# Hard Example Mining-Driven Label Alignment for LLMs in Aspect Sentiment Triplet Extraction

Anonymous ACL submission

## Abstract

In recent years, large language models (LLMs) have achieved remarkable success across various natural language processing tasks. However, recent studies indicate that LLMs still underperform smaller supervised fine-tuned models in the fine-grained Aspect Sentiment Triplet Extraction (ASTE) task. To investigate the reasons behind this phenomenon, we conducted an in-depth analysis of cases where LLM predictions failed. Our findings reveal that the primary cause for this suboptimal performance lies in the inconsistency between the internal extraction behaviors of LLMs and task-specific annotation standards. Motivated by this insight, we introduce HEMLA, a Hard Example Mining-driven Label Alignment framework. Specifically, we use LLM-generated responses as prompts to train a lightweight alignment model, while dynamically deciding whether to include each response in the training process based on a hard example mining strategy. Extensive experimental results demonstrate that our method consistently outperforms existing state-of-the-art approaches and offers a new paradigm for adapting LLMs to downstream tasks without fine-tuning the underlying LLMs[1].

## 1 Introduction

Aspect sentiment triplet extraction (ASTE) is a fine-grained sentiment analysis task that aims to extract aspect terms, opinion terms, and their associated sentiment polarities from a given review (Peng et al., 2020). Typically, aspect terms are nouns or noun phrases, while opinion terms are adjectives or adjectival phrases. Given a user review such as "*The place is small and cramped but the food is fantastic*", the objective of ASTE is to identify the following sentiment triplets: (*place*, *small*, *negative*), (*place*, *cramped*, *negative*), and (*food*, *fantastic*, *positive*).

[1]Our code will be released publicly upon acceptance.

| **Boundary Mismatch** |
|---|
| **Review:** In the summer months, the back garden area is really nice.<br>**Label:** (back garden area, nice, positive)<br>**GPT-4o:** (back garden area, really nice, positive) |
| **Triplet Merging** |
| **Review:** I thought the restaurant was nice and clean.<br>**Label:** (restaurant, nice, positive), (restaurant, clean, positive)<br>**GPT-4o:** (restaurant, nice and clean, positive) |

Figure 1: Representative examples of boundary mismatch and triplet merging errors produced by GPT-4o.

Previous studies have primarily relied on smaller pre-trained language models. Based on modeling paradigms, existing approaches can be categorized into sequence labeling, table-filling, span enumeration, and generative methods. In recent years, large language models (LLMs) have achieved remarkable success across a wide range of natural language processing (NLP) tasks. However, recent studies have shown that, on fine-grained ASTE tasks, LLMs underperform smaller supervised fine-tuned models (Su et al., 2024; Mukherjee et al., 2023). Similar findings have been reported in related information extraction tasks such as named entity recognition (Ding et al., 2024), relation extraction (Delbrouck et al., 2024; Qi et al., 2024), and event extraction (Hong and Liu, 2024). These results suggest that, despite their strong semantic understanding capabilities, LLMs still struggle to adhere to the requirements of fine-grained information extraction tasks.

To further examine this limitation, we evaluated GPT-4o[2] on the 15RES[3] dataset and conducted a statistical analysis of its error types. Figure 1 illustrates the two most prevalent error categories, while additional error types and detailed analyses are provided in Appendix A. Human evaluation results indicate that the primary factor underlying the model's suboptimal performance is a misalign-

[2]API version: gpt-4o-2024-08-06

[3]https://github.com/xuuuluuu/SemEval-Triplet-data

ment between the LLM's extraction behavior and task-specific annotation standards. This misalignment manifests in two main aspects. First, the LLM exhibits inconsistent internal extraction behavior, which likely arises from exposure to diverse roles during instruction fine-tuning, hindering the formation of a stable extraction strategy. Second, even when the LLM maintains internal consistency, its extraction behavior often diverges from human annotation preferences, leading to further discrepancies with the gold-standard labels.

To address this issue, we propose Hard Example Mining-driven Label Alignment (HEMLA), a novel framework designed to align LLM outputs with task-specific annotation standards. Specifically, we introduce a lightweight alignment model that bridges the gap between LLM predictions and gold-standard labels. To train the alignment model, we employ a dynamic optimization strategy based on hard example mining (HEM) (Shrivastava et al., 2016). LLM-generated responses are incorporated as prompts and are adaptively included or excluded based on the alignment model's training loss. This mechanism encourages stronger adherence to annotation standards and enhances the model's task-specific understanding.

Overall, the main contributions of this paper are summarized as follows:

- We propose a unified framework that aligns LLM predictions with gold-standard annotations without fine-tuning the underlying LLMs, offering a novel paradigm for adapting LLMs to downstream tasks.
- We introduce a dynamic training strategy based on HEM to adaptively optimize the alignment model, significantly improving label alignment accuracy and task-specific understanding.
- Extensive experiments demonstrate that our framework generalizes across different LLMs and consistently enhances their performance on the ASTE task.

## 2 Related Work

Existing methods for ASTE can be broadly categorized into four paradigms: sequence labeling, table filling, span enumeration, and generative modeling. Early work primarily adopts sequence labeling frameworks, either in pipeline or joint learning settings, to extract aspect terms, opinion terms, and sentiment polarities (Peng et al., 2020; Xu et al., 2020). Subsequent studies reformulate ASTE as a structured prediction problem, giving rise to table-filling approaches that model token-level interactions through grid tagging or table-based classification schemes (Wu et al., 2020; Chen et al., 2021b, 2022a; Zhang et al., 2022; Liang et al., 2023).

While these methods achieve strong performance by explicitly modeling fine-grained token-level dependencies, they often struggle to effectively capture long or multi-word aspect and opinion terms. To alleviate this limitation, span enumeration methods treat ASTE as a span-level classification task, enabling richer contextual representations, albeit at the cost of increased redundancy and span overlap (Xu et al., 2021, 2022; Chen et al., 2022b; Yu et al., 2023).

In recent years, generative approaches have been explored for ASTE by reformulating the task as a sequence generation problem (Zhang et al., 2021). Building upon this paradigm, subsequent studies incorporate auxiliary techniques—such as sequence labeling, contrastive learning, and multi-task learning—into the generative framework (Luo et al., 2023; Zhou and Qian, 2023; Mukherjee et al., 2023; Zhang et al., 2023), achieving competitive performance on standard ASTE benchmarks.

More recently, several studies have investigated the integration of large language models (LLMs) with smaller language models (SLMs). Yang et al. (2024) leverage predictions and confidence scores produced by SLMs as prompts to guide the reasoning process of LLMs, demonstrating strong performance across multiple natural language inference and question answering tasks. Building on this line of work, Li et al. (2025) introduce a discriminative model to assess the correctness of LLM-generated responses and use the resulting feedback to iteratively refine LLM predictions. Despite their effectiveness, these approaches primarily rely on the outputs of SLMs as pseudo labels to prompt LLMs. Such pseudo labels are inherently imperfect and may introduce noisy or erroneous signals during inference. More importantly, these methods do not explicitly address the inconsistency between LLM predictions and task-specific annotation standards, which may further exacerbate prediction bias even with auxiliary supervision.

In contrast, our HEMLA adopts a post-processing paradigm that operates directly on LLM outputs rather than relying on input-level prompting. Specifically, HEMLA employs a lightweight

alignment model trained via supervised fine-tuning to map raw LLM predictions to gold-standard annotations. By explicitly correcting systematic discrepancies between LLM outputs and task-specific annotation standards, our approach mitigates error propagation arising from noisy pseudo labels and provides an effective solution for adapting LLMs to structured prediction tasks without modifying the underlying LLMs.

## 3 Proposed Framework

As illustrated in Figure 2, our framework consists of three stages. First, a closed-source LLM generates initial predictions conditioned on the input text. Second, a lightweight alignment model is trained using a HEM strategy to map these predictions to gold-standard annotations. Finally, during inference, the trained alignment model is applied to align LLM outputs with task-specific annotation standards. In the following sections, we formally define the task and provide a detailed description of each stage in the proposed framework.

### 3.1 Task Definition

Given an input review $x = \{w_1, w_2, \cdots, w_n\}$, where $w_i$ is the $i$-th word and $n$ is the sequence length, the goal of ASTE is to extract all aspect-sentiment triplets. Formally, the triplets are defined as a set $\mathcal{T} = \{(a_i, o_i, s_i)\}_{i=1}^{|\mathcal{T}|}$, where $a_i$ and $o_i$ represent the aspect term and opinion term, respectively, and $s_i$ denotes the associated sentiment polarity. Here, $|\mathcal{T}|$ denotes the number of triplets in the review, and $s_i \in \{\text{Positive, Negative, Neutral}\}$.

### 3.2 Unaligned Prediction

As shown in Stage 1 of Figure 2, we first prompt the LLM $\mathcal{M}$ to generate an unaligned response for each review in the dataset. We adopt two prompting strategies: zero-shot and few-shot. In the zero-shot setting, only the task instruction is provided, whereas in the few-shot setting, the task instruction is accompanied by several reference examples.

The prompt templates under the zero-shot and few-shot settings are shown below. In both templates, $\{\cdot\}$ represents a placeholder, $x_i$ denotes the input text of the $i$-th reference example, $y_i$ corresponds to its gold-standard label, and $K$ is the number of demonstration examples. After obtaining the unaligned responses for all samples, we concatenate them with the corresponding input texts to construct augmented prompts for training the alignment model.

**Zero-shot Setting**

You are an expert in textual sentiment analysis. Given a review, extract its aspect term, opinion term, and the corresponding sentiment polarity, where the categories of sentiment polarity are positive, negative, and neutral. Please generate responses strictly in the following format: [["aspect1", "opinion1", "sentiment1"], ["aspect2", "opinion2", "sentiment2"], ...].
Review: $\{x_{test}\}$ Answer:

**Few-shot Setting**

You are an expert in textual sentiment analysis. Given a review, extract its aspect term, opinion term, and the corresponding sentiment polarity, where the categories of sentiment polarity are positive, negative, and neutral. Please generate responses strictly in the following format: [["aspect1", "opinion1", "sentiment1"], ["aspect2", "opinion2", "sentiment2"], ...]. Here are some examples:
Review: $\{x_1\}$ Answer: $\{y_1\}$
Review: $\{x_2\}$ Answer: $\{y_2\}$
.....
Review: $\{x_K\}$ Answer: $\{y_K\}$
Review: $\{x_{test}\}$ Answer:

### 3.3 Alignment Training

We find that directly using LLM-generated responses as prompts to fine-tune the alignment model can sometimes degrade performance (see Section 4.3), as these responses may contain erroneous information that misleads the model. To mitigate such noise while strengthening the model's task-specific alignment capability, we adopt a HEM strategy. The core idea of HEM is to allocate more learning resources to samples with higher training loss, which typically correspond to cases where the model struggles to make accurate predictions (Shrivastava et al., 2016). In our adaptation, instead of searching for hard examples across the entire training set, we construct two prompt variants for each original sample. We then compare the alignment model's training loss on these variants and prioritize the one yielding the higher loss during optimization, thereby encouraging the model to focus on more challenging training instances.

Specifically, given an input sequence $x = (w_1, w_2, \ldots, w_n)$, we feed it into an alignment model $\mathcal{M}_A$ to generate a target sequence $y = (y_1, y_2, \ldots, y_T)$. The alignment model follows a standard encoder-decoder architecture and defines a conditional distribution over output sequences:

$$p_\theta(y \mid x) = \prod_{t=1}^{T} p_\theta(y_t \mid y_{<t}, x). \quad (1)$$

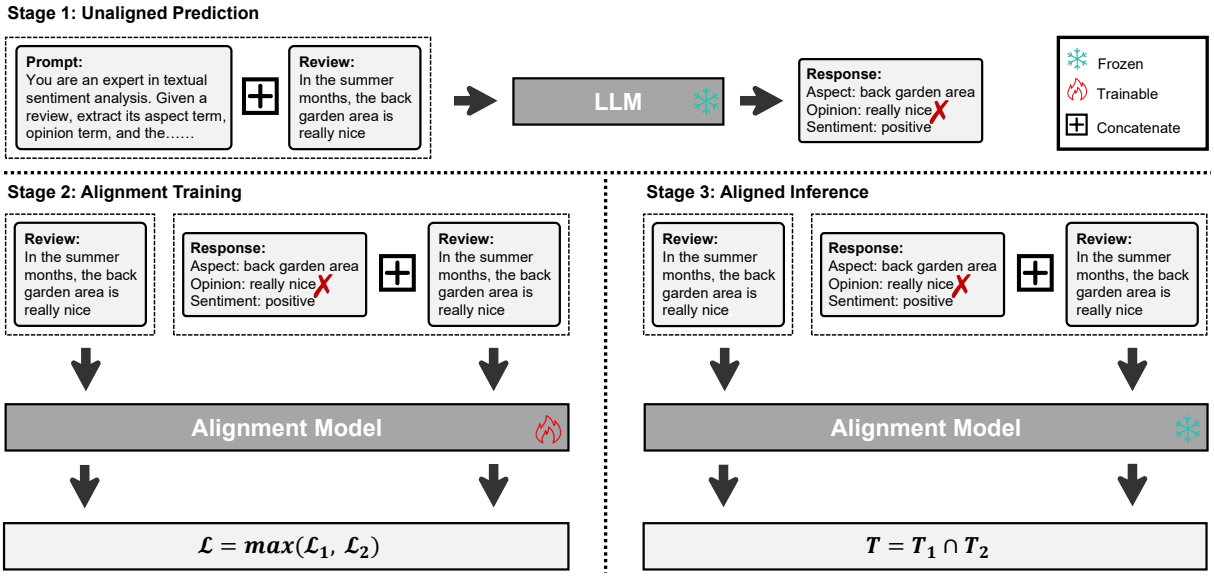


Figure 2: Framework of HEMLA. $\mathcal{L}1$ and $\mathcal{L}2$ are the losses computed from the original and augmented inputs, respectively, while $T_1$ and $T_2$ denote the triplet sets predicted by the corresponding two channels.

The model is trained by minimizing the token-level cross-entropy loss between the predicted distribution and the ground-truth sequence:

$$\mathcal{L}_1 = -\sum_{t=1}^{T} \log p_\theta(y_t \mid y_{<t}, x). \tag{2}$$

Next, we incorporate the response generated by the LLM into the alignment process. Given an input sequence $x$, we first obtain the LLM response

$$r = \mathcal{M}(x), \tag{3}$$

where $\mathcal{M}$ denotes the frozen LLM. We then concatenate the original input and the LLM response to form an augmented sequence $\tilde{x} = r \oplus x$, which is fed into the alignment model $\mathcal{M}_A$. Analogous to Equation 1, the alignment model defines a conditional distribution over the target sequence given the augmented input:

$$p_\theta(y \mid \tilde{x}) = \prod_{t=1}^{T} p_\theta(y_t \mid y_{<t}, \tilde{x}). \tag{4}$$

The corresponding token-level cross-entropy loss is computed as:

$$\mathcal{L}_2 = -\sum_{t=1}^{T} \log p_\theta(y_t \mid y_{<t}, \tilde{x}). \tag{5}$$

Together with the loss $\mathcal{L}_1$ computed from the original input $x$, we obtain two candidate losses for the same training instance. Following the HEM principle, which allocates more learning resources to training instances that are more difficult to optimize, we select the larger loss as the final training objective:

$$\mathcal{L} = \max(\mathcal{L}_1, \mathcal{L}_2). \tag{6}$$

We justify the selection of the input variant associated with the higher loss from two complementary perspectives. First, when incorporating the LLM response results in a higher loss (i.e., $\mathcal{L}_2 > \mathcal{L}_1$), it indicates that the alignment model has difficulty reconciling the response with the gold-standard label. In this case, selecting $\mathcal{L}_2$ encourages the model to focus on correcting challenging misalignment patterns introduced by the LLM response. Second, when the loss decreases after incorporating the LLM response (i.e., $\mathcal{L}_2 < \mathcal{L}_1$), it suggests that the response already closely approximates the gold label. Under this condition, the alignment model can largely rely on the response itself to infer the target, and prioritizing $\mathcal{L}_1$ instead helps strengthen the model's intrinsic understanding of the task without over-reliance on external cues. Accordingly, we select the larger loss between $\mathcal{L}_1$ and $\mathcal{L}_2$ as the optimization objective.

### 3.4 Aligned Inference

To accommodate the dual-channel architecture employed during alignment training, we adopt an intersection-based inference strategy to merge predictions from the prompt-based and prompt-free channels. Given an input instance, the alignment model produces two prediction sets, denoted as $T_1$

and $T_2$, corresponding to outputs conditioned on the original input and the augmented input with LLM responses, respectively. We then determine the final prediction set $T$ according to the following rule:

$$T = \begin{cases} T_1 \cap T_2, & \text{if } T_1 \cap T_2 \neq \varnothing, \\ T_2, & \text{otherwise.} \end{cases} \quad (7)$$

When the intersection of $T_1$ and $T_2$ is non-empty, we adopt the overlapping triplets as the final prediction, as they represent consistent outputs across both channels. If the intersection is empty, we fall back to $T_2$, which incorporates information from the LLM response. This design allows the model to leverage informative LLM signals while remaining robust to noisy or misleading content.

## 4 Experiments

### 4.1 Experiment Setup

**Datasets.** We evaluate our framework on four widely used benchmark datasets from ASTE-Data-V2 (Xu et al., 2020), a refined version of ASTE-Data-V1 (Peng et al., 2020). This benchmark includes three datasets from the restaurant domain and one from the laptop domain, all originating from the SemEval Challenges (Pontiki et al., 2014, 2015, 2016). Detailed statistics of these datasets are presented in Table 1. #R denotes the number of reviews, and #T denotes the number of triplets, where each review may contain multiple triplets. #POS, #NEG, and #NEU indicate the numbers of triplets labeled with positive, negative, and neutral sentiments, respectively. #A and #O denote the numbers of aspect terms and opinion terms, respectively.

**Implementation Details.** We employ models from the GPT series as the LLMs to be aligned, specifically GPT-3.5-turbo[4] and GPT-4o[5]. For the alignment model, we adopt T5-base[6]. The zero-shot and few-shot prompt templates used for the GPT models are described in Section 3.2. Model parameters are optimized using AdamW (Loshchilov and Hutter, 2019) with a batch size of 16, a learning rate of $1 \times 10^{-4}$, and 50 training epochs. Following previous work (Peng et al., 2020), we adopt micro-averaged precision (P), recall (R), and F1 score as evaluation metrics. For evaluation, we select the checkpoint that achieves the best F1 score on the development set for each run and report the average F1 score on the test set over five runs with different random seeds. All experiments are conducted on a single NVIDIA RTX 4090 GPU using CUDA 11.8 and PyTorch 2.3.0.

[4]API version: gpt-3.5-turbo-0125
[5]API version: gpt-4o-2024-08-06
[6]https://huggingface.co/google-t5/t5-base

| Dataset | Split | #R | #T | #POS | #NEU | #NEG | #A | #O |
|---|---|---|---|---|---|---|---|---|
| 14LAP | Train | 906 | 1460 | 817 | 126 | 517 | 1254 | 1460 |
| | Dev | 219 | 345 | 169 | 36 | 140 | 302 | 346 |
| | Test | 328 | 541 | 364 | 63 | 114 | 466 | 543 |
| 14RES | Train | 1266 | 2337 | 1691 | 166 | 480 | 2051 | 2061 |
| | Dev | 310 | 577 | 404 | 54 | 119 | 500 | 497 |
| | Test | 492 | 994 | 773 | 66 | 155 | 844 | 994 |
| 15RES | Train | 605 | 1013 | 783 | 25 | 205 | 935 | 1013 |
| | Dev | 148 | 249 | 185 | 11 | 53 | 236 | 249 |
| | Test | 322 | 485 | 317 | 25 | 143 | 460 | 485 |
| 16RES | Train | 857 | 1394 | 1015 | 50 | 329 | 1300 | 1394 |
| | Dev | 210 | 339 | 252 | 11 | 76 | 319 | 339 |
| | Test | 326 | 514 | 407 | 29 | 78 | 474 | 514 |

Table 1: Detailed statistics of ASTE-Data-V2.

**Baselines.** To evaluate the effectiveness of our framework, we compare against three categories of baseline approaches. First, we assess the in-context learning performance of LLMs under both zero-shot and few-shot (random) settings (Brown et al., 2020). Second, we compare against a range of LLM-free supervised methods, including GTS (Wu et al., 2020), BMRC (Chen et al., 2021a), Span-ASTE (Xu et al., 2021), EMC-GCN (Chen et al., 2022a), GAS (Zhang et al., 2021), COM-MRC (Zhai et al., 2022), RLI (Yu et al., 2023), Sim-STAR (Li et al., 2023), CONTRASTE (Mukherjee et al., 2023), DLSP (Liu et al., 2024), and Mini-ConGTS (Sun et al., 2024). Finally, we include recent LLM-based supervised approaches, namely SuperContext (Yang et al., 2024) and BAAlign (Li et al., 2025), for comparison. For a fair comparison, all LLM-based supervised methods employ the same SLM, T5-base. Detailed descriptions of all baseline methods are provided in Appendix B.

### 4.2 Main Results

Table 2 reports the experimental results of all compared methods on four ASTE benchmark datasets. Overall, our proposed HEMLA consistently outperforms all baseline approaches across all datasets.

Compared with the supervised T5-base model, incorporating our HEMLA framework leads to substantial performance gains, and these gains further increase as the capability of the underlying LLM improves. In particular, when equipped with GPT-4o, HEMLA achieves the best overall performance on all four datasets. Compared with state-of-the-art LLM-free supervised methods, HEMLA yields consistent improvements, especially on 15RES and

| Method | LLM | 14LAP | | | 14RES | | | 15RES | | | 16RES | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| ***In-context Learning*** | | | | | | | | | | | | | |
| Zero-shot (Brown et al., 2020) | GPT-3.5 | 29.26 | 34.57 | 31.70 | 47.65 | 51.01 | 49.27 | 40.81 | 49.90 | 44.90 | 47.99 | 57.98 | 52.51 |
| Few-shot (Brown et al., 2020) | GPT-3.5 | 37.58 | 43.07 | 40.14 | 54.47 | 61.27 | 57.67 | 38.93 | 58.35 | 46.70 | 45.04 | 59.14 | 51.14 |
| Zero-shot (Brown et al., 2020) | GPT-4o | 38.24 | 40.85 | 39.50 | 54.58 | 56.34 | 55.45 | 49.19 | 56.29 | 52.50 | 53.32 | 60.89 | 56.86 |
| Few-shot (Brown et al., 2020) | GPT-4o | 39.94 | 45.47 | 42.52 | 55.89 | 63.08 | 61.44 | 50.09 | 59.59 | 54.43 | 52.47 | 66.15 | 58.52 |
| ***LLM-free supervised methods*** | | | | | | | | | | | | | |
| GTS (Wu et al., 2020) | - | 59.40 | 51.94 | 55.42 | 68.09 | 69.54 | 68.81 | 59.28 | 57.93 | 58.60 | 68.32 | 66.86 | 67.58 |
| BMRC (Chen et al., 2021a) | - | 65.12 | 54.41 | 59.27 | 71.32 | 70.09 | 70.69 | 63.71 | 58.63 | 61.05 | 67.74 | 68.56 | 68.13 |
| Span-ASTE (Xu et al., 2021) | - | 63.44 | 55.84 | 59.38 | 72.89 | 70.89 | 71.85 | 62.18 | 64.45 | 63.27 | 69.45 | 71.17 | 70.26 |
| EMC-GCN (Chen et al., 2022a) | - | 61.70 | 55.26 | 58.81 | 71.21 | 72.39 | 71.78 | 61.54 | 62.47 | 61.93 | 65.62 | 71.30 | 68.33 |
| GAS (Zhang et al., 2021) | - | 64.52 | 57.27 | 60.68 | 72.81 | 71.56 | 72.18 | 63.36 | 60.62 | 61.96 | 69.26 | 71.15 | 70.19 |
| COM-MRC (Zhai et al., 2022) | - | 62.35 | 58.16 | 60.17 | 75.46 | 68.91 | 72.01 | 68.35 | 61.24 | 64.53 | 71.55 | 71.59 | 71.57 |
| RLI (Yu et al., 2023) | - | 63.32 | 57.43 | 60.96 | 77.46 | 71.97 | 74.34 | 60.08 | 70.66 | 65.41 | 70.50 | 74.28 | 72.34 |
| SimSTAR (Li et al., 2023) | - | 66.46 | 58.23 | 62.07 | 76.23 | 71.63 | 73.86 | 71.71 | 59.59 | 65.09 | 72.02 | 74.12 | 73.06 |
| CONTRASTE (Mukherjee et al., 2023) | - | 64.20 | 61.70 | 62.90 | 73.60 | 74.40 | 74.00 | 65.30 | 66.70 | 66.10 | 72.20 | 76.30 | 74.20 |
| DLSP (Liu et al., 2024) | - | 69.26 | 55.82 | 61.82 | 76.75 | 71.14 | 73.84 | 69.92 | 60.39 | 64.81 | 75.60 | 72.54 | 74.04 |
| MiniConGTS (Sun et al., 2024) | - | 66.82 | 60.68 | <u>63.61</u> | 76.10 | 75.08 | <u>75.59</u> | 66.50 | 63.86 | 65.15 | 75.52 | 74.14 | 74.83 |
| ***LLM-based supervised methods*** | | | | | | | | | | | | | |
| T5-base | - | 65.00 | 57.67 | 61.12 | 71.02 | 71.73 | 71.37 | 61.85 | 66.19 | 63.94 | 70.73 | 75.68 | 73.12 |
| SuperContext† (Yang et al., 2024) | GPT-3.5 | 59.88 | 56.56 | 58.17 | 69.96 | 71.93 | 70.93 | 61.99 | 68.25 | 64.97 | 67.07 | 76.07 | 71.29 |
| BAAlign† (Li et al., 2025) | GPT-3.5 | 60.37 | 58.06 | 59.20 | 72.70 | 70.72 | 71.70 | 62.26 | 65.98 | 64.06 | 67.54 | 74.90 | 71.03 |
| HEMLA (Ours) | GPT-3.5 | 67.78 | 57.94 | 62.48 | 76.42 | 71.67 | 73.97 | 72.13 | 63.51 | <u>67.54</u> | 77.06 | 74.51 | <u>75.77</u> |
| SuperContext† (Yang et al., 2024) | GPT-4o | 65.08 | 58.23 | 61.46 | 70.66 | 71.73 | 71.19 | 61.98 | 67.22 | 64.49 | 69.64 | 75.88 | 72.63 |
| BAAlign† (Li et al., 2025) | GPT-4o | 66.30 | 59.16 | 62.59 | 74.72 | 72.54 | 73.61 | 64.84 | 68.45 | 66.60 | 71.58 | 76.46 | 73.94 |
| HEMLA (Ours) | GPT-4o | 71.36 | 58.78 | **64.45** | 77.17 | 74.63 | **75.90** | 73.19 | 64.74 | **68.71** | 78.60 | 76.46 | **77.51** |

Table 2: Experimental results on four ASTE datasets (%). The best results are shown in bold, and the second-best results are underlined. † denotes results reproduced by us.

16RES, where HEMLA (GPT-4o) surpasses the previous best method MiniConGTS by 3.56% and 2.68% F1, respectively.

When compared with other LLM-based supervised methods, HEMLA demonstrates not only superior performance but also greater stability. As shown in Table 2, when using the relatively weaker GPT-3.5, both SuperContext and BAAlign exhibit performance degradation on 16RES compared to the T5-base baseline, indicating that naively incorporating LLMs may even harm the performance of the underlying small model. In contrast, HEMLA consistently improves performance under both GPT-3.5 and GPT-4o, highlighting its robustness across different LLMs. These results collectively demonstrate the effectiveness and stability of the proposed framework.

From the in-context learning results, we observe that incorporating demonstration examples generally improves LLM performance compared to the zero-shot setting. However, on 16RES, the few-shot performance of GPT-3.5 is inferior to its zero-shot counterpart. This observation suggests that, for structured extraction tasks such as ASTE, randomly selected demonstrations may negatively affect LLM reasoning, as the provided examples may fail to adequately reflect task-specific extraction standards. This finding further highlights the value of the proposed post-hoc LLM-based supervised framework.

### 4.3 Ablation Study

To evaluate the contribution of each component in our framework, we conduct an ablation study under several configurations, as summarized in Table 3. Specifically, HEM denotes applying the hard example mining strategy during training. NP refers to the prompt-free channel, whereas P denotes the prompt-based channel. HEMLA represents the full version of our framework, which integrates hard example mining during training and employs both NP and P inference channels. FT-P denotes directly fine-tuning the baseline T5-base model by using LLM-generated responses as prompts.

As shown in Table 3, compared with T5-base, directly fine-tuning the alignment model by treating LLM responses as prompts may lead to performance degradation. For example, on the 14LAP dataset, FT-P underperforms the T5-base baseline regardless of whether GPT-3.5 or GPT-4o is used. This result indicates that erroneous predictions contained in LLM responses can mislead the alignment model during training.

In contrast, the proposed HEMLA framework effectively mitigates this issue through a dual-channel dynamic training strategy, achieving con-

| Method | LLM | 14LAP | | | 14RES | | | 15RES | | | 16RES | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| T5-base | - | 65.00 | 57.67 | 61.12 | 71.02 | 71.73 | 71.37 | 61.85 | 66.19 | 63.94 | 70.73 | 75.68 | 73.12 |
| FT-P | GPT-3.5 | 60.28 | 56.01 | 58.05 | 71.39 | 71.53 | 71.46 | 64.14 | 66.39 | 65.25 | 72.66 | 74.92 | 73.78 |
| HEM-NP | GPT-3.5 | 65.41 | 56.67 | 60.68 | 73.54 | 71.03 | 72.26 | 65.26 | 63.92 | 64.58 | 75.53 | 75.26 | 75.41 |
| HEM-P | GPT-3.5 | 66.6 | 56.7 | 61.25 | 72.94 | 71.1 | 72.01 | 64.26 | 65.98 | 65.11 | 73.27 | 76.07 | 74.64 |
| HEMLA | GPT-3.5 | 67.78 | 57.94 | **62.48** | 76.42 | 71.67 | **73.97** | 72.13 | 63.51 | **67.54** | 77.06 | 74.51 | **75.77** |
| FT-P | GPT-4o | 63.07 | 56.19 | 59.43 | 73.12 | 72.23 | 72.67 | 66.05 | 66.19 | 66.13 | 72.05 | 76.26 | 74.12 |
| HEM-NP | GPT-4o | 67.36 | 58.78 | 62.78 | 73.13 | 73.94 | 73.54 | 64.08 | 66.79 | 67.41 | 73.47 | 74.85 | 74.18 |
| HEM-P | GPT-4o | 65.98 | 58.6 | 62.07 | 72.98 | 74.45 | 73.71 | 68.27 | 67.42 | 67.84 | 73.43 | 77.43 | 75.38 |
| HEMLA | GPT-4o | 71.36 | 58.78 | **64.45** | 77.17 | 74.63 | **75.90** | 73.19 | 64.74 | **68.71** | 78.60 | 76.46 | **77.51** |

Table 3: Ablation study results (%). Results are grouped by base LLM. For each base model, the best and second-best results are highlighted in bold and underlined, respectively.

sistent performance improvements over the baseline across all datasets. Moreover, while the HEM-NP and HEM-P variants exhibit advantages on different datasets, neither variant consistently outperforms the other. This observation suggests that the two inference channels capture complementary information, further highlighting the effectiveness of HEMLA's intersection-based matching strategy in integrating both channels to achieve superior overall performance.

### 4.4 Effect of Different Base LLMs

As shown in Tables 2 and 3, HEMLA achieves better overall performance when built upon the more powerful GPT-4o, compared to its GPT-3.5-based variant. This observation suggests a positive correlation between the capability of the base LLM and the performance of the resulting alignment model. To further examine this relationship, we evaluated HEMLA with six widely used LLMs—Qwen3[7], DeepSeek-V3[8], DeepSeek-R1[9], GPT-4o, LLaMA4[10], and Claude 4[11]—with the overall trends illustrated in Figure 3 and detailed results reported in Appendix C.

Overall, the alignment performance tends to improve with the ASTE capability of the underlying base model. However, when the performance differences among base LLMs are relatively small, the resulting alignment gains are not always consistent. In some cases, an alignment model built upon a slightly weaker base LLM can even outperform those derived from stronger counterparts. For instance, on the 15RES and 16RES datasets, although GPT-4o does not achieve the best initial ASTE performance, the corresponding alignment model attains the highest final results. We hypothesize that this phenomenon may be related to differences in error distributions across models. Specifically, the prediction errors produced by GPT-4o tend to be more concentrated and interpretable, which may enable the alignment model to more effectively capture error patterns and learn more robust alignment strategies.

### 4.5 Effect of LLM Predictions on Alignment Model

To further investigate the influence of LLM responses on the alignment model, we conduct a comparative analysis from two complementary perspectives: (1) the behavior of the alignment model when the LLM produces correct predictions, and (2) the behavior of the LLM when the alignment model produces correct outputs.

As illustrated in Figure 4, when the LLM successfully predicts correct aspect sentiment triplets, the alignment model is generally able to reproduce most of them. Nevertheless, approximately 10%-20% of these correct triplets are not preserved. This observation reveals a limitation of the current alignment mechanism, as it does not fully guarantee the retention of all correct information generated by the LLM. To further analyze the underlying causes, we conduct a case study on two representative failure examples from the 14LAP dataset, in which the alignment model fails to retain correct triplets produced by the LLM (see Appendix D).

Conversely, as shown in Figure 5, when the align-

[7] https://huggingface.co/Qwen/Qwen3-235B-A22B

[8] https://huggingface.co/deepseek-ai/DeepSeek-V3-0324

[9] https://huggingface.co/deepseek-ai/DeepSeek-R1-0528

[10] https://huggingface.co/meta-llama/Llama-4-Maverick-17B-128E-Instruct

[11] We use the Claude Sonnet 4 model (version 20250514), available at https://www.anthropic.com/claude.

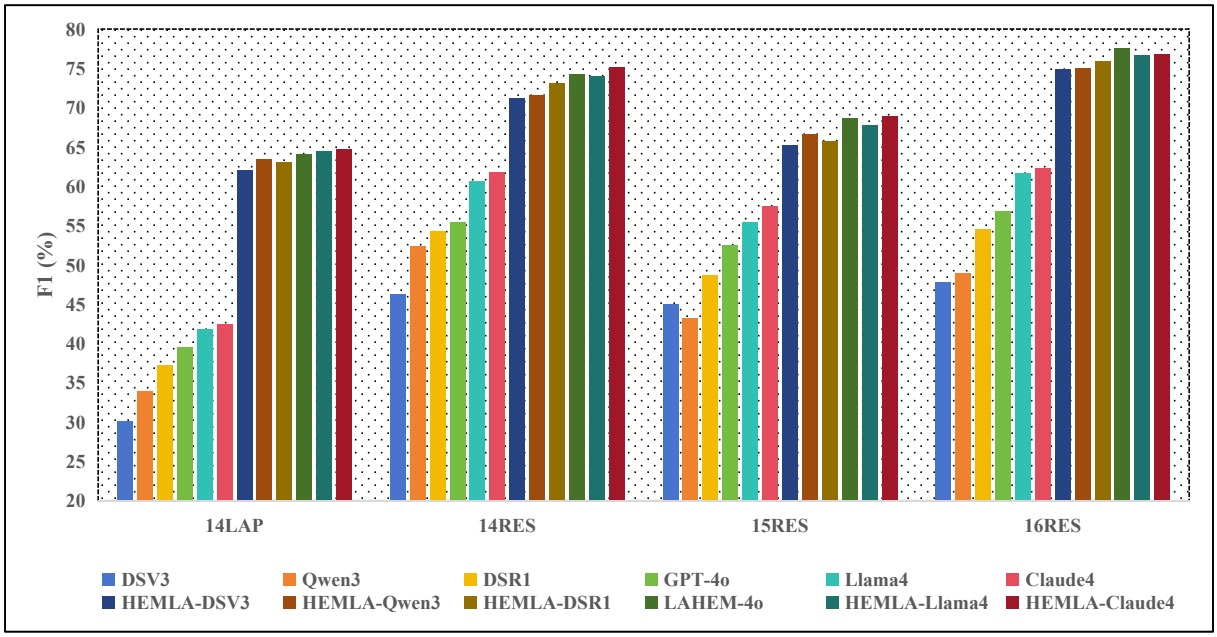


Figure 3: Performance of HEMLA with different base LLMs.

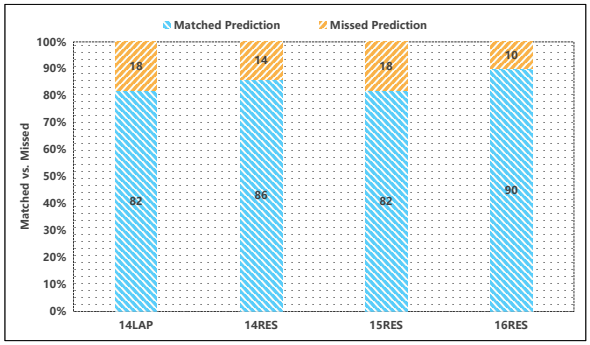


Figure 4: Performance of the alignment model when the LLM predicts correct triplets.

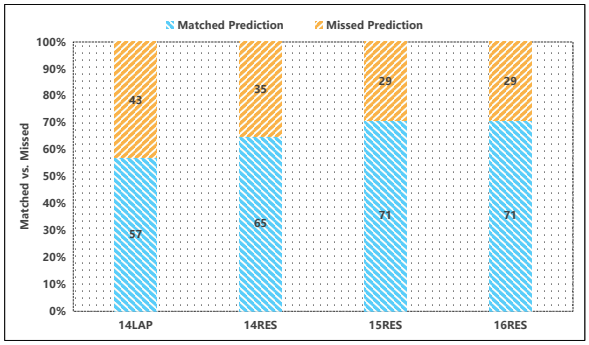


Figure 5: Performance of the LLM when the aligned model predicts correct triplets.

ment model successfully predicts correct triplets, more than half of them are also present in the corresponding LLM outputs. Interestingly, around 30%–40% of the correct triplets produced by the alignment model are not generated by the LLM. This finding suggests two complementary implications. First, the performance of our method benefits from the strength of the underlying LLM: when the LLM is more capable, the alignment model can effectively inherit and refine its predictions. Second, even when the LLM fails to generate correct triplets, the alignment model still demonstrates the ability to extract accurate aspect sentiment triplets directly from the input sentences. To further illustrate this capability, we present a case study on two examples from the 14LAP dataset in which the alignment model correctly predicts triplets that are missed by the LLM (see Appendix D).

## 5 Conclusion

In this paper, we propose HEMLA, a novel hard example mining-driven label alignment framework for ASTE that bridges the gap between LLM predictions and task-specific annotation standards. By dynamically incorporating LLM outputs as prompts based on training loss, HEMLA improves alignment with gold-standard labels and enhances task-specific understanding. Extensive experiments on four benchmark datasets demonstrate that HEMLA consistently outperforms both traditional supervised methods and recent LLM-based supervised approaches, achieving state-of-the-art performance without requiring LLM fine-tuning. These results highlight the effectiveness of HEMLA in adapting LLMs to domain-specific structured prediction tasks.

## Limitations

Despite its effectiveness, HEMLA has several limitations. First, while HEMLA is designed to correct systematic mismatches between LLM outputs and task-specific annotation standards, its effectiveness is still bounded by the overall quality of the base LLM predictions. As demonstrated in our experiments, HEMLA can mitigate certain errors and even recover correct triplets missed by the LLM. However, when the base model exhibits poor performance—such that its predictions contain a large amount of misleading signals—HEMLA is also affected by these signals, leading to degraded prediction performance. Second, HEMLA employs training loss as a heuristic signal for identifying hard examples and dynamically selecting between prompt variants. Although our ablation results demonstrate that this loss-guided strategy is effective in practice, training loss does not explicitly encode semantic correctness or annotation consistency. As a result, in some edge cases, loss-based selection may favor samples that are difficult to optimize rather than truly informative for alignment. Finally, our experiments focus on aspect sentiment triplet extraction. Although the proposed framework is designed in a task-agnostic manner, extending HEMLA to other structured prediction tasks may require task-specific adaptations in alignment objectives or prompt construction, which we leave for future work.

## References


Berk Atil, Alexa Chittams, Liseng Fu, Ferhan Ture, Lixinyu Xu, and Breck Baldwin. 2024. LLM stability: A detailed analysis with some surprises. *CoRR*, abs/2408.04667.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, and et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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

Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021a. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12666–12674. AAAI Press.

Yuqi Chen, Keming Chen, Xian Sun, and Zequn Zhang. 2022b. A span-level bidirectional network for aspect sentiment triplet extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4300–4309. Association for Computational Linguistics.

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

Jean-Benoit Delbrouck, Pierre J. Chambon, Zhihong Chen, Maya Varma, Andrew Johnston, Louis Blankemeier, Dave Van Veen, Tan Bui, Steven Quoc Hung Truong, and Curtis P. Langlotz. 2024. Radgraph-xl: A large-scale expert-annotated dataset for entity and relation extraction from radiology reports. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 12902–12915. Association for Computational Linguistics.

Yuyang Ding, Juntao Li, Pinzheng Wang, Zecheng Tang, Yan Bowen, and Min Zhang. 2024. Rethinking negative instances for generative named entity recognition. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 3461–3475. Association for Computational Linguistics.

Zijin Hong and Jian Liu. 2024. Towards better question generation in qa-based event extraction. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 9025–9038. Association for Computational Linguistics.

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

Shichen Li, Jiawei Zhang, Zhongqing Wang, and Peifeng Li. 2025. Aligning black-box llms for aspect sentiment quad prediction. In *Findings of the*

*Association for Computational Linguistics: EMNLP 2025*, pages 1012–1025.

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

Jingping Liu, Tao Chen, Hao Guo, Chao Wang, Haiyun Jiang, Yanghua Xiao, Xiang Xu, and Baohua Wu. 2024. Exploiting duality in aspect sentiment triplet extraction with sequential prompting. *IEEE Trans. Knowl. Data Eng.*, 36(11):6111–6123.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Keming Lu, Bowen Yu, Chang Zhou, and Jingren Zhou. 2024. Large language models are superpositions of all characters: Attaining arbitrary role-play via self-alignment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7828–7840. Association for Computational Linguistics.

Xianlong Luo, Meng Yang, and Yihao Wang. 2023. Tagging-assisted generation model with encoder and decoder supervision for aspect sentiment triplet extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2078–2093. Association for Computational Linguistics.

Rajdeep Mukherjee, Nithish Kannen, Saurabh Kumar Pandey, and Pawan Goyal. 2023. CONTRASTE: supervised contrastive pre-training with aspect-based prompts for aspect sentiment triplet extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 12065–12080. Association for Computational Linguistics.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8600–8607. AAAI Press.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, 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 *SemEval@NAACL-HLT*, pages 19–30. The Association for Computer Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 486–495. The Association for Computer Linguistics.

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

Kunxun Qi, Jianfeng Du, and Hai Wan. 2024. End-to-end learning of logical rules for enhancing document-level relation extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7247–7263. Association for Computational Linguistics.

Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. 2016. Training region-based object detectors with online hard example mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 761–769. IEEE Computer Society.

Guixin Su, Mingmin Wu, Zhongqiang Huang, Yongcheng Zhang, Tongguan Wang, Yuxue Hu, and Ying Sha. 2024. Refine, align, and aggregate: Multi-view linguistic features enhancement for aspect sentiment triplet extraction. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 3212–3228. Association for Computational Linguistics.

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

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme

for end-to-end fine-grained opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2576–2585. Association for Computational Linguistics.

Kang Xu, Fei Li, Dongdong Xie, and Donghong Ji. 2022. Revisiting aspect-sentiment-opinion triplet extraction: Detailed analyses towards a simple and effective span-based model. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:2918–2927.

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

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2339–2349. Association for Computational Linguistics.

Linyi Yang, Shuibai Zhang, Zhuohao Yu, Guangsheng Bao, Yidong Wang, Jindong Wang, Ruochen Xu, Wei Ye, Xing Xie, Weizhu Chen, and Yue Zhang. 2024. Supervised knowledge makes large language models better in-context learners. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Guoxin Yu, Lemao Liu, Haiyun Jiang, Shuming Shi, and Xiang Ao. 2023. Making better use of training corpus: Retrieval-based aspect sentiment triplet extraction via label interpolation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4914–4927. Association for Computational Linguistics.

Zepeng Zhai, Hao Chen, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022. COM-MRC: A context-masked machine reading comprehension framework for aspect sentiment triplet extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3230–3241. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 504–510. Association for Computational Linguistics.

Yice Zhang, Yifan Yang, Meng Li, Bin Liang, Shiwei Chen, and Ruifeng Xu. 2023. Target-to-source augmentation for aspect sentiment triplet extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12165–12177. Association for Computational Linguistics.

Yice Zhang, Yifan Yang, Yihui Li, Bin Liang, Shiwei Chen, Yixue Dang, Min Yang, and Ruifeng Xu. 2022. Boundary-driven table-filling for aspect sentiment triplet extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6485–6498. Association for Computational Linguistics.

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, Toronto, Canada, July 9-14, 2023*, pages 12038–12050. Association for Computational Linguistics.

## A Error Types and Analysis of GPT-4o Predictions

Although prior studies have shown that LLMs such as ChatGPT underperform compared to smaller supervised fine-tuned models on the ASTE task (Mukherjee et al., 2023; Su et al., 2024), few studies have examined the underlying causes of this performance gap. To address this gap, we conduct a detailed error analysis of GPT-4o[12] on the 15RES (Xu et al., 2020) dataset, identifying common failure patterns in LLM predictions. Based on our analysis, we categorize errors into ten primary types, as illustrated in Figure 6. **(1) Triplet Merging** refers to cases where multiple gold-standard triplets are incorrectly collapsed into a single predicted triplet, typically by merging multiple opinion expressions associated with the same aspect; **(2) Triplet Splitting** occurs when a single gold-standard triplet is erroneously decomposed into multiple predicted triplets due to redundant or fragmented extraction; **(3) Boundary Mismatch** arises when the predicted aspect or opinion spans do not align with the gold-standard boundaries, including over-extension or under-extension; **(4) Pairing Mismatch** denotes incorrect associations between aspect terms and opinion terms that deviate from the gold annotations; **(5) Polarity Mismatch**

[12]OpenAI, version 2024-08-06

| Type | Review | Gold Labels | GPT-4o |
|---|---|---|---|
| Triplet Merging | I thought the restaurant was nice and clean. | (restaurant, nice, positive)<br>(restaurant, clean, positive) | (restaurant, nice and clean, positive) |
| Triplet Splitting | The Dancing, White River and Millenium rolls are musts. | (Dancing, White River and Millenium rolls, musts, positive) | (Dancing, musts, positive)<br>(White River, musts, positive)<br>(Millenium rolls, musts, positive) |
| Boundary Mismatch | We had half/half pizza, mine was eggplant and my friend had the buffalo and it was sooo huge for a small size pizza! | (half/half pizza, huge, positive) | (pizza, huge, positive) |
| | In the summer months, the back garden area is really nice. | (back garden area, nice, positive) | (back garden area, really nice, positive) |
| Pairing Mismatch | Everything is so easy to use, Mac software is just so much simpler than Microsoft software. | (Mac software, easy, positive) | (Mac software, simpler, positive) |
| Polarity Mismatch | Yes , they use fancy ingredients, but even fancy ingredients…… | (ingredients, fancy, positive) | (ingredients, fancy, neutral) |
| Over-Abstraction | Not only is it an adventure getting to this somewhat hidden spot…… | (spot, hidden, neutral) | (location, somewhat hidden, neutral) |
| Spelling Normalization | The wait staff is very courteous and accomodating. | (wait staff, courteous, positive)<br>(wait staff, accomodating, positive) | (wait staff, courteous, positive)<br>(wait staff, accommodating, positive) |
| Abbreviation Normalization | I tried a couple other dishes but was nt too impressed. | (dishes, was nt too impressed, neutral) | (dishes, was not too impressed, neutral) |
| Spurious Prediction | Their sake list was extensive, but we were looking for Purple Haze, which was nt listed but made for us…… | (sake list, extensive , positive) | (sake list, extensive , positive)<br>(Purple Haze, was nt listed but made for us, positive) |
| Triplet Omission | Love Al Di La. | (Al Di La, Love, positive) | None |

Figure 6: Type of prediction errors made by GPT-4o. Discrepancies between the model predictions and the gold labels are highlighted in red.

refers to cases where the predicted sentiment polarity differs from the gold-standard label; **(6) Over-Abstraction** describes situations where the model predicts overly abstract or generalized concepts (e.g., *location*) instead of the concrete entities specified in the annotations; **(7) Spelling Normalization** involves altering the surface form of an expression (e.g., spelling or morphological corrections) in ways that conflict with the annotation standards; **(8) Abbreviation Normalization** occurs when the model converts between abbreviated and full-form expressions inconsistently with the gold annotations; **(9) Spurious Prediction** refers to triplets generated by the model that are not present in the gold-standard labels; and **(10) Triplet Omission** denotes cases where one or more gold-standard triplets are not predicted.

Overall, this prediction bias primarily stems from inconsistencies between the implicit extraction preferences of LLMs and the annotation guidelines adopted in human-labeled datasets. We attribute this phenomenon to both internal and external inconsistency issues within LLMs. Internally, LLMs are exposed to a wide range of textual styles, domains, and task formats during pretraining and instruction tuning (Lu et al., 2024; Atil et al., 2024), making it difficult for them to develop a unified extraction behavior. Externally, even when some degree of internal consistency is achieved, LLMs may still struggle to align with task-specific annotation guidelines without fine-tuning on downstream data. For instance, guidelines may differ on whether opinion terms should include adverbs, or whether aspect terms should incorporate definite or indefinite articles. Furthermore, sentiment polarity annotations are often subjective, with different annotators potentially assigning different sentiments to the same aspect–opinion pair, further complicating alignment between LLM outputs and human annotations. These internal and external inconsistencies collectively limit the performance of LLMs on the ASTE task.

## B Details of Baseline Methods

We compare our framework against three categories of baseline approaches: (i) in-context learning, (ii) LLM-free supervised methods, and (iii) LLM-based supervised methods.

- *In-context learning*: Following Brown et al. (2020), we evaluate the performance of GPT-3.5 and GPT-4o under both zero-shot and few-shot settings. The corresponding prompt templates and evaluation procedures are detailed in Section 3.2.

- *LLM-free supervised methods*: BMRC (Chen et al., 2021a) formulates the ASTE task as a machine reading comprehension (MRC) problem and proposes a bidirectional MRC framework. DLSP (Liu et al., 2024) also adopts a dual-channel MRC design but with a different output order. COM-MRC (Zhai et al., 2022) alleviates interference between multiple aspect terms via contextual masking. Span-ASTE (Xu et al., 2021) extracts aspect and opinion terms through span enumeration. RLI (Yu et al., 2023) improves sentiment classification accuracy by referencing similar

aspect-opinion spans. GTS (Wu et al., 2020) models ASTE as a grid tagging problem with a novel tagging scheme. EMC-GCN (Chen et al., 2022a) improves node representations using a five-channel graph convolutional network that incorporates word-pair relations and syntactic features. SimSTAR (Li et al., 2023) treats ASTE as a span-based table-filling task. MiniConGTS (Sun et al., 2024) builds upon a minimalist tagging framework and introduces a token-level contrastive loss. GAS (Zhang et al., 2021) presents a generative formulation, while CONTRASTE (Mukherjee et al., 2023) extends it with contrastive pre-training and multi-task fine-tuning.

- *LLM-based supervised methods*: SuperContext (Yang et al., 2024) leverages the predictions of a SLM along with their confidence scores as pseudo-labels to guide the reasoning process of LLMs. BAAlign (Li et al., 2025) samples multiple predictions from SLM as prompts for LLMs and employs a discriminator to assess the correctness of the LLM responses. The resulting judgments are then used as feedback signals to iteratively refine the LLM's outputs.

## C Detailed Results of HEMLA with Different Base LLMs

This appendix reports the detailed numerical results corresponding to Figure 3 in Section 4.4. Table 4 presents the detailed results of HEMLA when built upon different base LLMs across four ASTE benchmark datasets.

## D Case Study

In Figure 7, we present two examples where the LLM produces correct predictions, but our alignment model fails to do so. In Example 1, the alignment model correctly identifies both the aspect and opinion terms, but incorrectly classifies the sentiment polarity as negative. This error can be attributed to the relatively limited contextual understanding capability of the alignment model, which is significantly smaller than the LLM. As a result, it places disproportionate emphasis on the word "small", which is generally associated with negative sentiment, while failing to fully capture the overall semantic context. In Example 2, the HEM-P channel generates a correct triplet, whereas the HEM-NP channel omits it. As a result, when the final prediction is produced via the set-matching strategy, this triplet is excluded. While the set-matching strategy improves precision by enforcing agreement between the two channels, it inevitably reduces recall. In real-world deployment scenarios, the choice of fusion strategy should be guided by task-specific requirements. When high recall is preferred over precision, the current strategy can be adapted to perform a union over the outputs from both channels, thereby maximizing the recovery of all potential triplets.

Figure 8 presents two examples in which the alignment model produces correct predictions while the LLM generates incorrect responses. In Example 1, the LLM mistakenly includes the adverb "super" as part of the opinion term, which corresponds to the Boundary Mismatch issue highlighted earlier in Figure 1. In Example 2, the LLM erroneously merges two distinct triplets into one, reflecting the Triplet Merging problem also discussed in Figure 1. In both cases, our alignment model successfully aligns the misleading outputs from the LLM to the correct gold-standard triplets. These examples demonstrate the robustness of the proposed alignment method, which is capable of filtering out misleading information from the LLM while still effectively leveraging its useful knowledge to produce accurate triplet predictions.

| Model | 14LAP | | | 14RES | | | 15RES | | | 16RES | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| DSV3 | 28.20 | 32.16 | 30.05 | 45.13 | 47.59 | 46.33 | 40.23 | 50.93 | 44.95 | 43.13 | 53.70 | 47.83 |
| Qwen3 | 31.16 | 37.15 | 33.90 | 49.64 | 55.53 | 52.42 | 37.48 | 50.93 | 43.18 | 41.58 | 59.53 | 48.96 |
| DSR1 | 36.84 | 37.52 | 37.18 | 53.08 | 55.43 | 54.23 | 44.46 | 53.81 | 48.69 | 51.47 | 57.98 | 54.53 |
| GPT-4o | 38.24 | 40.85 | 39.50 | 54.58 | 56.34 | 55.45 | 49.19 | 56.29 | 52.50 | 53.32 | 60.89 | 56.86 |
| Llama4 | 47.21 | 37.52 | 41.81 | 65.05 | 56.74 | 60.61 | 54.12 | 56.91 | 55.48 | 61.55 | 61.67 | 61.61 |
| Claude4 | 39.25 | 46.21 | 42.44 | 57.79 | 66.40 | 61.80 | 50.55 | 66.60 | 57.47 | 56.16 | 70.04 | 62.34 |
| HEMLA-DSV3 | 64.98 | 59.33 | 62.03 | 72.50 | 70.02 | 71.24 | 64.33 | 66.19 | 65.24 | 73.86 | 75.88 | 74.86 |
| HEMLA-Qwen3 | 68.01 | 59.33 | 63.38 | 75.76 | 67.91 | 71.62 | 67.80 | 65.57 | 66.67 | 74.85 | 75.29 | 75.07 |
| HEMLA-DSR1 | 68.40 | 58.41 | 63.01 | 75.75 | 70.52 | 73.06 | 66.60 | 64.95 | 65.76 | 77.37 | 74.51 | 75.92 |
| HEMLA-4o | 71.36 | 58.78 | 64.45 | 77.17 | 74.63 | **75.90** | 73.19 | 64.74 | 68.71 | 78.60 | 76.46 | **77.51** |
| HEMLA-Llama4 | 70.70 | 59.33 | 64.52 | 74.79 | 73.23 | 74.01 | 68.87 | 66.60 | 67.71 | 76.91 | 76.46 | 76.68 |
| HEMLA-Claude4 | 69.81 | 60.26 | **64.68** | 77.06 | 73.33 | 75.15 | 71.21 | 66.80 | **68.94** | 78.46 | 75.10 | 76.74 |

Table 4: Detailed results of HEMLA with different base LLMs (%).

| | | |
|---|---|---|
| 1 | Review | It s silent and has a very small footprint on my desk. |
| | GPT-4o | (footprint, small, positive) |
| | HEM-NP | (footprint, small, negative) |
| | HEM-P | (footprint, small, negative) |
| | HEMLA | (footprint, small, negative) |
| | Label | (footprint, small, positive) |
| 2 | Review | It has all the expected features and more +plus a wide screen and more than roomy keyboard. |
| | GPT-4o | (features, expected, positive), (screen, wide, positive), (keyboard, roomy, positive) |
| | HEM-NP | (screen, wide, positive), (keyboard, roomy, positive) |
| | HEM-P | (features, expected, positive), (screen, wide, positive), (keyboard, roomy, positive) |
| | HEMLA | (screen, wide, positive), (keyboard, roomy, positive) |
| | Label | (features, expected, positive), (screen, wide, positive), (keyboard, roomy, positive) |

Figure 7: Examples where the LLM predicts correctly but the alignment model fails

| | | |
|---|---|---|
| 1 | Review | Boot time is super fast , around anywhere from 35 seconds to 1 minute. |
| | GPT-4o | (Boot time, super fast, positive) |
| | HEM-NP | (Boot time, fast, positive) |
| | HEM-P | (Boot time, fast, positive) |
| | HEMLA | (Boot time, fast, positive) |
| | Label | (Boot time, fast, positive) |
| 2 | Review | Everything is so easy and intuitive to setup or configure. |
| | GPT-4o | (setup, easy and intuitive, positive), (configure, easy and intuitive, positive) |
| | HEM-NP | (setup, easy, positive), (setup, intuitive, positive), (configure, easy, positive), (configure, intuitive, positive) |
| | HEM-P | (setup, easy, positive), (setup, intuitive, positive), (configure, easy, positive), (configure, intuitive, positive) |
| | HEMLA | (setup, easy, positive), (setup, intuitive, positive), (configure, easy, positive), (configure, intuitive, positive) |
| | Label | (setup, easy, positive), (setup, intuitive, positive), (configure, easy, positive), (configure, intuitive, positive) |

Figure 8: Examples where the alignment model predicts correctly but the LLM fails