DS^2-ABSA: Dual-stream Data Synthesis with Label Refinement for Few-shot Aspect-based Sentiment Analysis

Anonymous ACL submission

Abstract

Aspect-based sentiment analysis (ABSA) aims to recognize aspect terms and their corresponding sentiments within user reviews. A significant challenge in ABSA is the scarcity of labeled data, which limits the performance of existing models. To address this issue, previous efforts explore data augmentation techniques, which generate new samples by modifying existing ones. However, these methods fail to produce adequately diverse samples, impairing their effectiveness. Recently, researchers have applied large language models (LLMs) to ABSA using in-context learning, prompting LLMs with specific instructions and a few examples. Though promising, these methods often generate results that deviate from task requirements. To overcome these limitations, we propose DS^2-ABSA, a dual-stream data synthesis framework for few-shot ABSA. This framework leverages LLMs to synthesize ABSA samples from two distinct perspectives: *key-point-driven* and *instance-driven*. Furthermore, it integrates a *label refinement* module to refine the label quality of the synthesized data. Our framework not only produces more diverse samples than previous data augmentation methods but also exhibits better task-specific alignment than in-context learning methods. Extensive experiments demonstrate that DS^2-ABSA substantially outperforms previous methods.[1]

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 would be {(*battery life*, positive), (*screen resolution*, negative)}. With the advancement of deep learning technologies, numerous neural approaches have been developed for ABSA (Hu et al., 2019; Yan et al., 2021; Zhang et al., 2021; Scaria et al., 2024). While these methods demonstrate strong performance, they rely heavily 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., 2023, 2024d; Zhang et al., 2024).

Existing low-resource solutions comprise three main types. The first, **data augmentation**, produces additional samples by modifying existing ones. These modifications are typically implemented using heuristic rules (Wei and Zou, 2019) or masked language models (Li et al., 2020; Ding et al., 2020; Zhou et al., 2022). Although these methods can yield a large number of new samples, these samples exhibit limited diversity, thereby providing marginal benefit for subsequent model training. The second type, **in-context learning**, aligns large language models (LLMs) with the ABSA task through task-specific instructions and a few examples (Wang et al., 2024d; Zhang et al., 2024; Wang et al., 2024c). 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., 2023; Zhang et al., 2023b) to reduce reliance on downstream datasets. Nonetheless, these techniques require vast additional corpora and incur high training costs.

To overcome the above limitations, we propose **DS^2-ABSA**, a dual-stream data synthesis framework for few-shot ABSA. Our framework leverages LLMs to synthesize ABSA data via two distinct strategies: *key-point-driven* and *instance-driven*. The former engages LLMs to brainstorm a variety

[1]We will release our code and data via GitHub.

of potential ABSA attributes, which are then composed to create synthesized ABSA samples. The latter transforms 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 synthesizes data based on existing samples and provides better label 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 traditional 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. The label refinement module further enhances this advantage. Thirdly, unlike pre-training methods, our approach does not require any additional corpus and entails significantly lower training costs.

Our contributions are summarized as follows:

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

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

(3) Experimental results on four public datasets demonstrate our approach significantly outperforms previous methods. Moreover, we provide in-depth analyses of our approach, offering valuable technical insights.

2 Related Work

2.1 Few-shot Aspect-based Sentiment Analysis

Current state-of-the-art ABSA methods rely on fine-tuning text-to-text language models on sufficient labeled data (Yan et al., 2021; Zhang et al., 2021; Scaria et al., 2024). Nonetheless, annotating fine-grained sentiments in adequate reviews is expensive. To mitigate data dependence, traditional methods employ (1) data augmentation (Wei and Zou, 2019; Li et al., 2020; Hsu et al., 2021; Wang et al., 2021; Zhou et al., 2022), which suffers from low diversity; and (2) pre-training (Xu et al., 2019; Zhou et al., 2020; Li et al., 2021; Wang et al., 2023; Zhang et al., 2023b), which requires substantial extra corpora.

With the recent emergence of powerful LLMs, researchers have tried to leverage their in-context learning capabilities for ABSA (Wang et al., 2024d; Zhang et al., 2024; Wang et al., 2024c,b; Zhu et al., 2024). However, LLMs often fail to produce task-aligned outputs. While supervised fine-tuning offers a potential solution (Simmering and Huoviala, 2023), it incurs notable training expenses. 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

With the advent of LLMs, data synthesis has become a pivotal strategy to address data-related challenges such as high annotation costs and data privacy concerns (Long et al., 2024; Liu et al., 2024). It has been applied to various domains, including instruction tuning (Honovich et al., 2023; Li et al., 2024; Zhao et al., 2024), mathematical reasoning (Huang et al., 2024; Yu et al., 2024; Chan et al., 2024), text embedding (Zhang et al., 2023a; Wang et al., 2024a), and so on. These studies mainly concentrate on enhancing the quality, authenticity, and diversity of synthetic data (Long et al., 2024).

Existing data synthesis approaches can be grouped into three types: (1) instance-driven synthesis (Honovich et al., 2023; Zhao et al., 2024; Yu et al., 2024; Zhang et al., 2023a), which leverages examples to guide LLMs in generating relevant data; (2) key-point-driven synthesis (Yu et al., 2023; Li et al., 2024; Huang et al., 2024; Wang et al., 2024a), utilizing conditional prompts to produce data sticking to specific key points; and (3) knowledge-driven synthesis (Xu et al., 2024; Chan et al., 2024), incorporating external knowledge to steer the generation process. In practice, these strategies are often combined to achieve optimal data quality. After generation, methods like data

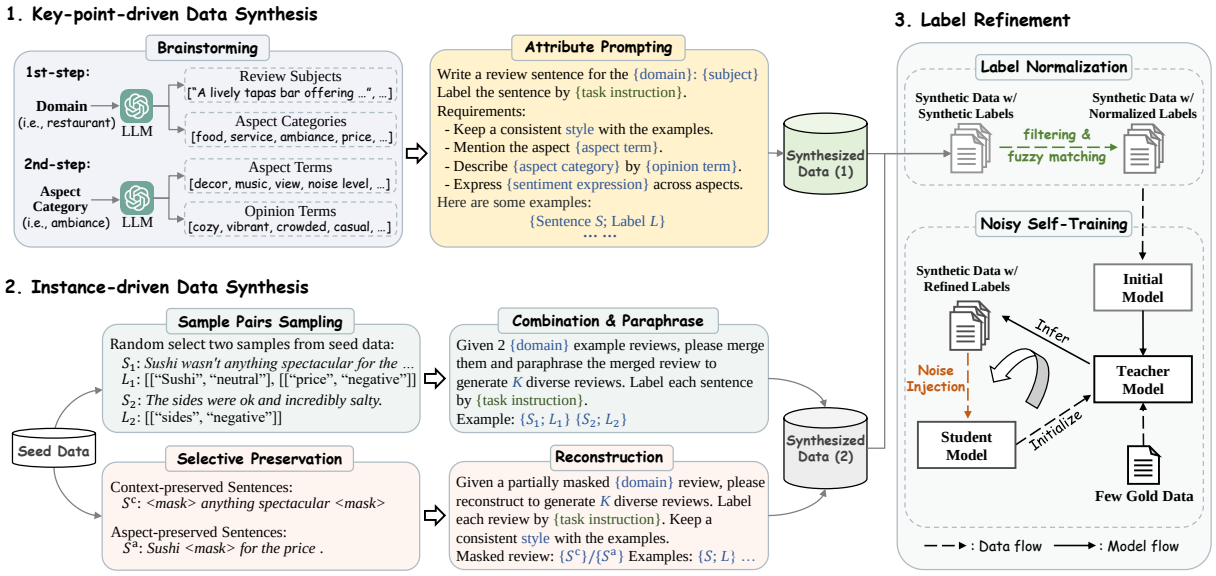

Figure 1: Overview of the proposed DS^2-ABSA.

filtering (Zhao et al., 2024; Seedat et al., 2024) are used for data curation. On this basis, our study pioneers the application of data synthesis to ABSA by integrating key-point-driven and instance-driven approaches, along with normalization and noisy self-training for quality refinement.

3 Method

As depicted in Figure 1, DS^2-ABSA consists of three modules: *key-point-driven* data synthesis, *instance-driven* data synthesis, and *label refinement*. The key-point-driven and instance-driven modules function as parallel streams to synthesize data. These synthesized data are then merged and fed into the label refinement module to enhance the quality of the labels. We provide detailed descriptions of each module below.

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 attributed 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 refinement strategy 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. By leveraging this rule, 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 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 conduct data generation by using **GPT-3.5 Turbo** (Ouyang et al., 2022).[2] Training samples with explicit aspects are selected as seed data for synthesis. For key-point-driven synthesis, we generate 20,000 samples using randomly combined attributes and 4 examples in prompts. For instance-driven synthesis, we limit the maximum combined samples to 1,000, set the aspect masking window sizes to 0 and 2, and the context masking probability to 0.6 with random masking twice. Besides, the number of generated samples in a single response, K, is set to 4. See Appendix A for detailed prompts used in data synthesis. During label refinement, p_{noise} is set to 0.1, the maximum number of iterations is 2, and additional hyper-parameters 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**.

[2]The specific version we used is *gpt-3.5-turbo-0125*.

4.2 Baselines

To validate the effectiveness of our method, we employ three ABSA models, including (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), utilizing instruction tuning with 2 positive examples, for which we apply *Tk-instruct-large* (Wang et al., 2022).

To benchmark against data augmentation methods, we implement *EDA* (Wei and Zou, 2019), *CA* (Li et al., 2020), and *MELM* (Zhou et al., 2022). Additionally, we evaluate against pre-training methods across various model types. Specifically, we use *BERT-PT* (Xu et al., 2019) and *SPT-ABSA* (Zhang et al., 2023b) for BERT models, while utilizing *DAPT* (Gururangan et al., 2020) and *FS-ABSA* (Wang et al., 2023) for text-to-text models. Moreover, to compare with LLM-based approaches, in addition to *GPT-3.5 Turbo*, we furthermore use Llama-3-8B-Instruct (Dubey et al., 2024) (denoted as *Llama-3-8B*) as the backbone. Besides, we apply the *In-context Learning* method proposed by (Wang et al., 2024c) and conduct *Supervised Fine-tuning* (Simmering and Huoviala, 2023). Detailed descriptions of these methods are provided in Appendix B.2.

4.3 Main Results

Comparison with Traditional Methods. The results in Table 2 show that the proposed framework DS^2-ABSA consistently outperforms existing data augmentation and pre-training methods by a considerable margin. For instance, the average F1 of DS^2-ABSA on Paraphrase exceeds the second-best FS-ABSA by 7.73% and 3.87% in the 2%-shot and 5%-shot settings, respectively. These findings underscore the superiority of our approach in low-resource scenarios. 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. By expanding training samples, data augmentation

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	$42.07_{+0.84}$
	MELM†	38.27	46.26	32.11	34.90	$37.89_{+13.75}$	42.86	57.39	39.76	51.04	$47.76_{+6.53}$
	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	$51.47_{+10.24}$
	DS^2-ABSA*	**47.30**	**60.39**	**49.49**	**59.40**	$\mathbf{54.15_{+30.01}}$	**47.97**	62.37	**49.26**	**58.40**	$\mathbf{54.50_{+13.27}}$
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	$53.91_{-0.48}$
	MELM†	48.57	57.11	41.84	48.74	$49.07_{+4.06}$	52.85	63.42	48.01	60.20	$56.12_{+1.73}$
	DAPT‡	48.03	56.56	44.56	47.84	$49.25_{+4.24}$	51.81	63.66	52.37	58.21	$56.51_{+2.13}$
	FS-ABSA‡	49.66	57.65	47.15	50.70	$51.29_{+6.28}$	52.02	63.12	53.30	61.52	$57.49_{+3.10}$
	DS^2-ABSA*	**56.86**	**64.92**	**53.15**	**61.16**	$\mathbf{59.02_{+14.01}}$	**60.83**	**68.41**	**54.32**	**61.87**	$\mathbf{61.36_{+6.97}}$
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	$59.81_{-0.97}$
	MELM†	56.36	64.24	52.22	59.81	$58.16_{+1.19}$	58.53	68.15	54.31	63.44	$61.11_{+0.33}$
	DAPT‡	55.34	64.90	55.39	60.46	$59.02_{+2.05}$	59.96	69.01	55.87	63.69	$62.13_{+1.35}$
	FS-ABSA‡	56.59	63.46	55.88	61.55	$59.37_{+2.40}$	**61.63**	68.68	55.83	66.24	$63.10_{+2.32}$
	DS^2-ABSA*	**58.15**	**69.65**	**59.78**	**63.89**	$\mathbf{62.87_{+5.90}}$	61.24	**71.94**	**60.56**	**68.81**	$\mathbf{65.64_{+4.86}}$

Table 2: Main results compared with traditional methods. Data augmentation, pre-training, and our data synthesis methods are marked with †, ‡, and *, respectively.

methods yield improvements in most situations. Nonetheless, due to the limited diversity, their benefits diminish as model scale and data volume increase, even leading to negative effects as observed with InstructABSA using EDA or CA in the 5%-shot setting.

(2) Pre-training approaches generally surpass data augmentation by leveraging extensive extra corpora that provide domain and sentiment knowledge. However, the distinct objectives between pre-training and downstream tasks limit their effectiveness. In contrast, our method generates high-quality task-specific data and achieves better results. Compared with SPT-ABSA, DS^2-ABSA surpasses by 3.03% in the 5-% shot setting without additional corpora and costly training, indicating both its significance and efficiency.

Comparison with LLM-based Methods. As presented in Table 3, we employ LLM-based approaches for comparison. We can conclude that:

(1) DS^2-ABSA outperforms other methods on almost all datasets, confirming its superiority. Specifically, in the 5%-shot setting, DS^2-ABSA(INST) using synthetic data from GPT-3.5 and Llama-3-8B surpasses their Supervised Fine-tuning counterparts by 1.81% and 2.99%, respectively, with lower hardware requirements. Additionally, DS^2-ABSA(PARA), using T5-base as the ABSA model, achieves competitive performance compared to Supervised Fine-tuning of LLMs in the 2%-shot setting. We attribute this to LLMs retaining robust general capabilities even after fine-tuning with few-shot data. By comparison, models trained on synthetic data are specifically optimized for ABSA, enabling them to achieve satisfactory results.

(2) Overall, utilizing Llama-3-8B or GPT-3.5 Turbo as the backbone yields similar results, indicating their comparable capabilities for synthesizing ABSA samples. Among all methods, In-context Learning shows the poorest performance due to the inability of LLMs to meet the structural requirements of ABSA. In contrast, fine-tuned LLMs align better with task-specific requirements, resulting in significant performance gains. Furthermore, the results reveal that Supervised Fine-tuning benefits more from an increased number of training samples compared to DS^2-ABSA, indicating its greater potential with abundant data.

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 re-

Backbone LLM	Method	2%-shot					5%-shot				
		Lap14	Res14	Res15	Res16	Avg	Lap14	Res14	Res15	Res16	Avg
GPT-3.5 Turbo	In-context Learning	38.26	59.35	47.84	52.43	49.71	38.21	59.71	46.30	55.21	49.86
	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	In-context Learning	41.92	59.00	45.30	55.00	50.31	42.54	59.57	46.42	55.68	51.05
	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 approaches. For our method, synthetic data generation using GPT-3.5 Turbo and Llama-3-8B 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 study on Paraphrase w/ DS^2-ABSA.

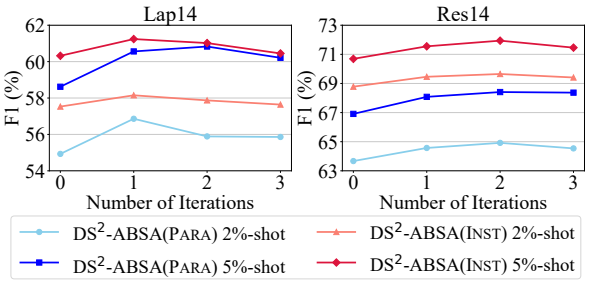

Figure 2: Effect of noisy self-training over iterations. Iteration 0 means noisy self-training is not conducted.

moving 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 on model performance, 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 model performance 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 because the teacher model may overfit the training data after multiple iterations, thereby losing some of the aspect-sentiment knowledge provided by LLMs in synthesized data.

4.6 Exploring Diversity and Label Quality

Analysis of Data Diversity. We conduct the visualization of semantic representations to explore the diversity of dual-stream synthetic data, limiting the display to a maximum of 5000 samples for clarity. As shown in Figure 3, traditional data augmentation methods generate sentences that are highly similar to the original data, exhibiting poor diversity. 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. In contrast, the key-point-driven synthesis produces data with a broader and more evenly distributed range of sentence representations, indicating richer semantic diversity. Besides, data synthesized through dual-stream methods show comple-

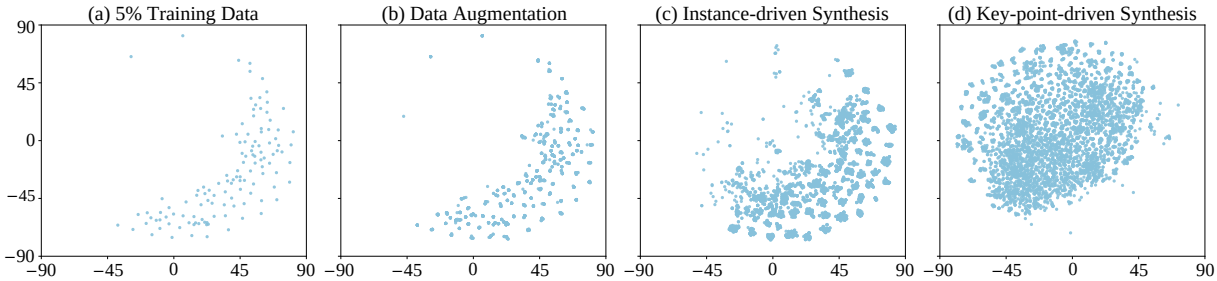

Figure 3: Data diversity comparison in the 5%-shot setting of Res14, including (a) few-shot data; (b) data augmentation (EDA, CA, MELM); (c) instance-driven synthesis; and (d) key-point-driven synthesis. We use instructor (Su et al., 2023) for embedding and t-SNE for visualization. See results of Lap14 in Appendix B.2.

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.

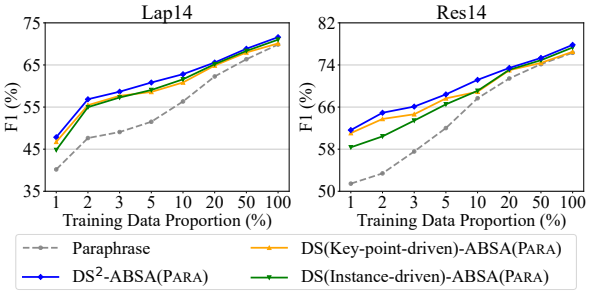

Figure 4: Effect of dual-stream synthesis methods using different training data proportions.

mentary patterns, further validating the significance of the proposed dual-stream 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 Effect of Training Data Sizes

To investigate the effectiveness of dual-stream synthesis across varying data sizes, we analyze the performance of DS^2-ABSA and each stream of synthetic data under different training data proportions, 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.

5 Conclusion

In this paper, we introduce DS^2-ABSA, a dual-stream data synthesis framework tailored for few-shot ABSA. By employing LLMs in both key-point-driven and instance-driven strategies, our framework generates diverse and well-aligned task-specific data that overcomes the limitations of data augmentation and in-context learning. Additionally, the incorporation of the label refinement module further enhances label quality, contributing to improved overall performance. Extensive experiments demonstrate that DS^2-ABSA significantly outperforms traditional and LLM-based methods, offering a promising solution for ABSA in low-resource scenarios. Furthermore, we believe that our method can serve as a valuable reference for applications in other fields.

Limitations

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

- For the ABSA task, annotation guidelines and specific examples for aspect term extraction and sentiment polarity identification are available, which could help improve the synthesis process. However, these have not been utilized in our current approach.
- Although the label refinement module improves label quality, it may not fully eliminate inaccuracies arising from inherent biases in LLMs. Such biases can impair downstream performance, as residual errors in the synthesized data may lead to sub-optimal model training and predictions.

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

References

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*.

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, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428.

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.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, et al. 2024. Synthetic data (almost) from scratch: Generalized instruction tuning for language models. *arXiv preprint arXiv:2402.13064*.

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.

Zhengyan Li, Yicheng Zou, Chong Zhang, Qi Zhang, and Zhongyu Wei. 2021. Learning implicit sentiment in aspect-based sentiment analysis with supervised contrastive pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 246–256.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, et al. 2024. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*.

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.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 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, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 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.

Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. 2024. Curated LLM: Synergy of LLMs and data curation for tabular augmentation in low-data regimes. In *Forty-first International Conference on Machine Learning*.

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, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121.

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. Association for Computational Linguistics.

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, Zhiyuan Wen, Qin Zhao, Min Yang, and Ruifeng Xu. 2021. Progressive self-training with discriminator for aspect term extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 257–268.

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, Swaroop Mishra, Pegah Alipoormolabashi, 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. 2023. 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.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for 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 (Volume 1: Long Papers)*, pages 2416–2429.

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*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Junlei Zhang, Zhenzhong Lan, and Junxian He. 2023a. Contrastive learning of sentence embeddings from scratch. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3932.

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. 2024. 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, Yifan Yang, Bin Liang, Shiwei Chen, Bing Qin, and Ruifeng Xu. 2023b. 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*.

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.

Senbin Zhu, Hanjie Zhao, Wxr Wxr, 18437919080@163.com 18437919080@163.com, 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;
- Descriptions of the baseline methods and more implementation details 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 8 and 9, respectively. For instance-driven synthesis, prompts for sample combination and selective reconstruction strategies are both presented in Table 10. Additionally, for LLM-based methods, including in-context learning and supervised fine-tuning, the utilized prompts are presented in Table 11.

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 6.

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 6: 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.

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

LLM-based Methods. (1) **In-context Learning** (Wang et al., 2024c): Retrieves in-context demonstrations for ABSA based on semantic similarity, syntactic relevance, and aspect-sentiment semantics. (2) **Supervised Fine-tuning** (Simmering and Huoviala, 2023): Fine-tunes LLMs on labeled ABSA datasets using task-specific instructions.

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 7. 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 7: Full-shot performance of ABSA models.

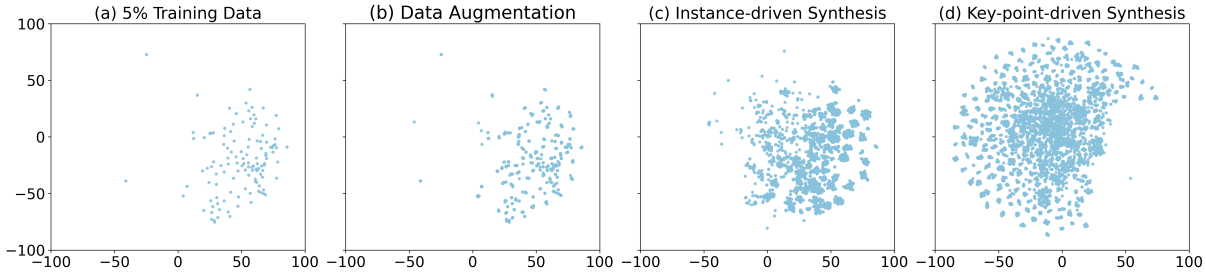

Figure 5: Data diversity comparison in the 5%-shot setting of Lap14, including (a) few-shot data; (b) data augmentation (EDA, CA, MELM); (c) instance-driven synthesis; and (d) key-point-driven synthesis.

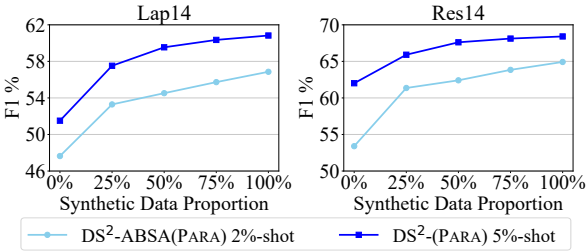

Figure 6: Impact of the number of synthetic data.

C.2 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 6, model performance gradually improves as the amount of synthetic data increases. Additionally, we observe that in the 2%-shot setting, changes in the quantity of synthetic data yield greater gains, with the growth rate not showing signs of decline even at 100% synthetic data. In contrast, in the 5%-shot setting, the growth rate gradually slows down. This indicates that when gold data is more scarce, increasing the amount of synthetic data has a greater impact on performance.

C.3 Data Diversity Results of Lap14

The results are illustrated in Figure 5. We find that the diversity of different methods on Lap14 is similar to that on Res14, where key-point-driven synthesis exhibits broader diversity while semantically complementing instance-driven synthesis.

C.4 Examples of Synthetic Data from LLMs

To enhance clarity, Table 12 and 13 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.

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 8: 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 9: 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 10: 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 11: 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 12: 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 13: Examples of instance-driven synthetic data from LLMs.