Aligning Black-Box LLMs for Aspect Sentiment Quad Prediction

Anonymous ACL submission

Abstract

Aspect-Based Sentiment Analysis (ABSA) focuses on extracting opinions about specific aspects, with Aspect Sentiment Quad Prediction (ASQP) being the most complex subtask. Large language models (LLMs) like GPT4 exhibit strong generalization yet struggle with ASQP due to a lack of task-specific alignment. Supervised small language models (SLMs), while effective in capturing task-specific patterns, lack the extensive knowledge of LLMs. To address this, we propose a framework that combines SLMs and LLMs using supervised in-context learning to align LLM outputs with human preferences. One SLM is supervised to generate candidate answers and guide LLMs with task-specific instructions, while another SLM acts as a reward model iteratively evaluates and refines LLM outputs. Experiments show that our framework significantly improves ASQP performance, demonstrating robustness, scalability, and potential for advancing alignment techniques in sentiment analysis.

1 Introduction

Aspect-based sentiment analysis (ABSA) is a fine-grained sentiment analysis task that aims to extract opinions expressed toward specific aspects of a given target (Hu and Liu, 2004). Among its subtasks, Aspect Sentiment Quad Prediction (ASQP) represents the most challenging task, requiring the identification of aspect-category-opinion-sentiment quads from the text (Zhang et al., 2021b; Cai et al., 2021a).

Common methods for solving ASQP often rely on structured extraction techniques (Zhang et al., 2021b; Bao et al., 2023, 2022; Cai et al., 2021a; Hu et al., 2022b). However, with the rise of large language models (LLMs) such as ChatGPT (Ouyang et al., 2022) and Claude (Anthropic, 2024), there is increasing interest in leveraging their strong generalization capabilities for ASQP. These LLMs have demonstrated remarkable performance across diverse applications (Kojima et al., 2022; Wang et al., 2023), but directly applying them to ASQP remains challenge (Zhang et al., 2023, 2024a). One common approach to align language models with human preferences is supervised fine-tuning. While effective, fine-tuning huge LLMs (e.g.,GPT-4, Claude) is infeasible for ASQP due to their black-box nature and the prohibitive computational costs of updating such massive models. Alternatively, in-context learning (Brown et al., 2020) has emerged as a practical strategy to guide black-box LLMs for downstream tasks.

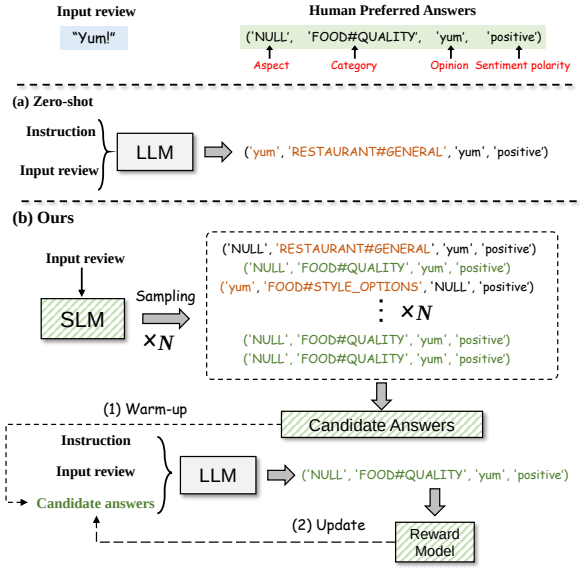

Figure 1: An illustration of the zero-shot approach and our proposed framework.

As shown in Figure 1(a), LLMs can follow instructions and generate outputs in forms that humans prefer. However, their answers often differ from what humans consider correct. This difference appears because LLMs rely on their pre-trained knowledge, which may not include the specific details required for ASQP. As a result, their outputs tend to be biased or incomplete. In con-

trast, as shown in Figure 1(b), supervised SLMs can learn patterns that align well with human preferences for ASQP. Yet, because SLMs have limited world knowledge, their single-pass answers may still be incorrect or incomplete. Surprisingly, we observe that by increasing the number of samples from one to ten, the probability of including a correct answer grows substantially, ultimately boosting the F1-score by more than 10%.

Based on these observations, we propose a framework that combines the strengths of supervised SLMs and black-box LLMs to address ASQP. Specifically, we use SLMs to learn human preferences and transfer them to LLMs through in-context learning. We firstly supervise fine-tuning a SLM to learn from human-annotated data and generates candidate answers during testing. By combining these candidate answers with well-designed instructions, we use the rich human supervision signals to guide LLMs toward aligning their outputs with human intent. Secondly, we supervised fine-tuning another SLM to act as a reward model to evaluate whether the LLMs' outputs align with human preferences (Ouyang et al., 2022; Rafailov et al., 2023). In particular, the LLM can perform multiple rounds of sampling, and the reward model evaluates the correctness of its outputs, adding correct answers to the candidate answer list. By iteratively repeating this process, the LLM can be effectively aligned with human preferences and progressively improve its ability to generate human-desired aspect sentiment quad predictions.

We conduct extensive experiments to evaluate the proposed framework across various dimensions. Our results reveal that directly applying in-context learning with black-box LLMs struggles to generate human-aligned ASQP answers, while supervised fine-tuning of SLMs effectively captures human preferences. Furthermore, integrating supervised signals of SLM into context for guiding black-box LLMs demonstrates significant improvements without requiring additional training data. Comparative studies highlight the advantages of our candidate answer strategy and iterative alignment approach.

The main contributions of this work can be summarized as follows:

- We introduce a framework combining supervised SLMs and in context learning to align black-box LLM outputs with human preferences for aspect sentiment quad prediction.
- We design a reward model to iteratively evaluate and refine LLM outputs, progressively improving their alignment with aspect sentiment quad prediction.
- Extensive experiments demonstrate the effectiveness of our framework in improving aspect sentiment quad prediction performance, highlighting its robustness, scalability, and potential for advancing alignment techniques.

2 Related Work

2.1 Aspect Sentiment Quad Prediction

Aspect-Based Sentiment Analysis (ABSA) has been extensively studied as a fine-grained sentiment analysis task (Ben-David et al., 2022; Li et al., 2022; Cai et al., 2021b; Zhang et al., 2022). The recently proposed Aspect Sentiment Quads Prediction (ASQP) extends ABSA by identifying four elements: the aspect, its category, the associated opinion, and the sentiment polarity.

With the advent of pre-trained generative models, methods such as GAS (Zhang et al., 2021b) and OTG (Bao et al., 2022) have been developed to address ASQP in an end-to-end manner, leveraging the power of generative models to predict all components simultaneously (Ma et al., 2024). These approaches reformulate ASQP as a sequence-to-sequence problem, allowing the model to predict all elements simultaneously. Recently, the rise of LLMs has further advanced ASQP. Previous work, such as Zhang et al. (2024b), utilized LLMs as scoring mechanisms to generate pseudo-labeled data for data augmentation. In our work, we simplify this process. Since LLMs are already strong scorers (Zhang et al., 2024b), we propose directly leveraging them with the guide of supervised SLMs to predict the final answers without additional domain-specific, unlabeled data and increases computational costs to retrain the supervised model.

2.2 In-context Learning Methods

In-Context Learning (ICL) is a practical approach for using LLMs like GPT-4 in tasks with limited labeled data (Brown et al., 2020; Kojima et al., 2022). By providing examples directly in the input, ICL allows the model to make predictions without needing to retrain, making it useful for zero-shot and few-shot tasks. However, ASQP introduces unique challenges. It requires identifying complex relationships, following predefined

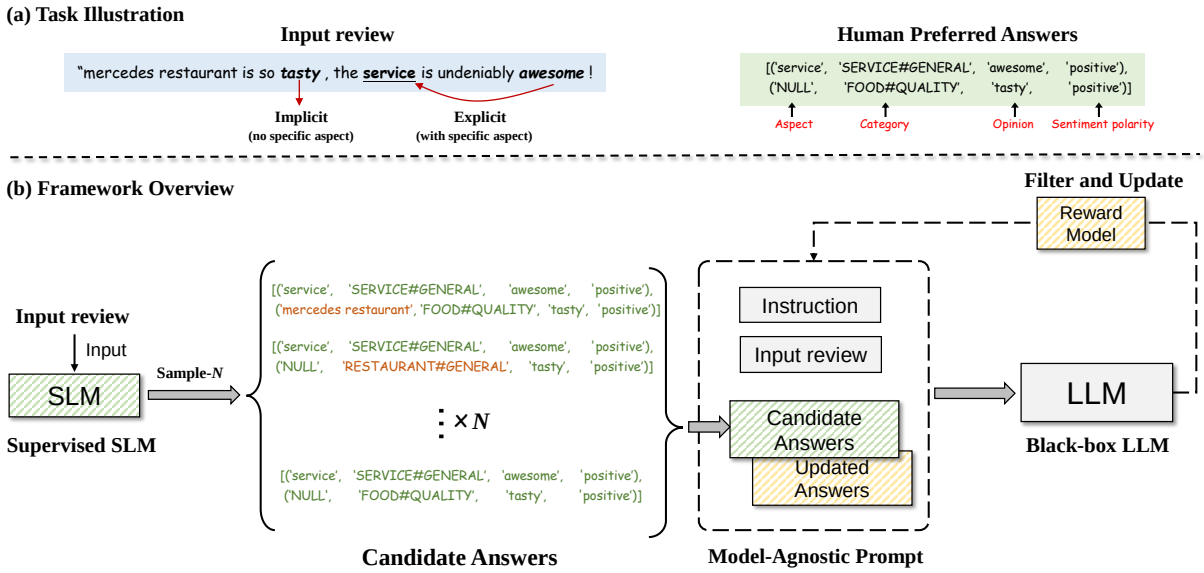

Figure 2: Illustration about the ASQP task and our framework. (a) shows a case of the ASQP task; (b) demonstrates the flowchart of our proposed framework for aligning a black-box LLM through supervised SLMs for ASQP.

categories, and ensuring outputs match human annotations. Simply applying ICL often produces inconsistent predictions because it depends heavily on the model's existing knowledge, which may not be well-suited to the task (Zhang et al., 2024a). Recent advancements, such as retrieval-augmented generation (Lewis et al., 2020; Liu et al., 2022) and knowledge-enhanced context methods (Yang et al., 2024b; Ma et al., 2023; Xu et al., 2024; Shen et al., 2023) address this by integrating task-specific knowledge retrieval into ICL, improving alignment with human preferences. These developments highlight promising directions for enhancing LLM-based in-context learning in complex structured prediction tasks.

Our framework differs from traditional ASQP and ICL methods by combining LLMs for prediction with supervised SLMs for dynamic guidance. Unlike ASQP methods that use small models (e.g., T5) for prediction and large models only for offline data augmentation, our framework enables dynamically interaction. In contrast to ICL methods that rely on static prompts without supervision, our framework provides adaptive inference.

3 Methods

In this section, we first introduce the aspect sentiment quad prediction problem definition, then quantitatively analyze the zero-shot black-box LLMs compared with supervised SLM. Finally, based on the insights of the analysis, we explore aligning the black-box LLMs through supervised and reinforcement-enhanced context for aspect sentiment quad prediction as shown in Figure 2(b).

3.1 Problem Definition

Aspect sentiment quad prediction is a fine-grained task in aspect-based sentiment analysis that aims to extract and classify quadruples. Formally, given an input text $T = \{w_1, w_2, \ldots, w_s\}$, where w_i represents the i-th token in a sequence of s tokens, the aspect sentiment quad prediction task aims to extract a set of quadruples:

$$\mathcal{Q} = \{(a_i, c_i, o_i, s_i) \mid i = 1, 2, \ldots, q\}, \quad (1)$$

where a_i is the aspect term, c_i is the predefined category, o_i is the opinion term, and $s_i \in \{\text{positive}, \text{neutral}, \text{negative}\}$ is the sentiment polarity associated with the aspect. The number of quadruples q depends on the content of the input text. If a_i and o_i are implicit, then $a_i = \text{NULL}$ and $o_i = \text{NULL}$. The ASQP task requires a model to predict the set $\mathcal{Q}$ for any given input text T while maintaining alignment between the extracted aspects, categories, opinions, and sentiments as shown in Figure 2(a).

3.2 Zero-Shot LLMs vs. Supervised SLMs

This section provides a quantitative analysis of the phenomenon that LLMs struggle with task definitions, often producing incorrect outputs, while supervised small models perform better, and sam-

pling multiple outputs improves the chances of getting the correct answer.

We use Top@1 and Top@10 as evaluation metrics to compare the effectiveness of SLMs with zero-shot LLMs. Top@1 measures the F1-score of the model's first prediction, while Top@10 considers whether the correct answer is present within the top 10 predictions. Figure 3 compares the average F1-score of zero-shot LLM, Top@1 (SLM), and Top@10 (SLM) across various datasets. Specifically, we select two powerful foundation models, as shown in Figure 3 (a) and (b), which can be deployed on a consumer-grade GPU as the SLM, while using the commonly adopted GPT-4o-mini as the LLM. The results show that zero-shot LLM perform poorly, while Top@1 predictions from supervised SLMs provide moderate improvements. In contrast, Top@10 predictions from SLMs achieve significantly higher F1-scores, aligning with earlier observations. This finding demonstrates that SLMs, by generating multiple outputs, can effectively identify the correct answer and better align with human preferences. These observations motivate us to explore a more effective approach for aligning LLM outputs with human-preferred answers (pre-defined gold answers) in the ASQP task through the supervised SLM. The observed performance gap between Top@1 and Top@10 in the supervised SLM paves a potential avenue.

Through the SLM's candidate answers, we can transform the open-domain QA format (i.e., asking the LLM for an answer) into a new format where the LLM is prompted with the SLM generated potential answers and asked to select one. Furthermore, the potential answers generated by the SLM inherently contain strong human-preference signals, which can serve as a valuable alignment mechanism, helping to bridge the gap between the LLM's output and human-preferred answers. Thus, we are motivated to align the strong black-box LLM through in-context learning by utilizing the SLM generated candidate answers and propose the two stage framework.

3.3 Stage-1: Supervised Context

In-context learning refers to the capability of a model to infer patterns or generate answers based on the input prompt, without explicit parameter updates. Our framework begins by training a supervised SLM on labeled data to learn human preferences and act as a candidate answers generator of the given samples. Specifically, we fine-tune

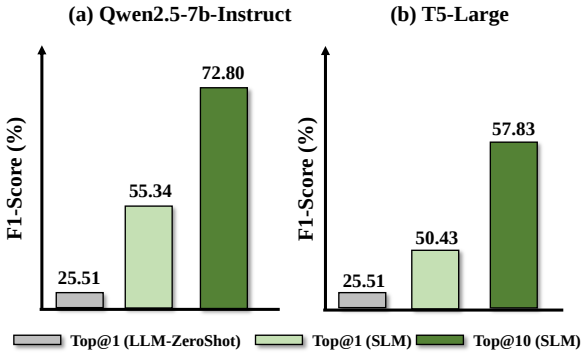

Figure 3: Analysis of the supervised SLMs using Top@1 and Top@10 metrics.

the SLM with supervision and then use it to perform multiple samplings with a high-temperature setting. The sampled outputs are subsequently used as candidate answers for in-context learning in a black-box LLM.

Formally, let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{D}$ represent the test dataset, where x_i is an input, and y_i is the corresponding ground truth. Given an test input x_i, a supervised SLM generates candidate answers $A_i = \{a_i^1, \ldots, a_i^n\}$, where $i \in [1..|D|]$ and $n \in [1..N]$. N is empirically set as 10. Then, the probability of the LLM generating the answer y_i by our method is defined as:

$$p_{\text{LLM}}\left(y_i \mid I^*, A_i, x_i\right), \tag{2}$$

where I^* represents a specific instruction guiding the LLM to identify the most suitable answer.

3.4 Stage-2: Reinforcement-Enhanced Context

Reinforcement learning methods, such as Direct Preference Optimization (Rafailov et al., 2023) and Proximal Policy Optimization (Schulman et al., 2017), have demonstrated their effectiveness in fine-tuning models based on reward signals (Ouyang et al., 2022; Anthropic, 2024). However, these approaches require updating model parameters, making them unsuitable for black-box LLMs where parameter access is restricted.

To address this limitation, we propose to build reinforcement-enhanced context. Specifically, we first train a reward model to guide the reinforcement process. The training data for the reward model is constructed as follows: we use the supervised SLM trained in the initial stage to perform sampling on the training set to generate candidate answers. These candidates are then provided to the

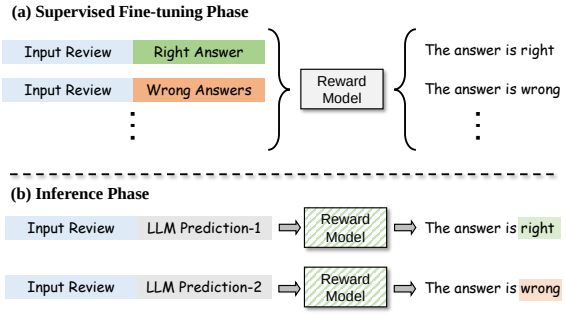

Figure 4: Illustration of our proposed reward model: (a) Supervised training phase, where input reviews come from the training dataset; (b) Inference phase, where input reviews come from the test dataset.

black-box LLM for predictions on the training set. Incorrect answers from the LLM are paired with the corresponding ground-truth answers to form a labeled dataset containing both positive and negative examples, which is subsequently used to train the reward model.

Once the reward model is trained, it evaluates the outputs of the black-box LLM to refine candidate answers. Formally, for each test input x_i, the LLM generates multiple predictions $M = \{y_i^1, \dots y_i^m\}$. The reward model then evaluates each pair $\{x_i, y_i^m\}$ and assigns a reward score $R_\theta(x_i, y_i^m)$ for each $y_i^m \in M$, where $|M|$ is empirically set as 10.

In stage-2, the reward model outputs statements such as "the answer is X." and we focus on the logit at the position X corresponding to the word "right" to determine correctness. Specifically, we maintain an answer only when the logit of the word "right" exceeds 0.8. This threshold (τ = 0.8) is empirically set to ensure that the reward model confidently recognizes the answer as correct. If the logit is below 0.8, we classify the answer as incorrect and disregard it as shown in Figure 4.

Formally, the new candidates with scores above a predefined threshold τ is defined:

$$a_i^m = \{y_i^m \mid R_\theta(\{x_i, y_i^m\}) \geq \tau\}, \quad (3)$$

where τ is the predefined threshold.

The selected candidates are added to the in-context candidate answers A_i^*, forming an updated candidate set:

$$A_i^* = \{a_i^1, \dots, a_i^n, a_i^{n+1}, \dots, a_i^{|M|}\}. \quad (4)$$

This process is iterative, refining the candidate examples over multiple steps to improve alignment with the task objectives. At each iteration t, the candidate set is updated as follows:

$$A_i^*[t+1] = A_i^*[t] \cup \{a_i^{T[t]+1}[t], a_i^{T[t]+2}[t], \dots, a_i^{T[t]+\Delta T[t]}[t]\}, \quad (5)$$

where $T[t]$ is the number of candidates at iteration t, and $\Delta T[t]$ represents the number of newly selected candidates in that iteration.

The newly selected candidates at iteration t are defined as:

$$a_i^k[t] = \{y_i^* \mid R_\theta(\{x_i, y_i^*\}) > \tau\}, \quad \forall k \in \{T[t]+1, \dots, T[t]+\Delta T[t]\}. \quad (6)$$

Finally, the probability of the LLM generating the correct answer y_i under the refined candidate set is defined as:

$$p_{\text{LLM}}(y_i \mid I^*, A_i^*[\mathcal{X}], x_i), \quad (7)$$

where $\mathcal{X}$ denotes the total number of iterations, empirically set to 2.

Datasets	Train		Dev		Test	
	#S	#Q	#S	#Q	#S	#Q
ACOS-Laptop	2934	4172	326	440	816	1161
ACOS-Rest	1530	2484	171	261	583	916
Rest-15	834	1354	209	347	537	795
Rest-16	1264	1989	316	507	544	799

Table 1: Statistics of four ASQP datasets (Cai et al., 2021a; Zhang et al., 2021a). #S and #Q represent the number of sentences and quads.

4 Experiments

In this section, we introduce our experimental setup and implementation details, present our frameworks' performance on several standard datasets compared to competitive baselines.

4.1 Setup

We conduct experiments on four aspect sentiment quad prediction datasets: ACOS-Laptop, ACOS-Restaurant, Rest15, and Rest16. These datasets are based on the SemEval Challenges (Pontiki et al., 2015, 2016), while the quad-level annotations are introduced in Cai et al. (2021a) and Zhang et al. (2021b). Table 1 provides detailed statistics for each dataset, including the number of sentences (S) and quads (Q) in the train, development, and test splits.

Methods	LLMs	F1-score (↑)				
		ACOS-Rest	ACOS-Laptop	Rest-15	Rest-16	Avg.
In-context Learning						
ZERO-SHOT (Brown et al., 2020)	GPT4O-MINI	31.28	11.18	25.24	34.31	25.50
ZERO-SHOT COT (Kojima et al., 2022)	GPT4O-MINI	23.01	7.56	21.55	26.73	19.71
FEW-SHOT (N=5) (Brown et al., 2020)	GPT4O-MINI	32.76	13.69	30.28	35.39	28.03
MAJORITY-VOTE (N=5, K=8)	GPT4O-MINI	34.09	15.22	31.62	36.40	29.33
RETRIEVAL-AUGMENTED (N=5)	GPT4O-MINI	42.15	21.87	38.46	41.27	35.94
Supervised Learning w/o LLM						
EXTRACT-CLASSIFY (Cai et al., 2021a)	—	38.54	35.80	52.96	44.61	42.98
GAS (Zhang et al., 2021b)	—	58.63	43.07	46.57	57.55	51.46
DLO (Hu et al., 2022b)	—	59.18	43.60	48.48	59.79	52.76
ILO (Hu et al., 2022b)	—	58.69	44.35	49.05	59.32	52.85
MVP (Gou et al., 2023)	—	61.54	43.92	51.04	60.39	54.22
MUL (Hu et al., 2023)	—	60.53	44.01	49.75	60.47	53.69
Supervised Learning w/ LLM						
SCORER (Zhang et al., 2024b)	GPT4	62.47	**46.01**	51.74	63.51	56.41
SUPERCONTEXT* (Yang et al., 2024b)	GPT4O	61.43	41.28	52.17	62.48	54.34
SCORER-GAS (Zhang et al., 2024b)	GPT4	61.44	45.19	50.38	61.08	54.52
SCORER-GAS*	GPT4	61.28	44.57	49.63	60.49	53.99
+ STAGE-1	GPT4O	62.86	<u>45.72</u>	51.88	61.77	55.56
+ STAGE-1	GPT4O-MINI	62.50	44.87	51.56	61.58	55.13
+ STAGE-1 & STAGE-2	GPT4O-MINI	63.58	45.22	52.34	62.88	56.01
QWEN2.5-7B-INSTRUCT (Yang et al., 2024a)	—	62.03	43.12	52.89	63.30	55.34
+ STAGE-1	GPT4O	<u>64.67</u>	44.78	<u>54.22</u>	<u>65.37</u>	<u>57.09</u>
+ STAGE-1	GPT4O-MINI	64.41	43.48	53.85	64.97	56.68
+ STAGE-1 & STAGE-2	GPT4O-MINI	**66.78**	45.68	**55.94**	**66.83**	**58.81**

Table 2: Performance comparison of different methods on ACOS-Rest, ACOS-Laptop, Rest-15, and Rest-16 datasets. The final column shows the average F1-Score across all datasets. * denotes the method we reproduced.

In this section, we select two powerful foundation models (T5-large, Qwen-2.5) that can be deployed on a consumer-grade GPU (e.g., NVIDIA RTX 3090, 4090) as the SLM. Specifically, we use SCORER-GAS (Zhang et al., 2024b), which is based on T5-large and additionally trained with pseudo-labeled data generated by the LLMs, and 7B-Instruct version (Yang et al., 2024a) for LoRA-based (Hu et al., 2022a) supervised fine-tuning. Additionally, the 0.5B-Instruct version is full-parameter fine-tuned to serve as the reward model. For black-box LLMs, we include the commonly used GPT-4o and GPT-4o-mini. Since the order of options may influence the experimental results (Pezeshkpour and Hruschka, 2024), we report results averaged over three runs for experiments involving candidate selection, with the candidate answers randomly shuffled in each run. Thus, due to resource constraints, we perform the complete experimental pipeline only on GPT-4o-mini.

For baseline comparison, we evaluate several commonly used supervised learning methods (Yang et al., 2024a; Cai et al., 2021a; Hu et al., 2022b; Gou et al., 2023; Hu et al., 2023) as well as in-context learning techniques (Brown et al., 2020; Wang et al., 2023; Kojima et al., 2022; Liu et al., 2022; Yang et al., 2024b). The baseline results in the supervised learning w/o LLM section are derived from Zhang et al. (2024b).

4.2 Main Results

As shown in Table 2, simply relying on in-context learning fails to effectively guide LLMs output answers that align with human expectations and thus have a poor performance. On the other hand, supervised fine-tuning with human-annotated labels allows models to learn the preferred types of predictions efficiently, resulting in better performance. Moreover, leveraging supervised models and LLMs leads to further improvements, highlighting the potential of leveraging LLMs for this task. Notably, SCORER-GAS based on our framework achieves competitive results with SCORER (Zhang et al., 2024b) relying on GPT-4o-mini and requiring no additional data for training a AI-reranker. Furthermore, as task performance improves, achieving further gains becomes increasingly difficult. While Qwen2.5-Instruct already performs well, our frame-

Methods	Rest	Laptop	Rest15	Rest16
Zero-shot	28.74	10.18	24.33	28.16
Same	62.14	43.10	53.12	63.81
Ours ($N = 5$)	63.78	42.82	53.35	64.12
Ours ($N = 10$)	64.41	**43.48**	**53.85**	**64.97**
Ours ($N = 20$)	**64.52**	43.44	53.67	64.88

Table 3: Performance comparison of different methods across datasets. Bold values indicate the best performance for each dataset. N denotes the number of candidate answers generated by SLM.

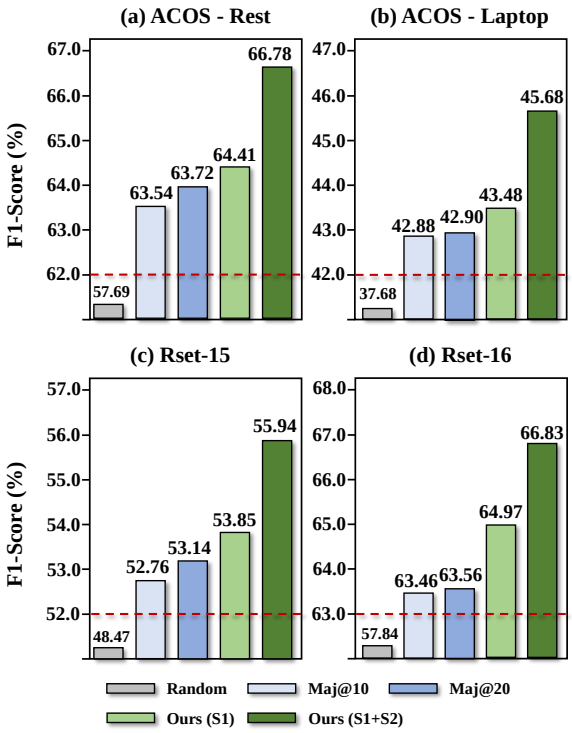

Figure 5: Influence of different answer selection strategies. 'Maj' refers to the Majority-vote strategy, where 'Maj@10' indicates the selection of the majority answer from 10 candidate answers.

work enhances it even further, demonstrating the effectiveness of our framework. Surprisingly, SuperContext (Yang et al., 2024b) still has certain limitations in addressing the ASQP task. A possible reason is that it provides only a single answer generated by the SLM in the prompt, leaving the search space for the LLM too broad. The results indicate that our framework by integrating supervised signals of SLM into context for guiding black-box LLMs demonstrates significant improvements and paves a new way to combine LLMs and supervised SMLs for ASQP.

5 Analysis and Discussion

The experiments showed that while Qwen2.5 had a Top@1 performance similar to SCORER-GAS, its larger gap between Top@1 and Top@10 led to better results within our framework. Moreover, leveraging PEFT techniques like LoRA (Hu et al., 2022a) and well-deigned inference framework like vLLM (Kwon et al., 2023), it can achieve T5-large-level GPU efficiency. Therefore, we consider it a more promising SLM backbone and select it for further analysis in this section.

5.1 Impact of Candidate Answer Strategy

We evaluate the impact of different candidate answer strategies on model performance, as shown in Table 3. The methods include Zero-shot (no options), Same (replicates Top@1 prediction at 10 times), and Ours. The results show that methods with candidate answers outperform Zero-shot, highlighting that providing supervised context helps narrow the search space of LLMs, leading to more accurate predictions. Compared to $N = 5$ and $N = 20$, $N = 10$ achieved relatively better results. Therefore, we select the number of candidate answers to ten. The reason why the performance does not improve significantly when increasing N beyond ten might be that the outputs of the SLM remain highly similar. Additionally, a longer prompt resulting from larger number of candidate answers could negatively impact the LLM's judgment.

5.2 Influence of Answer Selection Strategy

In this section, we compare our proposed framework with different answer selection strategies. Random selects the final answer randomly from the candidate answers. The majority-vote approach selects the answer that appears most frequently among the candidates. our framework uses supervised context (S1) and reinforcement-enhanced context (S1&S2) along with LLMs to make the final prediction.

Majority-vote is a simple but effective baseline, as it aggregates repeated predictions to reflect the performance of the supervised model. As demonstrated in Figure 5, majority-vote achieves strong results across all datasets, while increasing the number of aggregated predictions from Maj@10 to Maj@20 only lead to a marginal improvement. However, our method surpasses majority-vote by combining supervised context and reinforcement-enhanced context with LLMs. This highlights the advantages of our framework.

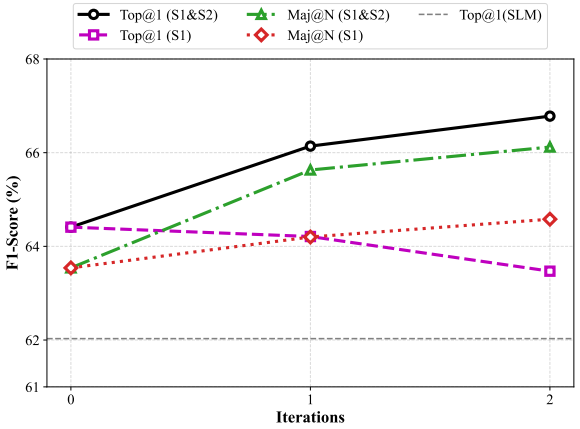

Figure 6: Analysis of the iterative alignment of the LLM. Maj@N indicates the selection of the majority answer from the current candidate answer pool. S1 and S2 denotes stage-1 and stage-2.

5.3 Analysis of Iterative Alignment

In this section, we utilize the ACOS-Rest dataset as the benchmark to explore the impact of reinforcement context on LLM outputs across multiple iterations.

As shown in Figure 6, methods incorporating Stage-2 consistently outperform other methods at each iteration. Furthermore, as the Reinforcement Context evolves in each iteration, both the Maj@N (S1&S2) and Maj@N (S1) improve, indicating that our framework effectively increases the proportion of correct answers among the candidate answers. However, the gap between Maj@N (S1&S2) and Maj@N (S1) suggests that there are still many uncertain candidates. Our proposed reward model helps filter out these uncertain candidate answers, leading to better performance in Maj@N (S1&S2). Notably, skipping the Stage-2 leads to performance degradation in Top@1. This is likely due to low-precision candidate answers affecting LLM judgment or rapid context growth causing the LLM misunderstanding of the instructions.

5.4 Case Study

In this section, we present a case study, where Table 4 illustrates one of the most common cases in which the proposed framework has led to improvements. Additionally, more detailed examples and error analysis can be found in Appendix A.3. As shown in Table 4, both the zero-shot method and Ours (Stage-1) produced outputs that appeared correct but did not align with human preferences. However, by leveraging a reward model to construct a refinement context, our framework was able to preserve more of the correct answers in the candidate pool. This iterative refinement process enabled the model to correct the error and output the correct prediction, thus demonstrating the effectiveness of our framework.

Case Study
ZERO-SHOT
Task Definition: {Task Definition}
Input: *This is a great place to get a delicious meal*
Final output:
((meal, FOOD#QUALITY, delicious, positive), (a place, RESTAURANT#GENERAL, great, positive)) ✗
OURS (Stage-1)
Task Definition: {Task Definition}
Input: *This is a great place to get a delicious meal*
Candidate answers:
a) ((meal, FOOD#QUALITY, delicious, positive) (place, RESTAURANT#GENERAL, great, positive))*3
b) (meal, FOOD#QUALITY, delicious, positive) *4
c) (place, RESTAURANT#GENERAL, great, positive) *3
Final output:
(place, RESTAURANT#GENERAL, great, positive) ✗
OURS (Stage-1 & Stage-2)
Task Definition: {Task Definition}
Input: *This is a great place to get a delicious meal*
Candidate answers:
a) ((meal, FOOD#QUALITY, delicious, positive) (place, RESTAURANT#GENERAL, great, positive))*11
b) (meal, FOOD#QUALITY, delicious, positive) *5
c) (place, RESTAURANT#GENERAL, great, positive) *8
Final output:
((meal, FOOD#QUALITY, delicious, positive), (place, RESTAURANT#GENERAL, great, positive)) ✓

Table 4: An example of case study. *X (e.g., *3) indicate the number of identical candidate answers.

6 Conclusion

We proposes a framework that integrates supervised SLMs with black-box LLMs to address the challenges of aspect sentiment quad prediction. Motivated by the complementary strengths of SLMs in capturing task-specific knowledge and LLMs in generalization, we designed a framework to align LLM outputs with human preferences through in-context learning and iterative refinement. Experimental results demonstrate that our framework significantly improves aspect sentiment quad prediction performance compared with in-context learning and supervised learning methods. In the future, we will explore extending this alignment framework to other fine-grained sentiment analysis tasks and further enhancing its adaptability to diverse datasets and tasks.

7 Limitations

Despite its effectiveness, our framework has certain limitations. First, the in-context learning process heavily relies on carefully designed instructions and high-quality candidate answers from SLMs. If these inputs are not well-crafted, the LLMs may fail to align with human preferences, leading to suboptimal performance. Second, while the framework allows flexible combinations of large and small models, it is not effective when the smaller model is underperforming, such as a simple perceptron, which cannot provide meaningful guidance for alignment. Future research could explore optimizing the efficiency of in-context learning setups and developing methods to enhance the robustness of instruction designs.

References

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Xiaoyi Bao, Xiaotong Jiang, Zhongqing Wang, Yue Zhang, and Guodong Zhou. 2023. Opinion tree parsing for aspect-based sentiment analysis. In *Findings of ACL*.

Xiaoyi Bao, Zhongqing Wang, Xiaotong Jiang, Rong Xiao, and Shoushan Li. 2022. Aspect-based sentiment analysis with opinion tree generation. In *Proceedings of IJCAI*.

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2022. PADA: example-based prompt learning for on-the-fly adaptation to unseen domains. *Trans. Assoc. Comput. Linguistics*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of NeuralIPS*.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021a. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of ACL*.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021b. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of ACL*.

Zhibin Gou, Qingyan Guo, and Yujiu Yang. 2023. MvP: Multi-view prompting improves aspect sentiment tuple prediction. In *Proceedings of ACL*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Mengting Hu, Yinhao Bai, Yike Wu, Zhen Zhang, Liqi Zhang, Hang Gao, Shiwan Zhao, and Minlie Huang. 2023. Uncertainty-aware unlikelihood learning improves generative aspect sentiment quad prediction. In *Findings of ACL*.

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022b. Improving aspect sentiment quad prediction via template-order data augmentation. In *Proceedings of EMNLP*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of NeuralIPS*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS*.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of NeurIPS*.

Shichen Li, Zhongqing Wang, Xiaotong Jiang, and Guodong Zhou. 2022. Cross-domain sentiment classification using semantic representation. In *Findings of EMNLP*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of DeeLIO*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Tianlai Ma, Zhongqing Wang, and Guodong Zhou. 2024. Transition-based opinion generation for aspect-based sentiment analysis. In *Findings of ACL*.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of EMNLP*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arxiv:2203.02155*.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of NAACL*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of NAACL-HLT*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of NAACL-HLT*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of NeurIPS*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. In *Proceedings of ICLR*.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving AI tasks with chatgpt and its friends in huggingface. In *Proceedings of NeurIPS*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of ICLR*.

Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley. 2024. Small models are valuable plug-ins for large language models. In *Findings of ACL*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Wei Ye, Xing Xie, Weizhu Chen, and Yue Zhang. 2024b. Supervised knowledge makes large language models better in-context learners. In *Proceedings of ICLR*.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of EMNLP*.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2024a. Sentiment analysis in the era of large language models: A reality check. In *Findings of NAACL*.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative aspect-based sentiment analysis. In *Proceedings of ACL*.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2023. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge & Data Engineering*.

Yice Zhang, Jie Zeng, Weiming Hu, Ziyi Wang, Shiwei Chen, and Ruifeng Xu. 2024b. Self-training with pseudo-label scorer for aspect sentiment quad prediction. In *Proceedings of ACL*.

Zheng Zhang, Zili Zhou, and Yanna Wang. 2022. SSEGCN: syntactic and semantic enhanced graph convolutional network for aspect-based sentiment analysis. In *Proceedings of NAACL*.

A Experiment Details

A.1 Prompt Details

This section provides details about the prompts used in our experiments, covering both zero-shot and few-shot settings for the laptop and restaurant domain.

The specific prompts are presented in Table 9, Table 10, Table 11, Table 12. Since the few-shot and zero-shot methods lack sufficient knowledge of ASQP, we incorporate additional knowledge and examples to provide a more comprehensive understanding of the ASQP task. For the Retrieval-Augmented method, we used the LangChain framework to implement the process. Specifically, LangChain was utilized to build a retrieval pipeline, where a dense retriever searched for relevant labeled examples from training dataset. The retrieved examples are then integrated into the prompt to guide the generation process. For the CoT method, we followed prior studies (Kojima et al., 2022) and added "Let's think step by step" after the zero-shot prompt. Additionally, our method incorporates candidate answers generated by the supervised model after the zero-shot prompt as shown in Table 13 and Table 14.

A.2 Implementation Details

In our experiments, all language models used a temperature of 0.7 for Top@10 candidate answers sampling and 0.2 for Top@1 candidate answers sampling. To enhance diversity in LLM-generated outputs, we slightly adjusted the generation order format of ASQP quads in the instruction during sampling, inspired by the previous work (Hu et al., 2022b).

We employ Qwen2.5-7B-Instruct (Yang et al., 2024a) and SCORER (Zhang et al., 2024b) as our primary supervised SLM and Qwen2.5-0.5B-Instruct act as the reward model. AdamW (Loshchilov and Hutter, 2018) is used as the optimizer, with a learning rate of 1×10^{-4} for LoRA-based supervised fine-tuning (Hu et al., 2022a) and 1×10^{-5} for the full-parameters supervised fine-tuning. During training, we employ early stopping based on the development set performance

Previous research (Pezeshkpour and Hruschka, 2024) has shown that the performance of LLMs in multiple-choice tasks can be influenced by the order of options. Therefore, for experiments involving candidate selection, we report results averaged over three runs, with the candidate answers randomly shuffled in each run.

Case Study

ZERO-SHOT
Task Definition: {Task Definition}
Input: *The owner is belligerent to guests that have a complaint.*
Final output:
(('owner', 'SERVICE#GENERAL', 'belligerent', 'negative'),
('guests', 'SERVICE#GENERAL', 'NULL', 'negative'),
('complaint', 'SERVICE#GENERAL', 'NULL', 'negative')) ✗

OURS (Stage-1)
Task Definition: {Task Definition}
Input: *The owner is belligerent to guests that have a complaint.*
Candidate answers:
a) ('owner', 'SERVICE#GENERAL', 'belligerent', 'negative')
Final output:
('owner', 'SERVICE#GENERAL', 'belligerent', 'negative') ✓

OURS
Task Definition: {Task Definition}
Input: *The owner is belligerent to guests that have a complaint.*
Candidate answers:
a) ('owner', 'SERVICE#GENERAL', 'belligerent', 'negative')
Final output:
('owner', 'SERVICE#GENERAL', 'belligerent', 'negative') ✓

Table 5: An example of case study.

A.3 Case Study

In this section, we present anther case studies to analyze our framework. In the table, symbols like ***X** (e.g., ***2**) indicate the number of identical candidate answers.

As shown in Table 5, the zero-shot method successfully followed the instructions and generated a response. However, it produced additional, unnecessary answers that did not align with human preferences. Our method, on the other hand, produced the correct result with the guidance of the supervised candidate answers.

As shown in Table 7, both the zero-shot method and Ours (Stage-1) produced outputs that appeared correct but did not align with human preferences. They both over-predicted the adverbs, failing to produce the human preferred correct result.

As shown in Table 8, we conducted an error analysis, and the results showed that all methods, including the supervised learning model, produced incorrect answers. Interestingly, the number of candidate answers generated by the supervised model indicated that even with supervision, the model's output remained uncertain. Furthermore, we observed that LLMs tend to exhibit bias when analyzing neutral sentiment, which aligns with findings from previous studies. Notably, our framework, af-

Methods	Rest	Laptop	Rest15	Rest16
Top@1	64.59	65.09	52.18	63.53
Top@10	72.56	69.08	57.90	67.47

Table 6: Top@1 and Top@10 results on the ASTE task

ter several rounds of reward model refinement, was able to explore the candidate answers but failed to identify the correct one.

B Exploring the Generalization Ability

B.1 Generalization to ASTE task

Our study primarily investigates the impact of increasing the number of samples on tasks such as Aspect Sentiment Quad Prediction (ASQP). However, we have also observed a similar trend in the Aspect Sentiment Triplet Extraction (ASTE) task, further reinforcing the generalization of our findings in section 3.2.

In the ASTE task, we refine the ASQP dataset to focus solely on aspect, opinion, and sentiment polarity, deliberately excluding the category attribute. Our experimental results Shown in Table 6 indicate that increasing the number of samples enhances the likelihood of correctly capturing the desired triplets. This trend aligns with our findings in ASQP, suggesting that the observed performance improvement is not limited to a single task but can extend to other sentiment analysis tasks.

In this section, we utilize the Qwen2.5-7B-Instruct (Yang et al., 2024a) model as the base model, further fine-tuned with LoRA (Hu et al., 2022a) under supervised learning settings. The observed improvements across different datasets support our hypothesis that increasing sample size consistently enhances extraction performance, reinforcing the broader applicability of our framework to other tasks.

B.2 Analysis of Different Model Combinations

we evaluate the scalability of our framework by experimenting with various base supervised models and LLMs. Specifically, we utilize the supervised SLMs with parameters of 0.5B, 1.5B, and 7B (LoRA) and pair them with different LLMs, including Qwen2.5-32B-Instruct, GPT4o-mini, and GPT-4o.

As illustrated in Figure 7, our framework consistently demonstrates improved performance as

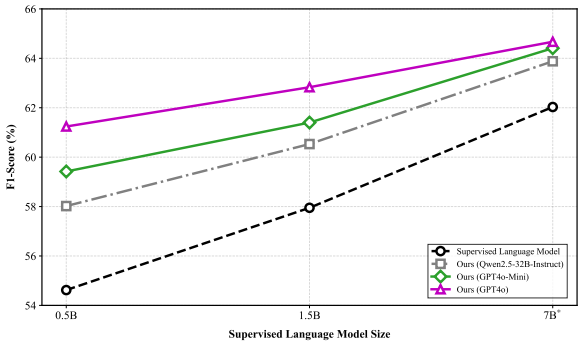

Figure 7: Illustration of the Top@1 performance of our proposed framework, composed of different model sizes

the size and capability of the supervised SLMs and LLMs increase. Specifically, transitioning from a 0.5B to a 7B supervised SLM results significantly boosts prediction F1-score across all datasets. Similarly, upgrading the LLM from Qwen2.5-32B-Instruct to GPT-4o yields further performance improvements. Experimental results show that as the performance of the SLM improves, our framework consistently achieves better results. Additionally, with the enhancement of LLMs, the performance of our framework also improves accordingly. These findings highlight the high scalability of our framework and pave the way for future research.

Case Study

ZERO-SHOT
Task Definition: {Task Definition}
Input: *Serves really good sushi.*
Final output:
('sushi', 'FOOD#QUALITY', 'really good', 'positive') ✗

OURS (Stage-1)
Task Definition: {Task Definition}
Input: *Serves really good sushi.*
Candidate answers:
a) ('sushi', 'FOOD#QUALITY', 'really good', 'positive') *6
b) ('sushi', 'FOOD#GENERAL', 'good', 'positive') *2
c) ('sushi', 'FOOD#QUALITY', 'good', 'positive') *1
d) ('sushi', 'FOOD#GENERAL', 'really good', 'positive') *1
Final output:
('sushi', 'FOOD#QUALITY', 'really good', 'positive') ✗

OURS (Stage-1 & Stage-2)
Task Definition: {Task Definition}
Input: *Serves really good sushi.*
Candidate answers:
a) ('sushi', 'FOOD#QUALITY', 'really good', 'positive') *8
b) ('sushi', 'FOOD#GENERAL', 'good', 'positive') *2
c) ('sushi', 'FOOD#QUALITY', 'good', 'positive') *14
d) ('sushi', 'FOOD#GENERAL', 'really good', 'positive') *1
Final output:
('sushi', 'FOOD#QUALITY', 'good', 'positive') ✓

Table 7: An example of case study.

Case Study

GOLD ANSWER:
(('portions', 'FOOD#STYLE_OPTIONS', 'not the biggest', 'negative'),
('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'neutral')) ✓

ZERO-SHOT
Task Definition: {Task Definition}
Input: *not the biggest portions but adequate .*
Final output:
('portions', 'FOOD#GENERAL', 'not the biggest', 'negative') ✗

OURS (w/o reward)
Task Definition: {Task Definition}
Input: *not the biggest portions but adequate .*
Candidate answers:
a) ('portions', 'FOOD#STYLE_OPTIONS', 'biggest', 'neutral') *1
b) ('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'neutral') *3
c) ('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'positive') *2
d) ('portions', 'FOOD#STYLE_OPTIONS', 'biggest', 'negative') *2
e) (('portions', 'FOOD#STYLE_OPTIONS', 'not', 'negative')
('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'neutral')) *2
Final output:
('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'neutral') ✗

OURS
Task Definition: {Task Definition}
Input: *not the biggest portions but adequate .*
Candidate answers:
a) ('portions', 'FOOD#STYLE_OPTIONS', 'biggest', 'neutral') *1
b) ('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'neutral') *16
c) ('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'positive') *2
d) ('portions', 'FOOD#STYLE_OPTIONS', 'biggest', 'negative') *2
e) (('portions', 'FOOD#STYLE_OPTIONS', 'not', 'negative')
('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'neutral')) *3
f) (('portions', 'FOOD#STYLE_OPTIONS', 'not the biggest', 'neutral')
('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'neutral')) *2
Final output:
('portions', 'FOOD#STYLE_OPTIONS', 'adequate', 'neutral') ✗

Table 8: An example case of error analysis

Zero-shot example of restaurant domain

Task Definition:
Aspect-Based Sentiment Analysis aims to extract the opinion target described by an entity and its aspect (collectively called aspect) from reviews and identify the sentiment toward the aspect.
Pre-defined Categories: Categories must follow the A#B format, where A is one of ['RESTAURANT', 'DRINKS', 'SERVICE', 'FOOD', 'AMBIENCE', 'LOCATION'], and B is one of ['GENERAL', 'STYLE_OPTIONS', 'QUALITY', 'PRICES', 'MISCELLANEOUS']. Each category must strictly adhere to these sets, e.g., FOOD#QUALITY.
Input
Instruction: From the restaurant review, identify all aspects, their opinion words, category, and sentiment ('positive', 'negative', 'neutral').
Answer Format: Your final answer can include multiple aspect-opinion pairs, formatted as follows:
"Final Answer: ['aspect_term$_1$', 'category$_1$', 'opinion_term$_1$', 'sentiment$_1$'], ['aspect_term$_2$', 'category$_2$', 'opinion_term$_2$', 'sentiment$_2$'] ...".If an aspect or opinion term is implicit, use 'NULL' to represent it.
Input: "Yum !"

Table 9: Zero-shot example of restaurant domain

Few-shot example of restaurant domain

Task Definition:
Aspect-Based Sentiment Analysis aims to extract the opinion target described by an entity and its aspect (collectively called aspect) from reviews and identify the sentiment toward the aspect.
Pre-defined Categories: Categories must follow the A#B format, where A is one of ['RESTAURANT', 'DRINKS', 'SERVICE', 'FOOD', 'AMBIENCE', 'LOCATION'], and B is one of ['GENERAL', 'STYLE_OPTIONS', 'QUALITY', 'PRICES', 'MISCELLANEOUS']. Each category must strictly adhere to these sets, e.g., FOOD#QUALITY.
Examples
The following are several examples to help you learn how to extract quadruples:
Input: "after all that , they complained to me about the small tip ."
Final Answer: ['NULL', 'SERVICE#GENERAL', 'complained', 'negative']
Input: "food was okay , nothing great ."
Final Answer: ['food', 'FOOD#QUALITY', 'okay', 'neutral'], ['food', 'FOOD#QUALITY', 'nothing great', 'neutral']
Input: "i had to ask her three times before she finally came back with the dish ive requested ."
Final Answer: ['NULL', 'SERVICE#GENERAL', 'NULL', 'negative']
Input: "went on a 3 day oyster binge , with fish bringing up the closing , and i am so glad this was the place it o trip ended , because it was so great !"
Final Answer: ['fish', 'RESTAURANT#GENERAL', 'great', 'positive'], ['NULL', 'RESTAURANT#GENERAL', 'glad', 'positive']
Input: "ive asked a cart attendant for a lotus leaf wrapped rice and she replied back rice and just walked away ."
Final Answer: ['cart attendant', 'SERVICE#GENERAL', 'NULL', 'negative']
Input
Instruction: From the restaurant review, identify all aspects, their opinion words, category, and sentiment ('positive', 'negative', 'neutral').
Answer Format: Your final answer can include multiple aspect-opinion pairs, formatted as follows:
"Final Answer: ['aspect_term$_1$', 'category$_1$', 'opinion_term$_1$', 'sentiment$_1$'], ['aspect_term$_2$', 'category$_2$', 'opinion_term$_2$', 'sentiment$_2$'] ...".If an aspect or opinion term is implicit, use 'NULL' to represent it.
Input: "Yum !"

Table 10: Few-shot example of restaurant domain

Zero-shot example of Laptop domain

Task Definition:
Aspect-Based Sentiment Analysis aims to extract the opinion target described by an entity and its aspect (collectively called aspect) from reviews and identify the sentiment toward the aspect.
Pre-defined Categories: Categories must follow the A#B format, where A is one of ['LAPTOP', 'HARD_DISC', 'OS', 'KEYBOARD', 'HARDWARE', 'PORTS', 'SUPPORT', 'COMPANY', 'MULTIMEDIA_DEVICES', 'POWER_SUPPLY', 'DISPLAY', 'BATTERY', 'FANS&COOLING', 'CPU', 'MEMORY', 'WARRANTY', 'OPTICAL_DRIVES', 'GRAPHICS', 'SOFTWARE', 'SHIPPING', 'MOTHERBOARD', 'MOUSE', 'Out_Of_Scope'], and B is one of ['PRICE', 'DESIGN_FEATURES', 'OPERATION_PERFORMANCE', 'USABILITY', 'GENERAL', 'QUALITY', 'PORTABILITY', 'CONNECTIVITY', 'MISCELLANEOUS']. Each category must strictly adhere to these sets, e.g., LAPTOP#GENERAL.
Input
Instruction: From the laptop review, identify all aspects, their opinion words, category, and sentiment ('positive', 'negative', 'neutral').
Answer Format: Your final answer can include multiple aspect-opinion pairs, formatted as follows:
"Final Answer: ['aspect_term$_1$', 'category$_1$', 'opinion_term$_1$', 'sentiment$_1$'], ['aspect_term$_2$', 'category$_2$', 'opinion_term$_2$', 'sentiment$_2$'] ...".If an aspect or opinion term is implicit, use 'NULL' to represent it.
Input: "the unit cost $ 275 to start with , so it is not worth repairing ."

Table 11: Zero-shot example of laptop domain.

Few-shot example of laptop domain

Task Definition:
Aspect-Based Sentiment Analysis aims to extract the opinion target described by an entity and its aspect (collectively called aspect) from reviews and identify the sentiment toward the aspect.
Pre-defined Categories: Categories must follow the A#B format, where A is one of ['LAPTOP', 'HARD_DISC', 'OS', 'KEYBOARD', 'HARDWARE', 'PORTS', 'SUPPORT', 'COMPANY', 'MULTIMEDIA_DEVICES', 'POWER_SUPPLY', 'DISPLAY', 'BATTERY', 'FANS&COOLING', 'CPU', 'MEMORY', 'WARRANTY', 'OPTICAL_DRIVES', 'GRAPHICS', 'SOFTWARE', 'SHIPPING', 'MOTHERBOARD', 'MOUSE', 'Out_Of_Scope'], and B is one of ['PRICE', 'DESIGN_FEATURES', 'OPERATION_PERFORMANCE', 'USABILITY', 'GENERAL', 'QUALITY', 'PORTABILITY', 'CONNECTIVITY', 'MISCELLANEOUS']. Each category must strictly adhere to these sets, e.g., LAPTOP#GENERAL.
Examples
The following are several examples to help you learn how to extract quadruples:
Input: "acer wants $ 170 to just look at it then add the repair cost on top of that ."
Final Answer: ['acer', 'SUPPORT#PRICE', 'NULL', 'neutral']
Input: "update : i repaired it myself for $ 12 ."
Final Answer: ['NULL', 'LAPTOP#GENERAL', 'NULL', 'neutral']
Input: "first one that they shipped was obviously defective , super slow and speakers were garbled ."
Final Answer: ['NULL', 'SHIPPING#GENERAL', 'defective', 'negative'], ['NULL', 'SHIPPING#GENERAL', 'slow', 'negative'], ['speakers', 'MULTIMEDIA_DEVICES#GENERAL', 'garbled', 'negative']
Input: "pro : light , reasonable price , fast ."
Final Answer: ['NULL', 'LAPTOP#DESIGN_FEATURES', 'light', 'positive'], ['NULL', 'LAPTOP#OPERATION_PERFORMANCE', 'fast', 'positive'], ['price', 'LAPTOP#PRICE', 'reasonable', 'positive']
Input: "overall , it is not horrible , but i wouldn ' t purchase this model again ."
Final Answer: ['model', 'LAPTOP#GENERAL', 'not horrible', 'negative']
Input
Instruction: From the laptop review, identify all aspects, their opinion words, category, and sentiment ('positive', 'negative', 'neutral').
Answer Format: Your final answer can include multiple aspect-opinion pairs, formatted as follows:
"Final Answer: ['aspect_term$_1$', 'category$_1$', 'opinion_term$_1$', 'sentiment$_1$'], ['aspect_term$_2$', 'category$_2$', 'opinion_term$_2$', 'sentiment$_2$'] ...".If an aspect or opinion term is implicit, use 'NULL' to represent it.
Input: "the unit cost $ 275 to start with , so it is not worth repairing ."

Table 12: Few-shot example of laptop domain

Our instruction for restaurant domain

Task Definition:
Aspect-Based Sentiment Analysis aims to extract the opinion target described by an entity and its aspect (collectively called aspect) from reviews and identify the sentiment toward the aspect.
Pre-defined Categories: Categories must follow the A#B format, where A is one of ['RESTAURANT', 'DRINKS', 'SERVICE', 'FOOD', 'AMBIENCE', 'LOCATION'], and B is one of ['GENERAL', 'STYLE_OPTIONS', 'QUALITY', 'PRICES', 'MISCELLANEOUS']. Each category must strictly adhere to these sets, e.g., FOOD#QUALITY.
Instruction:
From the restaurant review, identify all aspects, their opinion words, category, and sentiment ('positive', 'negative', 'neutral').
You will be given several possible answers and the correct answer is highly likely to be among the provided options. Please select the most appropriate option. Only if you believe none of the options are correct, provide your own answer.
Answer Format:
Your final answer can include multiple aspect-opinion pairs, formatted as follows:
"Final Answer: ['aspect_term$_1$', 'category$_1$', 'opinion_term$_1$', 'sentiment$_1$'], ['aspect_term$_2$', 'category$_2$', 'opinion_term$_2$', 'sentiment$_2$'] ...".
If an aspect or opinion term is implicit, use 'NULL' to represent it.
Input:
{Input review}
Candidate answers:
{candidate answers}

Table 13: Our instruction for restaurant domain

Our instruction for laptop domain

Task Definition:
Aspect-Based Sentiment Analysis aims to extract the opinion target described by an entity and its aspect (collectively called aspect) from reviews and identify the sentiment toward the aspect.
Pre-defined Categories: Categories must follow the A#B format, where A is one of {category_a}, and B is one of {category_b}. Each category must strictly adhere to these sets, e.g., BATTERY#GENERAL.
Instruction:
From the laptop product review, identify all aspects, their opinion words, category, and sentiment ('positive', 'negative', 'neutral').
You will be given several possible answers and the correct answer is highly likely to be among the provided options. Please select the most appropriate option. Only if you believe none of the options are correct, provide your own answer.
Answer Format:
Your final answer can include multiple aspect-opinion pairs, formatted as follows:
"Final Answer: ['aspect_term$_1$', 'category$_1$', 'opinion_term$_1$', 'sentiment$_1$'], ['aspect_term$_2$', 'category$_2$', 'opinion_term$_2$', 'sentiment$_2$'] ...".
If an aspect or opinion term is implicit, use 'NULL' to represent it.
{Input review}
Candidate answers:
{candidate answers}

Table 14: Our instruction for laptop domain