# Iterative Data Augmentation with Large Language Models for Aspect-based Sentiment Analysis

**Anonymous ACL submission**

## Abstract

Aspect-based Sentiment Analysis (ABSA) is an important sentiment analysis task, which aims to determine the sentiment polarity towards an aspect in a sentence. Due to the expensive and limited labeled data, data augmentation (DA) has become the standard for improving the performance of ABSA. However, current DA methods usually have some shortcomings: 1) poor fluency and coherence, 2) lack of diversity of generated data, and 3) reliance on some existing labeled data, hindering its applications in real-world scenarios. In response to these problems, we propose a systematic **Iter**ative **D**ata augmentation framework, namely **IterD**, to boost the performance of ABSA. The core of IterD is to leverage the powerful ability of large language models (LLMs) to iteratively generate more fluent and diverse synthetic labeled data, starting from an unsupervised sentence corpus. Extensive experiments on 4 widely-used ABSA benchmarks show that IterD brings consistent and significant performance gains among 5 baseline ABSA models. More encouragingly, the synthetic data generated by IterD can achieve comparable or even better performance against the manually annotated data.

## 1 Introduction

Aspect-based Sentiment Analysis (ABSA), which aims to determine the sentiment polarity towards an aspect in a sentence, is an important find-grained task in sentiment analysis (Liu and Zhang, 2012; Schouten and Frasincar, 2015). With the advancements of pretrained language models (PLMs), *e.g.*, BERT (Devlin et al., 2019) and its variants (Liu et al., 2019; He et al., 2020), numerous PLM-based ABSA models have been proposed and achieved promising results (Wang et al., 2020; Zhong et al., 2022). However, these methods usually require large-scale labeled data with fine-grained annotation, which is time-consuming and expensive to obtain for many emerging scenarios (Yu et al., 2023).

To alleviate this issue, a common approach is data augmentation (DA), which aims to enrich the training data and can be generally divided into two categories: word-level (Wei and Zou, 2019; Wu et al., 2019) and sentence-level DA (Sennrich et al., 2016; Wang et al., 2022). Specifically, word-level DA methods involve replacing or inserting words into sentences, leveraging techniques such as word synonym dictionaries (Wei and Zou, 2019) or contextual word embeddings (Wu et al., 2019). Conversely, sentence-level DA methods focus on generating new sentences using paraphrasing methods (Guo et al., 2019), generative models (Wang et al., 2022), or machine translation (Sennrich et al., 2016) techniques. These methods aim to introduce linguistic variations, while keeping the aspect and its sentiment polarity unchanged.

Despite achieving remarkable performance, we find that the aforementioned DA methods still have some limitations: 1) *Poor fluency and coherence*, as the word-level DA methods might distort the sentence meaning or structures, and current sentence-level DA methods usually struggle to generate fluent and coherent sentences (Yu et al., 2023). 2) *Lack of the diversity of generated data*, as most of the prior DA methods do not reconstruct the structure of original sentence, limiting the diversity of generated sentences. 3) *Reliance on some existing labeled data*, as these DA methods generally start from a set of existing labeled data, which could be unavailable in real-world scenarios, especially in emerging domains. Intuitively, the currently popular large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023) have the great potential to deal with the above issues of DA methods, as they can generate fluent and high-quality text following the human instructions (Wei et al., 2021). Hence, there raises a question: *whether we can leverage the powerful ability of LLMs for better augmenting the ABSA data*?

Motivated by this, we propose a novel **Iter**ative

Data augmentation approach, namely **IterD**, which aims to generate fluent and diverse ABSA training data. The core of IterD is to leverage the ability of LLMs to automatically generate the high-quality labeled ABSA data from the easy-to-obtain unsupervised sentence corpus. Specifically, given an unlabeled sentence corpus, IterD ❶ first extracts the aspect terms and expands them into a candidate aspect set. Then, IterD ❷ introduces an iterative generation module to automatically obtain the fluent ABSA data based on the aspect set. Lastly, to ensure the quality and diversity of the generated data, IterD ❸ designs a discriminator to filter the low-quality samples. The generation processes of IterD are systemic and do not rely on much existing ABSA data or human effort That is, our IterD can be easily applied in real-world scenarios.

We evaluate our IterD on a variety of widely-used ABSA benchmarks, including Laptop14, Restaurant14 (Pontiki et al., 2014), Restaurant15 (Pontiki et al., 2015) and Restaurant16 (Pontiki et al., 2016), and the results show that: 1) our IterD brings consistent and significant performance gains among 5 baseline ABSA models; 2) without relying on any labeled data, IterD can achieve comparable performance to that training with full labeled data; 3) IterD outperforms the other DA counterparts by a clear margin. More in-depth analyses delve into the mechanism of IterD, and reveal when and where to use it. To summarize, our contributions are three-fold: (1) We propose a novel iterative DA approach (IterD) for ABSA by leveraging the powerful ability of LLMs. (2) IterD is plug-and-play and can be easily applied in real-world scenarios. (3) Extensive results on 4 widely-used ABSA benchmarks show the effectiveness and superiority of IterD.

## 2 Related Work

**Aspect-based Sentiment Analysis**  ABSA has been extensively studied in the last decade (Liu and Zhang, 2012; Schouten and Frasincar, 2015). With the advancements of PLMs, a large amount of PLM-based ABSA models have emerged (He et al., 2019; Luo et al., 2019; Xu et al., 2018; He et al., 2018; Chen and Qian, 2020; Zhao and Yu, 2021), which involve designing the network structure or injecting external knowledge in different ways. These methods have achieved promising performance on several widely-used ABSA benchmarks (Pontiki et al., 2014, 2015; Jiang et al., 2019).

However, most of them highly rely on numerous labeled data, which is expensive to obtain in some scenarios (Yu et al., 2023).

**Data Augmentation for ABSA.**  To alleviate the above issue, a common approach is Data Augmentation (DA), which enlarges the training dataset by changing the original data or generating new data through various methods. In the context of ABSA, numerous DA methods have been proposed (Wei and Zou, 2019; Sennrich et al., 2016; Wang et al., 2022) and achieved remarkable performance. However, since most of them attempt to augment the data by simply modifying the sentence structure or using pretrained models for text infilling, they have some shortcomings, *e.g.*, poor fluency and lack of diversity. Moreover, current DA methods usually rely on some existing labeled data, and might not be able to expand to real-world scenarios, in which the labeled data is unavailable. To this end, we propose a new DA method, which is more effective and applicable, for alleviating the issue of data scarcity in ABSA.

**Large Language Model**  Recently, we have witnessed the great success of large language models (LLMs) (Ouyang et al., 2022; Touvron et al., 2023; Anil et al., 2023) in many downstream NLP tasks. Owing to the instruction-tuning approach (Wei et al., 2021), LLMs can generate fluent and high-quality contexts following the human's instruction. Unfortunately, in the context of ABSA, directly using LLMs is not an optimal choice. Prior empirical studies (Zhong et al., 2023) show that LLMs might under-perform the traditional BERT (Devlin et al., 2019) models in some fine-grained language understanding tasks, *e.g.*, ABSA. Thus, employing BERT-style PLMs is still a viable option for ABSA. Alternatively, in this paper, we attempt to take advantage of LLMs' instruction-following and in-context learning abilities and enforce them to generate more high-quality data for boosting the performance of existing ABSA models.

## 3 Methodology

In this section, we first briefly review the ABSA task and then present the details of our IterD, which contains three-stage processes: ❶ *Aspect Extraction and Extension*, ❷ *Pseudo Data Generation* and ❸ *Evaluating and Filtering*. The framework of IterD is illustrated in Figure 1.
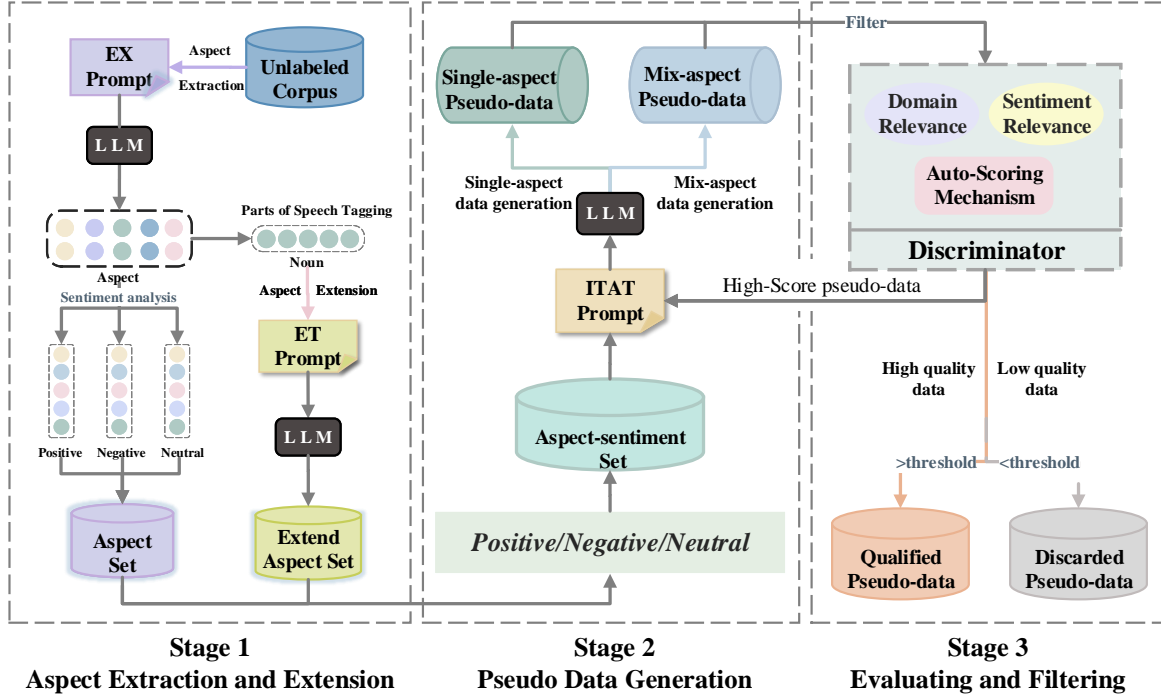
Figure 1: **Overview of our IterD framework**, covering three-stage processes: ❶ *Aspect Extraction and Extension*, ❷ *Pseudo Data Generation* and ❸ *Evaluating and Filtering*. Notably, "EX Prompt" and "ET Prompt" denote the aspect extraction and extension prompts, respectively. "ITAT Prompt" refers to the Iteration Teaching Analysis Prompt, which enforces the LLM to generate more diverse data. More detailed prompts can be found in Appendix A.4.

## 3.1 Problem Formulation

Given a sentence-aspect pair $\{S, T\}$, the goal of ABSA is to predict the sentiment polarity $y \in \{0, 1, 2\}$ of the sentence $S$ towards the aspect $T$, where 0, 1, and 2 denote the *positive*, *neutral* and *negative* polarities, respectively. Note that $T$ is the subsequence of $S$. As mentioned in §1, there are usually limited labeled sentence-aspect pairs. Thus, we aim to generate the synthetic dataset $\mathcal{G} = \{(S_i, T_i) | i > i\}$ from an unsupervised text corpus $U = \{S_1, S_2, S_3, ..., S_n\}$ with $n$ sentences.

## 3.2 Iterative Data Augmentation

**Aspect Extraction and Extension.** Starting from an unsupervised corpus $U$, we first attempt to extract the aspects relevant to a specific domain. Specifically, we carefully design an aspect extraction (denoted as "EX") prompt[1] to enforce the LLM to automatically extract domain-related aspects for each sentence $S_i \in U$. After doing that, we deduplicate the aspects and obtain the initial aspect set $A$. Considering that aspects are generally nouns and their variants, we perform the part-of-speech

processing with a Python library Textblob[2] on all candidate aspects of $A$ to remove those that are difficult to accurately generate the samples. Then, to further improve the diversity of extracted aspects, we introduce an aspect extension module to expand $A$. In particular, for the Noun aspects in $A$, we enforce the LLM to expand them with their homonyms and synonyms by an aspect extension (denoted as "ET") prompt. Lastly, the extend aspect set is merged into $A$. Moreover, for better generating the sentiment-aware data, we split the $A$ into three sub-sets with different sentiment polarities, *i.e.*, positive aspects $A_{pos}$, negative aspects $A_{neg}$, neutral aspects $A_{neu}$, by performing a word sentiment analysis on each aspect.

**Pseudo Data Generation.** After obtaining the domain-related aspects, we then generate the pseudo labeled data, *i.e.*, triplet $\{S_i, T_i, y_i\}$. Specifically, for each aspect sub-set, we append the aspects with their corresponding sentiment polarities to construct the aspect-sentiment set. For instance, for the aspect in $A_{pos}$, we append it with the positive polarity. Consequently, we can basically design a prompt to guide the data generation

---

[1] Due to the space limitations, we present the detailed prompts of IterD in Appendix A.4.
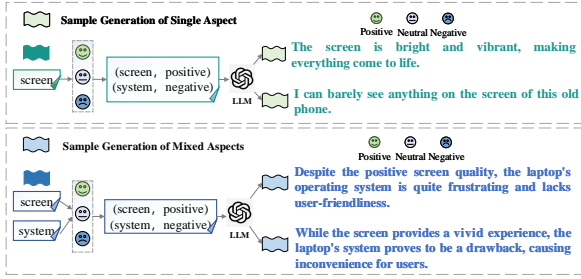
[2] https://pypi.org/project/textblob/

Figure 2: **Illustration of single-aspect/mix-aspect data generation**. For ease of illustration, we only show some cases in the laptop domain.



Figure 3: **Illustration of the discriminator.**

of LLMs based on the aspect-sentiment set. However, during the preliminary experiments, we found that as the generation of LLMs continued, LLMs suffer from the problem of repetitive generation, *i.e.*, the generated samples tend to be similar and low-diversity. Hence, we propose a more powerful Iteration Teaching Analysis Prompt (denoted as "ITAT"), which randomly selects samples from each round of generating samples as feedback to guide the next-round generation. By doing so, ITAT can prompt the LLMs to generate more richer and diverse pseudo triplet data.

Inspired by prior studies (Wang et al., 2022), we recognize that multi-aspect data, *i.e.*, data with multiple aspects in a sentence, is greatly beneficial to the training of ABSA models. To this end, in addition to the vanilla single-aspect pseudo data generation, we further utilize a mix-aspect pseudo data generation branch to obtain the more complex yet effective multi-aspect data. To have a close look, we provide the illustrations of single-aspect/mix-aspect pseudo data generation in Figure 2.

**Evaluating And Filtering.** Despite the powerful capability of LLMs, they might generate unexpectedly low-quality data, hindering the performance of DA. Thus, it is critical to evaluate the quality of generated data and filter the lower-quality one. To achieve this goal, we introduce a new discriminator, as illustrated in Figure 3, containing a judgment module and an auto-scoring mechanism. Specifically, in the judgment module, we employ the popular **LLM-as-a-Judge** method to enforce the LLM to determine the domain relevance and sentiment relevance of generated data. That is, LLM is utilized to verify whether the generated data is relevant to the given domain and sentiment. After filtering the data with lower domain relevance and sentiment relevance, we further use the auto-scoring mechanism to quantitatively measure the
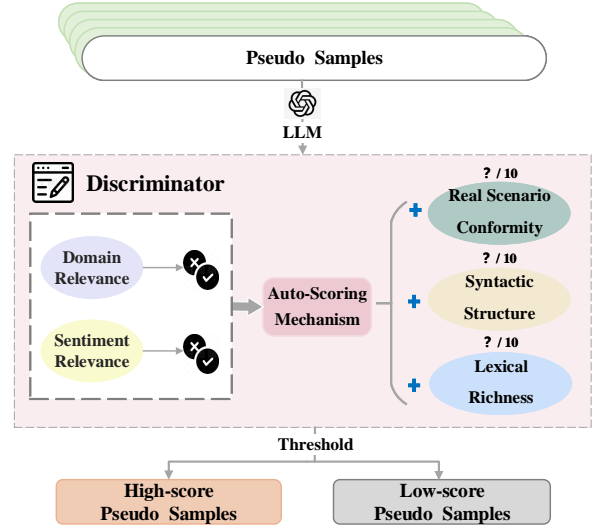
data quality, in terms of Syntactic Structure, Lexical Richness, and Real Scenario Conformity. The scoring mechanism takes a sample judgment on a scale of 1-10, where larger scores mean higher data quality. For filtering the low-quality data, we set a filtering threshold[3] $\mathcal{T}$. The data exceeding the threshold is used as final training data, while the others are discarded. Notably, for promoting the aforementioned ITAT strategies, we use the high-quality generated data as the feedback.

## 4 Experiments

### 4.1 Setup

**Task and Dataset.** We conduct the main experiments on 4 widely-used ABSA benchmarks, *i.e.*, Laptop14, Restaurant14 (Pontiki et al., 2014), Restaurant15 (Pontiki et al., 2015) and Restaurant16 (Pontiki et al., 2016). Following Tang et al. (2019), we remove a few instances with conflicting sentiment polarity. For the evaluation of aspects extracted by our IterD, we use "Precision" (P), "Recall" (R) and "Macro-F1" ($F_1$) as the metrics, while the "Accuracy" (Acc) and $F_1$ score are used to evaluate the final ABSA models. The details of all used datasets can be found in Appendix A.1.

**Implementation.** For simulating the real-world scenarios, we use the unlabeled sentences in training sets of the above ABSA benchmarks (*i.e.*, ignoring the aspect and polarity information) as the initial unsupervised corpus for our IterD. The aspects of the original training sets are used as gold

---

[3]The analysis of $\mathcal{T}$ can be found in §4.3

| Model | Dataset | Laptop14 | | Restaurant14 | | Restaurant15 | | Restaurant16 | |
|---|---|---|---|---|---|---|---|---|---|
| | | Acc | $F_1$ | Acc | $F_1$ | Acc | $F_1$ | Acc | $F_1$ |
| ATAE-LSTM | Original data | 79.50 | 75.50 | 83.42 | 75.03 | 83.39 | 68.59 | 91.41 | 77.08 |
| | Generated data | $79.22_{\downarrow0.29}$ | $75.64_{\uparrow0.14}$ | $80.36_{\downarrow3.06}$ | $70.52_{\downarrow4.51}$ | $83.27_{\downarrow0.12}$ | $70.42_{\uparrow1.83}$ | $89.22_{\downarrow2.19}$ | $76.89_{\downarrow0.19}$ |
| | Mixed data | $\mathbf{80.94}_{\uparrow1.44}$ | $\mathbf{77.54}_{\uparrow2.04}$ | $\mathbf{84.91}_{\uparrow1.49}$ | $\mathbf{77.88}_{\uparrow2.85}$ | $\mathbf{84.01}_{\uparrow0.74}$ | $\mathbf{71.43}_{\uparrow2.84}$ | $\mathbf{91.67}_{\uparrow0.26}$ | $\mathbf{79.25}_{\uparrow2.17}$ |
| ASGCN | Original data | 80.94 | 77.80 | 86.37 | 80.13 | 85.04 | 70.75 | 92.22 | 78.42 |
| | Generated data | $80.62_{\downarrow0.32}$ | $77.71_{\downarrow0.09}$ | $82.95_{\downarrow3.42}$ | $74.61_{\downarrow5.52}$ | $85.48_{\uparrow0.44}$ | $72.47_{\uparrow1.72}$ | $89.74_{\downarrow2.48}$ | $77.27_{\downarrow1.15}$ |
| | Mixed data | $\mathbf{82.03}_{\uparrow1.09}$ | $\mathbf{79.17}_{\uparrow1.37}$ | $\mathbf{87.23}_{\uparrow0.86}$ | $\mathbf{81.45}_{\uparrow1.32}$ | $\mathbf{86.21}_{\uparrow1.17}$ | $\mathbf{74.55}_{\uparrow3.80}$ | $\mathbf{93.11}_{\uparrow0.89}$ | $\mathbf{82.43}_{\uparrow4.01}$ |
| BERT-SPC | Original data | 78.68 | 74.82 | 84.82 | 78.08 | 83.95 | 69.91 | 90.42 | 76.61 |
| | Generated data | $77.02_{\downarrow1.66}$ | $73.97_{\downarrow0.85}$ | $85.24_{\uparrow0.42}$ | $72.34_{\downarrow5.74}$ | $83.39_{\downarrow0.57}$ | $69.70_{\downarrow0.21}$ | $88.66_{\downarrow1.76}$ | $72.75_{\downarrow3.86}$ |
| | Mixed data | $\mathbf{80.09}_{\uparrow1.41}$ | $\mathbf{77.13}_{\uparrow2.31}$ | $\mathbf{85.62}_{\uparrow0.80}$ | $\mathbf{78.45}_{\uparrow0.37}$ | $\mathbf{85.24}_{\uparrow1.29}$ | $\mathbf{70.65}_{\uparrow0.74}$ | $\mathbf{90.75}_{\uparrow0.33}$ | $\mathbf{77.37}_{\uparrow0.76}$ |
| R-GAT | Original data | 78.37 | 73.92 | 86.34 | 80.74 | 83.58 | 71.48 | 91.72 | 77.77 |
| | Generated data | $78.58_{\uparrow0.21}$ | $75.67_{\uparrow1.75}$ | $81.79_{\downarrow4.55}$ | $74.60_{\downarrow6.14}$ | $84.32_{\uparrow0.74}$ | $69.14_{\downarrow2.34}$ | $88.96_{\downarrow2.76}$ | $75.64_{\downarrow2.13}$ |
| | Mixed data | $\mathbf{80.56}_{\uparrow2.19}$ | $\mathbf{77.08}_{\uparrow3.16}$ | $\mathbf{87.50}_{\uparrow1.16}$ | $\mathbf{82.04}_{\uparrow1.30}$ | $\mathbf{85.06}_{\uparrow1.48}$ | $\mathbf{73.36}_{\uparrow2.28}$ | $\mathbf{92.05}_{\uparrow0.33}$ | $\mathbf{78.80}_{\uparrow1.03}$ |
| KGAN | Original data | 82.34 | 79.17 | 86.55 | 81.47 | 86.40 | 73.89 | 92.81 | 81.17 |
| | Generated data | $80.47_{\downarrow1.87}$ | $76.83_{\downarrow2.34}$ | $81.70_{\downarrow4.85}$ | $74.11_{\downarrow0.19}$ | $85.11_{\downarrow7.36}$ | $72.11_{\downarrow1.29}$ | $89.22_{\downarrow3.59}$ | $77.71_{\downarrow3.46}$ |
| | Mixed data | $\mathbf{82.49}_{\uparrow0.15}$ | $\mathbf{79.62}_{\uparrow0.45}$ | $\mathbf{87.50}_{\uparrow0.95}$ | $\mathbf{81.86}_{\uparrow0.39}$ | $\mathbf{87.13}_{\uparrow0.73}$ | $\mathbf{75.17}_{\uparrow1.28}$ | $\mathbf{92.95}_{\uparrow0.14}$ | $\mathbf{82.83}_{\uparrow1.66}$ |

Table 1: **Results of our IterD method on various baseline ABSA models**. Notably, "Original data" and "Generated data" denote that we train the models on the original ground-truth training data and our generated data, respectively. "Mixed data" means that we train on the mix of original and generated training data.

| Method | Metric | Laptop14 | Rest14 | Rest15 | Rest16 |
|---|---|---|---|---|---|
| Zero-Shot | P | 36.04 | 44.24 | 44.38 | 40.2 |
| | R | 69.27 | 65.65 | 72.82 | 65.04 |
| | $F_1$ | 47.41 | 52.86 | 55.15 | 49.69 |
| Few-shot | P | 46.79 | 59.85 | 60.34 | 57.31 |
| | R | 73.12 | 70.04 | 72.82 | 73.19 |
| | $F_1$ | 57.07 | 64.55 | 65.99 | 64.28 |
| Few-shot* | P | 45.72 | 48.00 | 50.25 | 46.36 |
| | R | **79.77** | **79.84** | **82.15** | **80.30** |
| | $F_1$ | 58.13 | 59.95 | 62.36 | 58.79 |

Table 2: **Evaluation on aspects extracted by IterD with different strategies.** Notably, "Zero-shot" refers to the aspects extracted in a zero-shot manner, "Few-shot" refers to few-shot extraction using domain-related demonstrations, and "Few-shot*" refers to the few-shot extraction using random demonstrations.

labels to evaluate our extracted aspects. After obtaining the augmented ABSA data, we train the models with these data and evaluate them on the test sets of the above benchmarks. Specifically, we use the powerful GPT-3.5-turbo [4] as the LLM in our IterD. For each benchmark, we enforce IterD to generate the ABSA data, the number of which is similar to that of the original training set.

**Baseline Models.** To investigate the effectiveness of our IterD, we mainly apply it to improve 5 representative baseline ABSA models, *i.e.*, ATAE-LSTM (Wang et al., 2016), ASGCN (Zhang et al., 2019), and BERT-SPC (Song et al., 2019), R-GAT (Wang et al., 2020) and KGAN (Zhong et al., 2022). For each model, we utilize the BERT-base-uncased [5] as the backbone and train it following the default settings in the original papers. Due to the space limitations, we present the details of all baseline models in Appendix A.2.

**Compared Methods.** We conduct the main results in 3 different settings, *i.e.*, 1) "Original data": training the ABSA models with the original labeled ABSA data, 2) "Generated data": training with only the synthetic data generated by our IterD and 3) "Mixed data": training with the mix of original data and our generated data. We additionally compare IterD with several cutting-edge DA methods, including Back-Translation (BT) (Sennrich et al., 2016), EDA (Wei and Zou, 2019), CBERT (Wu et al., 2019) and C3DA (Wang et al., 2022). The detailed descriptions of these compared DA methods can be found in Appendix A.3.

## 4.2 Main Results

### 4.2.1 Aspect Extraction Results

In our IterD, the performance of final ABSA models highly relies on the relevance between ex-

| Method | Laptop14 | | Restaurant14 | |
|---|---|---|---|---|
| | Acc | $F_1$ | Acc | $F_1$ |
| R-GAT | 78.37 | 73.92 | 86.34 | 80.74 |
| +BT | 79.70 | 75.01 | 86.85 | 81.02 |
| +EDA | 78.59 | 74.82 | 86.52 | 81.47 |
| +CBERT | 78.62 | 74.96 | 87.01 | 82.19 |
| +C3DA | 79.16 | 75.40 | 87.22 | **82.69** |
| **+IterD (Ours)** | **80.25** | **76.18** | **87.50** | 82.04 |

Table 3: **Comparison of different DA methods.**

| Model | Method | Acc | $F_1$ |
|---|---|---|---|
| ASGCN | IterD (Ours) | **82.03** | **79.17** |
| | -w/o Extension | 81.56 | 78.96 |
| | $\Delta(\downarrow)$ | ↓ 0.47 | ↓ 0.21 |
| R-GAT | IterD (Ours) | **80.56** | **77.08** |
| | -w/o Extension | 80.25 | 76.18 |
| | $\Delta(\downarrow)$ | ↓ 0.31 | ↓ 0.90 |

Table 4: **Ablation study of aspect extension module in IterD**. "-w/o Extension" means that we do not extend the aspect set in IterD. Laptop14 is used for evaluation.

| Method | ASGCN | | R-GAT | |
|---|---|---|---|---|
| | Acc | $F_1$ | Acc | $F_1$ |
| Single-aspect | 76.09 | 72.42 | 72.88 | 68.71 |
| +Mix-aspect | 79.53$_{\uparrow 3.44}$ | 76.33$_{\uparrow 3.91}$ | 75.71$_{\uparrow 2.83}$ | 72.79$_{\uparrow 4.08}$ |
| +Multi_Neu | **80.62**$_{\uparrow 4.53}$ | **77.71**$_{\uparrow 5.29}$ | **78.58**$_{\uparrow 5.70}$ | **75.67**$_{\uparrow 6.96}$ |

Table 5: **Analysis of different generation strategies**. "Single-aspect" denotes that we only generate the samples with a single aspect in a sentence, and "Mix-aspect" means that there are multiple aspects in a generated sentence. "Multi_Neu" refers to the samples that have multiple aspects with neutral polarity in a sentence. Here, we report the results on the Laptop14 benchmark.

tracted aspects and gold aspects. Here, to verify whether IterD can extract the relevant aspects, we evaluate the aspects extracted by different strategies ("Zero-shot", "Few-shot" and "Few-shot*") of IterD and report the contrastive results in Table 2. As seen, given some examples, IterD can extract more relevant aspects, indicating the superiority of few-shot learning. Interestingly, compared to the domain-related demonstrations, IterD with random demonstrations performs better. We conjecture that domain-related demonstrations might be too similar and hinder the diversity of extracted aspects, thus leading to sub-optimal performance. Notably, "Few-shot*" performs best, and we thus use it as the default setting in the following content.

### 4.2.2 Evaluation on the Generated Data

In this part, we perform the evaluation of the synthetic data generated by IterD. The contrastive results are presented in Table 1 and 3, from which we observe that:

**Models trained on the generated data partially outperforms those trained on the ground-truth data.** As seen, training with only the generated data achieves remarkable or even better performance than on the ground-truth data, *e.g.*, +1.75% $F_1$ score of R-GAT in the Laptop14. These results show that IterD can generate high-quality labeled ABSA data, similar to the manually annotated data.

**IterD brings consistent and significant performance gains among all baseline models and tasks.** By combining the ground-truth data with our generated data, we find that there are consistent and significant performance gains among all settings, up to +4.01% $F_1$ score. This indicates the effectiveness of our IterD.

**IterD outperforms the other DA counterparts by a clear margin.** In Table 3, we compare our method with the other DA counterparts on the R-GAT model. As seen, IterD performs better than

the others in most settings. It is also noteworthy that the other DA methods commonly rely on the full original data, but IterD only requires the unlabeled sentence corpus, which is more flexible and suitable for real-world scenarios.

### 4.3 Ablation Study

We evaluate the impact of each component of our IterD, including 1) aspect extension, 2) sample generation strategies, 3) discriminator for filtering the low-quality data, and 4) filtering threshold $\mathcal{T}$.

**Impact of aspect extension.** As mentioned in §3, we expand the aspect set to improve its diversity. Here, to verify its effectiveness, we compare IterD with a simple alternative, "-w/o Extension", *i.e.*, removing the aspect extension module. The contrastive results are shown in Table 4. It can be seen that removing the aspect extension causes clear performance degradation, indicating the effectiveness of aspect extension.

**Impact of different sample generation strategies.** In the sample generation phase of IterD, we use two different strategies, *i.e.*, single-aspect and mix-aspect generation. Specifically, the latter strategy is to simulate the multi-aspect problem (Wang et al., 2022) in ABSA. Notably, for a fair comparison, we generate the same number of training data for
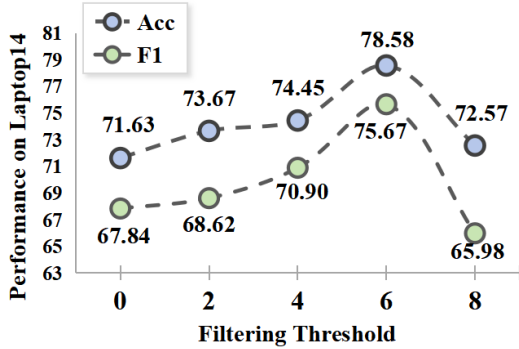
6

Figure 4: **Parameter analysis of filtering threshold** $\mathcal{T}$. We report the results of R-GAT training with the synthetic data generated by IterD with different $\mathcal{T}$.

| Model | Method | Acc | $F_1$ |
|---|---|---|---|
| ATAE-LSTM | Vanilla IterD | 76.06 | 72.80 |
| | +Discriminator | **79.22** | **75.64** |
| | $\Delta(\uparrow)$ | ↑ **3.16** | ↑ **2.84** |
| ASGCN | Vanilla IterD | 74.84 | 70.99 |
| | +Discriminator | **80.62** | **77.71** |
| | $\Delta(\uparrow)$ | ↑ **5.78** | ↑ **6.72** |
| KGAN | Vanilla IterD | 76.24 | 73.55 |
| | +Discriminator | **80.47** | **76.83** |
| | $\Delta(\uparrow)$ | ↑ **4.23** | ↑ **3.28** |

Table 6: **Ablation study of discriminator in IterD**. "Vanilla IterD" means that we directly use the generated data without filtering as final training data. Here, we report the results on the Laptop14 benchmark.

both strategies and present the compared results in Table 5. As seen, by generating more multi-aspect data, IterD brings consistent and significant performance gains against the vanilla single-aspect data. This is similar to the findings of Wang et al. (2022), as training on multi-aspect data can encourage the models to extract more fine-grained aspect-specific information, thus leading to better performance.

Moreover, in the preliminary experiments, we empirically found that IterD falls short in generating single-aspect data with neutral sentiment polarity. However, IterD can effectively generate the correct data of multiple aspects with the same neutral polarity during multi-aspect data generation. One possible reason is that LLMs struggle to distinguish neutral emotions, which is also found by prior empirical studies (Zhong et al., 2023). In Table, we further report the results of adding the corrected neutral multi-aspect data, denoted as "Multi_Neu". Obviously, these neural training data can further boost the ABSA performance effectively.
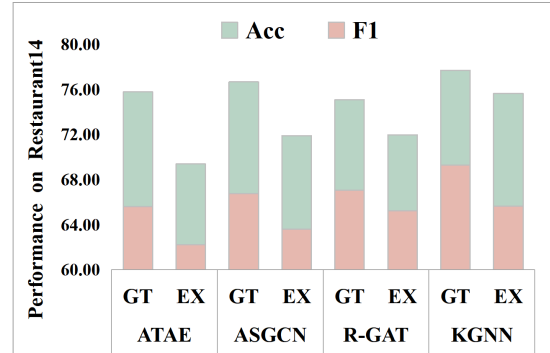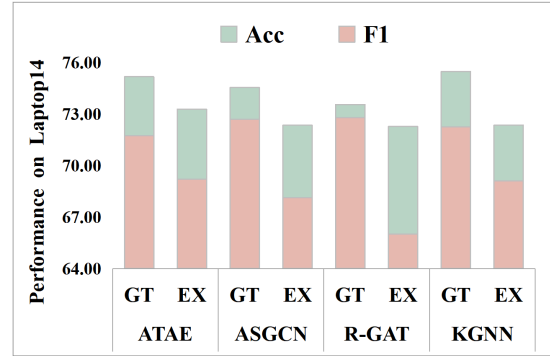




Figure 5: **Impact of accuracy of extracted aspects.** We replace the extracted aspects with gold ones in IterD and verify whether gold aspects can lead to better performance. "GT" and "EX" denote the gold and extracted aspects, respectively.

**Impact of discriminator.** In our IterD, we introduce a discriminator to filter the low-quality generated data. Here, we verify its effectiveness and report the contrastive results in Table 6. Compared to vanilla IterD, *i.e.*, directly using the generated data without filtering, IterD with the discriminator achieves much better performance. This highlights the importance of filtering the low-quality data, and indicates that data quality is more important than the data quantity for the field of ABSA.

**Parameter analysis on $\mathcal{T}$.** The $\mathcal{T}$, which is used to control the threshold for filtering data, is an important hyper-parameter in IterD. Here, we analyze its influence by evaluating the performance with different $\mathcal{T}$, spanning $\{0, 2, 4, 6, 8\}$. Figure 4 illustrates the contrastive results of R-GAT on Laptop14. With the increasing of $\mathcal{T}$ in a certain range (*i.e.*, 0 to 6), IterD continues achieving better performance. This indicates that filtering low-quality data is beneficial. Conversely, too large $\mathcal{T}$ values (*e.g.*, 8) lead to performance degradation, as filtering too much data might lead to limited available data for training. More specifically, $\mathcal{T} = 6$ performs best, thus leaving as the default setting.
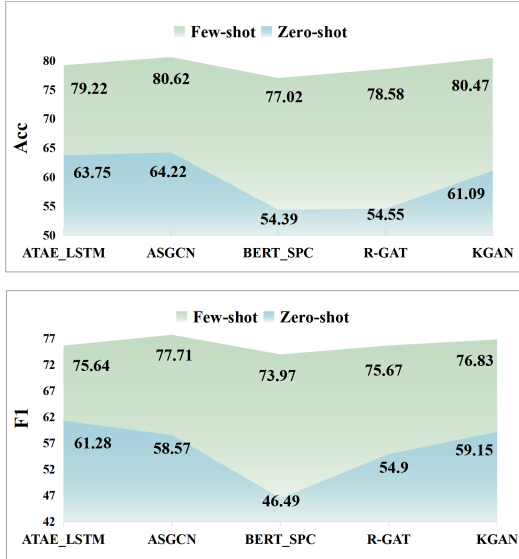
Figure 6: **Comparison of few-shot and zero-shot data generation in IterD.** We report the results on Laptop14.



Figure 7: **Analysis on the number of generated data.** "R" denotes the ratio of the number of generated data relative to that of original training data. R-GAT is used as the baseline model in this experiment.

## 4.4 Discussion and Analysis

In this part, we perform more in-depth analyses to further explore the underlying mechanism of IterD, covering 1) the impact of the accuracy of extracted aspects, 2) the effect of the few-shot generation prompt, and 3) an analysis of the number of generated data.

**Impact of the accuracy of extracted aspects.** Intuitively, based on more accurate aspects, IterD can generate more relevant training data and bring more performance gains. To verify it, we use the gold aspects in the original training sets as the upper bound to guide the generation of IterD. The contrastive results are illustrated in Figure 5, from which we find that IterD with gold aspects indeed achieves much better results. This indicates that *the performance of IterD relies on the accuracy of extracted aspects and more accurate aspects can result in better performance*.

**Effect of few-shot generation prompt.** In the iterative generation module of IterD, we use a few-shot prompt to guide the sample generation of LLMs. Here, we compare it with a zero-shot prompt, *i.e.*, removing the labeled examples in the prompt, and show the results in Figure 6. As seen, comparing the zero-shot prompt, IterD with the few-shot prompt achieves better and more stable performance, indicating that *adding some examples in the generation prompt is beneficial to generate more high-quality data*.
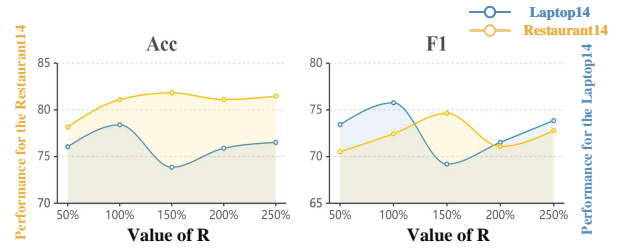
**Analysis of the number of generated data.** Here, we investigate the number of training data generated by IterD. Specifically, let $R$ be the number ratio of generated data relative to that of original training data, and we evaluate the performance of IterD with different $R$ ranging from 50% to 250%. Figure 7 illustrates the contrastive results of R-GAT on Laptop14 and Restaurant14 benchmarks. It can be found that the performance on both datasets shows a rising, falling, and then rising trend. With the increase in the amount of generated data, there will inevitably be more noisy samples in the generated data, which leads to performance degradation. However, with the generation of more reliable and stable quality samples, IterD brings performance improvements again. In general, these results show that *more generated data does not always lead to better performance, i.e., data quality is more important than quantity*.

## 5 Conclusion

In this paper, we propose a systemic iterative data augmentation framework (IterD), which leverages the powerful ability of LLMs to generate more high-quality labeled data. Starting from an unsupervised corpus, IterD first enforces the LLM to extract and expand the aspects and then designs an iterative LLM-based module to generate fluent and diverse labeled data. Lastly, IterD introduces a discriminator to filter the low-quality data. Extensive experiments on 4 popular ABSA benchmarks upon 5 baseline models show that the synthetic data generated by IterD can achieve comparable or even better performance against the original ground-truth data. Moreover, by combining the generated data and original data, IterD brings consistent and significant performance gains in all settings.

## Limitations

Our work has several potential limitations. First, despite its promising performance, our IterD may unexpectedly generate a low-quality sample with mixed sentiment polarities in a sentence. We will explore more effective prompting strategies for guiding the high-quality data generation of LLMs in future work. On the other hand, besides the data augmentation for the ABSA task, we believe that our method has the great potential to expand to more scenarios, *e.g.*, end-to-end ABSA, which are not fully explored in this work.

## Ethics and Reproducibility Statements

**Ethics.** We take ethical considerations very seriously and strictly adhere to the ACL Ethics Policy. This paper proposes a systematic DA method for generating more high-quality labeled data for ABSA. All models and evaluation datasets used in this study are publicly available and have been widely adopted by researchers. We believe that our proposed method will help alleviate ethical issues.

**Reproducibility.** In this paper, we discuss the detailed experimental setup and provide enough information to re-product our results, such as statistics and model descriptions. More importantly, ***we have provided our code in the supplementary materials*** to help reproduce our experimental results.

## References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint*.

Zhuang Chen and Tieyun Qian. 2020. Enhancing aspect term extraction with soft prototypes. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *ICLR*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting document knowledge for aspect-level sentiment classification. In *ACL*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *ACL*.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *EMNLP*.

B. Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining Text Data*.

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*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint*.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. Doer: Dual cross-shared rnn for aspect term-polarity co-extraction. In *ACL*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint*.

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. In *NeurIPS*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *ACL*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *ACL*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *ACL*.

Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *ACL*.

Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. 2019. Attentional encoder network for targeted sentiment classification. *arXiv preprint*.

Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, Linfeng Song, Le Sun, and Jiebo Luo. 2019. Progressive self-supervised attention learning for aspect-level sentiment analysis. In *ACL*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*.

Bing Wang, Liang Ding, Qihuang Zhong, Ximing Li, and Dacheng Tao. 2022. A contrastive cross-channel data augmentation framework for aspect-based sentiment analysis. In *COLING*.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *ACL*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *EMNLP*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *ICLR*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *ICCS*.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *ACL*.

Jianfei Yu, Qiankun Zhao, and Rui Xia. 2023. Cross-domain data augmentation with domain-adaptive language modeling for aspect-based sentiment analysis. In *ACL*.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *EMNLP*.

Anping Zhao and Yu Yu. 2021. Knowledge-enabled bert for aspect-based sentiment analysis. *Knowledge-Based Systems*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, Hua Jin, and Dacheng Tao. 2022. Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint*.

| Dataset | Type | Positive | | Neutral | | Negative | |
|---------|------|----------|------|---------|------|----------|------|
| | | Train | Test | Train | Test | Train | Test |
| Laptop14 | Original | 994 | 341 | 464 | 169 | 870 | 128 |
| | Generated | 1,051 | - | 358 | - | 919 | - |
| Rest14 | Original | 2,164 | 728 | 637 | 196 | 807 | 196 |
| | Generated | 2,377 | - | 548 | - | 1,291 | - |
| Rest15 | Original | 912 | 326 | 36 | 34 | 256 | 182 |
| | Generated | 1,572 | - | 405 | - | 1,631 | - |
| Rest16 | Original | 1,240 | 469 | 69 | 30 | 439 | 117 |
| | Generated | 2,215 | - | 184 | - | 1,209 | - |

Table 7: **Statistics of all used benchmarks**. Notably, "Original" denotes the original training and test sets of the benchmark, and "Generated data" denotes the synthetic data generated by our IterD. "Rest14", "Rest15" and "Rest16" refer to the Restaurant14, Restaurant15 and Restaurant16 benchmarks.

# A  Appendx

## A.1  Details of Tasks and Datasets

In this paper, we conduct main experiments on four public standard aspect-level datasets, *i.e.*, Laptop14, Restaurant14, Restaurant15, and Restaurant16. The Laptop14 and Restaurant14 datasets are from the SemEval2014 ABSA challenge (Pontiki et al., 2014), and Restaurant15 and Restaurant16 are from the SemEval2015 (Pontiki et al., 2015) and SemEval2016 (Pontiki et al., 2016) challenges, respectively. Following prior studies (Tang et al., 2019; Liu et al., 2023), we remove a few instances with conflicting sentiment polarity. To evaluate our IterD, we generate the synthetic data for each benchmark and compare the results training with the original data and generated data. Table 7 shows the statistics of all used data in this work.

## A.2  Details of baseline ABSA models

To investigate the effectiveness of our methods, we apply our augmented data to various ABSA baseline models, including:

- **ATAE-LSTM** (Wang et al., 2016): A LSTM-based model for ABSA using aspect embedding and attention mechanism.

- **ASGCN** (Zhang et al., 2019): It is the first ABSA model to represent sentences with dependency trees and use GCN to explore the syntactical information.

- **BERT-SPC** (Song et al., 2019): BERT-SPC feeds sequence "[CLS] + context + [SEP] + target + [SEP]" into the basic BERT model for sentence pair classification task.

- **R-GAT** (Wang et al., 2020): It uses a novel aspect-oriented dependency tree structure to reshape and prune ordinary dependency parse trees to better model syntax information.

- **KGAN** (Zhong et al., 2022): A novel knowledge graph augmented network encodes different types of information as multiview representations to enrich the semantic features.

### A.3 Details of compared DA methods

In the main experiments, we compare our IterD with the following widely-used DA methods:

- **Back Translation** (Sennrich et al., 2016): It is a sentence-level DA method, which first translates a sentence to another language and then translates it back to the original language.

- **EDA** (Wei and Zou, 2019): It is a simple word-level DA technique containing four operations: synonym substitution, random insertion, random exchange, and random deletion.

- **CBERT** (Wu et al., 2019): It integrates label information into the masked language modeling task to realize the prediction of replacement words, considering not only context but also label information

- **C3DA** (Wang et al., 2022): It uses a pre-trained generator to construct the synthetic multi-aspect training dataset.

### A.4 Details of Prompts in IterD

In this part, we show IterD prompts in detail, covering aspect extraction ("EX Prompt" in Table 8), aspect extension ("ET Prompt" in Table 9), sample generation ("ITAT Prompt" in Table 10), and discriminator ("Judgement Module" and "Auto Scoring Mechanism" in Table 11). Please refer to the tables for more details.

| Type | Prompts |
|---|---|
| **System Prompt** | You are extracting words from aspects of the text where sentiment has been expressed. |
| | We will perform an Aspect-Based Sentiment Analysis task. In this task, you are required to: |
| |   - Identify the aspects mentioned in the text |
| |   - Determine the sentiment polarity toward each aspect (positive, neutral, negative) |
| |   - Output format: [aspect, sentiment] |
| | {example} |
| | Now, complete the aspect extraction task for the text below: |
| | Input: {input} |
| **EX Prompt** | Output: |

Table 8: **Detailed prompts for aspect extraction**. The slot {example} denotes the example of aspect extraction results, and the slot {input} denotes the input unlabeled sentence.

| Type | Prompts |
|---|---|
| **System Prompt** | You are an AI assistant specializing in linguistics and sentiment analysis. |
| | We will perform an Aspect-Based Sentiment Analysis task. In this task, you need to expand the given aspect with its homonyms or synonyms. Generating 2-5 synonyms or cognates for a given aspect: |
| |   - example: |
| |     input: {example-input} |
| |     output: {example-input} |
| | Now, complete the aspect extend task for the text below: |
| | Input: {input} |
| **ET Prompt** | Output: |

Table 9: **Detailed prompts for aspect extension**. The slots {example-input} and {example-output} denote the example of aspect extension input-output pairs, *e.g.*, input: salads, output: {fish, noodles, bread, fruit salads}. The slot {input} denotes the input aspect.

| Type | Prompts |
|---|---|
| **System Prompt** | You are a critic who can generate comments on the specified aspect and sentiment. |
| | We would like you to complete a sentence generation task, and we will tell you how to generate appropriate sentences. Please follow these requirements: |
| |   -Teaching analysis – analyzing the given aspect and sentiment: |
| |   - Specify the sentiment of the aspect in the generated sample. |
| |   - Domain of sample generation: {domain} |
| |   - Generate a sentence containing a given aspect, clarify the meaning of the aspect, and generate sentences corresponding to the polarity of the sentiment. |
| |   - The generated sentence must be in length within {length} words. |
| |   - Generated sentences can contain only one period at a time and the sentence should not consist of an unspecified aspect |
| |   - examples: |
| |     Input: {example-input} |
| |     Output: {example-input} |
| | Now, complete this task in a natural human-like manner and generate only one sentence: |
| | Input: {input} |
| **ITAT Prompt** | Output: |

Table 10: **Detailed prompts for sample generation**. The slots {example-input} and {example-output} denote the example of input-output pairs. The slots {domain} and {length} are the given sample domain and length. The slot {input} denotes the input aspect-sentiment pair.

| Type | Prompts. |
|---|---|
| *Judgement Module* | |
| **System Prompt** | You are an AI assistant specializing in linguistics and sentiment analysis. |
| | You need to perform a task of sentiment judgment and domain judgment, the task requirements are shown below: |
| |    - Determine whether the potential sentiment hidden in the sentence by aspect is positive, negative, or neutral based on the context given in the sentence. |
| |    - Avoid confusing the neutral sentiment of the aspect with a positive or negative sentiment. |
| |    - Is this sentence related to {domain} ? If so, output "Y"; otherwise, output "N". |
| |    - Here are some examples of how aspect represents the sentiment in a sentence for your reference: |
| |       example-input:{[aspect, sentiment] } |
| |       example-output:{[sentence, #aspect, sentiment]} |
| | Now, please complete the task for the following input: |
| |    - input format: sentence, #aspect |
| |    - output format: sentiment; Y(N) |
| | Input: {input} |
| **Prompt** | Output: |
| *Auto Scoring Mechanism* | |
| **System Prompt** | You are an AI assistant specializing in linguistics and sentiment analysis. |
| | You are a psycholinguist who analyses sentiment and scores the above sentences in the following three areas: |
| |    1. Possessing complex syntactic structures, such as inverted sentences, imperative sentences, sentences with inflections, and sentences beginning with multiple combinations of adverbs, nouns, and subjects, the more complex the higher the score. |
| |    2. With a rich vocabulary, the richer the score, the higher the score. |
| |    3. User comments that match real-life scenarios, the more they match, the higher the score. |
| | Please give a score of 1-10 from each aspect accurately, and finally output a comprehensive average score selection of the highest-scoring sentences, the requirements of the output format are as follows: |
| |    [syntactic-structure: score; vocabulary-richness: score; real-scenario-conformity: score; comprehensive score: score] |
| **Prompt** | Please output in decimal form: |

Table 11: **Detailed prompts for discriminator**. The slots {domain} and {length} are the given sample domain and length. The slot {input} denotes the input sentence-aspect pair.