Generative Data Augmentation for Aspect Sentiment Quad Prediction

An Wang[†] **Junfeng Jiang**[‡] **Youmi Ma**[†] **Ao Liu**[†] **Naoaki Okazaki**[†]
[†]Tokyo Institute of Technology [‡]The University of Tokyo
an.wang@nlp.c.titech.ac.jp
jiangjf@is.s.u-tokyo.ac.jp
{youmi.ma@nlp., ao.liu@nlp., okazaki@}c.titech.ac.jp

Abstract

Aspect sentiment quad prediction (ASQP) analyzes the aspect terms, opinion terms, sentiment polarity, and aspect categories in a text. One challenge in this task is the scarcity of data owing to the high annotation cost. Data augmentation techniques are commonly used to address this issue. However, existing approaches simply rewrite texts in the training data, restricting the semantic diversity of the generated data and impairing the quality due to the inconsistency between text and quads. To address these limitations, we augment quads and train a quads-to-text model to generate corresponding texts. Furthermore, we designed novel strategies to filter out low-quality data and balance the sample difficulty distribution of the augmented dataset. Empirical studies on two ASQP datasets demonstrate that our method outperforms other data augmentation methods and achieves state-of-the-art performance on the benchmarks.[1]

1 Introduction

Aspect-based sentiment analysis (ABSA) aims to mine opinions expressed regarding specific aspects of a given text. Recently, Zhang et al. (2021a) proposed a challenging compound subtask of ABSA called aspect sentiment quad prediction (ASQP), which predicts four kinds of elements (aspect category, aspect term, opinion term, sentiment polarity) as quadruplets (quads). A single text may contain multiple quads. For example, the text "*The pizza is delicious but expensive.*" mentions one aspect term (*pizza*) and two opinion terms (*delicious* and *expensive*). Because these two opinions are related to the same aspect, the text includes two quads: (taste, pizza, delicious, positive) and (price, pizza, expensive, negative).

Traditional methods (Cai et al., 2020; Wan et al., 2020; Cai et al., 2021) address such compound subtasks of ABSA in a discriminative manner. Recent studies (Zhang et al., 2021b; Hu et al., 2022) have primarily concentrated on sequence-to-sequence frameworks for ASQP because of their superior performance. Specifically, These frameworks transform the input text into a sequence of linearized quads.

Source Text:
It is one the nicest outdoor restaurants I have ever seen in NY.
Label: [ambience general, outdoor restaurants, nicest, positive]

Augmented Text (Previous Methods):
Back Translation:
It is one of the most beautiful outdoor restaurants I have ever seen in NY.
Random Deletion:
It is one the nicest restaurants I have ever seen in NY.
Synonym Replacement:
It is one the best outdoor restaurants I have ever seen in NY.

Our Method:
Augmented Label:
[ambience general, space, Great, positive],
[ambience general, outdoor restaurants, nothing special, positive]
Augmented Text:
Great space for meetings, but the outdoor restaurants is nothing special.

Figure 1: Examples of text data augmentation methods. We observe that the augmented texts from previous methods fail to include all spans in the label and the augmented texts are semantically very similar to the source text. Our method addresses these problems by generating texts from augmented labels.

Despite the success of the field of ASQP, the scarcity of annotated data is still a remaining challenge. For instance, Rest15 and Rest16 ASQP datasets only consist of 834 and 1,264 training samples respectively. However, manual annotation is costly and time consuming. One solution for expanding the number of training samples is data augmentation. EDA (Wei and Zou, 2019) adopted some typical data augmentation techniques such as random swapping, inserting, deleting words, and synonym replacement to improve text classification. Back-translation (Yu et al., 2018) obtained augmented data by translating the original text in English into another language and then translating

[1] The source code is available at https://github.com/AnWang-AI/AugABSA

*Proceedings of the The 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 128–140
July 13-14, 2023 ©2023 Association for Computational Linguistics

it back into English. However, applying these operations to ASQP datasets usually disrupts crucial spans, such as aspect or opinion terms, resulting in label mismatches with the original input text. Additionally, traditional data augmentation methods only focus on augmenting texts that preserve semantic information similar to the original text in the training dataset. Therefore, the ability of these methods to help models generalize to unseen data is limited.

In this study, we propose a novel **Gen**erative **D**ata **A**ugmentation method (GenDA) by proposing a quads-to-text (Q2T) generation task—the reverse task of ASQP, which aims to generate a text based on the input quads. We synthesize a large number of quads by mixing the labels from the ASQP training dataset. Then, we feed these labels to the trained Q2T model which uses a sequence-to-sequence model to generate new parallel data with high diversity. Figure 1 shows some examples of the traditional text augmentation methods and our method. In addition, we propose a data filtering strategy concerning the unalignment of the aspect and opinion terms between text and quads to remove low-quality augmented data. Furthermore, we propose a new measurement, Average Context Inverse Document Frequency (AC-IDF), to evaluate the difficulty of augmented samples and a strategy to balance the difficulty distribution. Finally, we can augment sufficient training data with good diversity and high quality.

To evaluate our method, we conducted empirical studies using two ASQP datasets. We applied the proposed data augmentation with the previous ASQP model. These studies demonstrate that our method outperforms other data augmentation methods and achieves state-of-the-art performance on the benchmark. In addition, the experimental analysis also verifies that our method successfully generates data with stronger diversity. Additionally, we conducted a detailed ablation study to confirm the effectiveness of each component of our method and provide insights into how they contribute to the performance of our method.

The contributions of this study are summarized as follows: (1) We propose the synthesis of diverse parallel data using a Q2T model for ASQP. To the best of our knowledge, this is the first study to achieve data augmentation by text generation for ABSA. (2) We propose a data filtering strategy to remove low-quality augmented data and a measurement to evaluate the difficulty of the augmented samples, which is used to balance the augmented dataset. (3) Our experiments demonstrate that the proposed method achieves state-of-the-art performance on the two ASQP datasets.

2 Preliminaries

2.1 Task Definition of ASQP

Aspect sentiment quad prediction aims to predict all sentiment-related quadruplets (ac, at, ot, sp) from a given text x. The elements of each quadruplet are aspect category (ac), aspect term (at), opinion term (ot), and sentiment polarity (sp). In particular, the aspect category belongs to a specific category set AC and the sentiment polarity falls into sentiment classes $\{\texttt{POS}, \texttt{NEU}, \texttt{NEG}\}$ denoting positive, neutral, and negative sentiments toward the aspect. Note that if the aspect and opinion terms are not explicitly mentioned in the given text, they are set as `NULL`.

2.2 Generative ASQP Methods

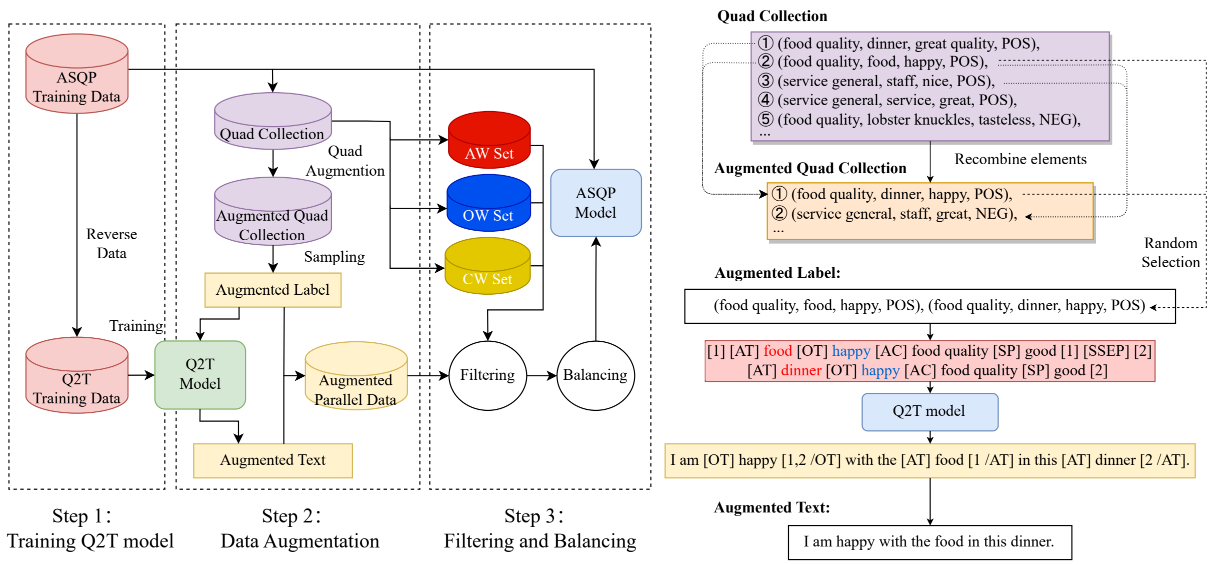
Although early work handled ABSA in a discriminative manner, recent studies (Zhang et al., 2021a,b; Hu et al., 2022) have mainly focused on generative ASQP methods because of their better performances.

PARAPHRASE (Zhang et al., 2021a) formulated ASQP into a paraphrasing problem. They transformed sentiment-related quadruplets into a natural language. Specifically, given a quad (ac, at, ot, sp), they designed the following template: "ac is sp because at is ot.", where ac and sp are projected onto the natural language format. When the input text contains multiple quads, the quads are transformed into different templated sentences separately and then concatenated with a special marker [SSEP]. Hu et al. (2022) explored the effect of the order of each quad element in the template. In addition, they proposed a more effective target template: "[AT] at [OT] ot [AC] ac [SP] sp", where [AT], [OT], [AC], and [SP] are special tokens.

Inspired by previous generative ASQP methods, we consider the reverse process of text-to-quads and further propose a generative data augmentation method based on it.

3 Methodology

To alleviate the problem of annotated data scarcity and to generate augmented data with strong diver-


ASQP Training Data
Reverse Data
Training
Q2T Training Data
Q2T Model
Quad Collection
Quad Augmention
Augmented Quad Collection
Sampling
Augmented Label
Augmented Parallel Data
Augmented Text
AW Set
OW Set
CW Set
ASQP Model
Filtering
Balancing
Step 1: Training Q2T model
Step 2: Data Augmentation
Step 3: Filtering and Balancing
Quad Collection
① (food quality, dinner, great quality, POS),
② (food quality, food, happy, POS),
③ (service general, staff, nice, POS),
④ (service general, service, great, POS),
⑤ (food quality, lobster knuckles, tasteless, NEG),
...
Augmented Quad Collection
Recombine elements
① (food quality, dinner, happy, POS),
② (service general, staff, great, NEG),
...
Random Selection
Augmented Label:
(food quality, food, happy, POS), (food quality, dinner, happy, POS)
[1] [AT] food [OT] happy [AC] food quality [SP] good [1] [SSEP] [2] [AT] dinner [OT] happy [AC] food quality [SP] good [2]
Q2T model
I am [OT] happy [1,2 /OT] with the [AT] food [1 /AT] in this [AT] dinner [2 /AT].
Augmented Text:
I am happy with the food in this dinner.

(a) Pipeline of our method.

(b) Example of generating augmented text.

Figure 2: Overview of our proposed method. In Step 3 of Figure (a), AW Set, OW Set, and CW Set represent aspect word set, opinion word set, and context word set, respectively. They are utilized to aid the filtering process. Figure (b) shows an example of synthesizing augmented parallel data consisting of a text, "I am happy with the food in this dinner," and an associated label, "(food quality, food, happy, POS), (food quality, dinner, happy, POS)". The dotted line indicates the source of the quad or label.

sity and high quality for the ASQP task, we propose a novel generative data augmentation method. Figure 2 presents an overview of the proposed method. The proposed method consists of three main steps. (1) We reverse the data in the ASQP dataset to create a new training set, which we then use to train a quads-to-text model. (2) We aim to generate data that are semantically different from the training data. Hence, we collect all labels from the training set and propose mixing them to create an augmented quad set. We then randomly sample several mixed quads and feed them into the quads-to-text model to generate the corresponding source text. (3) To further improve the performance of our data augmentation, we propose two strategies to filter out generated texts that do not match the given quads and balance the sample difficulty distribution of the augmented data. Finally, we combine the augmented data with the original training set to train an ASQP model.

3.1 Quads-to-Text Task

Before introducing our generative data augmentation method, we first define a new text generation task, the quads-to-text (Q2T) task, and then design a Q2T model based on a pre-trained sequence-to-sequence model.

3.1.1 Task Definition of Quads-to-Text Task

To obtain parallel augmented data for our generative ASQP data augmentation, we first propose a quads-to-text task. Q2T aims to generate text describing the given quads. Given n quads $\{q_1, q_2, ..., q_n\}$, where $q_i = (ac_i, at_i, ot_i, sp_i)$, the task requires generating a text x that includes and only includes the input quads.

3.1.2 Quads-to-Text Model

To handle the Q2T problem, we utilize the pre-trained sequence-to-sequence model following other works on controllable text generation (Zhang et al., 2022). Unlike conventional text generation methods, our designed Q2T model not only generates texts but also provides a mechanism to conveniently judge whether the generated statement meets the task requirements of Q2T. In our method, we mainly focus on the input and output designs of the model.

For the input sequence of the model, we formulate the given quads as template sentences similarly to Hu et al. (2022). The difference is that we insert special indexing markers before and after each sentence to distinguish multiple quads. Specifically, the i-th quad (ac_i, at_i, ot_i, sp_i) is transformed into a templated text:

$$\text{[i][AT] } at_i \text{ [OT] } ot_i \text{ [AC] } ac_i \text{ [SP] } sp_i \text{ [i]}$$

The final transformed texts are linked with a special marker [SSEP] following previous work (Zhang et al., 2021b; Hu et al., 2022).

For the output sequence of the model, instead of only generating the source text, the Q2T model can generate text with annotations. The annotations identify aspect terms and opinion terms in the text. In addition, the annotation also includes the relation information between aspects and opinions. The model annotates aspect and opinion terms of i-th quad in the text using special markers "[AT]", "[i /AT]", "[OT]", and "[i /OT]". Special tokens [AT] and [OT] denote the beginning of an aspect and opinion term while [i /AT] and [i /OT] denote the ending position. When there are multiple aspects in a text that are described by the same opinion or there are multiple opinions describing the same aspect, they can be grouped together using a comma-separated list of numbers within square brackets, such as [1,2 AT] to indicate that the first and second opinion describe the same aspect. We will explain the function of these annotations in detail in Section 3.2.2.

3.1.3 Training

To make the Q2T model generate text that meets our requirements, we first build Q2T datasets based on ASQP datasets. ASQP aims to predict quads from the given text, thus, we obtain Q2T datasets by simply inverting the input and label of the ASQP dataset. To enhance the ability to understand the meaning of the special index markers, we augment Q2T data by permuting the order of quads in the templated input of Q2T model. After training the Q2T model, the model can be used to obtain more abundant augmented data for the ASQP task.

3.2 Augmention Strategy

In this section, we first propose a novel method for obtaining a diverse augmented dataset based on the Q2T model. We then propose a filtering strategy and a difficulty balancing strategy to further improve the performance of data augmentation.

3.2.1 Synthesizing Augmented Quads

To obtain diverse data that are meaningfully different from the data in the original ASQP training dataset, we propose to diversify the input of the Q2T as shown in figure 2.

First, we collect all quads from the ASQP training dataset as a quad collection, denoted as $\mathcal{S}_{origin} = \{(ac_i, at_i, ot_i, sp_i)\}$. Subsequently, for those quads that **share the same aspect category** ac_i, we randomly exchange their aspect term at and opinion term ot with sentiment polarity sp to create new quads. The opinion term and sentiment polarity from the same original quad will be bound together to avoid getting new quads where elements conflict with each other. For example, given two quads: (price, pizza, cheap, POS) and (price, steak, expensive, NEG), we can synthesize new quads (price, steak, cheap, POS) and (price, pizza, expensive, NEG). Finally, we balance the number of synthesized quads for each aspect category to obtain the augmented quad collection, denoted as $\mathcal{S}_{augment}$. Subsequently, each time we randomly select $1 \sim 3$ quads from $\mathcal{S}_{origin} \cup \mathcal{S}_{augment}$, and feed them to the Q2T model for data augmentation. During the training of the ASQP model, we remove the annotations such as [AT] in the augmented text.

3.2.2 Data Filtering

For ASQP data augmentation, a common problem is that the augmented texts may not be faithful to the given quads. Specifically, the generated texts from the Q2T model may contain fewer or more quads compared to input quads. Using unfaithful text as ground truth for given quads to train the ASQP model will introduce noise that decreases the performance. Thus we propose a two-step filtering strategy to remove these low-quality data.

The first step of filtering involves checking the consistency between the output text of the Q2T model and the input quads. As introduced in Section 3.1.2, our Q2T model annotates aspect terms and opinion spans using special markers when generating texts. This allows us to collect aspect-opinion pairs from the output text and then check the consistency between the detected pairs and input quads. We filter out the examples with inconsistent aspect and opinion terms.

However, the generated texts that pass the first filtering step may contain additional aspect or opinion terms that are not annotated with special markers. Training the ASQP model with such data may lead to a lower recall. To address this issue, we propose the second step of data filtering. The process involves building two keyword sets (an aspect word set and an opinion word set) and a context word set. Specifically, we begin by collecting all the texts from the training data. Because the aspect and opinion terms are annotated, we categorize the words in the text based on labels into three groups:

aspect words, opinion words, and context words. After that, we gather all the aspect words to create the aspect word set. Similarly, we collect all the opinion words to form the opinion word set and all the context words to construct the context word set. If the unmarked part (i.e., the context) of a generated text contains any word that belongs to the keyword sets but does not exist in the context word set, we consider this example as containing additional aspect/opinion terms and remove it from the augmented dataset.

3.2.3 Difficulty Balancing

In addition to the existence of low-quality data, another problem we observe is that more than half of the generated texts are simple expressions. These generated texts are far simpler than most texts in the ASTE dataset. A text can be divided into three different parts, aspect terms, opinion terms, and context. Even if being given different quads as inputs, the Q2T model usually generates text with relatively similar context, such as 'The at is ot'. When most augmented training data are too simple, the model may not learn the complex patterns required to make accurate predictions on unseen data. Therefore, it is necessary to balance the distribution of the sample difficulty of the augmented dataset.

To assess the sample difficulty, we propose a new measurement factor, called the Average Context Inverse Document Frequency (AC-IDF). The difficulty of a text can be defined as the level of language proficiency required to understand the text (Fulcher, 1997). A text that uses many uncommon words is considered more difficult than one that uses simple and common language. Therefore, one way to measure the difficulty of a text is to calculate the average IDF score of the words in the text. Furthermore, because aspects and opinions are directly copied from the input of the model, it is critical to evaluate the difficulty of the context part of the text. Therefore, we propose using the context difficulty to measure the learning difficulty of the sample for our model.

Specifically, given a text collection X from the dataset, we remove all aspects and opinions terms to obtain only the context words. We denote the preprocessed text collection by $\bar{X}$. Then, for each text $\bar{x}_i$ after preprocessing, we calculate the AC-IDF$_i$ of the text as follows:

$$\text{AC-IDF}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \text{IDF}(t_{ij}), \tag{1}$$

$$\text{IDF}(t_{ij}) = \ln \frac{|\bar{X}|}{1 + |\{\bar{x} \in \bar{X} : t_{ij} \in \bar{x}\}|}, \tag{2}$$

where t_{ij} is the j-th word in $\bar{x}_i$, n_i is the number of words in $\bar{x}_i$, $|\bar{X}|$ denotes the size of $\bar{X}$, and $|\{\bar{x} \in \bar{X} : t_{ij} \in \bar{x}\}|$ represents the total number of texts where t_{ij} appears.

We build a subset according to the AC-IDF of the generated texts so that the difficulty of the selected data follows a uniform distribution. Specifically, we set several intervals according to the sample difficulty and randomly sample similar amounts of data from the entire augmented dataset for each interval. Finally, we create a subset whose data obey an approximate uniform distribution with respect to the sample difficulty. Thus, the model learns to predict quads from diverse and balanced data to improve performance.

4 Experiment

4.1 Datasets

We evaluate our method on two ASQP datasets: `Rest15` and `Rest16` (Zhang et al., 2021b), which originates from the SemEval Challenges (Pontiki et al., 2015, 2016). Their domain is of restaurant reviews. Detailed statistics are shown in Appendix A. We also evaluate our method on four Aspect Sentiment Triplet Extraction (ASTE) datasets (Peng et al., 2020) in Appendix 5.

4.2 Experiment Setting

In accordance with previous studies (Zhang et al., 2021a; Hu et al., 2022), our method also employs T5-base (Raffel et al., 2020) as the pre-trained backbone for both Q2T and ASQP tasks. The parameter count is twice the size of the backbone model (one for the Q2T model and one for the ASQP model), which is equivalent to 2 × 220 million parameters. We set the batch size to 8 and the learning rate to 1e-4. During the inference stage, greedy decoding is used to generate the output sequence. The amount of augmented data is four times that of training data. The experiments are run for a maximum of 20 epochs. All reported results are the average of five runs initialized with different random seeds. We use precision, recall, and micro F1 scores as the evaluation metric. A sentiment quad prediction is considered accurate only when all of its predicted elements match the ground truth exactly. We also report the standard errors of our base model and proposed data augmentation method.

4.3 Main Results

4.3.1 Compared Methods

Previous ASQP methods can be categorized into two types: BERT (Devlin et al., 2019) based methods and T5 (Raffel et al., 2020) based methods. The BERT based methods include **HGCN** (Cai et al., 2020), **TASO** (Wan et al., 2020), and **Extract-Classify-ACOS** (Cai et al., 2021). T5 based methods include **GAS** (Zhang et al., 2021b), **PARAPHRASE** (Zhang et al., 2021a), **DLO** and **ILO** (Hu et al., 2022). We report the performance of these methods directly copied from their paper. **PARAPHRASE + Marked Template** is a variant of the PARAPHRASE method. It uses a different target template with special markers which are proposed by Hu et al. (2022). We implement it by ourselves and adopt it as our base model to apply our data augmentation method.

4.3.2 Analysis

Table 1 shows the evaluation results on the ASQP task. We observe that our proposed data augmentation method, **GenDA**, clearly improves the performance of the base model by +2.22 and +2.18 F1 score on `Rest15` an `Rest16`. GenDA achieves state-of-the-art performance on the ASQP benchmark. Note that GenDA has a higher precision score while maintaining a good recall compared with other methods. This observation indicates that our proposed data augmentation method helps to improve the robustness of our model, and therefore, predicts the sentiment quadruplets more precisely.

Our base model **PARAPHRASE + Marked Template** achieves better performance than the original PARAPHRASE method but does not outperform DLO and ILO. The reason why we do not choose DLO or ILO as our base model is that these two methods are relatively complex and not suitable for integrating our data augmentation methods.

4.4 Effects of Augmentation Methods

To demonstrate the effectiveness of the data augmentation method we proposed, we also compare it with several representative data augmentation methods on the ASQP benchmark. For all data augmentation methods, the amount of augmented data is four times that of training data.

EDA (Wei and Zou, 2019) adopts four operations including synonym replacement, random insertion, random swap, and random deletion to the input texts. We additionally design two ASQP-specific variants of EDA: **CEDA** applies EDA only in the context of input text whereas **AOEDA** applies EDA on the aspect terms and opinion terms of the input text. Note that the terms in quads will also be revised correspondingly. **AEDA** (Karimi et al., 2021) is an simpler data augmentation method that randomly inserts punctuation into the input texts. **Back Translation** (Yu et al., 2018) augments data by translating text from English to another language and then back to English. We used the machine translation models proposed by Ng et al. (2019) in our experiment.

Comparison results are reported in Table 2. Compared with existing data augmentation methods, we observe that applying EDA and Back Translation on the base model brings no noticeable improvement and can even reduce performance. We attribute it to the fact that sometimes these methods disrupt the matching of input text and labels because they may revise some important spans including aspects or opinion terms. To explore whether directly modifying traditional data augmentation methods to adapt the ASQP task can improve model performance, we evaluate AOEDA and CEDA, two simple variants of EDA which avoid mismatches between text and labels of augmented data. The results show that both two variants could improve the performance slightly, but the improvement is limited, likely because the existing data augmentation methods cannot provide training samples with high diversity. Finally, our method GenDA significantly outperforms all traditional data augmentation methods under all evaluation metrics. Compared with the best F1 scores of previous data augmentation methods, the improvement of our method reaches 1.26 and 1.04. These results demonstrate the effectiveness of our data augmentation method.

4.5 Analysis of the Text Diversity

We analyze the text diversity of different data augmentation methods. In Figure 3, we visualize the text representations of the entire `Rest 16` training dataset and 4000 augmented data generated by different data augmentation methods. Specifically, we first adopt a BERT-based encoder to transform each text into a representative vector and then use t-

PLM	Method	Rest 15			Rest 16		
		P	R	F1	P	R	F1
BERT	HGCN-Linear (Cai et al., 2020)	24.43	20.25	22.15	25.36	24.03	24.68
	HGCN-TFM (Cai et al., 2020)	25.55	22.01	23.65	27.40	26.41	26.90
	TASO-Linear (Wan et al., 2020)	41.86	26.50	32.46	49.73	40.70	44.77
	TASO-CRF (Wan et al., 2020)	44.24	28.66	34.78	48.65	39.68	43.71
	Extract-Classify-ACOS (Cai et al., 2021)	35.64	37.25	36.42	38.40	50.93	43.77
T5	GAS (Zhang et al., 2021b)	45.31	46.70	45.98	54.54	57.62	56.04
	PARAPHRASE (Zhang et al., 2021a)	46.16	47.72	46.93	56.63	59.30	57.93
	DLO (Hu et al., 2022)	47.08	49.33	48.18	57.92	**61.80**	59.79
	ILO (Hu et al., 2022)	47.78	**50.38**	49.05	57.58	61.17	59.32
	PARAPHRASE + Marked Template	47.40 ±0.20	48.18 ±0.44	47.79 ±0.30	57.85 ±0.30	59.58 ±0.42	58.70 ±0.35
	+ GenDA	**49.74** ±0.28	50.29 ±0.35	**50.01** ±0.31	**60.08** ±0.34	61.70 ±0.12	**60.88** ±0.13

Table 1: Evaluation results (%) on `Rest 16` and `Rest 15` datasets of ASQP for comparing with previous state-of-the-art methods. The best and second-best performances are highlighted in bold and underlined, respectively.

Type	Method	Rest 15			Rest 16		
		P	R	F1	P	R	F1
Baseline	PARAPHRASE + Marked Template	47.40	48.18	47.79	57.85	59.58	58.70
Previous Data Augmentation	+ EDA	47.77	48.27	47.85	57.70	58.85	58.27
	+ CEDA	47.44	48.63	48.19	58.47	60.43	59.43
	+ AOEDA	47.78	48.40	48.09	58.22	60.30	59.24
	+ AEDA	48.17	48.65	48.40	58.40	59.70	59.04
	+ Back Translation	47.08	47.30	47.19	58.58	59.86	59.21
Ablation	+ GenDA	**49.74**	**50.29**	**50.01**	**60.08**	**61.70**	**60.88**
	+ GenDA (original label)	48.88	49.48	49.18	59.23	61.08	60.14
	+ GenDA w/o Filtering & Balancing	47.95	48.55	48.25	58.33	60.45	59.37
	+ GenDA w/o Balancing	48.34	49.21	48.77	58.84	60.61	59.71
	+ GenDA w/o Filtering	48.63	49.18	48.91	59.39	61.05	60.21

Table 2: Evaluation results (%) on `Rest 16` and `Rest 15` datasets of ASQP for comparing different data augmentation methods and ablations. All involved data augmentation methods use PARAPHRASE + Marked Template as the base model for a fair comparison. GenDA (original label) denotes only using original labels from the training dataset instead of augmented quads for Q2T.

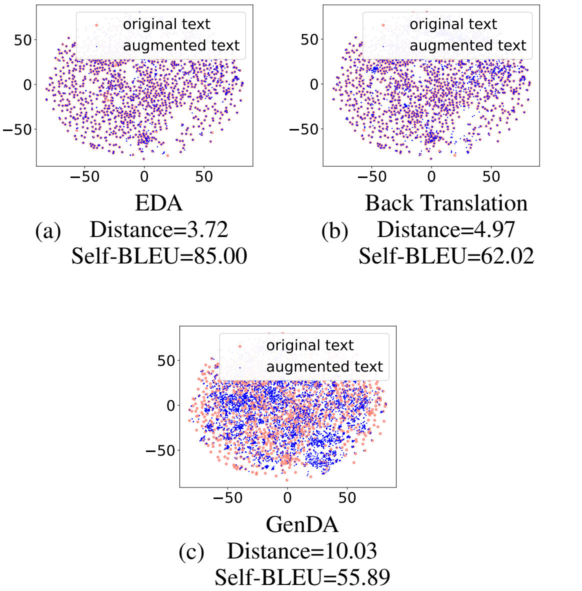
SNE (Van der Maaten and Hinton, 2008) to reduce dimension for visualizing the distributions.

To quantify the difference between the training dataset and the augmented dataset, we calculate the average Euclidean distance between each point in the augmented dataset and its nearest neighbor in the training dataset. We also provide the Self-BLEU scores (Zhu et al., 2018) to evaluate the diversity of each augmented dataset. Lower Self-BLEU means better diversity.

From Figure 3, we observe that the semantic representations of most EDA-augmented texts are coincident with original texts, showing the smallest average distance. The high Self-BLEU score of EDA further indicates the low diversity of EDA-augmented texts. Back Translation achieves a much lower Self-BLEU score than EDA, but the visualization shows a high semantic similarity between the original and augmented data. By contrast, our proposed method GenDA achieves the largest distance score and lowest Self-BLEU, demonstrating that it can generate texts that are more diverse and less likely to semantically overlap with the original texts.

4.6 Ablation Studies

To investigate the effectiveness of each component of our proposed method, we conduct an ablation study on two ASQP datasets as shown in Table 2. Even without adopting our filtering or balancing strategies, our model can outperform the base model. After applying our filtering strategy, we observe an improvement because it filters out noisy and irrelevant data. The balancing strategy also brings a performance gain, which indicates that addressing the sample difficulty imbalance issue in the augmented datasets is beneficial for models to learn. Note that compared to the filtering strategy, the balancing strategy contributes more to performance gains, which means that sample difficulty imbalance has a worse impact on performance than the low-quality problem. Furthermore, when the filtering and balancing strategies are applied jointly, our model achieves a further performance



(a) EDA Distance=3.72 Self-BLEU=85.00

(b) Back Translation Distance=4.97 Self-BLEU=62.02

(c) GenDA Distance=10.03 Self-BLEU=55.89

Figure 3: Visualization of text semantic representation. Each subfigure shows the distribution of original texts (in salmon color) from the Rest16 training dataset and corresponding augmented texts (in blue color) obtained using different methods. In each subcaption, we report the distance between two datasets and the Self-BLEU score (%) computed on each augmented dataset.

gain. In addition, when only the original labels are input, the model's f1 scores noticeably decline by 0.83 and 0.74 compared to when augmented labels are input.

5 Effects on ASTE task

We conduct experiments on the ASTE task to verify that our method is also effective on other ABSA subtasks. We compare our method with strong previous work.

ASTE methods Previous ASTE methods can be categorized into three types: pipeline-based methods, end-to-end discrimination methods, and text-generation methods. The pipeline-based methods include CMLA (Wang et al., 2017), RINATE+ (Dai and Song, 2019), Li-unified-R (Li et al., 2019), P-pipeline (Peng et al., 2020), and Two-Stage (Huang et al., 2021). End-to-end discrimination methods include BMRC (Chen et al., 2021), SPAN-ASTE (Xu et al., 2021), EMC-GCN (Chen et al., 2022), and COM-MRC (Zhai et al., 2022). Text-generation methods for ASTE include GAS (Zhang et al., 2021b) and PARAPHRASE (Zhang et al., 2021a).

We select three types of ASTE methods for comparison: 1) Pipeline based methods including CMLA (Wang et al., 2017), RINATE+ (Dai and Song, 2019), Li-unified-R (Li et al., 2019), P-pipeline (Peng et al., 2020) and Two-Stage (Huang et al., 2021); 2) End-to-end discrimination methods: BMRC (Chen et al., 2021), SPAN-ASTE (Xu et al., 2021), EMC-GCN (Chen et al., 2022) and COM-MRC (Zhai et al., 2022); and 3)Text-generation methods: GAS (Zhang et al., 2021b) and PARAPHRASE (Zhang et al., 2021a).

Analysis Table 3 shows the evaluation results of baselines and our methods on four datasets of ASTE task, including `Lap14`, `Rest14`, `Rest15`, and `Rest16`. Compared to ASQP, ASTE only needs to predict three kinds of elements. In our method, the target template of ASTE is changed to

[i] [AT] at_i [OT] ot_i [SP] sp_i [i],

for the i-th triplet (at_i, ot_i, sp_i). Other designs for the ASTE task are the same as the ASQP task. We find that with this slight revision, our methods outperform the best results by 1.53, 1.73, 1.27, and 2.53 f1 score on these four datasets respectively, achieving new state-of-the-art performance.

6 Related Work

6.1 Aspect-based Sentiment Analysis

ABSA aims to analyze fine-grained sentiment elements including not only the sentiment polarity but also the aspect term, opinion term, and aspect category. Intuitively, these elements are related. Therefore, recent studies tried to model them jointly, such as constructing aspect-sentiment pairs (Cai et al., 2020) or triples (Peng et al., 2020). Furthermore, there is a growing interest in modeling these four elements simultaneously, with two promising directions being proposed. Cai et al. (2021) proposed a two-stage method that first extracts aspect and opinion terms, and then uses them to classify aspect category and sentiment polarity. Another framework is based on a generation model (Zhang et al., 2021a,b), which predicts the quadruplet in an end-to-end manner by paraphrasing the input text to a target template. Since they additionally exploit the information from label semantics, the generation-based method achieves dominantly better performance in the field of ABSA.

6.2 Data Augmentation

Data augmentation is a common technique in language and vision domains to improve model performance. Previous data augmentation methods

Backbone	Method	L14	R14	R15	R16
BERT	CMLA (Wang et al., 2017)	33.16	42.79	37.01	41.72
	RINATE+ (Dai and Song, 2019)	34.95	20.07	29.97	23.87
	Li-unified-R (Li et al., 2019)	42.34	51.00	47.82	44.31
	P-pipeline (Peng et al., 2020)	42.87	51.46	52.32	54.21
	Jet (Xu et al., 2018)	51.04	62.40	57.53	63.83
	GTS (Wu et al., 2020)	55.21	64.81	54.88	66.08
	Two-Stage (Huang et al., 2021)	58.58	68.16	58.59	67.52
	BMRC (Chen et al., 2021)	57.82	67.99	60.02	65.75
	SPAN-ASTE (Xu et al., 2021)	59.38	71.85	63.27	70.26
	EMC-GCN (Chen et al., 2022)	58.81	71.78	61.93	68.33
	COM-MRC (Zhai et al., 2022)	60.17	72.01	64.53	71.57
T5	GAS (Zhang et al., 2021b)	58.19	70.52	60.23	69.05
	PARAPHRASE (Zhang et al., 2021a)	61.13	72.03	62.56	71.70
	GenDA	**62.66**	**73.76**	**65.80**	**74.23**

Table 3: Evaluation results (%) on four datasets of ASTE for comparing with previous state-of-the-art methods. The best and second-best performances are highlighted in bold and underlined, respectively.

can be categorized into three types. The first type only augments the input, such as image flipping, rotation, and scaling (Bjerrum, 2017) for images, and text modification (Wei and Zou, 2019) as well as back translation (Yu et al., 2018) for natural language. The second type only augments the output, such as generating target-side soft pseudo sequences (Xie et al., 2022). These approaches are particularly relevant for generation tasks where the order of words is important. The third type augments both the input and the output, such as the mixup approach (Zhang et al., 2018) which generates virtual training examples through linear combinations of feature vectors and their associated targets. To the best of our knowledge, our work is the first to propose a data augmentation method of the third type specifically for subtasks of ABSA. Unlike previous methods in this realm that augment only the input (Li et al., 2020) or output (Hu et al., 2022), our method augments both input and output, leading to augmenting more diverse samples. Our method reduces the model's reliance on a limited set of examples and enables it to better generalize to unseen data, thereby mitigating the problem of overfitting and achieving better performance on test data.

7 Conclusion

In this paper, we have proposed a new approach to tackle the problem of data scarcity in the ASQP task. To address this challenge, we present a generative data augmentation method based on a pretrained quads-to-text model. Our method generates new parallel data by synthesizing a large number of quads from the training dataset and generating corresponding pseudo texts. Moreover, we propose a data filtering strategy to remove low-quality generated data and a measurement to balance the difficulty of augmented samples. Our empirical studies on two ASQP datasets have demonstrated the superiority of our method compared to other data augmentation methods and the effectiveness of each component in our method. Our approach not only is an innovative solution to the problem of data scarcity in ASQP, but also provides a potential direction for future work in other related fields, such as relation extraction and event extraction.

Limitation

Firstly, because our data augmentation method relies on the quality of the quads-to-text (Q2T) model's generation, the performance of our method may be limited by the quality of the generated text. Besides, the quads-to-text (Q2T) model is trained by the original ASQP dataset, thus it may fail to generate expressions that do not appear in the dataset. Additionally, training an extra Q2T model brings additional computational costs. Furthermore, as the model inputs are randomly sampled from the augmented quad collection, some quad combinations may not be suitable for text generation, which could affect the effectiveness of data augmentation.

Ethics Statement

There are no ethical problems in this paper. All of the datasets are publicly available.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 19H01118. We sincerely appreciate Sakae Mizuki and Marco Cognetta for their generous assistance and valuable discussions. We also express our gratitude to all reviewers for their insightful comments and constructive feedback.

References

Esben Jannik Bjerrum. 2017. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076*.

Hongjie Cai, Yaofeng Tu, Xiangsheng Zhou, Jianfei Yu, and Rui Xia. 2020. Aspect-category based sentiment analysis with hierarchical graph convolutional network. In *Proceedings of the 28th international conference on computational linguistics*, pages 833–843.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. 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 340–350.

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

Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 12666–12674.

Hongliang Dai and Yangqiu Song. 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. *arXiv preprint arXiv:1907.03750*.

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

Glenn Fulcher. 1997. Text difficulty and accessibility: Reading formulae and expert judgement. *System*, 25(4):497–513.

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022. Improving aspect sentiment quad prediction via template-order data augmentation. *arXiv preprint arXiv:2210.10291*.

Lianzhe Huang, Peiyi Wang, Sujian Li, Tianyu Liu, Xiaodong Zhang, Zhicong Cheng, Dawei Yin, and Houfeng Wang. 2021. First target and opinion then polarity: Enhancing target-opinion correlation for aspect sentiment triplet extraction. *arXiv preprint arXiv:2102.08549*.

Akbar Karimi, Leonardo Rossi, and Andrea Prati. 2021. AEDA: An easier data augmentation technique for text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6714–6721.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8600–8607.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

Maria Pontiki, Dimitrios Galanis, Harris 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9122–9129.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

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, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. *arXiv preprint arXiv:2010.04640*.

Shufang Xie, Ang Lv, Yingce Xia, Lijun Wu, Tao Qin, Tie-Yan Liu, and Rui Yan. 2022. Target-side input augmentation for sequence to sequence generation. In *International Conference on Learning Representations*.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1805.04601*.

Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning span-level interactions for aspect sentiment triplet extraction. *arXiv preprint arXiv:2107.12214*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

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

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

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

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

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

Dataset	#Text	
	Rest15	Rest16
Train	834	1264
Validation	209	316
Test	537	544

Table 4: Statistics of datasets of ASQP task.

Dataset	#Text			
	Laptop14	Rest14	Rest15	Rest16
Train	1300	920	593	842
Validation	323	228	148	210
Test	496	339	318	320

Table 5: Statistics of datasets of ASTE task.

A Dataset Statistic

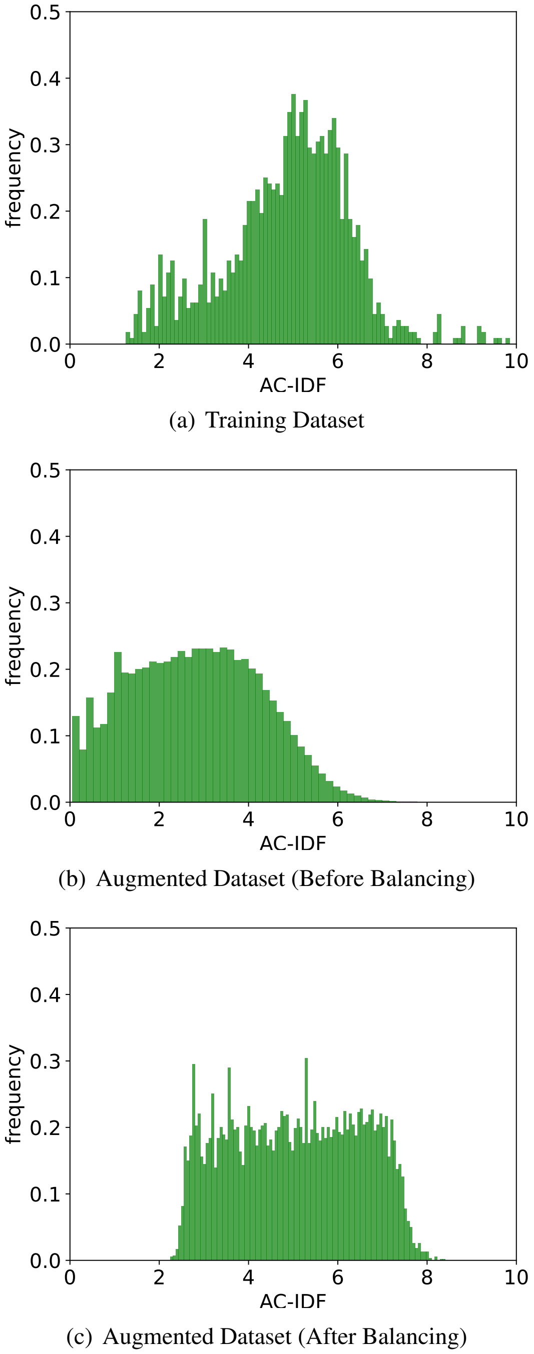
We conduct experiments on two publicly available ASQP datasets, namely Rest15 and Rest16 (Zhang et al., 2021a). In these datasets, each sample includes a text as input, with sentiment quads as ground truth. Datasets are split to train, validation, and test sets officially. Table 4 presents the relevant statistics. We also conduct experiments for Aspect Sentiment Triplet Extraction (ASTE) task, which aims to predict (aspect, opinion, sentiment polarity) triplets from the given text. Table 5 presents statistics of four ASTE datasets.

B Experimental Environment and Runtime

All our experiments are conducted with a single NVIDIA Tesla V100 GPU. Our method was implemented using the Hugging Face transformers library (Wolf et al., 2019). The training process of our method on GPU for one run took approximately 50 minutes including 20 minutes for training the Q2T model and 30 minutes for the ASQP model.

C Distribution of Context Difficulty

We present the frequency distribution histogram of the AC-IDF values of texts in the training dataset and augmented datasets in Figure 4. The AC-IDF frequency distribution of the training dataset follows a Gaussian distribution, with most data points falling between AC-IDF values of 4 and 6. However, for the augmented dataset generated without the balancing strategy, most of the data points fall between AC-IDF values of 0 and 4. This indicates that most of the generated texts are relatively simple and differ significantly from the distribution of the training dataset. After applying the balancing strategy, the augmented dataset shows a more uniform distribution of data points between AC-IDF values of 3 and 7. This indicates that the balancing strategy has effectively created a more balanced distribution of sample difficulty.

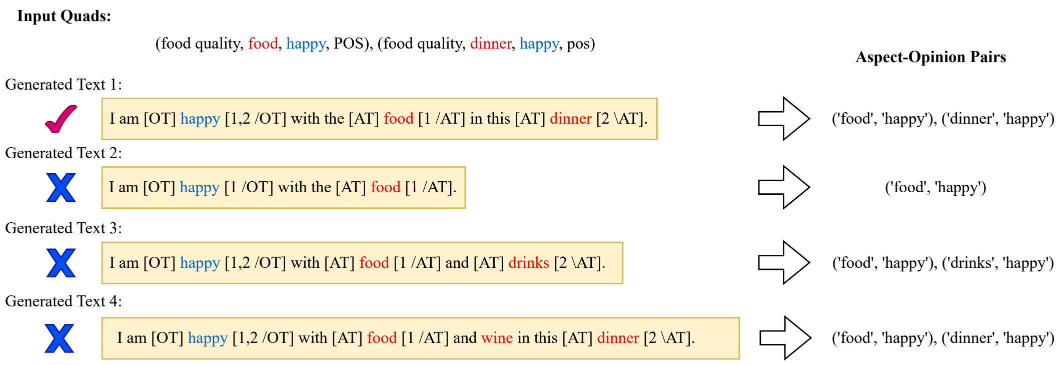


(a) Training Dataset

(b) Augmented Dataset (Before Balancing)

(c) Augmented Dataset (After Balancing)

Figure 4: Frequency distribution histogram for AC-IDF of texts in the training dataset and augmented datasets.


Input Quads:

(food quality, food, happy, POS), (food quality, dinner, happy, pos)

Generated Text 1:

I am [OT] happy [1,2 /OT] with the [AT] food [1 /AT] in this [AT] dinner [2 \AT].

('food', 'happy'), ('dinner', 'happy')

Generated Text 2:

I am [OT] happy [1 /OT] with the [AT] food [1 /AT].

('food', 'happy')

Generated Text 3:

I am [OT] happy [1,2 /OT] with [AT] food [1 /AT] and [AT] drinks [2 \AT].

('food', 'happy'), ('drinks', 'happy')

Generated Text 4:

I am [OT] happy [1,2 /OT] with [AT] food [1 /AT] and wine in this [AT] dinner [2 \AT].

('food', 'happy'), ('dinner', 'happy')

Figure 5: Examples of generative data.

Case 1
Sentence: If there is a line every day of the week for the entire time a place is open, you know it is great. **Predicted Quadruplet:** (restaurant miscellaneous, place, great, positive) **Gold Quadruplet:** (restaurant general, place, great, positive)
Case 2
Sentence: To be honest, I've had better frozen pizza. **Predicted Quadruplet:** (food quality, frozen pizza, better, negative) **Gold Quadruplet:** (food quality, pizza, better, negative)

Table 6: Two error examples of our methods.

D Examples of Generative Text

We present four examples of generative text: one correct example and three low-quality examples. The provided examples illustrate the issues of low-quality generated text and the motivation of our data filtering strategies. The first example is a high-quality one, faithful to the input quads. The second and third examples are low-quality ones that can be filtered out by the first step of the proposed two-step filtering strategy, which checks consistency between the output text and input quads. The fourth example is another low-quality one, which contains an additional aspect that is not present in the input but not annotated by the special markers. Such noisy texts would escape the first-step filtering but can be identified by the second-step filtering.

E Error Analysis

After conducting a comprehensive analysis of the error cases, we present two specific examples to shed light on the challenges encountered by our approach, as illustrated in Figure 6. In the first case, our model incorrectly identifies the predicted aspect category as "restaurant miscellaneous" instead of the correct label "restaurant general." This error highlights a limitation of our model in accurately categorizing certain aspects where the classification boundaries become ambiguous. In the second case, we observe a flaw in aspect extraction. The predicted aspect is "frozen pizza," whereas the correct aspect should have been "pizza." This error reveals that our model sometimes faces difficulties in extracting the precise aspect when there are subtle variations or distinctions within the aspect terms. Consequently, our data augmentation approach may not effectively assist the model when it encounters such challenging instances.