Multi-View Similarity Retrieval for In-Context Learning in Aspect Sentiment Triplet Extraction

Anonymous ACL submission

Abstract

Although large language models (LLMs) have achieved remarkable success in many NLP tasks, their performance on Aspect Sentiment Triplet Extraction (ASTE) remains inferior to fully supervised methods, even with in-context learning (ICL). We attribute this gap to inconsistent extraction behaviors and insufficient alignment with annotation standards. To address this issue, we propose a multi-view similarity retrieval (MVSR) framework for selecting in-context learning demonstrations by jointly considering semantic and syntactic information. This strategy improves structural alignment between demonstrations and target inputs, leading to more consistent and accurate triplet extraction. Experiments on four ASTE benchmarks show that our method consistently outperforms existing ICL baselines. In the 10-shot setting, it improves F1 by 2.27%, 2.03%, and 2.00% on 14RES, 15RES, and 16RES, respectively, and even surpasses several supervised fine-tuning baselines. These results highlight the importance of structural information in demonstration selection for structured prediction with LLMs[1].

1 Introduction

Aspect Sentiment Triplet Extraction (ASTE) is a structured information extraction task that aims to identify aspect terms, opinion terms, and their corresponding sentiment polarities from text (Peng et al., 2020; Xu et al., 2020; Wu et al., 2020). For example, given the sentence "*The service was excellent, but the food was terrible*", ASTE extracts the triplets (service, excellent, positive) and (food, terrible, negative).

Most prior work on ASTE has focused on fully supervised approaches, where models are trained on annotated data to accurately extract sentiment triplets. Depending on the modeling paradigm, existing methods include sequence labeling (Mao et al., 2021; Mukherjee et al., 2023), span enumeration (Xu et al., 2021; Chen et al., 2022b), table-filling (Chen et al., 2022a; Zhang et al., 2022), and generative frameworks (Zhang et al., 2021; Zhou and Qian, 2023). With the recent success of large language models (LLMs) across a wide range of NLP tasks, increasing attention has been paid to their potential for information extraction (Delbrouck et al., 2024; Ding et al., 2024; Hong and Liu, 2024). However, for ASTE, LLMs still substantially underperform smaller fully supervised models (Mukherjee et al., 2023; Su et al., 2024; Sun et al., 2024), revealing a persistent performance gap that is not yet well understood.

Example 1:
Sentence: Boot time is super fast, around anywhere ...
Label: (Boot time, fast, positive)
GPT-4o: (Boot time, super fast, positive)

Example 2:
Sentence: the hardware problems have been so bad ...
Label: (hardware , bad, negative)
GPT-4o: (hardware problems, bad, negative)

Figure 1: Examples of GPT-4o predictions that differ from gold labels but remain semantically similar.

To better understand this performance gap, we conduct a human evaluation of predictions generated by GPT-4o[2]. The results show that a substantial portion of predictions marked as incorrect by automatic metrics are in fact semantically equivalent to the gold annotations. Figure 1 presents two representative examples illustrating such discrepancies. These errors primarily arise from mismatches between model outputs and annotation standards, including boundary mismatches and other issues detailed in Appendix A.

We attribute these errors to two sources of in-

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

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

consistency inherent in LLMs. **Internally**, during pretraining and instruction tuning, LLMs are exposed to diverse writing styles, domains, and task formulations (OpenAI, 2023), which can hinder the emergence of stable and consistent extraction behaviors. **Externally**, LLMs often diverge from human annotators in their interpretation of the ASTE task. When provided with only brief task descriptions and a limited number of reference demonstrations—which may be noisy or fail to cover the full range of extraction scenarios—LLMs may struggle to consistently adhere to annotation conventions. Together, these internal and external inconsistencies limit the effectiveness of LLMs on ASTE.

To mitigate these issues, we adopt the in-context learning (ICL) paradigm (Brown et al., 2020), retrieving representative demonstrations to help LLMs better internalize the extraction objective and reduce behavioral variability. However, most existing ICL methods rely on model-internal signals, such as predicted probabilities (Zhang et al., 2025; Liang et al., 2025) or hidden states (Peng et al., 2025; Wang et al., 2025), which are inaccessible for proprietary LLMs like GPT. In contrast, model-agnostic ICL approaches typically either select demonstrations based on model performance over a candidate pool (Wu et al., 2024; Liu et al., 2024a) or improve task understanding through carefully designed prompting strategies (Honda and Oka, 2025; Cho et al., 2025). Nevertheless, these methods largely overlook the role of structural information in enabling LLMs to consistently interpret task requirements and generate reliable structured outputs. The work most closely related to ours is AMR-RE (Han et al., 2025), which incorporates abstract meaning representations (AMR) for structure-aware demonstration retrieval in relation extraction. While AMR provides a high-level semantic abstraction of sentences, it primarily models predicate–argument relations and tends to obscure surface-level syntactic structure and word-level alignments. Such information is critical for retrieving structurally representative demonstrations in ASTE, where fine-grained boundary and correspondence consistency play a central role.

To address these limitations, we propose **Multi-View Similarity Retrieval (MVSR)**, a demonstration selection strategy that jointly considers semantic and syntactic similarity. By retrieving demonstrations that are aligned with the input in both content and structure, MVSR encourages more stable and consistent extraction behavior in LLMs. We evaluate MVSR on four widely used ASTE benchmarks and show that it consistently outperforms existing ICL retrieval methods, and under favorable settings, even surpasses several supervised fine-tuning baselines.

Our main contributions are summarized as follows:

- We propose a multi-view similarity retrieval method that jointly models semantic and syntactic similarity to select structurally aligned demonstrations for ICL.
- Experiments on four ASTE benchmarks show that the proposed method improves the consistency of LLMs in triplet extraction and generally outperforms existing ICL baselines.
- We provide a systematic analysis of how structural similarity measures, the number of demonstrations, and their ordering affect ICL performance, offering practical insights for demonstration selection.

2 Related Work

2.1 Aspect Sentiment Triplet Extraction

Most prior work on ASTE has focused on fully supervised settings. Existing approaches can be broadly categorized into four modeling paradigms: sequence labeling, table filling, span enumeration, and generative methods.

The ASTE task was first introduced by Peng et al. (2020), who proposed a pipeline framework based on sequence labeling. Building on this formulation, Xu et al. (2020) incorporated positional information and developed an end-to-end joint model. Subsequent studies (Mao et al., 2021; Chen et al., 2021; Zhai et al., 2022) further advanced this line of work by reformulating ASTE as a machine reading comprehension problem, while still largely adhering to the sequence labeling paradigm.

To alleviate error propagation in pipeline architectures, Wu et al. (2020) proposed a grid-based table-filling framework that models aspect-opinion-sentiment interactions in a unified manner. Following this direction, Chen et al. (2022a), Zhang et al. (2022), and Liang et al. (2023) introduced enhanced labeling schemes that incorporate syntactic features, boundary information, and multi-dimensional representations. Although these end-to-end models effectively reduce error accumulation, they often rely on dense token-pair interac-

tions and can struggle with sentences containing multiple aspect-opinion pairs.

Span-based approaches have been proposed to further improve joint modeling. Xu et al. (2021) introduced a span enumeration framework for ASTE, which was later extended by Chen et al. (2022b) and Xu et al. (2025) through contrastive learning and syntactic information to reduce feature ambiguity among neighboring spans. Additionally, Yang et al. (2024) combined span enumeration with table-filling to capture interactions at multiple granularities, while Zou et al. (2026) incorporated dependency and constituency structures using graph neural networks.

More recently, generative approaches based on causal decoder architectures have attracted increasing attention due to their strong generalization ability and compatibility with large language models. Zhang et al. (2021) first explored a generative formulation of ASTE, though its performance was constrained by limited model capacity and task complexity. Subsequent work (Zhou and Qian, 2023; Mukherjee et al., 2023) improved generative methods by integrating auxiliary sequence labeling tasks and contrastive pretraining, achieving performance competitive with, or even surpassing, other end-to-end paradigms.

2.2 In-Context learning

Existing research on ICL can be broadly divided into model-aware and model-agnostic approaches. As our work targets scenarios where model parameters are inaccessible, such as proprietary GPT-series models, we focus on model-agnostic methods. Early studies in this setting primarily retrieved demonstrations based on semantic similarity between inputs (Liu et al., 2022).

More recent work largely rely on LLM behavior over a candidate pool to guide demonstration selection. For example, Zhou et al. (2024) select demonstrations that are correctly predicted by the LLM, while Wu et al. (2024) organize candidate instances according to their estimated difficulty. Building on this idea, Mo et al. (2024) partition the candidate pool based on prediction correctness and sample demonstrations from each subset. Subsequent work augments demonstrations with explanatory content to enhance task understanding (Xu et al., 2024; Cho et al., 2025), while related approaches (He et al., 2024; Honda and Oka, 2025) similarly incorporate explanations without explicitly modeling demonstration difficulty.

Despite their effectiveness, these methods generally require querying the LLM to obtain prior predictions or explanations for candidate instances, which substantially limits scalability and practicality. To address this issue, Liu et al. (2024a) and Gao et al. (2025) construct training datasets based on LLM behavior and train retrieval models via contrastive learning, enabling demonstration selection without LLM inference at test time. However, such retrievers still depend on LLM-generated data during training and often require re-training or fine-tuning when applied to new tasks.

More recently, Han et al. (2025) proposed a structure-aware retrieval method based on AMR for relation extraction. While AMR captures sentence semantics in a structured form and is effective at modeling high-level relational structures, it primarily encodes predicate-argument structures and does not explicitly preserve surface-level syntactic structures or word-level correspondences. As a result, such abstractions are less suitable for tasks like ASTE, which require fine-grained alignment between semantic content and surface structure for effective demonstration retrieval.

3 Methodology

As illustrated in Figure 2, MVSR adopts a two-stage pipeline consisting of retrieval and inference. In the retrieval stage, we first extract semantic and syntactic features for all candidate examples using a semantic encoder and a syntactic parser. Given a test instance, MVSR computes its similarity to each candidate from multiple views, ranks all candidates based on the aggregated similarity scores, and selects the top-K instances as in-context demonstrations. In the inference stage, the selected demonstrations are concatenated with the test instance in the same similarity-based order and fed into the LLM, which generates the final aspect sentiment triplet predictions.

3.1 Problem Definition

Given an input text, the goal of ASTE is to extract a set of sentiment triplets $T = \{(a_i, o_i, s_i)\}_{i=1}^{|T|}$, where a_i denotes an aspect term, o_i an opinion term, and $s_i \in \{\text{Positive}, \text{Negative}, \text{Neutral}\}$ the corresponding sentiment polarity.

Under the ICL paradigm, given an input instance x and a candidate pool C[3] of labeled examples,

[3]In this work, the candidate pool C corresponds to the training set.

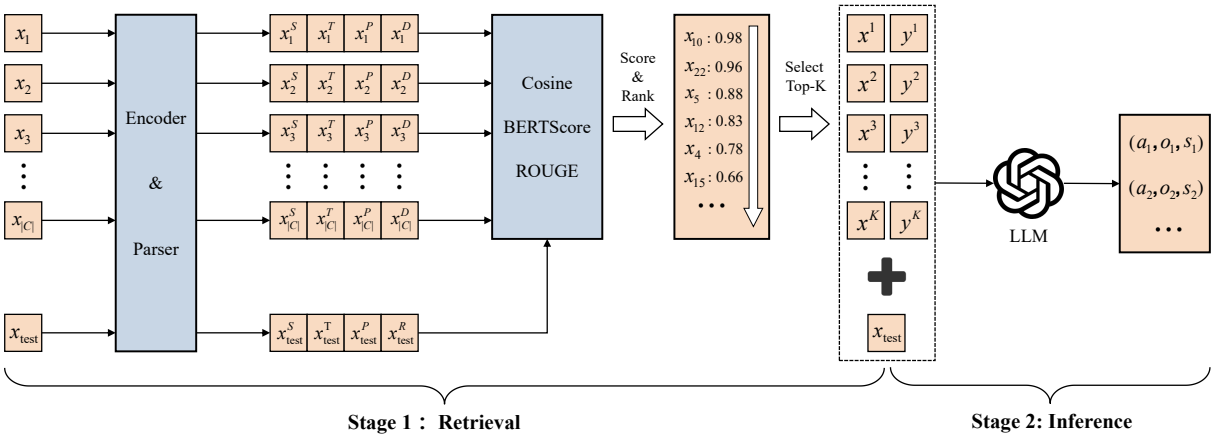

Figure 2: Overview of MVSR for ASTE. Encoder and Parser denote the semantic encoder and syntactic parser, respectively. x_i^S, x_i^T, x_i^P, and x_i^D represent sentence-level semantic features, token-level semantic features, part-of-speech features, and dependency relation features, respectively.

the objective is to select K demonstrations from C to guide a LLM in generating correct sentiment triplets. Formally, given an LLM $\mathcal{M}$, the prediction process can be expressed as:

$$y = \mathcal{M}\big(\{(x_i, y_i)\}_{i=1}^{K} \oplus x\big), \tag{1}$$

where $(x_i, y_i) \in C$, $\oplus$ denotes concatenation, and y represents the serialized output of the extracted sentiment triplets. For simplicity, task instructions and prompt templates are omitted.

3.2 Retrieval Stage

To support similarity-based demonstration retrieval, we represent each candidate example from both semantic and syntactic perspectives. These representations are used to compute similarity between a test instance and candidate examples, enabling the selection of structurally aligned demonstrations. The feature extraction module consists of two components: a semantic encoder for capturing contextual semantics and a syntactic parser for modeling surface-level structural information.

3.2.1 Semantic Encoder

We employ a pre-trained BERT encoder (Devlin et al., 2019) to obtain semantic representations of the input text. Given an input sequence $x = \{w_1, w_2, \ldots, w_n\}$, we prepend the special token [CLS] and append [SEP]. The resulting sequence is fed into the BERT encoder to produce contextualized representations $H = \{h_{\text{CLS}}, h_1, \ldots, h_n, h_{\text{SEP}}\}$. We use h_{CLS} as a sentence-level semantic representation and $\{h_i\}_{i=1}^{n}$ as token-level representations, which are later used for both global and fine-grained similarity computation.

3.2.2 Syntactic Parser

While semantic similarity is effective for identifying demonstrations with similar content, it is often insufficient for structured extraction tasks such as ASTE, where LLMs are prone to structural errors (as shown in Figure 1). As illustrated in Figure 3, Example 2 achieves a higher semantic similarity score with the test instance than Example 1[4] (0.8939 vs. 0.8372). However, Example 1, whose syntactic structure more closely matches that of the test sentence, provides more effective demonstrations for inducing consistent extraction behavior. This example highlights a key limitation of semantic-only retrieval: sentences that are semantically similar may differ substantially in syntactic structure, potentially misleading LLMs during ICL.

In general, sentences with similar syntactic structures tend to exhibit more aligned part-of-speech (POS) tags and dependency relations. Motivated by this observation, we incorporate syntactic information as a complementary signal to semantic similarity. Specifically, we use the spaCy parser[5] to extract POS tags and dependency relations, which form the basis of our syntactic similarity computation.

We explore both tree-based and sequence-based formulations for modeling syntactic similarity and ultimately adopt the sequence-based approach due to its greater stability and stronger empiri-

[4]Semantic similarity is computed using the method of SBERT (Reimers and Gurevych, 2019).

[5]https://spacy.io/

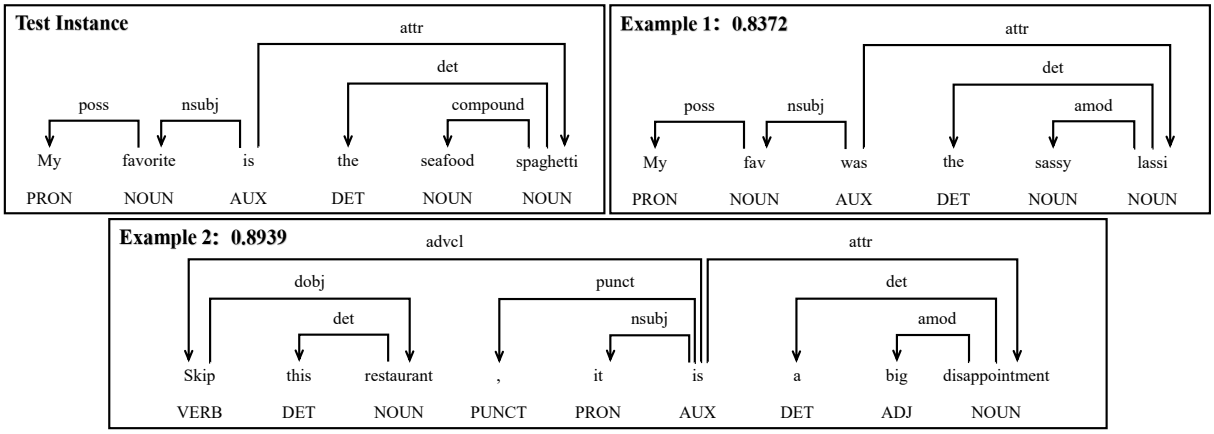

Figure 3: Syntactic structure and semantic similarity scores of two candidate examples.

cal performance. Formally, given a sentence x, we serialize its syntactic structures into a POS sequence $x^P = \{p_1, p_2, \ldots, p_n\}$ and a dependency sequence $x^D = \{d_1, d_2, \ldots, d_n\}$. For example, for the test instance in Figure 3, the corresponding POS and dependency sequences are $x^P = \{\texttt{PRON}, \texttt{NOUN}, \texttt{AUX}, \texttt{DET}, \texttt{NOUN}, \texttt{NOUN}\}$ and $x^D = \{\texttt{poss}, \texttt{nsubj}, \texttt{root}, \texttt{det}, \texttt{compound}, \texttt{attr}\}$, respectively.

Although dependency trees preserve richer hierarchical structure, our experiments show that serialized syntactic sequences—despite discarding explicit tree relations—consistently lead to more effective demonstration retrieval in the ICL setting. We provide a detailed analysis of this counterintuitive result in Appendix D.

3.2.3 Similarity Computation

We compute the similarity between a test instance and a candidate example from four complementary views: sentence-level semantics, token-level semantics, POS tags, and dependency relations. This multi-view formulation enables MVSR to capture semantic correspondence while promoting syntactic structural alignment, leading to more reliable demonstration selection.

Sentence-level similarity. Given the sentence-level representations h_{CLS} and $\hat{h}_{\text{CLS}}$, we measure global semantic similarity using cosine similarity:

$$\text{Sim}_S = \frac{h_{\text{CLS}} \cdot \hat{h}_{\text{CLS}}}{\|h_{\text{CLS}}\| \, \|\hat{h}_{\text{CLS}}\|} \tag{2}$$

Token-level similarity. Given the token representations $H = \{h_1, h_2, \ldots, h_n\}$ and $\hat{H} = \{\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_m\}$, we compute token-level semantic similarity using BERTScore[6] (Zhang et al., 2020), which measures fine-grained semantic alignment at the token level:

$$\text{Sim}_T = \text{BERTScore}(H, \hat{H}) \tag{3}$$

POS similarity. Given the POS sequences $x^P = \{p_1, p_2, \ldots, p_n\}$ and $\hat{x}^P = \{\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_m\}$, we quantify their similarity using the average ROUGE[7] score (Lin, 2004):

$$\text{Sim}_P = \frac{1}{|N|} \sum_{n \in N} \text{ROUGE-}n(x^P, \hat{x}^P), \tag{4}$$

where $N = \{1, 2, 3, L\}$. ROUGE-1, ROUGE-2, and ROUGE-3 capture local n-gram overlap, while ROUGE-L measures longest common subsequence similarity, jointly reflecting both local and global consistency of syntactic structures.

Dependency similarity. Analogously, dependency similarity is computed as the average ROUGE score between the serialized dependency relation sequences x^D and $\hat{x}^D$:

$$\text{Sim}_D = \frac{1}{|N|} \sum_{n \in N} \text{ROUGE-}n(x^D, \hat{x}^D) \tag{5}$$

Final score. The overall similarity is computed as a weighted linear combination of the four similarity components:

$$\text{Sim} = \lambda_1 \text{Sim}_S + \lambda_2 \text{Sim}_T + \lambda_3 \text{Sim}_P + \lambda_4 \text{Sim}_D, \tag{6}$$

where λ_i controls the contribution of each view. Candidate examples are ranked according to this score, and the top-K instances are selected as in-context demonstrations.

[6] We use the F1 variant of BERTScore.

[7] We use the F1 variant of ROUGE.

3.3 Inference Stage

After retrieving the top-K demonstrations $\{(x_i, y_i)\}_{i=1}^{K}$, we combine them with the test instance and insert the resulting sequence into a predefined prompt template containing task instructions and formatted examples. The constructed prompt is then fed into the LLM, which generates the sentiment triplets in an ICL setting. The prompt template used in MVSR is shown below, where x_i and y_i denote the input and corresponding label of the i-th demonstration.

Prompt Template

You are an expert in textual sentiment analysis. Given a text, 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: $\{x_{\text{test}}\}$ Answer:

We further analyze the effects of varying the number and ordering of demonstrations, with detailed results reported in Appendix E.

4 Experiments

4.1 Experiment Setup

Datasets. We conduct experiments on four widely used benchmark datasets: 14LAP, 14RES, 15RES, and 16RES, all sourced from the ASTE-DATA-V2 (Peng et al., 2020) collection and covering both laptop and restaurant domains. Each dataset is split into training, development, and test sets. Under the ICL setting, we use the training set as the candidate pool, without applying any task-specific fine-tuning. Detailed dataset statistics are provided in Table 1.

Baselines. To evaluate the effectiveness of the proposed approach, we compare it against several retrieval-based ICL baselines, including Random, SBERT (Reimers and Gurevych, 2019), BERTScore (Zhang et al., 2020), BM25 (Robertson and Zaragoza, 2009), F-ICL (Xu et al., 2024), CoQ (Wu et al., 2024) and AMR-RE (Han et al., 2025), under both 5-shot and 10-shot settings. We further compare it with a range of fully supervised fine-tuning methods, including JET (Xu et al., 2020), GAS (Zhang et al., 2021), GTS (Wu et al., 2020), Span-ASTE (Xu et al., 2021), EMC-GCN (Chen et al., 2022a), COM-MRC (Zhai et al., 2022), RLI (Yu et al., 2023), SimSTAR (Li et al., 2023), CONTRASTE (Mukherjee et al., 2023), DLSP (Liu et al., 2024b), and MiniConGTS (Sun et al., 2024). In addition, we report the zero-shot performance of the backbone model to assess its inherent capabilities. For detailed descriptions of the baselines and implementation details, please refer to Appendices B and C.

Dataset		#S	#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 four benchmark datasets. #S and #T denotes the total number of sentences and triplets. #POS, #NEG, and #NEU indicate the number of positive, negative, and neutral sentiments triplets, respectively. #A and #O refer to the number of aspect terms and opinion terms.

4.2 Main Results

4.2.1 Comparison with ICL Methods

Table 2 summarizes the performance of MVSR and all baseline methods on the test sets. Overall, MVSR consistently outperforms existing ICL approaches on most datasets. An exception is observed on the 14LAP dataset, where BERTScore and BM25 achieve slightly better performance. Both BERTScore and BM25 rely on surface-level lexical or semantic similarity, whereas MVSR additionally incorporates syntactic information. In this case, the inclusion of syntactic features appears less beneficial. Further analysis indicates that 14LAP exhibits the lowest structural similarity between its training and test sets, as measured by average syntactic similarity scores (see Appendix E.3). This structural divergence between the training and test sets may reduce the effectiveness of syntactic alignment, partially accounting for MVSR's comparatively weaker performance on this dataset.

Shot	Method	14LAP			14RES			15RES			16RES		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
0-shot	GPT-4o-mini	32.91	38.45	35.46	52.44	57.34	54.78	43.25	55.46	48.60	48.92	61.87	54.64
5-shot	Random (Brown et al., 2020)	35.29	41.46	38.13	54.36	60.80	57.40	46.71	59.52	52.34	51.77	64.53	57.45
	SBERT (Reimers and Gurevych, 2019)	41.75	48.92	45.04	56.10	63.68	59.65	45.30	59.24	51.34	55.06	68.38	61.00
	BERTScore (Zhang et al., 2020)	42.19	51.57	46.41	57.22	65.43	61.04	47.40	61.31	53.46	54.06	69.52	60.83
	BM25 (Robertson and Zaragoza, 2009)	42.99	51.57	**46.89**	57.98	64.47	61.06	52.58	59.31	55.76	54.33	68.74	60.69
	F-ICL (Xu et al., 2024)	32.05	39.93	35.56	53.50	62.27	57.55	44.13	57.32	49.97	52.48	67.90	59.20
	CoQ (Wu et al., 2024)	39.91	47.87	43.53	52.44	55.23	53.80	46.31	60.82	52.58	59.26	63.18	61.16
	AMR-RE (Han et al., 2025)	39.39	45.29	42.13	57.83	65.39	61.38	47.19	58.97	52.43	55.10	67.32	60.60
	MVSR (Ours)	42.36	51.20	46.36	59.73	68.58	**63.85***	51.24	65.29	**57.42***	57.28	72.18	**63.87***
10-shot	Random (Brown et al., 2020)	37.06	42.82	39.73	55.56	61.97	58.59	46.71	59.93	52.50	53.71	65.89	59.18
	SBERT (Reimers and Gurevych, 2019)	41.65	49.35	45.18	58.25	66.80	62.23	48.18	62.68	54.48	55.87	72.18	62.99
	BERTScore (Zhang et al., 2020)	44.75	52.89	**48.45**	59.82	68.01	63.65	50.00	63.37	55.90	57.02	72.44	63.81
	BM25 (Robertson and Zaragoza, 2009)	44.40	53.23	48.42	60.00	69.45	64.38	52.28	61.24	56.42	57.23	72.11	63.82
	F-ICL (Xu et al., 2024)	32.10	39.93	35.58	53.24	62.07	57.32	44.81	58.76	50.85	52.41	67.70	59.08
	CoQ (Wu et al., 2024)	40.07	43.25	41.60	59.74	65.69	62.58	47.38	61.44	53.50	54.22	70.04	61.12
	AMR-RE (Han et al., 2025)	43.23	49.54	46.17	59.40	66.10	62.57	50.34	61.65	55.42	57.37	70.43	63.23
	MVSR (Ours)	44.00	52.62	47.92	62.06	71.96	**66.65***	52.43	66.05	**58.45***	59.19	74.12	**65.82***

Table 2: Comparison between MVSR and other ICL methods under 0-shot, 5-shot, and 10-shot settings, with all methods evaluated using GPT-4o-mini as the backbone model. The best results are shown in bold, and the second-best results are underlined. The $*$ marker denotes statistically significant improvements of MVSR over the second-best result ($p < 0.01$).

We further compare MVSR with several recent ICL methods, including F-ICL, CoQ, and AMR-RE, which have demonstrated strong performance on classification, question answering, and relation extraction tasks, respectively. All three methods perform consistently worse than MVSR on ASTE and exhibit greater performance variability across datasets. While most ICL approaches benefit from increasing the number of demonstrations, F-ICL and CoQ experience performance degradation on certain datasets as more examples are introduced. One possible explanation is that F-ICL constructs demonstrations based on incorrectly predicted instances, which may introduce noise and hinder the learning of consistent extraction behaviors. Although CoQ employs chain-of-thought prompting to generate contextually coherent demonstrations, its selected examples are often overly simplistic and less informative for complex extraction scenarios. Moreover, CoQ applies a fixed demonstration set across all test instances, limiting its adaptability to input variation. AMR-RE, which is more closely related to MVSR, incorporates structured information into demonstration retrieval. However, its AMR-based representations emphasize abstract semantic relations and do not explicitly model surface-level syntactic structure or word-level alignment. Consequently, compared with MVSR, AMR-RE provides less effective guidance for learning consistent extraction behavior, resulting in inferior performance on ASTE.

4.2.2 Comparison with Fine-Tuning Methods

As shown in Table 3, when paired with a stronger backbone model such as GPT-4o, MVSR achieves competitive performance in the 32-shot setting and outperforms several supervised fine-tuning baselines. This observation indicates that incorporating structurally aligned demonstrations into ICL can substantially narrow the performance gap between ICL-based methods and fully supervised approaches. In particular, by selecting demonstrations that better reflect the structural regularities of the target task, MVSR enables LLMs to generate more consistent and accurate structured outputs. These results further suggest that LLMs, when guided by appropriate demonstration selection strategies, can be effectively applied to structured prediction tasks without requiring task-specific fine-tuning.

More broadly, these findings indicate that structure-aware demonstration retrieval is especially beneficial in scenarios where annotated data is limited, expensive to obtain, or where fine-tuning large models is impractical. By leveraging only a small number of labeled examples, MVSR provides a flexible alternative that can adapt to new tasks without additional training overhead. Nevertheless, despite its strong performance, MVSR does not yet match state-of-the-art supervised fine-tuning methods. Our objective is therefore not to replace supervised learning, but to complement it by offering a cost-effective and flexible ICL-based solution that broadens the applicability of LLMs to complex

Type	Method	14LAP			14RES			15RES			16RES		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
FT	JET (Xu et al., 2020)	55.39	47.33	51.04†	70.56	55.94	62.40†	64.45	51.96	57.53†	70.42	58.37	63.83†
	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†
	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†
	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	56.26	58.81	71.21	72.39	71.78	61.54	62.47	61.93†	65.62	71.30	68.33†
	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., 2024b)	69.26	55.82	61.82	76.75	71.14	73.84	69.92	64.81	**67.26**	75.62	74.04	74.04
	MiniConGTS (Sun et al., 2024)	66.82	60.68	**63.61**	76.10	75.08	**75.59**	66.50	63.86	65.15	75.52	74.14	**74.83**
ICL	MVSR (Ours)	53.67	56.75	55.17	65.90	71.73	68.69	60.84	65.36	63.02	65.88	75.88	70.52

Table 3: Comparison between MVSR (32-shot, GPT-4o) and supervised fine-tuning methods. The best results are shown in bold, and † indicates performance lower than that of MVSR.

Shot	Method	14LAP			14RES			15RES			16RES		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
5-Shot	MVSR	42.36	51.20	46.36	59.73	68.58	**63.85**	51.24	65.29	**57.42**	57.28	72.18	**63.87**
	-POS	42.66	51.02	46.46	58.91	67.92	63.06	49.68	63.30	55.67	55.18	69.46	61.50
	-DEP	44.13	52.56	**47.98**	59.20	67.08	62.93	49.44	63.09	55.43	56.29	71.15	62.81
	-POS&DEP	42.14	51.20	46.23	55.97	63.95	59.69	47.17	60.75	53.11	56.11	71.11	62.68

Table 4: Ablation study under the 5-shot setting using GPT-4o-mini. The best results are shown in bold.

structured prediction tasks.

4.3 Ablation Study

To assess the contribution of individual components in MVSR, we conduct ablation studies on four benchmark datasets, with results reported in Table 4. Specifically, the –POS variant removes the POS-based similarity component, the –DEP variant removes the dependency-based similarity component, and the –POS&DEP variant excludes both syntactic similarity signals. Overall, the full MVSR model achieves the best performance on most datasets, indicating the effectiveness of incorporating syntactic information for demonstration retrieval.

On the 14LAP dataset, however, removing syntactic similarity does not result in performance degradation. Notably, the –DEP variant slightly outperforms the full model. This observation is consistent with our analysis in Section 4.2.1, which shows that 14LAP exhibits a relatively large structural divergence between the training and test sets. In such cases, syntactic alignment may be less beneficial, and incorporating syntactic similarity can introduce noise into the retrieval process, leading to reduced performance.

5 Conclusion

In this paper, we propose MVSR, a multi-view demonstration selection strategy for ICL on the ASTE task. By jointly leveraging semantic and syntactic information, MVSR retrieves structurally aligned demonstrations that more effectively guide LLMs. Experimental results across four benchmarks demonstrate that MVSR consistently outperforms existing ICL baselines. Further analyses show that syntactic similarity measures, as well as the number and ordering of demonstrations, play a critical role in ICL performance. These findings highlight the importance of structure-aware demonstration selection and provide new insights into applying LLMs to structured information extraction tasks.

Limitations

Despite MVSR's strong performance compared to existing ICL methods, a noticeable gap remains between our approach and fully supervised models on the ASTE task. This gap reflects a broader limitation of ICL-based approaches: even with structure-aware demonstration retrieval, they do not yet fully match the effectiveness of dedicated task-specific training. Addressing this limitation may require hybrid paradigms that combine the flexibility of ICL with limited supervised signals, such as weak

supervision or semi-supervised learning.

In addition, while MVSR incorporates both semantic and syntactic information for demonstration retrieval, the fusion of these signals remains relatively coarse. Specifically, our current implementation relies on a weighted linear combination of multiple similarity measures, which may not sufficiently capture complex interactions between semantic content and syntactic structure. More expressive fusion strategies—such as adaptive or context-aware weighting, learned similarity fusion modules, or joint semantic–syntactic representation learning—represent promising directions for future work and may further improve retrieval quality and downstream performance.

References

Philip Bille. 2005. A survey on tree edit distance and related problems. *Theor. Comput. Sci.*, 337(1-3):217–239.

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, and 12 others. 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. 2021. 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.

Hyundong Justin Cho, Karishma Sharma, Nicolaas Paul Jedema, Leonardo F. R. Ribeiro, Jonathan May, and Alessandro Moschitti. 2025. Tuning-free personalized alignment via trial-error-explain in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5864–5885. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. 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.

Xiang Gao, Ankita Sinha, and Kamalika Das. 2025. Learning to search effective example sequences for in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6137–6146. Association for Computational Linguistics.

Peitao Han, Lis Pereira, Fei Cheng, Wan Jou She, and Eiji Aramaki. 2025. AMR-RE: Abstract Meaning Representations for retrieval-based in-context learning in relation extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 333–342. Association for Computational Linguistics.

Xuanli He, Yuxiang Wu, Oana-Maria Camburu, Pasquale Minervini, and Pontus Stenetorp. 2024. Using natural language explanations to improve robustness of in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13477–13499. Association for Computational Linguistics.

Ukyo Honda and Tatsushi Oka. 2025. Exploring explanations improves the robustness of in-context learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 23693–23714. 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.

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.

Siqi Liang, Sumyeong Ahn, Paramveer Dhillon, and Jiayu Zhou. 2025. Dual debiasing for noisy in-context learning for text generation. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 12855–12868. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. 2024a. Se^2: Sequential example selection for in-context learning. *CoRR*, abs/2402.13874.

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 Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114. Association for Computational Linguistics.

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

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. 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 13543–13551. AAAI Press.

Ying Mo, Jiahao Liu, Jian Yang, Qifan Wang, Shun Zhang, Jingang Wang, and Zhoujun Li. 2024. C-ICL: Contrastive in-context learning for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10099–10114. 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.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Mateusz Pawlik and Nikolaus Augsten. 2015. Efficient computation of the tree edit distance. *ACM Trans. Database Syst.*, 40(1):3:1–3:40.

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.

Keqin Peng, Liang Ding, Yuanxin Ouyang, Meng Fang, Yancheng Yuan, and Dacheng Tao. 2025. Enhancing input-label mapping in in-context learning with contrastive decoding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 997–1004. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

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.

Shaobo Wang, Xiangqi Jin, Ziming Wang, Jize Wang, Jiajun Zhang, Kaixin Li, Zichen Wen, Zhong Li, Conghui He, Xuming Hu, and Linfeng Zhang. 2025. Data whisperer: Efficient data selection for task-specific LLM fine-tuning via few-shot in-context learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 23287–23305. Association for Computational Linguistics.

Yiquan Wu, Anlai Zhou, Yuhang Liu, Yifei Liu, Adam Jatowt, Weiming Lu, Jun Xiao, and Kun Kuang. 2024. Chain-of-quizzes: Pedagogy-inspired example selection in in-context-learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10136–10142. 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.

Guangtao Xu, Zhihao Yang, Bo Xu, Ling Luo, and Hongfei Lin. 2025. Span-based syntactic feature fusion for aspect sentiment triplet extraction. *Inf. Fusion*, 120:103078.

Hongling Xu, Qianlong Wang, Yice Zhang, Min Yang, Xi Zeng, Bing Qin, and Ruifeng Xu. 2024. Improving in-context learning with prediction feedback for sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3879–3890. Association for Computational Linguistics.

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.

Xuan Yang, Tao Peng, Haijia Bi, and Jiayu Han. 2024. Span-level bidirectional retention scheme for aspect sentiment triplet extraction. *Inf. Process. Manag.*, 61(5):103823.

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*, 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*, pages 3230–3241. Association for Computational Linguistics.

Kaizhong Zhang and Dennis E. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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, 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.

Zheng Zhang, Shaocheng Lan, Lei Song, Jiang Bian, Yexin Li, and Kan Ren. 2025. Learning to select in-context demonstration preferred by large language model. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 11345–11360. Association for Computational Linguistics.

Anlai Zhou, Sunshine Jiang, Yifei Liu, Yiquan Wu, Kun Kuang, and Jun Xiao. 2024. Latent learningscape guided in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8090–8101. 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.

Wang Zou, Xia Sun, Maofu Liu, Yaqiong Xing, Xiaodi Zhao, and Jun Feng. 2026. Leveraging dependency and constituent graphs for aspect sentiment triplet extraction. *Inf. Fusion*, 127:103723.

A Error Analysis of GPT-4o

Figure 4 presents a taxonomy of error types observed in GPT-4o predictions, together with representative examples. We summarize the errors as follows. **(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 both over-extension and under-extension. **(4) Pairing Mismatch** denotes incorrect associations between aspect terms and opinion terms that deviate from the gold annotations. **(5) Polarity Mismatch** refers to cases where the predicted sentiment polarity differs from the gold-standard label. **(6) Over-Abstraction** describes situations in which 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 in a manner inconsistent with the gold annotations. **(9) Spurious Prediction** refers to triplets generated by the model that are not present in the gold-standard labels. **(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 (**??**), 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 Descriptions of the Baseline Methods

The detailed descriptions of the baselines are as follows:

- **Random**: Randomly selects K training examples as demonstrations.
- **SBERT**: Uses cosine similarity between the sentence embeddings of test and training instances (computed via SBERT) to select the top K similar examples.
- **BERTScore**: Computes token-level semantic similarity between training and test instances using BERTScore, and selects the top K examples accordingly.
- **BM25**: Ranks training examples based on BM25 scores with respect to the test instance, and selects the top K as demonstrations.

Type	Review	Gold Labels	GPT-4o
Triplet Merging	I thought the restaurant was nice and clean.	(restaurant, nice, positive) (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) (White River, musts, positive) (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) (wait staff, accomodating, positive)	(wait staff, courteous, positive) (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) (Purple Haze, was nt listed but made for us, positive)
Triplet Omission	Love Al Di La.	(Al Di La, Love, positive)	None

Figure 4: Representative examples of prediction errors made by GPT-4o, categorized by error type.

- **F-ICL**: First evaluates the zero-shot performance of the backbone model on the training set, then partitions the training instances into two subsets: *easy* (correctly predicted) and *hard* (incorrectly predicted). It samples $K/2$ instances from each subset and assigns gold labels to the mispredicted instances in the hard subset, explicitly indicating that the model's previous predictions were incorrect.

- **CoQ**: Performs chain-of-questioning inference on the training set using the backbone model. Once K consecutive correct predictions are obtained, they are combined to form a demonstration chain[8].

- **AMR-RE**: Retrieves demonstrations based on the similarity of the structured AMR representations between test and training instances.

- **JET**: Proposes a position-aware sequence labeling approach to model the positional relationships between aspects and opinions.

- **GTS**: Formulates ASTE as a grid tagging problem with a novel tagging scheme.

- **GAS**: Addresses ASTE using a generative modeling paradigm.

- **Span-ASTE**: Extracts aspect and opinion terms through span enumeration.

- **EMC-GCN**: Enhances node representations using a five-channel graph convolutional network that incorporates word-pair relations and syntactic features.

- **COM-MRC**: Formulates ASTE as a machine reading comprehension (MRC) task and introduces a masking mechanism to mitigate the influence of irrelevant contextual information.

- **RLI**: Enhances sentiment classification by referencing similar aspect–opinion spans.

- **SimSTAR**: Treats ASTE as a span-based table-filling task.

- **CONTRASTE**: Extends GAS with contrastive pre-training and multi-task fine-tuning.

- **DLSP**: Proposes a dual-channel decoding strategy within the MRC framework.

- **MiniConGTS**: Builds upon a minimalist tagging framework and introduces a token-level contrastive loss.

C Implementation Details

We use GPT-4o (version 2024-08-06) and GPT-4o-mini (version 2024-07-18) as the backbone models, and adopt micro-averaged precision (P), recall (R), and F1 score as evaluation metrics. We use bert-base-uncased[9] as the encoder for extracting semantic features during the demonstration retrieval stage. For syntactic feature extraction, we use the spaCy

[8]To ensure fairness and reduce computational overhead, we modify the original inference procedure by sampling 10 candidate demonstration chains and selecting the one that achieves the best performance on the development set.

[9]https://huggingface.co/google-bert/bert-base-uncased

parser with the `en_core_web_sm` model. To improve reproducibility, we set the decoding temperature to 0 to disable sampling and perform greedy decoding. All hyperparameters λ_i in MVSR are set to 0.25. All experiments are conducted on a single NVIDIA RTX 4090 GPU using CUDA 11.8 and PyTorch 2.3.0.

D Effect of Structural Similarity Algorithms

In Section 3.2, we describe how MVSR leverages serialized syntactic information to compute structural similarity. Although tree-based similarity metrics are generally considered more faithful to hierarchical sentence structure, our empirical results indicate that employing more fine-grained structural matching algorithms can, in practice, degrade performance in the ICL setting.

Specifically, we replace the sequence-based structural similarity with the tree edit distance (TED) metric proposed by Zhang and Shasha (1989), while keeping all other components unchanged. As shown in Table 5, this modification results in consistently lower performance and increased variability across datasets.

We attribute the inferior performance of the tree-based approach to two main factors. First, sentences with highly similar full syntactic trees are relatively rare in real-world data. Consequently, tree-based similarity metrics often assign uniformly low similarity scores, limiting the contribution of structural information during demonstration retrieval. Second, tree edit distance is highly sensitive to mismatches near the root of the tree. Even when large subtrees share substantial structural overlap, such root-level differences can incur disproportionately large penalties, thereby obscuring meaningful structural similarities (Bille, 2005; Pawlik and Augsten, 2015). Together, these effects impede the retrieval of structurally relevant demonstrations and ultimately diminish ICL performance.

E Analysis

E.1 Effect of the Number of Demonstrations

To evaluate the robustness of MVSR, we examine its performance under varying numbers of demonstrations using GPT-4o and GPT-4o-mini as backbone models. As shown in Table 6, MVSR generally benefits from increasing the number of demonstrations. However, when GPT-4o-mini is used, performance gains begin to plateau once the number of demonstrations exceeds 20. On the 14RES and 15RES datasets, adding additional demonstrations even leads to performance degradation, suggesting that the model's capacity may become saturated beyond this point. In contrast, GPT-4o continues to benefit from additional demonstrations, indicating a stronger ability to model longer contexts and induce consistent extraction behavior.

Table 6 further shows that GPT-4o exhibits a higher rate of parsing failures than GPT-4o-mini, particularly as the number of demonstrations increases. Manual inspection reveals that, in most failure cases, GPT-4o generates correct sentiment triplets but prepends them with explanatory text, thereby violating the required output format, while true structural extraction errors remain relatively rare. This behavior may be attributed to GPT-4o's training objective, which places greater emphasis on conversational and role-playing capabilities. As a result, even under explicit formatting constraints, GPT-4o occasionally produces dialog-style responses.

E.2 Effect of Demonstration Order

We examine the effect of demonstration ordering on model performance. As shown in Table 7, ORIGINAL denotes the descending similarity order used by MVSR, REVERSED corresponds to the ascending order, and SHUFFLE represents a randomly permuted sequence of demonstrations. The results indicate that both reversing and shuffling the demonstration order lead to consistent performance degradation.

To further analyze this order sensitivity, we examine the attention distribution of a causal decoder model. Since commercial LLMs such as GPT do not expose their internal parameters, we conduct this analysis using LLaMA-3-8B-Instruct[10], an open-source model with a comparable autoregressive architecture.

Figure 5 visualizes the attention distribution for a representative test example. For clarity, we segment the prompt into the start token, task description, individual demonstrations, and the final test sentence, and compute sentence-level attention scores accordingly. The visualization shows that each demonstration primarily attends to the start token, itself, and the immediately preceding demonstration, indicating a strong sequential dependency among demonstrations.

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

Shot	Method	14LAP			14RES			15RES			16RES		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
5-shot	MVSR	42.36	51.20	**46.36**	59.73	68.58	**63.85**	51.24	65.29	**57.42**	57.28	72.18	**63.87**
	MVSR-TED	41.01	48.06	44.26	58.02	66.20	61.84	49.59	62.68	55.37	52.58	67.51	59.11
10-shot	MVSR	44.00	52.62	**47.92**	62.06	71.96	**66.65**	52.43	66.05	**58.45**	59.19	74.12	**65.82**
	MVSR-TED	43.87	51.57	47.41	60.14	68.01	63.83	51.06	64.33	56.93	56.79	71.60	63.34

Table 5: Performance of MVSR with different structural similarity algorithms (GPT-4o-mini).

BackBone	Shot	14LAP		14RES		15RES		16RES	
		F1	Error	F1	Error	F1	Error	F1	Error
GPT-4o-mini	0-Shot	35.46	0	54.78	1	48.60	2	54.64	0
	1-Shot	42.55 ↑7.09	0	60.66 ↑5.88	1	54.56 ↑5.96	0	58.71 ↑4.07	0
	3-Shot	44.04 ↑1.49	0	64.02 ↑3.36	0	55.32 ↑0.76	0	63.27 ↑4.56	0
	5-Shot	46.36 ↑2.32	0	63.85 ↓0.17	1	57.42 ↑2.10	0	63.87 ↑0.60	0
	10-Shot	47.92 ↑1.56	0	**66.65** ↑2.63	1	**58.45** ↑1.03	0	65.82 ↑1.95	0
	20-Shot	50.16 ↑2.24	0	66.28 ↓0.37	1	58.03 ↓0.42	0	66.78 ↑0.96	0
	32-Shot	**50.17** ↑0.01	0	66.15 ↓0.50	1	58.30 ↓0.15	0	**67.52** ↑0.74	0
GPT-4o	0-Shot	38.04	5	50.05	12	49.07	4	55.35	2
	1-Shot	43.60 ↑4.56	0	62.11 ↑12.06	0	56.07 ↑7.00	0	64.00 ↑8.65	0
	3-Shot	46.29 ↑2.69	4	63.15 ↑1.04	3	57.33 ↑1.26	3	65.79 ↑1.79	1
	5-Shot	49.65 ↑3.36	11	65.74 ↑2.59	12	60.93 ↑3.60	3	66.19 ↑0.40	10
	10-Shot	51.83 ↑2.18	37	65.81 ↑0.07	40	60.96 ↑0.03	23	66.18 ↓0.01	24
	20-Shot	52.95 ↑1.12	22	68.30 ↑2.49	23	62.60 ↑1.64	15	68.44 ↑2.25	11
	32-Shot	**55.17** ↑2.22	29	**68.69** ↑0.39	28	**63.02** ↑0.42	26	**70.52** ↑2.08	16

Table 6: Effect of the number of demonstrations on model performance. ↑ and ↓ indicate improvements or degradations relative to the best-performing configuration in the previous rows. Error represents the number of model outputs that could not be parsed due to formatting issues.

Under the descending similarity order adopted by MVSR, more relevant demonstrations appear earlier in the prompt, enabling subsequent demonstrations to be conditioned on higher-quality contextual information. In contrast, reversing or shuffling the order may place less relevant or noisy demonstrations at the beginning of the prompt, which can negatively affect the encoding of later, more informative examples. This observation helps explain why the REVERSED and SHUFFLE variants consistently underperform the ORIGINAL ordering in Table 7.

E.3 Analysis of Structural Similarity in Datasets

In Section 4.2.1, we observe that on the 14LAP dataset, incorporating structural information leads to inferior performance of MVSR compared to strong similarity-based baselines such as BM25 and BERTScore. To better understand the underlying cause of this behavior, we conduct a quanti-

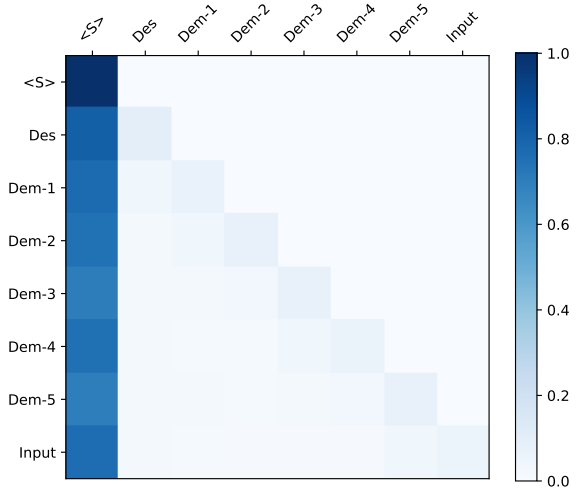

Figure 5: Attention distribution of the causal decoder model LLaMA. <S> denotes the start token, Des represents the task description, Dem-i refers to the i-th demonstration, and Input indicates the test text.

Shot	Order	14LAP			14RES			15RES			16RES		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
5-Shot	ORIGINAL	42.36	51.20	**46.36**	59.73	68.58	**63.85**	51.24	65.29	**57.42**	57.28	72.18	**63.87**
	REVERSED	41.32	49.72	45.13	59.45	68.86	63.81	50.00	63.51	55.95	56.07	71.76	62.93
	SHFFULE	42.26	50.19	45.88	59.31	67.64	63.21	50.77	64.74	56.91	56.53	71.86	63.27

Table 7: Effect of demonstration order on model performance (GPT-4o-mini).

Dataset	POS	DEP	POS&DEP
14LAP	0.5594	0.5235	1.0441
14RES	0.6183	0.5706	1.1566
15RES	0.5877	0.5487	1.1043
16RES	0.6254	0.5922	1.1828

Table 8: Sentence structural similarity between training and test sets measured by POS-based, dependency-based, and combined representations on four ASTE datasets.

tative analysis of structural similarity between the training and test sets across different datasets.

Specifically, for each test instance, we compute its structural similarity with all training instances and select the top five most similar examples. The average similarity score of these top-ranked instances is used as the structural similarity measure for the test instance. We then average this measure over all test instances to obtain a dataset-level structural similarity score.

We report three types of structural similarity scores in Table 8. POS denotes the POS-based similarity, DEP represents the dependency-based similarity, and POS&DEP corresponds to their combination. As shown in the table, the 14LAP dataset consistently exhibits substantially lower structural similarity scores than the other datasets, indicating a larger structural divergence between its training and test sets. Under such conditions, the syntactic information introduced by MVSR becomes less reliable for demonstration retrieval, which may partially account for its reduced performance on this dataset.

Furthermore, based on our dataset-level analysis of structural similarity between the training and test sets, we suggest that domain-level structural divergence may contribute to the generally lower performance observed on 14LAP—not only for ICL-based approaches but also for fully supervised fine-tuning methods—relative to the other three datasets (see Section 4.2).

To provide a more intuitive illustration of the impact of structural similarity, we randomly sample two test instances from each of the 14LAP and 15RES datasets and identify, for each test instance, the training examples with the highest structural similarity scores. As shown in Figure 6, test instances from 15RES exhibit higher structural similarity to their corresponding demonstrations, with more closely aligned sentence structures and sentiment triplet structures. In contrast, the demonstrations retrieved for 14LAP display lower structural similarity, resulting in weaker structural alignment and less effective guidance during ICL.

Dataset	ID	Source	Input	Label	Score
14LAP	1	Test	Boot time is super fast, around anywhere from 35 seconds to 1 minute.	(Boot time, fast, positive)	POS: 0.4988 DEP: 0.4297
		Train	Navigation through the computer is far superior compared to Windows operating systems, as well.	(Navigation, superior, positive)	
	2	Test	Did not enjoy the new Windows 8 and touchscreen functions.	(Windows 8, not enjoy, negative) (touchscreen functions, not enjoy, negative)	POS: 0.4784 DEP: 0.5201
		Train	Oh yeah, don't forget the expensive shipping to and from HP.	(shipping, expensive, negative)	
14RES	3	Test	I have to say they have one of the fastest delivery times in the city.	(delivery times, fastest, positive)	POS: 0.7060 DEP: 0.6660
		Train	This is one of the best comfort food places in the city.	(comfort food, best, positive)	
	4	Test	The food was extremely tasty, creatively presented and the wine excellent.	(food, tasty, positive) (food, creatively presented, positive) (wine, excellent, positive)	POS: 0.6939 DEP: 0.5929
		Train	The staff was very attentive, the ambience lovely, and the food superb.	(staff, attentive, positive) (ambience, lovely, positive) (food, superb, positive)	

Figure 6: Four example pairs from the 14LAP and 14RES datasets. Test denotes instances from the test set, Train denotes instances from the training set, and score indicates the structural similarity between each pair.