DS^2-ABSA: Dual-Stream Data Synthesis with Label Refinement for Few-Shot Aspect-Based Sentiment Analysis

Hongling Xu[1,3], **Yice Zhang**[1,3], **Qianlong Wang**[1,3], **Ruifeng Xu**[1,2,3*]
[1] Harbin Institute of Technology, Shenzhen, China
[2] Peng Cheng Laboratory, China
[3] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
xuhongling@stu.hit.edu.cn, xuruifeng@hit.edu.cn

Abstract

Recently developed large language models (LLMs) have presented promising new avenues to address data scarcity in low-resource scenarios. In few-shot aspect-based sentiment analysis (ABSA), previous efforts have explored data augmentation techniques, which prompt LLMs to generate new samples by modifying existing ones. However, these methods fail to produce adequately diverse data, impairing their effectiveness. Additionally, some studies apply in-context learning for ABSA by using specific instructions and a few selected examples as prompts. Though promising, LLMs often yield labels that deviate from task requirements. To overcome these limitations, we propose DS^2-ABSA, a dual-stream data synthesis framework targeted for few-shot ABSA. It leverages LLMs to synthesize data from two complementary perspectives: *key-point-driven* and *instance-driven*, which effectively generate diverse and high-quality ABSA samples in low-resource settings. Furthermore, a *label refinement* module is integrated to improve the synthetic labels. Extensive experiments demonstrate that DS^2-ABSA significantly outperforms previous few-shot ABSA solutions and other LLM-oriented data generation methods. Our code and synthetic data are available at https://github.com/HITSZ-HLT/DS2-ABSA.

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to identify aspect terms and determine their sentiments within user reviews (Pontiki et al., 2014). For example, given the review "*the battery life is great, but the screen resolution is disappointing*," the output of End-to-End ABSA (E2E-ABSA) would be {(*battery life*, positive), (*screen resolution*, negative)}. Previous studies have proposed various deep learning methods (Fei et al., 2022; Tian et al., 2023; Scaria et al., 2024; Zheng et al., 2024), demonstrating strong performance when trained on extensive manually labeled data. However, annotating sufficient data is extremely time-consuming and labor-intensive in practice, prompting researchers to explore ABSA approaches in low-resource scenarios (Varia et al., 2023; Wang et al., 2023b, 2024d; Zhang et al., 2024a).

Existing low-resource solutions comprise three main types. The first, **data augmentation**, produces additional samples by modifying existing ones, which are typically implemented through masked language modeling (Li et al., 2020; Zhou et al., 2022) or large language models (LLMs) (Dai et al., 2023; Peng et al., 2024). Although these methods can yield a large number of new samples, the diversity of them remains limited, thereby providing marginal benefit for subsequent model training. The second type, **in-context learning**, aligns LLMs with the ABSA task through task-specific instructions and a few examples (Zhang et al., 2024a; Wang et al., 2024c,d). However, even with well-crafted instructions and carefully chosen demonstrations, LLMs tend to deviate from task-specific requirements, frequently generating reasonable, yet incorrect results. Apart from these two types, some researchers explore **pre-training** techniques (Wang et al., 2023b; Zhang et al., 2023) to reduce reliance on downstream datasets. Nonetheless, these techniques require vast additional corpora and incur high training costs.

Inspired by the recent advances in data synthesis (Li et al., 2023; Wang et al., 2023a), we propose **DS^2-ABSA**, a dual-stream data synthesis framework for few-shot ABSA. Unlike existing methods, our study employs multi-granularity-guided synthesis targeted to E2E-ABSA. Specifically, we first leverage LLMs to generate data via two distinct strategies: *key-point-driven* and *instance-driven*. The former engages LLMs to brainstorm a variety of potential ABSA attributes, which are then composed to create new samples. The latter trans-

* Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15460–15478

forms existing samples through operations including sample combination and selective reconstruction. These two strategies are complementary: the former synthesizes data that covers a broader range of review scenarios and offers greater diversity, while the latter generates data based on existing samples and provides better relevance and quality. Moreover, we integrate a *label refinement* module to enhance the quality of labels in the synthetic data. This module applies a label normalization process alongside a noisy self-training algorithm that employs a few gold samples to guide the re-estimation of the synthetic labels.

Compared to previous methods, our approach offers the following advantages. Firstly, by leveraging key-point-driven and instance-driven strategies, it can generate more diverse data than data augmentation methods, providing greater potential benefits for subsequent model training. Secondly, in contrast to in-context learning methods, our approach introduces a novel way of utilizing LLMs, resulting in better task-specific alignment, and the label refinement module further enhances this advantage. Thirdly, unlike pre-training methods, our approach requires no additional corpus, entailing significantly lower training costs and avoiding the potential challenges of data acquisition.

Our contributions are summarized as follows:

(1) We propose a dual-stream synthesis framework that leverages LLMs to generate ABSA samples from two distinct perspectives. To the best of our knowledge, this is the first exploration of data synthesis for E2E-ABSA.

(2) We develop a label refinement module that effectively enhances the label quality of the synthesized data through label normalization and noisy self-training.

(3) Experimental results on four public datasets demonstrate that DS^2-ABSA significantly outperforms existing low-resource ABSA solutions, as well as other LLM-based data augmentation and synthesis methods.

2 Related Work

2.1 Few-shot ABSA

Current state-of-the-art E2E-ABSA methods primarily rely on fine-tuning text-to-text language models (Zhang et al., 2021; Yu et al., 2023; Scaria et al., 2024; Zheng et al., 2024) or incorporating syntactic knowledge (Fei et al., 2022; Tian et al., 2023; Liu et al., 2024) using sufficient labeled data. However, annotating fine-grained sentiments in adequate reviews is expensive. To mitigate data dependency, early methods mainly adopt data augmentation (Li et al., 2020; Ding et al., 2020; Hsu et al., 2021; Zhou et al., 2022) and pre-training (Xu et al., 2019; Zhou et al., 2020; Liu et al., 2023; Wang et al., 2023b; Zhang et al., 2023).

Recently, the advent of powerful LLMs has inspired new approaches for few-shot ABSA, including (1) LLM-based data augmentation (Dai et al., 2023; Peng et al., 2024), which often lacks diversity in low-resource settings; (2) in-context learning (Zhang et al., 2024a; Wang et al., 2024b,c,d; Zhu et al., 2024), where LLMs often fail to produce task-aligned outputs; and (3) knowledge distillation (Zhou et al., 2024; Zhang et al., 2024b), which similarly requires substantial extra review corpora like pre-training, posing challenges in domains with limited data availability and privacy concerns. To tackle these issues, we propose a novel approach that utilizes LLMs to synthesize samples with rich diversity and high quality, whose labels are then refined to facilitate downstream training.

2.2 Data Synthesis

Data synthesis is a pivotal strategy to address challenges like high annotation costs and privacy issues (Long et al., 2024). With the advent of LLMs, it has been applied to various fields, such as text classification (Ye et al., 2022; Li et al., 2023), instruction tuning (Wang et al., 2023a; Zhao et al., 2024), and mathematical reasoning (Huang et al., 2024; Yu et al., 2024; Chan et al., 2024).

Existing work can be grouped into three types: (1) instance-driven synthesis (Zhao et al., 2024; Yu et al., 2024), which leverages examples to guide LLMS in synthesizing relevant data; (2) key-point-driven synthesis (Huang et al., 2024; Wang et al., 2024a), utilizing conditional prompts to produce data satisfying specific attributes; and (3) knowledge-driven synthesis (Xu et al., 2024; Chan et al., 2024), incorporating external knowledge to steer the synthesis. In practice, these strategies are often combined to achieve optimal results. After generation, techniques such as data filtering (Wang et al., 2023a) are used for data curation. On this basis, our study pioneers the data synthesis for E2E-ABSA by designing key-point-driven and instance-driven strategies, along with a refinement module for label re-estimation.

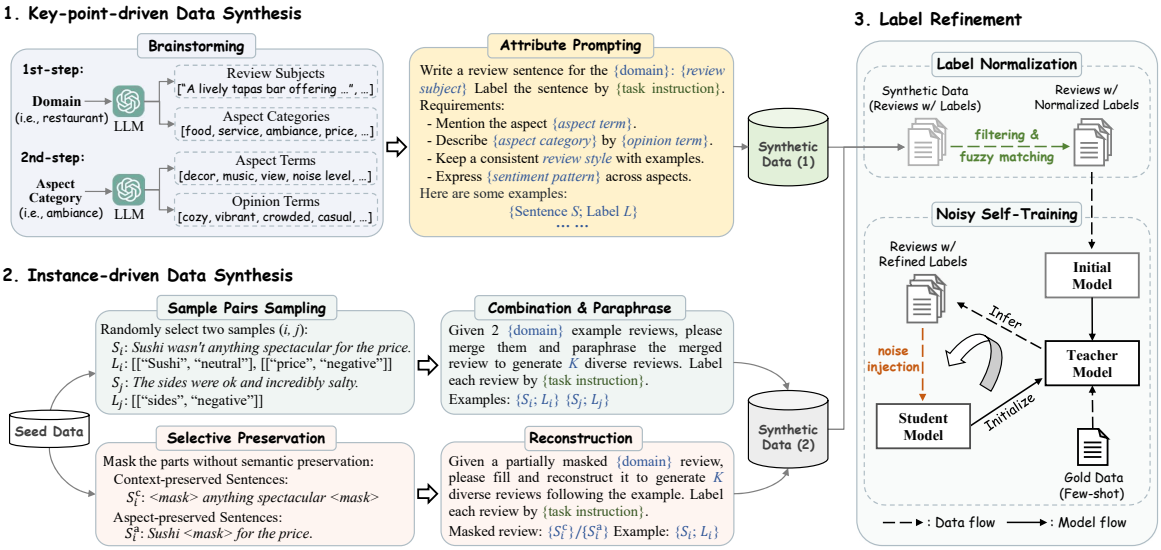


Figure 1: Overview of the proposed DS^2-ABSA. The process begins with parallel dual-stream data synthesis: the key-point-driven stream leverages LLMs to brainstorm a set of critical ABSA attributes for conditional generation, while the instance-driven stream applies a small seed dataset to perform multi-level transformations. The resulting data are then combined and processed through normalization and self-training for noise handling.

3 Method

As depicted in Figure 1, we propose a novel data synthesis framework to improve the few-shot E2E-ABSA. Here, we detail our DS^2-ABSA pipeline, where *key-point-driven* and *instance-driven* data synthesis generate complementary ABSA samples from different perspectives. These synthetic data are merged and fed into the label refinement module to enhance label quality.

3.1 Key-point-driven Data Synthesis

This module aims to generate reviews based on key points, hereafter referred to as attributes. To this end, we define several critical attributes, such as aspect and opinion terms, and guide LLMs to *brainstorm* numerous candidates for each attribute. Afterward, we sample a set of attributes from these candidates and employ LLMs to generate reviews based on these attributes using *attribute prompting*.

Brainstorming. Building on Zhang et al. (2022), we define the four core attributes that form a review sample: (a) *review subject*, a general description of the restaurant or product, such as "*a lively tapas bar offering ...*"; (b) *aspect category*, indicating the generalized dimension being evaluated, such as '*ambiance*' and '*service*'; (c) *aspect term*, referring to the specific opinion target explicitly mentioned in the review, such as '*decor*' and '*noise*'; and (d) *opinion term*, representing the descriptive expression conveying sentiment to the opinion target, such as '*charming*' and '*cozy*'.

Next, we implement a coarse-to-fine generation strategy to produce a range of potential values for each attribute. Firstly, given a specific domain (such as restaurants or laptops), we prompt LLMs to brainstorm and generate representative review subjects and aspect categories. Secondly, for each aspect category generated, we guide LLMs to generate a diverse array of aspects and opinion terms. We then collect the values for each attribute, forming the corresponding candidate pools. Following Wang et al. (2024a), we employ GPT-4 (OpenAI, 2023) for brainstorming to ensure both the quantity and quality of the attribute candidates.

Attribute Prompting. The attribute generation consists of three steps. Firstly, we randomly sample a set of attributes from the brainstormed candidate pools, denoted as (rs_i, ac_i, at_i, ot_i). Secondly, we instruct LLMs to generate a review that concerns the review subject rs_i and includes the aspect category ac_i, aspect term at_i, and opinion term ot_i. It is important to note that we do not sample multiple sets of attributes and combine them to generate reviews, as this method could lead to potential conflicts among different attribute sets and reduce the coherence of the generated reviews. Finally, we require LLMs to generate the corresponding ABSA labels based on the provided attributes and the generated reviews.

Our observations indicate that, despite utilizing

diverse attributes, the generated reviews tend to exhibit a uniform style and express sentiments too simplistically, thereby deviating from the true data distribution. To address these issues, we introduce two control attributes in our prompts: *review style* and *sentiment pattern*. The review style attribute involves using a few real reviews as exemplars to guide LLMs in generating reviews that mimic a similar style. The sentiment pattern attribute dictates the method of expressing sentiments, with options including 'consistent,' 'mixed,' and 'implicit.' This allows for control over how reviews express sentiments—whether consistently across different aspects, in a varied manner, or implicitly, where sentiments are conveyed indirectly through context rather than explicit opinion words. The full prompt is presented in Appendix A.

3.2 Instance-driven Data Synthesis

This module synthesizes new data by transforming existing data, differing from the key-point-driven module that synthesizes data from scratch. The main advantage of this approach is that the synthesized data maintains strong in-domain relevance with the reference samples. More importantly, during the data synthesis process, LLMs can access the labels of the reference samples, thereby ensuring a higher quality of the synthesized labels. Specifically, we employ two operations to facilitate this transformation: *sample combination* and *selective reconstruction*.

Sample Combination. This operation randomly selects two samples from the seed data and instructs LLMs to merge them, thereby creating a new sample. Such a combination can effectively increase sample diversity, as it simulates a broader range of review scenarios. However, a potential issue with this approach is that this may lead to semantic discontinuities and content conflicts. To address this, we additionally require LLMs to paraphrase the merged samples, aiming to produce more coherent and consistent samples.

Selective Reconstruction. This operation is inspired by existing data augmentation methods (Wei and Zou, 2019; Li et al., 2020; Hsu et al., 2021). It begins by preserving a portion of segments in a given review and then directs LLMs to reconstruct the complete review. We develop two selective preservation strategies: *context preservation* and *aspect preservation*. Context preservation masks the aspect terms and their surrounding m words. Aspect preservation randomly masks segments of the given review except for the aspect terms, with a total masking portion of p_{mask}. Subsequently, these masked reviews are input into LLMs, tasked with generating complete reviews. Compared to traditional data augmentation methods, the advantage of this operation is that it preserves fewer review segments and leverages the capabilities of LLMs to generate more diverse samples.

3.3 Label Refinement

The ABSA data synthesized by LLMs inevitably contain inaccurate labels due to misalignment with task requirements (Wang et al., 2024d). To reduce their impact, our study introduces a novel label re-estimation method that rectifies erroneous labels by *label normalization* and *noisy self-training*.

Label Normalization. For initial refinement, we introduce a rule-based approach to normalize the synthetic labels. In the task-specific requirements of ABSA, aspect terms must appear explicitly as complete sub-sequences within the text. Based on this requirement, we compare the extracted aspects in synthetic labels with their corresponding sentences in the synthetic data, removing any that do not appear as matches. We then apply fuzzy matching for matched but incomplete aspects, substituting them with n-grams that minimize the Levenshtein distance per unit length.

Noisy Self-training. We implement the noisy self-training algorithm (Xie et al., 2020; Liu et al., 2021; Jiang et al., 2023) to re-estimate the synthetic labels using a few gold data. The process begins by training an initial model on normalized data, followed by fine-tuning with gold data to obtain the teacher model $\mathcal{T}_0$, which is expected to better align with task requirements.

Next, we iterate through the noisy student training process. In the i-th iteration, we first use the previous teacher model $\mathcal{T}_{i-1}$ to label the synthetic sentences, refining errors like aspect boundary inaccuracies and sentiment misinterpretations. A new student model $\mathcal{S}_i$ is then trained on the refined data, with noise injected by randomly deleting or masking tokens in 50% of samples at a disturbance probability p_{noise} to improve robustness. $\mathcal{S}_i$ is subsequently fine-tuned on a few gold data to produce the updated teacher model $\mathcal{T}_i$. The iterative process repeats until the validation performance stabilizes, after which the final teacher model serves as the ABSA model for evaluation.

4 Experiments

4.1 Settings

Datasets. We evaluate the proposed method on four ABSA benchmark datasets, including Lap14 and Res14 from Pontiki et al. (2014), Res15 from Pontiki et al. (2015), and Res16 from Pontiki et al. (2016). These datasets cover two domains: *restaurant* and *laptop*. The data statistics are presented in Table 1, where we randomly split 20% of the training set for validation.

Dataset		Samples	Aspects	#Pos	#Neu	#Neg	#Con
Lap14	train	2,436	1,922	808	387	691	36
	dev	609	436	179	73	175	9
	test	800	654	341	169	128	16
Res14	train	2,432	2,972	1,774	509	621	68
	dev	609	721	390	124	184	23
	test	800	1,134	728	196	196	14
Res15	train	1,052	956	721	29	199	7
	dev	263	243	181	5	53	4
	test	685	542	319	27	179	17
Res16	train	1,600	1,363	952	51	337	23
	dev	400	380	268	10	96	6
	test	676	612	460	28	113	11

Table 1: Statistics of the four ABSA datasets. #Pos, #Neu, #Neg, and #Con represent positive, neutral, negative, and conflict aspects, respectively.

Implementation Details. In experiments, we adopt two few-shot settings: **2%-shot** and **5%-shot**, wherein a corresponding proportion of training data is randomly sampled to simulate low-resource scenarios. Unless specified otherwise, we employ **GPT-3.5 Turbo** (Ouyang et al., 2022) for data generation (the specific version is *gpt-3.5-turbo-0125*). Training samples with explicit aspects are selected as seed data. In key-point-driven synthesis, we generate 20,000 samples using randomly combined attributes and 4 examples in prompts. In instance-driven synthesis, we limit the maximum combined samples to 1,000, set the aspect masking window m to 0 and 2, and the context masking probability p_{mask} to 0.6 with random masking twice. Besides, the number of generated samples in a single response K is 4. See Appendix A for the detailed prompts applied in the dual-stream data synthesis. During the label refinement process, p_{noise} is set to 0.1, the maximum number of iterations is 3, and additional hyperparameters can be found in Appendix B.1. All experiments are conducted on NVIDIA A6000 GPUs. We run fine-tuning experiments with three random seeds and report the average **F1 score**.

4.2 Baselines

To validate the effectiveness of our method, we first select three representative ABSA models for training: (1) **TAG-BERT** (Hu et al., 2019), leveraging *bert-base-uncased* followed by a CRF layer for BIO tagging; (2) **Paraphrase** (Zhang et al., 2021), converting ABSA into an "{aspect} is {sentiment}" paraphrase generation task using *T5-base*; and (3) **InstructABSA** (Scaria et al., 2024), performing instruction tuning with 2 positive examples, for which we apply *Tk-instruct-large* (Wang et al., 2022). We then conduct extensive comparisons with two categories of approaches, as listed below. **Detailed descriptions of these methods and more baselines are available in Appendix B.2.**

Low-resource Enhancement Methods. These techniques improve few-shot ABSA by incorporating additional data or knowledge and are model-agnostic. The methods include: (1) data augmentation: *MELM* (Zhou et al., 2022), *AugGPT* (Dai et al., 2023), *CoTAM* (Peng et al., 2024); (2) pre-training: *BERT-PT* (Xu et al., 2019), *BERT-SPT* (Zhang et al., 2023), *FS-ABSA* (Wang et al., 2023b); (3) distillation: *UniNER* (Zhou et al., 2024); and (4) data synthesis: *ZeroGen* (Ye et al., 2022), *Self-Instruct* (Wang et al., 2023a). Comparisons with them highlight the advantages of our data synthesis framework for few-shot ABSA.

LLM-based ABSA methods. These methods directly utilize LLMs to extract aspect-sentiment pairs, including *Zero-shot Prompting* (Zhang et al., 2024a), *In-context Learning* (Wang et al., 2024c), and *Supervised Fine-tuning* (Simmering and Huoviala, 2023). In addition to GPT-3.5 Turbo, we also evaluate GPT-4 (OpenAI, 2023) (the specific version is *gpt-4-0125-preview*) and Llama-3-8B-Instruct (Dubey et al., 2024). Comparisons with these approaches demonstrate the potential performance gains of leveraging LLM-synthesized data to train specialized ABSA models compared to direct prompting or fine-tuning.

4.3 Main Results

Comparison with Low-resource Enhancement Methods. The results in Table 2 indicate that DS^2-ABSA consistently outperforms existing few-shot solutions and other LLM-oriented data generation methods. For instance, the average F1 of DS^2-ABSA on TAG-BERT exceeds the second-best Self-Instruct by 5.69% and 2.99% in the 2%-

ABSA Model	Method	2%-shot					5%-shot				
		Lap14	Res14	Res15	Res16	Avg(△)	Lap14	Res14	Res15	Res16	Avg(△)
TAG-BERT (Hu et al., 2019)	Origin	15.83	37.49	23.04	20.19	24.14	35.29	51.64	34.52	43.48	41.23
	MELM†	38.27	46.26	32.11	34.90	37.89$_{+13.75}$	42.86	57.39	39.76	51.04	47.76$_{+6.53}$
	AugGPT†	35.29	48.92	28.19	39.69	38.02$_{+13.88}$	37.26	57.93	37.88	53.26	46.58$_{+5.35}$
	CoTAM†	39.15	56.05	31.06	42.36	42.16$_{+18.02}$	45.21	59.07	43.99	54.18	50.61$_{+9.38}$
	BERT-PT‡	40.66	49.39	35.42	46.67	43.04$_{+18.90}$	47.95	61.92	35.75	54.19	49.95$_{+8.72}$
	SPT-ABSA‡	35.92	47.56	31.64	42.74	39.47$_{+15.33}$	46.71	**63.58**	40.42	55.17	51.47$_{+10.24}$
	ZeroGen*	41.84	55.33	41.25	48.86	46.82$_{+22.68}$	45.87	56.64	41.92	53.77	49.55$_{+8.32}$
	Self-Instruct*	41.85	56.94	42.34	52.71	48.46$_{+24.32}$	46.13	59.54	44.57	55.80	51.51$_{+10.28}$
	DS^2-ABSA*	**47.30**	**60.39**	**49.49**	**59.40**	**54.15$_{+30.01}$**	**47.97**	62.37	**49.26**	**58.40**	**54.50$_{+13.27}$**
PARAPHRASE (Zhang et al., 2021)	Origin	47.64	53.40	39.23	39.75	45.01	51.51	62.01	51.45	52.58	54.39
	MELM†	48.57	57.11	41.84	48.74	49.07$_{+4.06}$	52.85	63.42	48.01	60.20	56.12$_{+1.73}$
	AugGPT†	46.81	57.80	49.18	47.28	50.27$_{+5.26}$	48.74	62.97	52.71	59.13	55.89$_{+1.50}$
	CoTAM†	47.28	58.71	46.75	53.15	51.47$_{+6.46}$	52.55	62.37	52.06	59.37	56.59$_{+2.20}$
	FS-ABSA‡	49.66	57.65	47.15	50.70	51.29$_{+6.28}$	52.02	63.12	53.30	61.52	57.49$_{+3.10}$
	UniNER‡	53.92	64.02	**53.76**	58.36	57.52$_{+12.51}$	54.75	67.34	**55.41**	60.04	59.39$_{+5.00}$
	ZeroGen*	48.85	57.71	49.41	55.10	52.77$_{+7.76}$	51.92	63.18	53.58	59.19	56.97$_{+2.58}$
	Self-Instruct*	50.55	61.02	51.27	56.56	54.85$_{+9.84}$	53.67	64.89	51.82	60.36	57.69$_{+3.30}$
	DS^2-ABSA*	**56.86**	**64.92**	53.15	**61.16**	**59.02$_{+14.01}$**	**60.83**	**68.41**	54.32	**61.87**	**61.36$_{+6.97}$**
INSTRUCTABSA (Scaria et al., 2024)	Origin	52.05	63.36	53.67	58.78	56.97	57.54	67.59	55.08	62.91	60.78
	MELM†	56.36	64.24	52.22	59.81	58.16$_{+1.19}$	58.53	68.15	54.31	63.44	61.11$_{+0.33}$
	AugGPT†	53.40	63.26	54.26	61.27	58.05$_{+1.08}$	55.11	66.54	57.25	64.93	60.96$_{+0.18}$
	CoTAM†	54.07	64.77	51.42	62.45	58.18$_{+1.21}$	56.41	67.35	55.48	63.84	60.77$_{-0.01}$
	FS-ABSA‡	56.59	63.46	55.88	61.55	59.37$_{+2.40}$	61.63	68.68	55.83	66.24	63.10$_{+2.32}$
	UniNER‡	56.95	68.90	56.82	62.68	61.34$_{+4.37}$	59.81	70.33	58.34	67.95	64.11$_{+3.33}$
	ZeroGen*	54.72	64.22	53.85	62.56	58.84$_{+1.87}$	56.93	66.61	56.53	64.06	61.03$_{+0.25}$
	Self-Instruct*	56.78	66.98	56.15	61.84	60.44$_{+3.47}$	**61.89**	68.84	57.39	67.07	63.80$_{+3.02}$
	DS^2-ABSA*	**58.15**	**69.65**	**59.78**	**63.89**	**62.87$_{+5.90}$**	61.24	**71.94**	**60.56**	**68.81**	**65.64$_{+4.86}$**

Table 2: Main results compared with low-resource enhancement approaches (more baseline results are presented in Table 11). Data augmentation and data synthesis methods are marked with † and *, respectively. Pre-training and distillation methods, marked with ‡, rely on large amounts of additional unsupervised data.

and 5%-shot settings, respectively. These findings underscore the superiority of our approach in low-resource scenarios, which effectively synthesize higher-quality ABSA samples. Additionally, several further observations are listed below:

(1) Original ABSA models fall short with small gold data, as exemplified by TAG-BERT achieving only 24.14% average F1 on 2%-shot data. Both data augmentation and synthesis methods yield considerable improvements in most situations by expanding training samples, with the latter generally surpassing the former due to the limited diversity of augmented data, particularly in 2%-shot scenarios.

(2) Pre-training and distillation methods achieve competitive performance by leveraging large-scale external corpora that provide domain and sentiment knowledge. However, these approaches are often impractical for new domains or under privacy constraints that restrict access to such corpora. In contrast, DS^2-ABSA synthesizes and refines data using only small seed data, surpassing FS-ABSA and UniNER without relying on additional corpora or costly training, demonstrating both its significance and domain generalizability.

Comparison with LLM-based ABSA Methods. As presented in Table 3, DS^2-ABSA outperforms LLM-based methods on nearly all datasets, confirming its superiority in low-resource settings. First, it surpasses in-context learning methods by a large margin, indicating the difficulty of aligning ABSA task requirements through direct prompting, even with GPT-4. Second, the proposed framework, employing either GPT-3.5 Turbo or Llama-3-8B-Instruct for data synthesis, significantly outperforms their supervised fine-tuning counterparts with lower hardware requirements. For example, in the 5%-shot setting, DS^2-ABSA(INST) using these two LLMs exceeds their fine-tuning performance by 1.81% and 2.99%, respectively. We attribute this to LLMs retaining robust general capabilities even after fine-tuning. In comparison, DS^2-ABSA models are specifically optimized for

Backbone LLM	Method	2%-shot					5%-shot				
		Lap14	Res14	Res15	Res16	Avg	Lap14	Res14	Res15	Res16	Avg
GPT-4	Zero-shot Prompting	41.35	60.61	49.42	53.25	51.16	41.35	60.61	49.42	53.25	51.16
	In-context Learning	45.24	65.10	56.07	57.78	56.05	46.84	66.09	56.35	59.08	57.09
GPT-3.5 Turbo	In-context Learning	38.69	60.35	48.36	54.45	50.46	40.02	60.60	46.50	56.28	50.85
	Supervised Fine-tuning	53.09	65.17	56.60	**65.96**	60.21	55.47	**72.30**	58.83	68.73	63.83
	DS^2-ABSA(PARA)*	56.86	64.92	53.15	61.16	59.02	60.83	68.41	54.32	61.87	61.36
	DS^2-ABSA(INST)*	**58.15**	**69.65**	**59.78**	63.89	**62.87**	**61.24**	71.94	**60.56**	**68.81**	**65.64**
Llama-3-8B-Instruct	In-context Learning	41.62	59.84	46.50	55.01	50.74	40.70	61.07	47.13	56.22	51.28
	Supervised Fine-tuning	52.52	63.83	55.65	61.98	58.50	56.47	68.85	59.64	68.11	63.27
	DS^2-ABSA(PARA)*	57.29	65.73	52.68	60.30	59.00	60.57	68.83	54.62	62.44	61.62
	DS^2-ABSA(INST)*	**60.59**	**69.94**	**58.34**	**62.90**	**62.94**	**63.21**	**72.94**	**60.51**	**68.36**	**66.26**

Table 3: Main results compared with LLM-based ABSA approaches. For DS^2-ABSA, synthetic data generation using GPT-3.5 Turbo and Llama-3-8B-Instruct are marked with * and *, respectively. DS^2-ABSA(PARA) denotes PARAPHRASE w/ DS^2-ABSA, and DS^2-ABSA(INST) denotes INSTRUCTABSA w/ DS^2-ABSA.

Method	2%-shot		5%-shot	
	Lap14	Res14	Lap14	Res14
DS^2-ABSA(PARA)	**56.86**	**64.92**	**60.83**	**68.41**
w/o key-point-driven	54.98	60.43	59.05	66.52
w/o instance-driven	55.45	63.74	58.59	67.65
w/o normalization	55.08	63.78	60.20	67.86
w/o noise injection	56.51	64.15	60.61	67.88

Table 4: Ablation on PARAPHRASE w/ DS^2-ABSA.

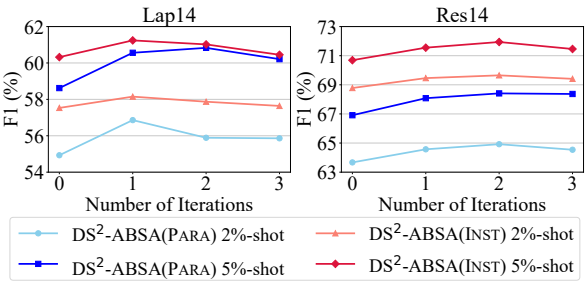


Figure 2: Effect of noisy self-training over iterations. Iteration 0 means noisy self-training is not conducted.

ABSA, enabling them to achieve superior results. Furthermore, using Llama-3-8B-Instruct and GPT-3.5 Turbo as the backbones of DS^2-ABSA yield similar results, demonstrating that our framework performs consistently well across LLMs with comparable capabilities.

4.4 Ablation Study

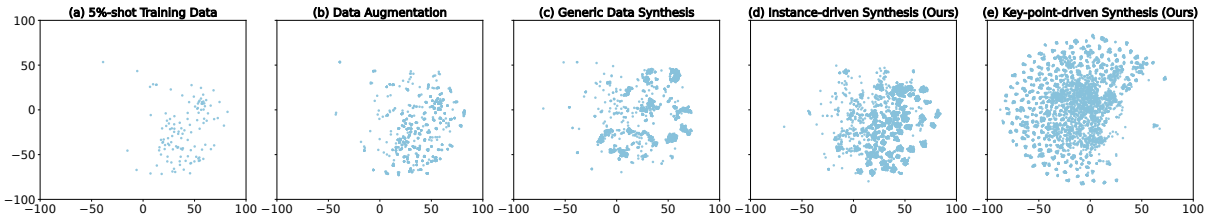
As shown in Table 4, we conduct ablation experiments to assess the impact of different components in DS^2-ABSA. The observations indicate that removing any strategy of the dual-stream synthesis results in a significant performance drop, underscoring the importance of integrating both streams. Particularly, in the 2%-shot setting, discarding key-point-driven synthesis would lead to an average F1 decrease by 3.15%, demonstrating the importance of generating highly diverse data in scenarios with extremely limited samples. Additionally, removing components in the label refinement process, such as label normalization or noise injection, also leads to decreased performance. These confirm that the rule-based approach alleviates inaccuracies in synthetic data, while noisy training enhances the robustness of the student model.

4.5 Effect of Noisy Self-training

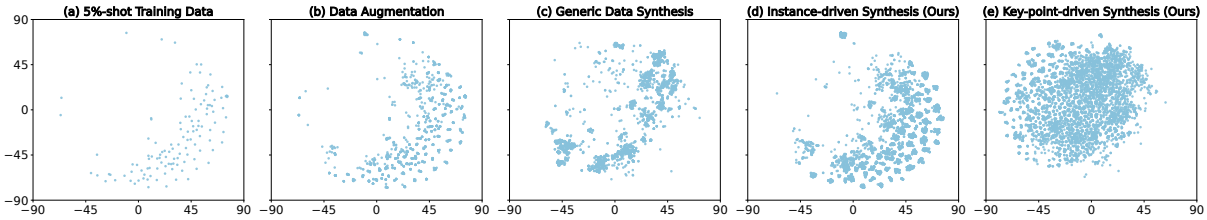
To explore the impact of noisy self-training, we analyze how performances of DS^2-ABSA(PARA) and DS^2-ABSA(INST) vary with the number of iterations, with results depicted in Figure 2. We observe that the F1 generally exhibits an initial increase as the number of iterations grows, confirming that self-training helps mitigate noise issues in synthetic labels. Notably, the most significant gain occurs during the first iteration, with average F1 improvements of 1.03% and 1.22% under the 2%-shot and 5%-shot settings, respectively. Furthermore, models typically reach optimal performance after one or two iterations. We speculate that this is due to the teacher model overfitting the training data after multiple iterations, which leads to a loss of the aspect-sentiment knowledge originally provided by LLMs in the synthetic data.

4.6 Exploring Diversity and Label Quality

Analysis of Data Diversity. As visualized in Figure 3b, different data generation methods vary in their ability to enhance data diversity. Data aug-



(a) Lap14 Datset



(b) Res14 Dataset

Figure 3: Data diversity comparison under the 5%-shot setting, including (a) few-shot gold data; (b) data augmentation (MELM, AugGPT, CoTAM); (c) generic data synthesis (ZeroGen, Self-Instruct); (d) instance-driven synthesis; and (e) key-point-driven synthesis. We use Instructor (Su et al., 2023) for text embedding and t-SNE for visualization, displaying at most 5k samples for clarity.

Method	Lap14			Res14		
	Asp	Senti	Pair	Asp	Senti	Pair
Key-point-driven	60.08	64.14	49.60	57.32	71.42	50.52
w/ label refinement	68.68	69.64	63.01	67.99	81.85	64.94
Instance-driven	70.90	80.17	60.41	77.72	86.12	71.93
w/ label refinement	86.26	85.87	76.93	88.16	88.26	82.71

Table 5: Analysis of text-label alignment in the 5%-shot setting using DS^2-ABSA(PARA). Asp, Senti, and, Pair represent F1 for aspects, macro-F1 for sentiments, and F1 for aspect-sentiment pairs, respectively.

mentation techniques generate reviews highly similar to the original data, exhibiting poor diversity. Meanwhile, generic data synthesis slightly broadens the range of semantic embeddings. In contrast, the instance-driven technique enhances fine-grained diversity by sample merging and aspect- or context-preserved synthesis, resulting in embeddings that form larger clusters around the seed data. The key-point-driven synthesis further enriches semantic diversity, producing data with a broader and more evenly distributed range of sentence representations. Additionally, data synthesized through dual-stream methods show complementary patterns, further validating the significance of the proposed framework.

Analysis of Label Quality. To analyze the label quality of dual-stream synthetic data and validate the effect of label refinement, we train INSTRUCTABSA with all training data and engage this model as the standard ABSA model for assessment. The results are displayed in Table 5. We find that instance-driven synthesis consistently achieves higher text-label consistency, significantly outperforming key-point-driven synthesis. Before refinement, the average F1 difference between them for aspect-sentiment pairs reaches 15.61%. Furthermore, label refinement greatly enhances the alignment, with improvements of aspect-sentiment pair F1 exceeding 10% for generated data from each stream, demonstrating its excellence.

4.7 Domain Generalization

To further assess the domain generalizability of our approach, we conduct experiments on two additional domains: *Book* and *Clothing*. For these domains, we utilize the datasets provided by Cai et al. (2023) for training and evaluation, and obtain unlabeled review corpora from Ni et al. (2019) to implement the pre-training and distillation baselines. As illustrated in Table 6, our method consistently outperforms other competitive few-shot techniques in both 2%-shot and 5%-shot settings. Notably, DS^2-ABSA(PARA) achieves a 2.26% average improvement over the second-best method, without relying on any additional domain-specific corpora. These results demonstrate that our method is not confined to specific domains. Instead, its domain-agnostic design, which leverages dual-stream synthesis and label refinement, enables effective transfer to other domains and significantly outperforms existing low-resource ABSA methods.

Method	2%-shot		5%-shot	
	Book	Clothing	Book	Clothing
PARAPHRASE	39.19	50.46	44.98	60.55
FS-ABSA‡	42.44	55.30	48.20	65.91
UniNER‡	42.82	55.47	46.36	66.22
ZeroGen*	42.91	56.88	48.01	64.95
Self-Instruct*	43.50	57.82	49.04	65.12
DS^2-ABSA*	**44.49**	**59.40**	**52.95**	**67.67**

Table 6: Analysis of domain generalization. Pre-training and distillation methods are marked with ‡, while data synthesis approaches are marked with *.

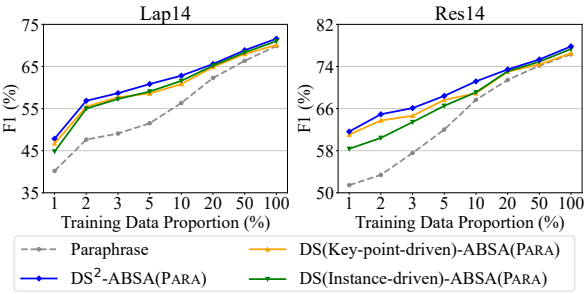


Figure 4: Effect of dual-stream synthesis methods using different training data proportions.

4.8 Discussions

Effect of Training Data Sizes. To investigate the effectiveness of DS^2-ABSA across varying training data sizes, we examine the performance of the dual-stream framework and its individual streams under different proportions of training data, as illustrated in Figure 4. Initially, the original PARAPHRASE displays the worst effects, whereas DS^2-ABSA(PARA) exhibits the best, proving the efficacy of dual-stream synthesis. Additionally, in conditions of scarce training data, key-point-driven synthesis outperforms instance-driven synthesis due to its ability to generate highly diverse samples via brainstormed attributes without relying on much seed data. As data volume increases, instance-driven synthesis gradually surpasses key-point-driven synthesis, suggesting that its effectiveness is positively correlated with the quantity of seed data available.

Effect of the Number of Synthetic Data. We investigate the impact of the quantity of synthetic data on model performance by sampling different proportions of synthetic data. As shown in Figure 5, model performance consistently improves as the amount of synthetic data increases. Notably, in the 2%-shot setting, the performance gains are more pronounced, with no signs of saturation even at

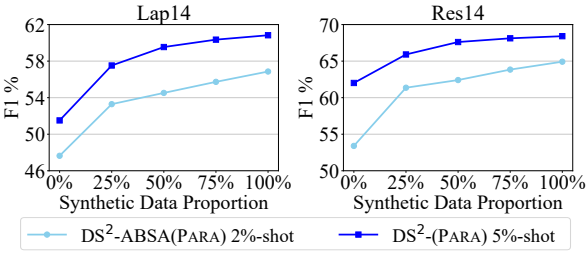


Figure 5: Impact of the number of synthetic data.

100% synthetic data. This suggests that synthetic data is particularly beneficial when gold data is extremely scarce. In contrast, in the 5%-shot setting, the rate of improvement gradually slows down as more synthetic data is added, indicating diminishing returns when more gold data is available. These observations indicate that increasing the amount of synthetic data has a more significant impact on performance when gold data is limited, highlighting its importance in low-resource scenarios.

5 Conclusion

In this paper, we introduce DS^2-ABSA, a dual-stream data synthesis approach with label refinement tailored for few-shot E2E-ABSA. By leveraging LLMs with both key-point-driven and instance-driven strategies, our framework effectively generates diverse and well-aligned ABSA samples without requiring additional corpora, thereby overcoming the limitations of existing approaches. The label refinement module further enhances data quality, contributing to improved overall performance. Extensive experiments on four datasets demonstrate that DS^2-ABSA significantly outperforms a range of low-resource enhancement techniques and LLM-based methods, offering a promising solution for few-shot ABSA. Furthermore, we believe that our pipeline can serve as a valuable reference for data synthesis in other fields.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China 62176076, Natural Science Foundation of Guang Dong 2023A1515012922, the Shenzhen Foundational Research Funding JCYJ20220818102415032, the Major Key Project of PCL2023A09, Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies 2022B1212010005 and CIPSC-SMP-ZHIPU Large Model Cross-Disciplinary Fund ZPCG20241119405.

Limitations

Despite the proposed DS^2-ABSA framework offering an effective solution for few-shot E2E-ABSA, several limitations still remain.

- Although prior work provides annotation guidelines and specific examples for ABSA, we have not explored their incorporation into our current data synthesis process.
- While the label refinement module improves data quality, it may not fully eliminate inaccuracies arising from inherent biases in LLMs. Such biases can impair downstream performance, as residual errors in the synthetic data may lead to sub-optimal model training and predictions.
- The prompts in key-point-driven synthesis rely on careful manual design. While this process does require an initial investment of effort, it's essential for yielding diverse reviews with promising label quality. Moreover, the consistent improvements observed across multiple domains indicate that the prompts can be effectively reused, making the cost largely a one-time effort that simplifies adaptation to new ABSA domains.

We believe that addressing these issues provides a promising direction for further improvement.

References

Hongjie Cai, Nan Song, Zengzhi Wang, Qiming Xie, Qiankun Zhao, Ke Li, Siwei Wu, Shijie Liu, Jianfei Yu, and Rui Xia. 2023. Memd-absa: A multi-element multi-domain dataset for aspect-based sentiment analysis.

Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094*.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6045–6057.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hao Fei, Fei Li, Chenliang Li, Shengqiong Wu, Jingye Li, and Donghong Ji. 2022. Inheriting the wisdom of predecessors: A multiplex cascade framework for unified aspect-based sentiment analysis. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4121–4128.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Ting-Wei Hsu, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. Semantics-preserved data augmentation for aspect-based sentiment analysis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4417–4422.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yiming Huang, Xiao Liu, Yeyun Gong, Zhibin Gou, Yelong Shen, Nan Duan, and Weizhu Chen. 2024. Key-point-driven data synthesis with its enhancement on mathematical reasoning. *arXiv preprint arXiv:2403.02333*.

Fan Jiang, Tom Drummond, and Trevor Cohn. 2023. Noisy self-training with synthetic queries for dense retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11991–12008.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. 2020. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7056–7066.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461.

Juhua Liu, Qihuang Zhong, Liang Ding, Hua Jin, Bo Du, and Dacheng Tao. 2023. Unified instance and knowledge alignment pretraining for aspect-based sentiment analysis. *IEEE/ACM transactions on audio, speech, and language processing*, 31:2629–2642.

Shunyu Liu, Jie Zhou, Qunxi Zhu, Qin Chen, Qingchun Bai, Jun Xiao, and Liang He. 2024. Let's rectify step by step: Improving aspect-based sentiment analysis with diffusion models. In *Proceedings of the*

2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 10324–10335.

Yang Liu, Sheng Shen, and Mirella Lapata. 2021. Noisy self-knowledge distillation for text summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 692–703.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11065–11082.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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)*, pages 188–197.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Letian Peng, Yuwei Zhang, and Jingbo Shang. 2024. Controllable data augmentation for few-shot text mining with chain-of-thought attribute manipulation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1–16.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, et al. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, et al. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, et al. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35.

Kevin Scaria, Himanshu Gupta, Siddharth Goyal, Saurabh Sawant, Swaroop Mishra, and Chitta Baral. 2024. Instructabsa: Instruction learning for aspect based sentiment analysis. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 720–736.

Paul F Simmering and Paavo Huoviala. 2023. Large language models for aspect-based sentiment analysis. *arXiv preprint arXiv:2310.18025*.

Hongjin Su, Weijia Shi, Jungo Kasai, et al. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121.

Yuanhe Tian, Weidong Chen, Bo Hu, Yan Song, and Fei Xia. 2023. End-to-end aspect-based sentiment analysis with Combinatory Categorial Grammar. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13597–13609.

Siddharth Varia, Shuai Wang, Kishaloy Halder, Robert Vacareanu, Miguel Ballesteros, Yassine Benajiba, Neha Anna John, Rishita Anubhai, Smaranda Muresan, and Dan Roth. 2023. Instruction tuning for few-shot aspect-based sentiment analysis. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 19–27, Toronto, Canada.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.

Qianlong Wang, Keyang Ding, Xuan Luo, and Ruifeng Xu. 2024b. Improving in-context learning via sequentially selection and preference alignment for few-shot aspect-based sentiment analysis. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2462–2466.

Qianlong Wang, Hongling Xu, Keyang Ding, Bin Liang, and Ruifeng Xu. 2024c. In-context example retrieval from multi-perspectives for few-shot aspect-based sentiment analysis. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8975–8985.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023a. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormo labashi, et al. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Zengzhi Wang, Qiming Xie, Yi Feng, Zixiang Ding, Zinong Yang, and Rui Xia. 2024d. Is chatgpt a good sentiment analyzer? a preliminary study. In *First Conference on Language Modeling*.

Zengzhi Wang, Qiming Xie, and Rui Xia. 2023b. A simple yet effective framework for few-shot aspect-based sentiment analysis. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1765–1770.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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)*, pages 6382–6388.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335.

Ran Xu, Hejie Cui, Yue Yu, Xuan Kan, Wenqi Shi, Yuchen Zhuang, May Dongmei Wang, Wei Jin, Joyce Ho, and Carl Yang. 2024. Knowledge-infused prompting: Assessing and advancing clinical text data generation with large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15496–15523.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669.

Chengze Yu, Taiqiang Wu, Jiayi Li, Xingyu Bai, and Yujiu Yang. 2023. Syngen: A syntactic plug-and-play module for generative aspect-based sentiment analysis. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

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

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024a. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Trans. on Knowl. and Data Eng.*, 35(11):11019–11038.

Yice Zhang, Guangyu Xie, Hongling Xu, Kaiheng Hou, Jianzhu Bao, Qianlong Wang, Shiwei Chen, and Ruifeng Xu. 2024b. Distilling fine-grained sentiment understanding from large language models.

Yice Zhang, Yifan Yang, Bin Liang, Shiwei Chen, Bing Qin, and Ruifeng Xu. 2023. An empirical study of sentiment-enhanced pre-training for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9633–9651.

Chenyang Zhao, Xueying Jia, Vijay Viswanathan, Graham Neubig, and Tongshuang Wu. 2024. Self-guide: Better task-specific instruction following via self-synthetic finetuning. In *First Conference on Language Modeling*.

Guangmin Zheng, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2024. Instruction tuning with retrieval-based examples ranking for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4777–4788.

Jie Zhou, Junfeng Tian, Rui Wang, Yuanbin Wu, Wenming Xiao, and Liang He. 2020. Sentix: A sentiment-aware pre-trained model for cross-domain sentiment analysis. In *Proceedings of the 28th international conference on computational linguistics*, pages 568–579.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. MELM: Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations*.

Senbin Zhu, Hanjie Zhao, Yuxiang Jia, and Hongying Zan. 2024. ZZU-NLP at SIGHAN-2024 dimABSA task: Aspect-based sentiment analysis with coarse-to-fine in-context learning. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 112–120.

Appendix for "DS^2-ABSA: Dual-Stream Data Synthesis with Label Refinement for Few-Shot Aspect-Based Sentiment Analysis"

We organize the appendix into three sections:

- Prompts utilized for synthetic data generation and LLM-based methods are presented in Appendix A;
- More implementation details and the descriptions of all baseline methods are presented in Appendix B;
- Supplemental discussions and results can be referenced in Appendix C.

A Prompt Design

For key-point-driven synthesis, we illustrate the prompts utilized in brainstorming and attribute prompting in Table 12 and 13, respectively. For instance-driven synthesis, prompts for sample combination and selective reconstruction strategies are both presented in Table 14. Additionally, for LLM-based methods, including in-context learning and supervised fine-tuning, the utilized prompts are presented in Table 15.

B Additional Experimental Settings

B.1 Hyper-parameter Settings of Refinement

We adjust different hyper-parameters for various ABSA models to achieve optimal performance. The settings are presented in Table 7.

Data Type	ABSA Model	BS	LR	Epochs
Synthetic Data	TAG-BERT	32	3e-6	5
	PARAPHRASE	32	3e-5	5
	INSTRUCTABSA	24	2e-5	5
Few-shot Gold Data	TAG-BERT	8	5e-6	20
	PARAPHRASE	8	1e-4	20
	INSTRUCTABSA	8	1e-4	20

Table 7: Hyper-parameter Settings for different data types and models. BS and LR denote batch size and learning rate, respectively.

B.2 Baseline Descriptions

Data Augmentation. (1) **EDA** (Wei and Zou, 2019): A simple data augmentation technique that employs synonym replacement, random insertion, deletion, and swapping to enhance data variety and expand the dataset. (2) **CA** (Li et al., 2020): Utilizes span-masking and embeds label information as conditions to generate augmented sentences with varied contextual content. (3) **MELM** (Zhou et al., 2022): A masked language modeling approach that linearizes and embeds labels into text sequences while performing masked entity prediction to augment training data. (4) **AugGPT** (Dai et al., 2023): Utilizes LLMs to generate rephrased training samples. Each training sample is rephrased into six augmented versions, which are then combined with the original data for fine-tuning the ABSA model. (5) **CoTAM** (Peng et al., 2024): Manipulates task-specific attributes, such as sentiment, through a three-step process: decomposition, manipulation, and reconstruction. For ABSA, it modifies the sentiment of each aspect to generate controlled augmented data.

Pre-training. (1) **BERT-PT** (Xu et al., 2019): Adapts pre-trained BERT to domain-specific review data through task-specific post-training. (2) **SentiX** (Zhou et al., 2020): Proposes multi-level pre-training tasks to learn domain-invariant sentiment knowledge. (3) **SPT-ABSA** (Zhang et al., 2023): A sentiment-specific pre-training method for ABSA that integrates various sentiment knowledge from reviews. (4) **DAPT** (Gururangan et al., 2020): Applies domain-adaptive pre-training using span corruption on 100k domain-specific reviews per domain, following Wang et al. (2023b). (5) **FS-ABSA** (Wang et al., 2023b): Combines domain-adaptive pre-training and text-infilling fine-tuning to optimize few-shot ABSA, narrowing the gap between pre-training and downstream tasks.

Distillation. (1) **UniNER** (Zhou et al., 2024): Performs targeted distillation by extracting aspect-sentiment pairs from texts using LLMs, followed by sentiment-based conversational fine-tuning to distill the knowledge into ABSA models. For a fair comparison with DS^2-ABSA, we collect 20k reviews for each domain, matching the data volume.

Data Synthesis. (1) **ZeroGen** (Ye et al., 2022): Prompts LLMs to generate text by providing sentiment labels. For ABSA, we input the target sentiment into the LLM and use prompting to generate domain-specific reviews annotated with aspect-sentiment pairs. (2) **Self-Instruct** (Wang et al., 2023a): A typical pipeline for generating input-output pairs and performing data filtering. Here, we provide four ABSA examples to guide LLMs in generating additional reviews and labels. The synthesized data is filtered for diversity, retaining only those samples with a ROUGE-L similarity $<$ 0.7 to existing reviews in the pool.

LLM-based ABSA Methods. (1) **Zero-shot Prompting**: Designing prompts to guide GPT-4 (OpenAI, 2023) for generating aspect-sentiment predictions directly without using any gold data. (2) **In-context Learning** (Wang et al., 2024c): Retrieves in-context demonstrations for ABSA based on semantic similarity, syntactic relevance, and aspect-sentiment semantics. (3) **Supervised Fine-tuning** (Simmering and Huoviala, 2023): Fine-tunes LLMs on few-shot ABSA gold data using task-specific instructions. Here, fine-tuning GPT-3.5 Turbo is implemented by Openai API.

C Additional Discussions

C.1 Effect of ABSA Models on Full Data

We conduct experiments using full training data to evaluate different ABSA models, and the results are displayed in Table 8. We observe that INSTRUCTABSA achieves the best performance across all datasets, demonstrating its effectiveness. Furthermore, while our synthetic data falls short compared to golden data, it still performs well in few-shot settings. Specifically, compared with the experimental results in Tabel 2, DS^2-ABSA(INST) can achieve 84.22% of the full-dataset performance in average F1 using only 2% of the training data, which increases to 87.81% with 5%-shot data, highlighting the quality of the synthetic data. These demonstrate the effectiveness of our method in low-resource scenarios.

Model (Params)	Lap14	Res14	Res15	Res16	Avg
TAG-BERT(109M)	61.98	74.25	60.71	69.03	66.49
PARAPHRASE(220M)	68.63	76.31	66.51	73.85	71.33
INSTRUCTABSA(770M)	70.48	79.28	71.96	77.29	74.75

Table 8: Full-shot performance of ABSA models.

C.2 Sentiment Distribution of Opinions

Section 3.1 introduces our key-point prompting strategy, which guides the LLM to generate opinion–sentiment pairs simultaneously (see Table 12 for the prompt). To assess the impact of this design, we compare two prompting strategies for brainstorming opinion terms: (a) generating opinions only, and (b) generating pairs (ours). As shown in Table 9, strategy (a) exhibits a strong positive sentiment bias, with over 50% of generated terms being positive and very few neutral ones. In contrast, strategy (b) significantly reduces this bias, yielding a more balanced sentiment distribution that better aligns with real-world data. These results indicate that our approach mitigates polarity skew at the opinion-term level, enabling the generation of more diverse synthetic comments and ultimately supporting more reliable data quality for downstream ABSA tasks.

C.3 Impact of Backbone LLMs

To examine whether DS^2-ABSA benefits from stronger backbone LLMs, we extend the analysis in Table 3 by incorporating the more advanced GPT-4 model for synthetic data generation. As shown in Table 10, using GPT-4 consistently enhances downstream ABSA performance across both DS^2-ABSA(PARA) and DS^2-ABSA(INST) models under the 5%-shot setting. On average, GPT-4 yields a 1.68% F1 improvement over GPT-3.5 Turbo and a 1.43% gain over Llama-3-8B-Instruct. These results not only demonstrate the generalizability of our method across different LLMs, but also support the hypothesis that stronger language models produce higher-quality synthetic data, enabling scalable improvements in low-resource ABSA as LLM capabilities continue to advance. Nevertheless, due to the substantially higher expense of GPT-4, most of our experiments are conducted with GPT-3.5 Turbo, which offers a more practical trade-off between performance and cost.

C.4 Examples of Synthetic Data from LLMs

To enhance clarity, Table 16 and 17 present examples of data and labels synthesized by LLMs under different input conditions, which cover various strategies including attribute prompting, sample combination, and selective reconstruction.

Domain	Strategy	Sentiment Polarity (%)		
		Positive	Neutral	Negative
Laptop	(a) opinion only	52.47	6.84	40.68
	(b) opinion–sentiment (ours)	38.17	21.37	40.46
Restaurant	(a) opinion only	51.97	9.45	38.58
	(b) opinion–sentiment (ours)	39.39	22.08	38.53

Table 9: Sentiment polarity distribution of brainstormed opinion terms under two prompting strategies: (a) outputting opinions only, and (b) generating [opinion, sentiment] pairs (ours).

Method	Backbone LLM	Lap14	Res14	Res15	Res16	Avg
DS^2-ABSA(PARA)	GPT-3.5 Turbo	60.83	68.41	54.32	61.87	61.36
	Llama-3-8B-Instruct	60.57	68.83	54.62	62.44	61.62
	GPT-4	**62.37**	**70.58**	**55.63**	**64.62**	**63.30**
DS^2-ABSA(INST)	GPT-3.5 Turbo	61.24	71.94	60.56	68.81	65.64
	Llama-3-8B-Instruct	63.21	72.94	60.51	68.36	66.26
	GPT-4	**64.37**	**74.06**	**63.24**	**69.54**	**67.80**

Table 10: Comparison of different backbone LLMs used for data synthesis in the proposed DS^2-ABSA framework under the 5%-shot setting. Results for GPT-3.5 Turbo and Llama-3-8B-Instruct are from Table 3.

ABSA Model	Method	2%-shot					5%-shot				
		Lap14	Res14	Res15	Res16	Avg(Δ)	Lap14	Res14	Res15	Res16	Avg(Δ)
TAG-BERT (Hu et al., 2019)	Origin	15.83	37.49	23.04	20.19	24.14	35.29	51.64	34.52	43.48	41.23
	EDA†	26.31	43.14	21.93	35.15	31.63$_{+7.49}$	40.77	52.96	30.06	45.92	42.43$_{+1.20}$
	CA†	31.36	42.80	25.27	32.91	33.09$_{+8.95}$	39.14	53.18	29.22	46.75	42.07$_{+0.84}$
	SentiX‡	36.35	46.07	28.30	39.80	37.63$_{+13.49}$	44.18	60.62	35.05	53.35	48.30$_{+7.07}$
	DS^2-ABSA*	**47.30**	**60.39**	**49.49**	**59.40**	**54.15$_{+30.01}$**	**47.97**	**62.37**	**49.26**	**58.40**	**54.50$_{+13.27}$**
PARAPHRASE (Zhang et al., 2021)	Origin	47.64	53.40	39.23	39.75	45.01	51.51	62.01	51.45	52.58	54.39
	EDA†	47.76	55.11	43.03	45.10	47.75$_{+2.74}$	51.40	62.14	49.97	56.41	54.98$_{+0.59}$
	CA†	50.33	52.17	43.07	43.74	47.33$_{+2.32}$	51.44	61.63	49.01	53.55	53.91$_{-0.48}$
	DAPT‡	48.03	56.56	44.56	47.84	49.25$_{+4.24}$	51.81	63.66	52.37	58.21	56.51$_{+2.13}$
	DS^2-ABSA*	**56.86**	**64.92**	**53.15**	**61.16**	**59.02$_{+14.01}$**	**60.83**	**68.41**	**54.32**	**61.87**	**61.36$_{+6.97}$**
INSTRUCTABSA (Scaria et al., 2024)	Origin	52.05	63.36	53.67	58.78	56.97	57.54	67.59	55.08	62.91	60.78
	EDA†	52.71	63.84	53.40	59.68	57.41$_{+0.44}$	56.47	67.06	53.73	63.29	60.14$_{-0.64}$
	CA†	54.39	62.79	53.28	57.30	56.93$_{-0.04}$	57.77	66.81	53.03	61.61	59.81$_{-0.97}$
	DAPT‡	55.34	64.90	55.39	60.46	59.02$_{+2.05}$	59.96	69.01	55.87	63.69	62.13$_{+1.35}$
	DS^2-ABSA*	**58.15**	**69.65**	**59.78**	**63.89**	**62.87$_{+5.90}$**	**61.24**	**71.94**	**60.56**	**68.81**	**65.64$_{+4.86}$**

Table 11: More baseline results are listed here. Data augmentation, pre-training, and data synthesis methods are marked with †, ‡, and *, respectively.

Target	Prompt
Review Subject	Brainstorm a list of {domain} descriptions (at least 200). Please adhere to the following guidelines: - Names are not required. - Summarize the core features and specialties in a short, neutral sentence. Your output should be a Python list of strings, with each element being a description.
Aspect Category	Brainstorm a list of commonly used aspect categories in {domain} reviews. Please adhere to the following guidelines: - Aspect categories should cover various potential aspects that opinions can be expressed about within the corresponding domain. - Aspect categories are coarse-grained overviews, not including specific things. Your output should be a Python list of strings, with each element being a brief word denoting an aspect category.
	Please filter the list to retain only distinct and representative aspect categories within the {domain} domain. Output the reason for selection along with the filtered Python list.
Aspect Term	Brainstorm a list of commonly used aspect terms for the aspect category {aspect category} within the {domain} domain. Please adhere to the following guidelines: - Aspect terms should cover various potential things that opinions can be expressed about within the corresponding category. - Aspect terms are fine-grained and concrete things. - Aspect terms are single or multiword terms naming particular aspects of the target entity. Your output should be a Python list of strings, with each element being an aspect term.
Opinion Term	Brainstorm a list of commonly used opinion terms for the aspect category {aspect category} within the {domain} domain. Please adhere to the following guidelines: - Opinion terms refer to the expression carrying subjective emotions. - Provide diverse words and phrases covering positive, negative, and neutral sentiments. Your output should be a Python list of lists, with each element being an [opinion, sentiment] pair.

Table 12: Brainstorming prompts.

Method	Prompt
Attribute Prompting	Write a review sentence for the {domain}: {review subject} Label the sentence by extracting the aspect term(s) and identifying their corresponding sentiment polarity (positive, negative, or neutral). Requirements: - Keep a consistent style and annotation standard with the examples. - Mention the aspect term '{aspect}'. - Describe {aspect category} by the opinion term '{opinion}'. - Express {sentiment expression} across aspects. Here are some examples: Sentence: {sentence} Label: {label} ... ... Sentence:

Table 13: Prompt template for attribute prompting.

Method	Prompt
Sample Combination	Given 2 {domain} example reviews with the labels, please combine them to generate 4 diverse sentences. Label each sentence by extracting the aspect term(s) and determine their corresponding sentiment polarity. Requirements: - Keep a consistent style and annotation standard with the examples. - Maintain the same format as the example. - Combine the aspects and meanings of both examples in every generated sentence. Examples: Sentence: {sentence} Label: {label} Sentence: {sentence'} Label: {label'} 4 Diverse Combined Sentences with Labels: 1. Sentence:
	Given a {domain} example review with the label, please paraphrase it to generate 4 diverse sentences. Label each sentence by extracting the aspect term(s) and determine their corresponding sentiment polarity. Requirements: - Keep a consistent style and annotation standard with the example. - Maintain the same format as the example. - The meaning of the example sentence should be unchanged. Example: Sentence: {sentence} Label: {label} 4 Diverse Paraphrased Sentences with Labels: 1. Sentence:
Selective Reconstruction	Given a partially masked {domain} review sentence, please reconstruct it to generate 4 diverse sentences. Label each sentence by extracting the aspect term(s) and determine their corresponding sentiment polarity. Masked Sentence: {mask sentence} Requirements: - Keep a consistent style and annotation standard with the example. - Maintain the same format as the example. - The unmasked part of the should be unchanged. Example: Sentence: {sentence} Label: {label} 4 Diverse Reconstructed Sentences with Labels: 1. Sentence:

Table 14: Prompts for instance-driven data synthesis.

Method	Prompt
In-context Learning & Supervised Fine-tuning	Given a review, extract the aspect term(s) and determine their corresponding sentiment polarity (positive, negative, or neutral). Format the label as follows: [['aspect1', 'sentiment1'], ['aspect2', 'sentiment2'], ...]. If there are no aspect terms, use an empty list []. *(Here are some examples: . . .)* Sentence: {test input} Label:

Table 15: Prompts for LLM-based approaches. Note that the examples are utilized only in in-context learning.

Method	Input Prompt / Output
Attribute Prompting	Write a review sentence for the laptop: A laptop offers adaptive performance settings based on usage. Label the sentence by extracting the aspect term(s) and identifying their corresponding sentiment polarity (positive, negative, or neutral). Requirements: - Keep a consistent style and annotation standard with the examples. - Mention the aspect term 'lid rigidity'. - Describe software by the opinion term 'efficient'. - Express a consistent sentiment across aspects. Here are some examples: Sentence: The laptop is relatively simple to use, though I bought Macs for Dummies, which is well worth $2 Label: [['use', 'positive']] Sentence: The computer is currently in West Verginia doe to the method of shipping choosen by Toshiba. Label: [['shipping', 'negative']] Sentence: It weighed like seven pounds or something like that. Label: [['weighed', 'neutral'], ['seven pounds', 'neutral']] Sentence: I need graphic power to run my Adobe Creative apps efficiently. Label: [['graphic power', 'neutral'], ['Adobe Creative apps', 'neutral']] Sentence:
	The laptop offers impressive lid rigidity and efficient software for adaptive performance settings based on usage. Label: [['lid rigidity', 'positive'], ['efficient software', 'positive'], ['adaptive performance settings', 'positive']]

Table 16: Examples of key-point-driven synthetic data from LLMs.

Method	Input Prompt / Output
Sample Combination	Given 2 restaurant example reviews with the labels, please combine them to generate 4 diverse sentences. Label each sentence by extracting the aspect term(s) and determine their corresponding sentiment polarity. Requirements: - Keep a consistent style and annotation standard with the examples. - Maintain the same format as the examples. - Combine the aspects and meanings of both examples in every generated sentence. Examples: 1. Sentence: Fast service. Label: [['service', 'positive']] 2. Sentence: The miso soup lacked flavor and the fish was unfortunately not as well prepared as in the past. Label: [['miso soup', 'negative'], ['flavor', 'negative'], ['fish', 'negative']] 4 Diverse Combined Sentences with Labels: 1. Sentence:
	The miso soup lacked flavor but the service was fast and efficient. Label: [['miso soup', 'negative'], ['flavor', 'negative'], ['service', 'positive']] 2. Sentence: The fish was unfortunately not as well prepared as in the past, however, the service was quick and attentive. Label: [['fish', 'negative'], ['service', 'positive']] 3. Sentence: Fast service made up for the lack of flavor in the miso soup. Label: [['service', 'positive'], ['miso soup', 'negative'], ['flavor', 'negative']] 4. Sentence: Despite the miso soup lacking flavor, the quick service redeemed the overall dining experience. Label: [['miso soup', 'negative'], ['flavor', 'negative'], ['service', 'positive']]
Selective Reconstruction	Given a partially masked restaurant review sentence, please reconstruct it to generate 4 diverse sentences. Label each sentence by extracting the aspect term(s) and determine their corresponding sentiment polarity. Masked Sentence: <mask> anything spectacular <mask> Requirements: - Keep a consistent style and annotation standard with the example. - Maintain the same format as the example. - The unmasked part of the should be unchanged. Example: Sentence: Sushi wasn't anything spectacular for the price. Label: [['Sushi', 'neutral'], ['price', 'negative']] 4 Diverse Reconstructed Sentences with Labels: 1. Sentence:
	Sushi wasn't anything spectacular for the price. Label: [['Sushi', 'neutral'], ['price', 'negative']] 2. Sentence: The ambiance lacked anything spectacular, but the service was impeccable. Label: [['ambiance', 'negative'], ['service', 'positive']] 3. Sentence: The dessert was anything spectacular, unlike the bland main course. Label: [['dessert', 'positive'], ['main course', 'negative']] 4. Sentence: The presentation was anything spectacular, making up for the slightly high prices. Label: [['presentation', 'positive'], ['prices', 'neutral']]"

Table 17: Examples of instance-driven synthetic data from LLMs.