathcal{S}^{(1)}_{o_j}$.

Subsequently, we also concatenate this local syntactic information with the original sentence, integrating the output $\mathcal{O}_2$ and the instruction $\mathcal{I}_2$. This integrated data is then processed and tuned to optimize model performance further:

$$\begin{aligned}\mathcal{O}_{2,j\in\{1,\dots,N\}} =& \text{ aspect}\colon \{a_j\}, \text{opinion}\colon \{o_j\} \\ & \text{category}\colon \{c_j\}, \text{sentiment}\colon \{s_j\}\end{aligned} \tag{5}$$

We then fine-tune the LLMs while minimizing the following loss function:

$$\begin{aligned}\min_{M^*} \mathcal{L}(M^*(\mathcal{I}_2, x, \mathcal{S}^{(1)}_{a_1}, \mathcal{S}^{(1)}_{o_1} \mid \dots \mid \mathcal{S}^{(1)}_{a_N}, \mathcal{S}^{(1)}_{o_N}, \mathcal{P}, \\ \mathcal{O}_{2,1} \mid \dots \mid \mathcal{O}_{2,N}))\end{aligned} \tag{6}$$

2.4 Fine-grained Structure Instruction Tuning

Element Link Prediction. To tailor our model to specialized structured knowledge tasks, we introduce the auxiliary task about element link prediction. Specifically, we developed a structure-aware matching task that presents genuine sentiment elements (e.g., aspects) to LLMs and uses their corresponding sentiment elements (e.g., opinions) as labels. This approach guides the model to accurately associate them based on the provided sentences and syntactic information. Such alignment not only enhances the model's accuracy but also its capability to manage complex relationships within the data.

Node Classification. Building upon our method, we introduce an auxiliary task focused on sentiment element classification. This task is integral to our structure-aware strategy, directing the model to accurately assign labels to each sentiment element (e.g., sentiment polarity or other relevant categories) within the structural framework. Through this task, the model effectively learns semantic information and local syntactic structures, thereby enhancing its capability to comprehend and analyze intricate language nuances.

These tasks are integral to our fine-grained structure instruction tuning method. All our instruction prompts are summarized in Appendix A.2.

3 Experiments

3.1 Datasets

We validate our methods on Restaurant and Laptop datasets (Cai et al., 2021a). There are 2,286 sentences in Restaurant domain, and 4,076 sentences in Laptop domain. Following the setting from Cai et al. (2021a), we divide the original dataset into a training set, a validation set, and a testing set.

3.2 Implement Details

We used Qwen2.5-7B-Instruct[1] and Qwen2.5-32B-Instruct[2] as the base language model. In addition, to verify how well our framework works on different LLMs, we also conducted ablation experiments on Llama3-8B-instruct[3] in (§3.4). The model was trained with an initial learning rate of 5e-5 for 5 epochs. We set the batch size to 4 and used a gradient accumulation of 2. We applied LoRA (Hu et al., 2021) with a LoRA rank of 32 for efficient fine-tuning, allowing our 7B model to be trained and inferred on a single NVIDIA GeForce RTX

[1] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[2] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

[3] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

Model	Restaurant		Laptop	
	R	F1	R	F1
• ***Extractive-based methods***				
TAS-BERT(Wan et al., 2020)	46.3	33.5	19.2	27.3
Extract-Classify(Cai et al., 2021b)	52.9	44.6	29.4	35.8
One-ASQP(Zhou et al., 2023)	56.2	60.6	39.5	41.5
• ***Generative-based methods***				
Paraphrase(Zhang et al., 2021)	59.8	59.8	42.5	43.0
DLO(Hu et al., 2022)	59.8	59.9	43.8	43.6
+AToss(Seo et al., 2024)	59.9	60.5	43.9	44.5
MvP(Gou et al., 2023)	57.8	59.5	43.6	43.7
+AToss(Seo et al., 2024)	58.3	60.6	43.1	44.2
qwen2.5-7B	64.1	63.6	44.7	44.6
$\mathcal{S}^2$**IT-7B**	64.9	66.1	44.7	45.9
qwen2.5-32B	65.0	64.2	45.0	45.3
$\mathcal{S}^2$**IT-32B**	**66.6**	**67.37**	**45.4**	**46.7**

Table 1: Recall and F1 scores (%) on the Restaurant and Laptop datasets compared to other baselines.

3090 24G. All training details, including hyperparameters, will be presented in Appendix A.1.

3.3 Main results

Our method significantly outperforms previous state-of-the-art baselines in terms of F1 scores across five datasets in supervised settings as shown in Table 1, becoming SOTA for all these tasks. In particular, our performance on the restaurant dataset of the challenging ACOS task exceeds the baseline model by 2.71 points. This improvement is attributed to our two-step framework, which effectively reduces the coupling in the tuple generation task, helping the model extract correct aspect-opinion pairs from the sentence and make accurate classifications from both syntactic and semantic perspectives.

3.4 Ablation Experiment.

Effect Analysis of syntax integration. To verify the impact of syntax integration, we conducted additional experiments to evaluate the effectiveness of syntax information in our framework. Table 2 compares the base model (qwen2.5-7B), the full $\mathcal{S}^2$IT framework, and $\mathcal{S}^2$IT with syntax information entirely removed. The results highlight the importance of syntax information in achieving superior performance.

Effect Analysis of Syntax Imformation. Global structural information effectively helps the model understand the syntactic relationship between aspects and opinions in a sentence. As shown in Figure 3, incorporating global syntactic features significantly improved the model's performance. We observed that, compared to the model without syntactic injection, our approach increased the F1 score by 2.9% on the Restaurant dataset and by 3.1% on the Laptop dataset in Step 1, proving the importance of global structure in handling complex dependencies. Local structure helps the model capture fine-grained details by aggregating local syntactic information. In Step 2, incorporating local structure improved the F1 score by 1.4% on the Restaurant dataset and 2.2% on the Laptop dataset. Thus, understanding nearby sentiment elements significantly enhances the model's ability significantly to perform accurate sentiment classification.

Model	Restaurant		Laptop	
	R	F1	R	F1
qwen2.5-7B	64.1	63.6	44.7	44.6
$\mathcal{S}^2$**IT-7B**	**64.9**	**66.1**	**45.0**	**45.9**
w/o syntax information	63.0	64.8	43.6	44.8

Table 2: Full $\mathcal{S}^2$IT framework, with Fine-grained Structural Tuning, but completely excluding syntax information

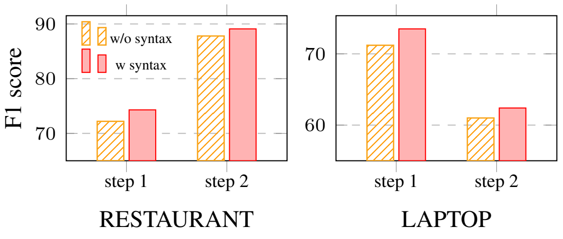

Figure 3: The impact of syntactic information at each stage on the Laptop and Restaurant datasets. Step 1 demonstrated the improvements that syntactic information brings to the extraction of $\mathcal{P}$, Step 2 showed how much the model depends on syntactic information when it has the correct $\mathcal{P}$.

Effect Analysis of Structure Instruction Tuning. To validate the effectiveness of structural instruction tuning, we conducted ablation experiments on two datasets using two different LLMs, focusing on the impact of element linking and node classification on the model's ability to understand structural information. As shown in Table 3, the effectiveness of our tasks was confirmed. We observed a significant performance drop across both models on different datasets after removing all structural instruction tuning tasks. This result indicates that

Model	Restaurant		Laptop	
	R	F1	R	F1
$\mathcal{S}^2$IT+qwen2.5(7B)	64.9	66.1	44.7	45.9
w/o Element Link	63.7	65.3	43.4	44.7
w/o Node Classification	63.1	65.0	43.3	44.5
w/o both	62.7	64.3	42.9	44.5
$\mathcal{S}^2$IT+llama3(8B)	64.8	66.0	44.2	45.7
w/o Element Link	62.7	64.9	43.6	44.8
w/o Node Classification	63.5	65.8	42.3	43.9
w/o both	62.8	64.3	43.4	44.6

Table 3: The ablation experiments on Structure Instruction Tuning.

Model	Restaurant		Laptop	
	R	F1	R	F1
qwen2.5-7B	64.1	63.6	44.7	44.6
qwen2.5-7B w/ **nl-syn**	**65.3**	**64.4**	**45.0**	**44.9**
qwen2.5-7B w/ symbol-syn	61.9	61.0	42.8	42.9

Table 4: Performance Comparison of Different Dependency Tree Templates on Restaurant and Laptop Datasets. Here, nl-syn refers to dependency tree syntactic information described using natural language templates, while symbol-syn refers to dependency tree syntactic information presented in the traditional bracketed tree format.

the structural instruction tuning tasks effectively enhance the model's understanding of hierarchical structures, especially when handling tasks like complex sentiment classification task.

Effect of Structural Prompt Templates. To further validate our structural prompt templates, we designed an interesting experiment comparing the overall impact of dependency tree templates composed of natural language (nl-syn) versus classical dependency tree templates composed of symbols (symbol-syn) on our method. The results in Table 4 reveal that symbol-based templates significantly degrade performance. We attribute this decline to the inconsistency between symbol templates and the pretraining objectives of LLMs, which adversely affects downstream task performance. This experiment demonstrates the rationale for converting syntactic information into natural language to better align with the capabilities of LLMs.

4 Related Work

Extractive Methods Previous methods leverage language models such as BERT to generate word-level textual embeddings and train additional classification heads (Wan et al., 2020). Cai et al. (2021b); Zhou et al. (2023) employ a joint multi-task sequence labeling approach to extract elements within tuples. However, their specially designed extraction framework exhibits poor generalization across various ABSA tasks. Consequently, unified generative methods have become the current mainstream.

Generative Methods Recent approaches have favored designing a unified end-to-end framework for performing ABSA tasks using generative language models. In essence, these methods serialize target tuples into natural language and employ them as labels to fine-tune the language models (Zhang et al., 2021).

Depending on the task requirements, sentiment tuples can be transformed into various sequences for training. Yan et al. (2021) use the indices of the target sentiment elements within a sentence as labels to guide the model in generating the indices of aspects and opinions. Bao et al. (2022, 2023) introduced a tree generation template that guides the model in capturing the semantic relationships between sentiment elements and utilizes a generative model to extract linearized trees.

A key challenge is how to add syntactic structure knowledge to language models. Li et al. (2023a) and Yu et al. (2023) incorporates GNNs into the encoder layer of T5 and BART. These methods all face the challenge of incorporating syntactic structural information into decoder-only models. Our framework effectively integrates syntactic knowledge into decoder-only models in a seamless and fluent manner, without requiring any structural modifications to the models.

5 Conclusion

In this paper, we presented $\mathcal{S}^2$IT, a novel framework designed to enhance Aspect Sentiment Quad Prediction (ASQP) by incorporating syntactic structure knowledge into large language models (LLMs) through a multi-step tuning process. The core idea of our framework is to simultaneously decompose quadruple prediction and syntactic structure learning during the multi-step tuning process of language models. The significant performance gains and adaptability to different language models further validate the effectiveness of our fine-tuning framework.

Limitations

While our framework achieved state-of-the-art results on the Restaurant and Laptop datasets, there is still room for improvement in constructing syntactic information and designing structural instruction tuning tasks. We have only implemented classic structure-aware tasks such as link prediction and node classification, yet these have significantly boosted performance. Nonetheless, enhancing large language models (LLMs) to better understand and reason about structural relationships remains an ongoing challenge.

Acknowledgments

This research was financially supported by the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), under Grant No. GML-KF-24-23, National Science and Technology Major Project (2021ZD0111501), National Science Fund for Excellent Young Scholars (62122022), Natural Science Foundation of China (62406078, 62476163, U24A20233), the major key project of PCL (PCL2021A12), Guangdong Basic and Applied Basic Research Foundation (2023B1515120020), Collaborative Education Project of the Ministry of Education (202407). This research was enabled by the computational resources and support of the High Performance Computing Platform at the School of Computer Science, Guangdong University of Technology.

References

Xiaoyi Bao, Zhongqing Wang, and Guodong Zhou. 2023. Exploring graph pre-training for aspect-based sentiment analysis. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Xiaoyi Bao, Wang Zhongqing, Xiaotong Jiang, Rong Xiao, and Shoushan Li. 2022. Aspect-based sentiment analysis with opinion tree generation. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4044–4050. International Joint Conferences on Artificial Intelligence Organization. Main Track.

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

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

Bingfeng Chen, Qihan Ouyang, Yongqi Luo, Boyan Xu, Ruichu Cai, and Zhifeng Hao. 2024. S^2gsl: Incorporating segment to syntactic enhanced graph structure learning for aspect-based sentiment analysis. *Preprint*, arXiv:2406.02902.

Xuxin Cheng, Zhihong Zhu, Bowen Cao, Qichen Ye, and Yuexian Zou. 2023. MRRL: Modifying the reference via reinforcement learning for non-autoregressive joint multiple intent detection and slot filling. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10495–10505, Singapore. Association for Computational Linguistics.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. Mvp: Multi-view prompting improves aspect sentiment tuple prediction. *Preprint*, arXiv:2305.12627.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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

Jinyang Li, Binyuan Hui, Reynold Cheng, Bowen Qin, Chenhao Ma, Nan Huo, Fei Huang, Wenyu Du, Luo Si, and Yongbin Li. 2023a. Graphix-t5: Mixing pre-trained transformers with graph-aware layers for text-to-sql parsing. *arXiv preprint arXiv:2301.07507*.

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*, pages 248–261, Singapore. Association for Computational Linguistics.

Shuo Liang, Wei Wei, Xian-Ling Mao, Fei Wang, and Zhiyong He. 2022. BiSyn-GAT+: Bi-syntax aware graph attention network for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1835–1848, Dublin, Ireland. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Yongsik Seo, Sungwon Song, Ryang Heo, Jieyong Kim, and Dongha Lee. 2024. Make compound sentences simple to analyze: Learning to split sentences for aspect-based sentiment analysis. *Preprint*, arXiv:2410.02297.

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z. Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9122–9129.

Yifeng Xie, Zhihong Zhu, Xuxin Cheng, Zhiqi Huang, and Dongsheng Chen. 2023. Syntax matters: Towards spoken language understanding via syntax-aware attention. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore. Association for Computational Linguistics.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian-guang Lou, and Shuai Ma. 2024. Re-reading improves reasoning in large language models.

Hang Yan, Junqi Dai, Tuo ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. *Preprint*, arXiv:2106.04300.

Chengze Yu, Taiqiang Wu, Jiayi Li, Xingyu Bai, and Yujiu Yang. 2023. Syngen: A syntactic plug-and-play module for generative aspect-based sentiment analysis. *Preprint*, arXiv:2302.13032.

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

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *Preprint*, arXiv:2203.01054.

Junxian Zhou, Haiqin Yang, Yuxuan He, Hao Mou, and Junbo Yang. 2023. A unified one-step solution for aspect sentiment quad prediction. *Preprint*, arXiv:2306.04152.

Shen Zhou and Tieyun Qian. 2023. On the strength of sequence labeling and generative models for aspect sentiment triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12038–12050, Toronto, Canada. Association for Computational Linguistics.

Appendix

A Training

A.1 Details of Hyperparameters

In this part, we outline the key hyperparameters used for fine-tuning the three different LLMs in our experiments: Qwen2.5-7B-instruct, Qwen2.5-32B-instruct and Llama3-8B-instruct. The 32B model was trained on an NVIDIA A100 80G. LoRA was applied to fine-tune the models, with ranks set to 16, or 32, and alpha values set to either 32. Dropout rates were adjusted between 0.1 to mitigate overfitting.

Hyperparameter	**Value**
Learning Rate	5e-5
Num Train Epochs	5
LoRA Rank	32
LoRA Alpha	32
LoRA Dropout	0.1
Batch Size	4
Gradient Accumulation Steps	2
LR Scheduler Type	cosine

Table 5: Hyperparameters used in the Qwen2.5-7B-instruct.

Hyperparameter	**Value**
Learning Rate	5e-5
Num Train Epochs	5
LoRA Rank	32
LoRA Alpha	32
LoRA Dropout	0.1
Batch Size	2
Gradient Accumulation Steps	4
LR Scheduler Type	cosine

Table 6: Hyperparameters used in the Qwen2.5-32B-instruct.

Hyperparameter	**Value**
Learning Rate	5e-5
Num Train Epochs	5
LoRA Rank	16
LoRA Alpha	32
LoRA Dropout	0.1
Batch Size	4
Gradient Accumulation Steps	2
LR Scheduler Type	cosine

Table 7: Hyperparameters used in the Llama3-8B-instruct.

A.2 Details of Instruction Tuning

We provide comprehensive details of the instruction tuning process. Specifically, we present examples of tasks used during Supervised Fine-Tuning

(SFT). These examples illustrate the types of instructions and responses the model was trained on, highlighting the integration of domain-specific knowledge and structural information to align the model with the desired tasks.

task: (extract aspect, opinion)

Instruction:
Given a sentence and related dependency relations, extract aspect and opinion (both implicit and explicit) from the sentence and return pair(aspect, opinion). Pay attention to the one or multi hop dependency relationships between aspect and opinion.
Input:
sentence: service ok but unfriendly , filthy bathroom .
dependency relation: root depend service | service modify ok | bathroom depend but | bathroom modify unfriendly | bathroom modify filthy | service depend bathroom
Output:
aspect: service, opinion: ok | aspect: service, opinion: unfriendly | aspect: bathroom, opinion: filthy
< | end_of_sentence | >

task: (extract opinion, aspect)

Instruction:
Given a sentence and related dependency relations, extract opinion and aspect (both implicit and explicit) from the sentence and return pair(opinion, aspect). Pay attention to the one or multi hop dependency relationships between aspect and opinion.
Input:
sentence: service ok but unfriendly , filthy bathroom .
dependency relation: root depend service | service modify ok | bathroom depend but | bathroom modify unfriendly | bathroom modify filthy | service depend bathroom
Output:
opinion: ok, aspect: service | opinion: unfriendly, aspect: service | opinion: filthy, aspect: bathroom
< | end_of_sentence | >

task: (linking aspect to opinion)

Instruction:
Given a sentence, related dependency relations and known aspects, determine the opinion (both implicit and explicit) related to the each aspect from dependency relation and return the pair(aspect, opinion).
Input:
sentence: service ok but unfriendly , filthy bathroom .
dependency relation: root depend service | service modify ok | bathroom depend but | bathroom modify unfriendly | bathroom modify filthy | service depend bathroom
candidates: aspect: service | aspect: service | aspect: bathroom
Output: aspect: service, opinion: ok | aspect: service, opinion: unfriendly | aspect: bathroom, opinion: filthy
< | end_of_sentence | >

task: (linking opinion to aspect)

Instruction:
Given a sentence, related dependency relations and known opinions, determine the aspect (both implicit and explicit) related to the each opinion from dependency relation and return the pair(opinion, aspect).
Input:
sentence: service ok but unfriendly , filthy bathroom .
dependency relation: root depend service | service modify ok | bathroom depend but | bathroom modify unfriendly | bathroom modify filthy | service depend bathroom
candidates: opinion: ok | opinion: unfriendly | opinion: filthy
Output:
opinion: ok, aspect: service | opinion: unfriendly, aspect: service | opinion: filthy, aspect: bathroom
< | end_of_sentence | >

task: (classification (aspect, opinion) to (category, sentiment))

Instruction:
Given a sentence, related dependency relations (will be presented in the form of subgraph) and (aspect, opinion) candidates, determine the category of the aspect and the sentiment (positive, neutral, negative) of the opinion and return the quadruple(aspect, opinion, category, sentiment).
Input:
sentence: service ok but unfriendly , filthy bathroom .
subgraph: aspect: service, which is connected to (bathroom, ok) within one hop. opinion: ok, which is connected to (service) within one hop. | aspect: service, which is connected to (bathroom, ok) within one hop. opinion: unfriendly, which is connected to (bathroom) within one hop. | aspect: bathroom, which is connected to (unfriendly, filthy, service, but) within one hop. opinion: filthy, which is connected to (bathroom) within one hop.
candidate: aspect: service, opinion: ok | aspect: service, opinion: unfriendly | aspect: bathroom, opinion: filthy
Output:
aspect: service, opinion: ok, category: service general, sentiment: negative | aspect: service, opinion: unfriendly, category: service general, sentiment: negative | aspect: bathroom, opinion: filthy, category: ambience general, sentiment: negative
< | end_of_sentence | >

task: (classification aspect to category)

Instruction:
Given a sentence, related dependency relations (will be presented in the form of subgraph) and known aspects (both implicit and explicit) , determine the category related to the each aspects from dependency relation and return pair (aspect, category).
Input:
sentence: service ok but unfriendly , filthy bathroom .
subgraph: aspect: service, which is connected to (bathroom, ok) within one hop. | aspect: service, which is connected to (bathroom, ok) within one hop. | aspect: bathroom, which is connected to (unfriendly, filthy, service, but) within one hop.
candidate aspect: service | service | bathroom
Output:
aspect: service, category: service general | aspect: service, category: service general | aspect: bathroom, category: ambience general
< | end_of_sentence | >

task: (classification aspect to sentiment)

Instruction:
Given a sentence, related dependency relations (will be presented in the form of subgraph) and known aspects (both implicit and explicit) , determine the sentiment related to the each aspects from dependency relation and return pair (aspect, sentiment).
Input:
sentence: service ok but unfriendly , filthy bathroom .
subgraph: aspect: service, which is connected to (bathroom, ok) within one hop. | aspect: service, which is connected to (bathroom, ok) within one hop. | aspect: bathroom, which is connected to (unfriendly, filthy, service, but) within one hop.
candidate aspect: service | service | bathroom
candidates: aspect: service | aspect: service | aspect: bathroom
Output: aspect: service, sentiment: negative | aspect: service, sentiment: negative | aspect: bathroom, sentiment: negative
< | end_of_sentence | >

task: (classification opinion to category)

Instruction:
Given a sentence, related dependency relations (will be presented in the form of subgraph) and known opinions (both implicit and explicit) , determine the category related to the each opinions from dependency relation and return pair (opinion, category).
Input:
sentence: service ok but unfriendly , filthy bathroom .
subgraph: opinion: ok, which is connected to (service) within one hop. | opinion: unfriendly, which is connected to (bathroom) within one hop. | opinion: filthy, which is connected to (bathroom) within one hop.
candidate opinion: ok | unfriendly | filthy
Output:
opinion: ok, category: service general | opinion: unfriendly, category: service general | opinion: filthy, category: ambience general
< | end_of_sentence | >

task: (classification opinion to sentiment)

Instruction:
Given a sentence, related dependency relations (will be presented in the form of subgraph) and known opinions (both implicit and explicit) , determine the category related to the each opinions from dependency relation and return pair (opinion, category).
Input:
sentence: service ok but unfriendly , filthy bathroom .
subgraph: opinion: ok, which is connected to (service) within one hop. | opinion: unfriendly, which is connected to (bathroom) within one hop. | opinion: filthy, which is connected to (bathroom) within one hop.
candidate opinion: ok | unfriendly | filthy
Output:
opinion: ok, sentiment: negative | opinion: unfriendly, sentiment: negative | opinion: filthy, sentiment: negative
< | end_of_sentence | >

B Inference

In this sention, we will introduce the details of our model during the inference process. The unidirectional dependency of sentiment elements in generative models has been thoroughly discussed in MVP (Gou et al., 2023) and SLGM (Zhou and Qian, 2023). Due to the unidirectional attention mechanism in decoder-only LLMs, this issue becomes more pronounced(Xu et al., 2024). Therefore, in the Global Syntax-guided Extraction, we adopt a bidirectional generation approach for the $\langle$a,o$\rangle$ pair, similar to SLGM. Beam search is set to a size of 4.