applied sciences

MDPI

*Article*

# Fine-Grained Sentiment-Controlled Text Generation Approach Based on Pre-Trained Language Model

**Linan Zhu, Yifei Xu, Zhechao Zhu, Yinwei Bao and Xiangjie Kong ***

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
* Correspondence: xjkong@ieee.org

**Abstract:** Sentiment-controlled text generation aims to generate texts according to the given sentiment. However, most of the existing studies focus only on the document- or sentence-level sentiment control, leaving a gap for finer-grained control over the content of generated results. Fine-grained control allows a generated review to express different opinions toward multiple aspects. Some previous works attempted to generate reviews conditioned on aspect-level sentiments, but they usually suffer from low adaptability and the lack of an annotated dataset. To alleviate these problems, we propose a novel pre-trained extended generative model that can dynamically refer to the prompt sentiment, together with an auxiliary classifier that extracts the fine-grained sentiments from the unannotated sentences, thus we conducted training on both annotated and unannotated datasets. We also propose a query-hint mechanism to further guide the generation process toward the aspect-level sentiments at every time step. Experimental results from real-world datasets demonstrated that our model has excellent adaptability in generating aspect-level sentiment-controllable review texts with high sentiment coverage and stable quality since, on both datasets, our model steadily outperforms other baseline models in the metrics of BLEU-4, METETOR, and ROUGE-L etc. The limitation of this work is that we only focus on fine-grained sentiments that are explicitly expressed. Moreover, the implicitly expressed fine-grained sentiment-controllable text generation will be an important puzzle for future work.

**Keywords:** artificial intelligence; natural language processing; controllable text generation; review generation; pre-trained language model; fine-grained sentiment



**Citation:** Zhu, L.; Xu, Y.; Zhu, Z.; Bao, Y.; Kong, X. Fine-Grained Sentiment-Controlled Text Generation Approach Based on Pre-Trained Language Model. *Appl. Sci.* **2023**, *13*, 264. https://doi.org/10.3390/app13010264

Academic Editor: Antonio Moreno

Received: 15 November 2022
Revised: 11 December 2022
Accepted: 20 December 2022
Published: 26 December 2022



## 1. Introduction

In recent years, Transformer-based pre-trained language models (LMs) have greatly improved the state-of-the-art of natural language processing tasks as well as natural language generation (NLG). Large-scale autoregressive Transformer models [1] that leverage large amounts of unannotated data and a simple log-likelihood training objective have achieved remarkable results in many text-generation tasks, such as machine translation, text summarization, and text style transfer. Meanwhile, for other real-world text-generation applications, such as review generation and essay writing, users prefer the generated text to be more controllable. However, since the LMs are trained on unannotated data, controlling attributes of generated text becomes difficult without modifying the model architecture to allow for extra input attributes or fine-tuning with attribute-specific data [2,3]. Therefore, some approaches, such as Plug-and-Play-Language-Models (PPLM) [4], control generated text through attribute models without changing the architecture or weights of pre-trained LMs. These models usually regard controllable text generation as generating tasks conditioned on the attributes, such as topic and sentiment at the sentence- or document-level, leaving a gap for finer-grained (e.g., aspect-level) control over the content of generated texts.

The fine-grained sentiment-conditioned text-generation task aims to automatically generate a highly relevant statement when given a series of fine-grained sentiments (e.g.,

aspect-opinion, aspect-sentiment) as input. Zang and Wan [5] first introduced the aspect-sentiment information to perform aspect-level sentiment-controllable review generation. They conducted conditional training by adopting a supervised method requiring a large dataset annotated with sentence-level aspect-sentiment labels. However, very few datasets provide such sufficient fine-grained labels, and it is also labor-intensive and time-consuming to conduct annotation on all data instances. Chen et al. [6] proposed a mutual learning framework leveraging large unlabeled data through interactive learning between the generator and the classifier. Besides the aspect-sentiment, aspect-opinion pairs also express aspect-level sentiment information. Therefore, inspired by them, in this work, we introduce the aspect-opinion information into the fine-grained sentiment-controllable text generation.

The aspect-opinion pairs represent the fine-grained sentiments that could be expressed within a review sentence, where the aspect term refers to the target of an opinion, and the opinion term refers to the sentimental words that describe the aspect term. For example, in the sentence of Figure 1, ("hotdog", "better") is an aspect-opinion pair, where "hotdog" is an aspect term, and "better" is an opinion term, together they form the backbone of fine-grained sentiment in the review text. Therefore, the aspect-opinion conditioned generation task aims to generate a review text $X$ that correctly contains the sentiment information from $n$ non-repeated aspect-opinion pairs $(\mathrm{a}, \mathrm{o})_{1:n}$. Most previous works [5,7,8] used the aspect-polarity pairs rather than the aspect-opinion pairs, and they used a straightforward data-to-text modeling approach, which is much more difficult due to the discrete and sparsity of the input data. To tackle this problem, relying on the natural characteristics of aspect-opinion pairs directly presented in sentences, our approach proposed a query-hint mechanism as a dynamic prompt strategy to guide the generation direction. Furthermore, in order to guarantee the quality of the generated results, in the generator, we incorporate a GPT-2 345M model [9] as the "super generator," then by extending this state-of-the-art model with our proposed query-hint mechanism and our sentiment control loss function to guide the generating process toward the given controlling information. Moreover, to further enhance the generator's performance, with the assistance of a classifier by extracting the fine-grained sentiments, we leveraged a large unlabeled dataset to train the generator. The experimental results demonstrate the effectiveness of these components.

| Sentence: |
| --- |
| Their hotdog is better compared with tasteless bread. |
| Aspect-Opinion Pairs: |
| {(hotdog, better), (bread, tasteless)} |

**Figure 1.** An illustrative example of how the aspect-opinion pairs are expressed in a review sentence. The terms highlighted in red and blue are aspect terms and opinion terms, respectively.

**Our Contributions:**

- We propose our conditional generative model by extending a pre-trained state-of-the-art Transformer-based generative model with our introduced query-hint mechanism and sentiment control loss function to further guide the text generation at a finer-grained level.
- To better model a text-to-text schema, we introduce the aspect-opinion pair as the fine-grained sentiment unit to control the constrained text generation.
- Through employing an auxiliary classifier, we leverage a large unannotated dataset to re-train and fine-tune an end-to-end conditioned text generative model.

The remainder of this paper is organized as follows. Section 2 discusses the related works in controlled text generation, including the review generation and the aspect-level sentiment-controlled generation, which is less studied. Section 3 introduces our proposed approach that achieved finer-grained sentiment control in generation. In Section 4, the experimental settings are detailed, and evaluation metrics and results are also discussed to

demonstrate the validity of our approach. Finally, we conclude this work in Section 5 while discussing future work.

## 2. Related Work

### 2.1. Controlled Text Generation

Recently, there has been many studies that aim to generate text conditioned on input attributes with neural networks. Some of the earlier efforts have studied this controlled text generation by training a conditional generative model [10,11] while fine-tuning pre-trained models with Reinforcement Learning (RL) [3] and training a Generative Adversarial Network [12] have also shown inspiring results. The Conditional-Transformer-Language (CTRL) model [2] is a recent approach that trains a language model conditioned on a variety of control codes (e.g., "Reviews" and "Legal" control the model to generate reviews and legal texts, respectively), which prepended meta-data to the text during generation. Although it uses a GPT-2-like architecture to generate high-quality text, the result is at the cost of fixing the control codes and training a very large model. PPLM [4] composed a pre-trained LM with attribute controllers guiding text generation toward the desired attribute. At the same time, its flexible design allows it to control the generating process through relatively small "pluggable" attribute models while keeping parameters in the LM fixed. Chan et al. [13] incorporated a pre-trained GPT-2 model with a Content-Conditioner (CoCon) to control the generated text under the guidance of target text content. Yu et al. [14] proposed a simple and flexible method, infusing attribute representations into a pre-trained unconditional LM without changing the LM parameters to achieve sentiment- and topic-controlled generation. Different from our fine-grained sentiment-controlled text-generation (FSCTG) task, these works focus on sentence-based sentiment and topic control in text generation. In the FSCTG task, the text-generation process is controlled by a series of fine-grained sentiments (e.g., aspect-opinion or aspect-sentiment).

### 2.2. Review Generation

Review generation [7,15], a generation task aiming to automatically generate review text, is a related area that generates reviews conditioned on the given information. While most of the previous approaches [7,8] have framed review generation as A2T (Attribute-to-Text problem), leaving a gap between attributes (e.g., user, product, and rating) and linguistic data. To tackle this problem, Kim et al. [16] proposed AT2T (Attribute-matched-Text-to-Text) by augmenting inductive biases of attributes with matching reference reviews to learn the rich representations of attributes.

### 2.3. Aspect-Level Sentiment Control

Nevertheless, most of these works only focus on sentence-level sentiments and ignore the aspect-level sentiment control, and very few researchers have studied generating reviews from fine-grained sentiments due to the lack of announced data. Zang and Wan [5] gave the first attempt to generate reviews from aspect-sentiment scores, which requires the reviews with sentence-level aspect-sentiment score annotations. This makes it impractical in real-world applications due to the lack of labeled data. To tackle this problem, Chen et al. [6] proposed a semi-supervised aspect-level sentiment-controllable review generation method, under their proposed mutual learning framework with the assistance of a classifier, it can take advantage of large-scale unlabeled data to achieve aspect-level sentiment control in review generation with few labeled data. Fei et al. [17] combined fine-grained sentiment classification and generation tasks as a joint dual learning system, strengthening the mutual connection of both tasks. To overcome the defect of sparsity and discrete nature brought by the input data in the data-to-text scheme, Yuan et al. [18] proposed a hierarchical template-transformer (HTT); they split the generation task into two corresponding pipeline subtasks, i.e., opinion phrase generation and review composition, which were jointly trained on the HTT. Although in different ways, they all trained an efficient end-to-end generative model. However, they did not attempt to dynamically adjust the attention weights during the

model's generation process since some contents (e.g., the completion of sentiment words generation) are informative to the global generation and need to be notified.

## 3. Method

In this section, we introduce our fine-grained sentiment-controllable text-generation task together with a conditional generative model named **A**spect-**l**evel **Se**ntiment **Cond**itioner (**AlSeCond**), which was trained with both labeled and unlabeled data to learn a fine-grained sentiment review generator with the assistance of a classifier.

First, we give the formalization of our fine-grained sentiment-controllable text-generation task. Specifically, given the fine-grained sentiment units (i.e., aspect-polarities or aspect-opinions) as the input $s$, the model generates a target text $X$ that covers the input sentiments. As a straightforward approach, as other studies have used [5,7,8], the data-to-text modeling can be much more difficult when compared with the text-to-text modeling due to the discrete and sparsity of the input data [17]. Therefore, in this work, we consider a translation of this task to the text-to-text formulation. More conveniently, given aspect and polarity, it is effortless to retrieve opinion phrases from aspect sentiment triplets (AST [19], i.e., the triplet of aspect, opinion, and sentiment polarity) extracted from the review text. This work, therefore, set $s = \{(a_1, o_1), (a_2, o_2), \ldots, (a_n, o_n)\}$ and aims to generate a review text $X$ comprising $m$ words ($X = \{x_1, x_2, \ldots, x_m\}$), which presents each aspect phrase $a_i$ and its corresponding opinion phrase $o_i$ ($i \in \{1, 2, \ldots, n\}$) properly.

In this task, we have a labeled dataset $L$ and an unlabeled dataset $U$. In the labeled dataset $L$, each labeled datum $\ell \in L$ comprises a review text and a list of aspect-opinion phrase pairs $s$, i.e., $\ell = \langle X, s \rangle$, while in the unlabeled dataset $U$, each $u \in U$ only contains a review text, i.e., $u = \langle X \rangle$.

In the following subsections, we first introduce our main framework for how to train a generator on both labeled and unlabeled datasets. Then, we explain our generator and classifier in detail.

### *3.1. Main Framework*

To make full use of both the limited labeled dataset and the large unlabeled dataset, inspired by Chen et al. [6], in the case of a text generator $G$, our proposed method additionally employs a sentiment classifier $C$, which is incorporated to extract all aspect sentiment triplets (aspect, opinion, polarity) in each sentence through a sequence-labeling schema, thus yielding pseudo labels for the unlabeled dataset. We assume that the generator can enhance itself by leveraging a large dataset with pseudo labels predicted by the classifier.

In order to benefit from both the data size of the unlabeled dataset and the correctness of the labeled dataset, we train our model sequentially using these two datasets. Specifically, as shown in Figure 2, following Chen et al. [6], we adopt three steps to make full use of the large unlabeled dataset:

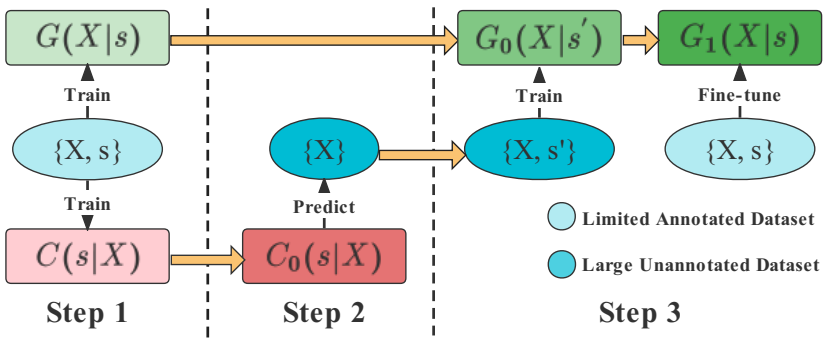


**Figure 2.** Illustration of the training steps for the generator and classifier. Note that "X", "s", "G", and "C" represent the review text, fine-grained sentiment, generator, and classifier, respectively.

- **Step 1:** We train both our generator and classifier on a limited labeled dataset to get *G0* and *C0*, respectively.
- **Step 2:** The *C0* is then used to extract the fine-grained sentiments in the large unlabeled dataset, thus yielding the pseudo labels for the next step's training.
- **Step 3:** Again, the generator is trained on the unlabeled dataset that is attached with pseudo labels. Finally, the generator is fine-tuned with the labeled dataset (used in Step 1) to receive the final generator *G1*.

As a result, we obtain an enhanced generator *G1* trained on both the limited labeled dataset and the large unlabeled dataset.

### *3.2. Generator*

Unconditional language models (LMs) are trained on the huge amount of unlabeled text data to optimize the probability of $p(x_i|x_1{:}x_{i-1})$ in an auto-regressive manner [20,21] where $x_i$ is the next token and $x_1{:}x_{i-1}$ are the previous tokens. While in the controlled text generation, the conditional distribution $p(x_i|a, x_1{:}x_{i-1})$ is optimized, where $a$ is the attribute for the model to control the generation.

To make use of the LM pre-trained with large unlabeled datasets, we need to infuse attribute $a$ into the unconditional distribution $p(x_i|x_1{:}x_{i-1})$. What is more, the pre-trained Transformer-based language model GPT-2 [9] has demonstrated remarkable natural text generation in an auto-regressive manner in recent years. Thereby, to improve the generated texts' quality, our generative model incorporates a pre-trained GPT-2 model as the "super-generator," and we further use the fine-grained sentiment infusion blocks, which are stacked in the AlSeCond to extend this pre-trained state-of-the-art language model's decoder blocks.

Essentially, the GPT-2 model is stacked with numerous Transformer-Decoder blocks, each consisting of layer normalization [22], multi-head self-attention [1], and position-wise feed-forward operations. Therefore, our AlSeCond blocks extend this kind of decoder block and incorporate a sentiment infusion operation together with our proposed query-hint mechanism to conditionally infuse the fine-grained sentiments into the next-token prediction process.

The sentiment infusion operation is performed inside the AlSeCond's blocks. Figure 3 briefly illustrated how our AlSeCond model works. Specifically, the target fine-grained sentiment pairs **s0** are appended sequentially as a prompt to the head of the regular sequence **s1** to form the **S**. This special appended sequence **S** is then encoded to $h$ ($h = [h^0; h^1]$, $h^0$, $h^1$ is the hidden representation of **s0** and **s1**, respectively) through numerous AlSeCond blocks, thus $h_t^1$ self-attends to the hidden states of the regular sequence $h^1$ for previous $t$ time steps and, further, all time steps of the fine-grained sentiment pairs $h^0$. Therefore, the sentiment representation $h^0$ is infused into the intermediate representation $h^1$ to control the next token logits ($o$) and hence the generation process.

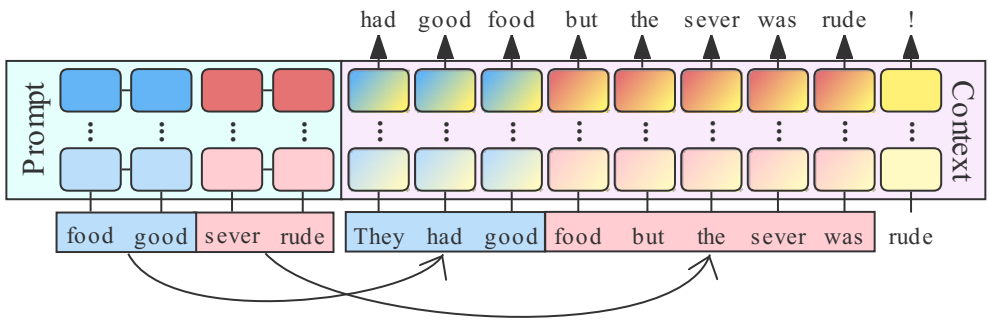


**Figure 3.** Illustration of how the AlSeCond model works. The curved arrow indicates where the sentiment unit should be hinted to the review sequence. The gradient color in the square indicates that this step is affected by the query-hint mechanism with prompt values brought by it.

Our AlSeCond's block (illustrated in the pink block in Figure 4) is a special Transformer-Decoder block that incorporates our proposed query-hint mechanism to guide the controlled generation process. Specifically, for fine-grained sentiment-appended hidden states, $h = [h^0; h^1]$ ($h^0$ and $h^1$ are the hidden states for the sentiment and regular sequence, respectively.), its key, value, and a special hinted query matrix ($K, V, Q' \in R^{(l_s+t)\times d}$, $l_s$, $t$ is the length of the appended sentiments and regular sequence, respectively) are computed to perform a query-hinted self-attention. Furthermore, during the computation of the hinted query ($Q'$) matrix, we infuse $K^0 \in R^{l_s \times d}$, the sentiments' part of $K$, into $Q^1 \in R^{t\times d}$ at their corresponding time step as the query-hint:

$$\begin{aligned} Q &= [Q^0; Q^1] = h \times W_q^T, \quad K = [K^0; K^1] = h \times W_k^T \\ Q' &= [Q^0; Q'^1], \qquad\qquad Q'^1 = f_{hint}(K^0, Q^1) \times W_{q'}^T \end{aligned} \tag{1}$$

$$f_{hint}(K^0, Q^1) = Q^1 + M_h \times \begin{bmatrix} Mean(K_{0:l_1}) \\ Mean(K_{l_1:l_2}) \\ \dots \\ Mean(K_{l_{n-1}:l_n}) \end{bmatrix}$$

where $f_{hint}(\cdot)$ is our proposed function, it strategically allocated the sentiments' representation to $Q^1$ as the query-hint information, and $M_h \in R^{t\times n}$ is an adjacency matrix, representing which sentiment pair should be hinted for each time step in $Q^1$, and $n$ is the number of sentiment pairs, $l_a$ ($a \in \{1, 2, \dots, n\}$) is the end index of the $a$-th sentiment pair in **S**. As a result, we guide the text generation by infusing the sentiment information into the generation process through the query-hinted self-attention operation.

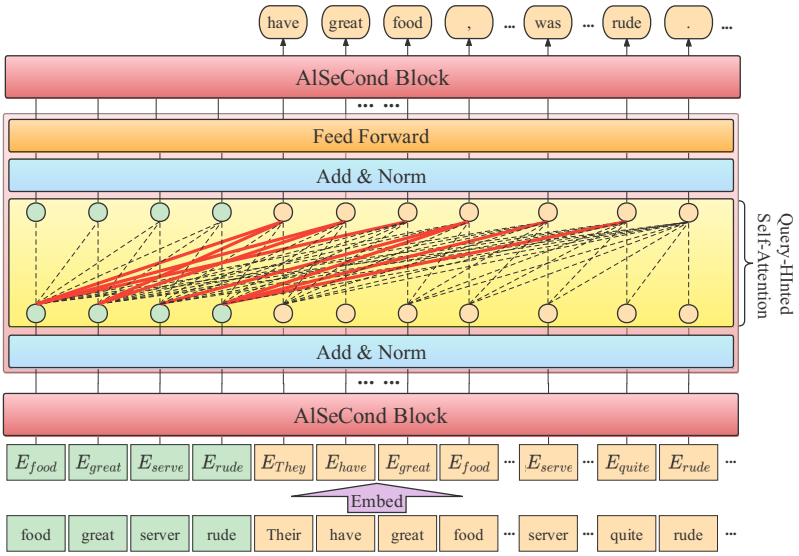


**Figure 4.** Architecture of the generator. This model is stacked with 24 AlSeCond blocks with the same structure. The dashed lines in the block represent the general attention, while the red solid lines represent the attention that is hinted at with prompt key values.

### *3.3. Query-Hint Mechanism*

Since the distance from the prompt and the next-token prediction correlates negatively with the prompt's influence [23], which makes it difficult to use a prompt to guide a non-adjacent piece of text, especially when the generation time step is far away from the prompt. In other words, prompt and regular sentences share equal importance, which is inadequate for prompt-based generative models because the prompt tokens propagate less dominant information to the next-token prediction as the sequence expands. Our idea is similar to Xia et al. [24], where the actual importance of information from different sentiment units is unequal to each token in a sentence, so they need to be attended to differently. Therefore,

as mentioned in Section 3.2, we introduced a query-hint mechanism to further remind each generation time step about the following content. The main idea of this mechanism is to let the generation process understand what text to generate in order to catch the next sentiment text.

Specifically, for each general sentiment pair, its aspect and opinion phrases have their own corresponding subsequence to provide query-hints. As shown in Figure 5 (e.g., 1 to 1), a sentiment pair's member starts query-hint at the beginning of the sentence or the end step of the previous sentiment pair and closes before its own full-presenting. The hinted steps form a "hint-unit" (framed in the red dotted line in Figure 5).

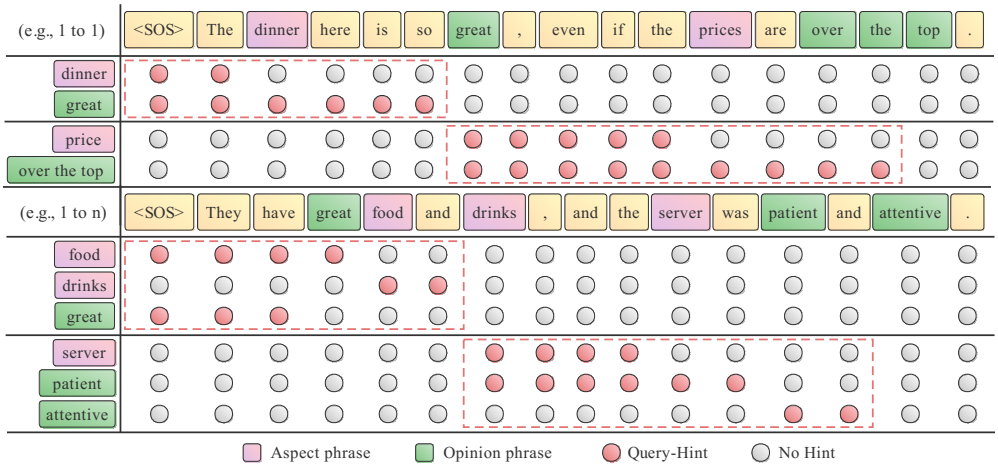


**Figure 5.** Strategy of the query-hint mechanism, this illustration demonstrates two different instances of query-hint strategy, i.e., "1 to 1" and "1 to $n$," which correspond to the one-to-one and one-to-many situations for aspect-opinion pairs, respectively.

In the source sentences, however, there are also some sentiment pairs that share the same phrase either in aspect or opinion (e.g., (food, great), (drinks, great)). Therefore, in order to make query-hint consistent in the training and generation process, given $n$ sentiment pairs that share the same aspect/opinion phrase, their query-hints are merged into one "hint-unit". As shown in Figure 5 (e.g., 1 to $n$), within the "hint-unit", each aspect/opinion phrase gives the query-hint sequentially.

Although our proposed strategy of query-hint in the training process is almost identical to the generation process, there is still a slight difference between them. During the training process, the corresponding time steps in the sentence are provided with query-hint according to the position of each sentiment information presented in the sentence. While in the generation process, since the part of the sentence that has not been generated is unknown, query-hint should be allocated according to the generated part of the sentence.

### *3.4. Loss Functions*

**Generation loss function:** through an LM training objective, we train our conditional generative model with the general generating loss term conditioned on previous $x_{:t-1}$ and input sentiment information $s$:

$$\mathcal{L}_G = -\sum_t log[p(x_t{}'|s, x_{:t-1})]_{I^x(x_t)} \tag{2}$$

where $x_t{}'$ is the predicted token at time step $t$. $I^x(\cdot)$ is the index function of a vector.

**Sentiment control loss function:** To encourage the generator to output texts incorporating the input sentiment information (phrases), we train the generator additional with our proposed sentiment-control loss function. The main idea of this loss function is to maximize the probability value of the one with the highest probability in terms of given aspect/opinion word from all the next-word predictions of a sentence. Specifically, for

every aspect phrase $a$ and opinion phrase $o$ presented in the source text, the training loss is defined as:

$$\begin{aligned}
&\mathcal{L}_{Senti} = \mathcal{L}_a + \mathcal{L}_o \\
&\mathcal{L}_a = -\sum_a \sum_t log[Q(x', Mask_{a,t})]_{I^x(x_{a,t})} \\
&\mathcal{L}_o = -\sum_o \sum_t log[Q(x', Mask_{o,t})]_{I^x(x_{o,t})} \\
&Q(x, Mask) = Mask \odot p_{max}(x) + (1 \oplus Mask) \times \phi_{mean} \\
&p_{max}(x) = MaxPooling([p(x_1|s, x_{:0}); p(x_2|s, x_{:1}); \cdots; p(x_t|s, x_{:t-1})])
\end{aligned} \tag{3}$$

where $\mathcal{L}_a$ and $\mathcal{L}_o$ are the losses for aspect and opinion term inclusion, respectively. $Mask_{a,t/o,t}$ is a one-hot vector with the size of $\mathcal{V}$ (vocabulary size), and only the element in the index of $a_t/o_t$ is 1. $\phi_{mean}$ is a hyper-parameter controlling how much the prediction of aspect/opinion terms should be enhanced. $p_{max}(\cdot)$ is a max-pooling operation with a kernel size of $l_t \times 1$ ($l_t$ is the length of the target text). $\odot$ and $\oplus$ represent the element-wise product and XOR, respectively.

As a result, our final loss function comprehensively considers the loss of generation quality and the loss of sentiment control:

$$\mathcal{L}_{total} = \lambda_G \mathcal{L}_G + \lambda_{Senti} \mathcal{L}_{Senti} \tag{4}$$

where $\lambda$ values are hyper-parameters controlling how much the loss terms dominate the training.

### *3.5. Classifier*

In this section, first, we give the task definition of Aspect Opinion Pair Extraction (AOPE), then we briefly introduced the model architecture of our sentiment classifier **C**.

The task of AOPE aims to extract aspect terms and their corresponding opinion terms as pairs [25–27]. This task can be defined as follows: Given a sentence with $m$ words $X = \{x_1, x_2, \ldots, x_m\}$, the goal of this task is to extract all aspect-opinion pairs $\tau = \{(a, o)_n\}_{n=1}^{|\tau|}$ from $X$, where $\{(a, o)_n\}$ is an aspect-opinion pair presented in $X$ and the notations $a$ and $o$ denote an aspect term and an opinion term, respectively.

For the overall architecture of our classifier, the two-dimensional interaction-based multi-task learning framework (2D-IMLF) is shown in Figure 6. Given an input sentence, two highly related works of the extraction task (aspect term extraction and opinion term extraction) are adopted to learn aspect-related and opinion-related features, respectively. Then, to capture different interactive features of aspect terms and opinion terms, a 2D interactive representation is obtained by tensor composition. Finally, the classifier model regards the AOPE task as a grid tagging problem and in the end, obtains the final results by applying a decoding algorithm [28].

As shown in Figure 6, we first use a group of CNN layers to encode the input sentence and get their hidden state:

$$\begin{aligned}
H_k^c &= Conv1D_k(X) \\
H_*^c &= [H_1^c; H_2^c; \ldots; H_k^c] \\
H^c &= Conv1D_3(Conv1D_5(H_*^c))
\end{aligned} \tag{5}$$

where $k \in \{1, 2, 3, \ldots\}$ represents the kernel size of an 1D-CNN. Then, a Bi-LSTM layer together with multi-head self-attention is incorporated to extract the context information from the sentences:

$$\begin{aligned}
H_t^l &= BiLSTM(H_{t-1}^l, H_t^c) \\
H_c &= MultiHeadAttention(H^l)
\end{aligned} \tag{6}$$

Afterward, we concatenate the hidden state $H_c$ with their transferring state $H_c^T$ to get a grid-formed feature. We then obtain the prediction probabilities of $P_a^c$ and $P_o^c$ for aspect and opinion terms, respectively, from the final logits $P$:

$$\begin{aligned} \hat{O}_c &= [H_c; H_c^T] \\ P &= Linear(\hat{O}_c) \end{aligned} \tag{7}$$

Finally, by using a grid-formed tagging schema [28], we can easily obtain a series of aspect-opinion pairs.

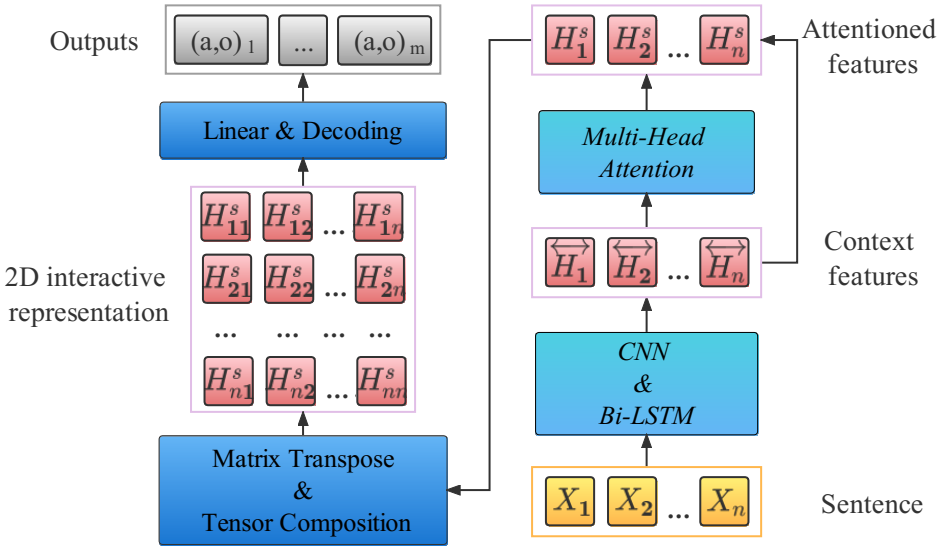


**Figure 6.** Architecture of the classifier. This model incorporates 2D interaction representation and grid-formed tagging schema [28] to extract all aspect and opinion phrases in a sentence.

## 4. Experiments

In this section, we first introduce datasets and settings in our experiment and then report the evaluation metrics and results.

### *4.1. Dataset and Settings*

We conduct experiments on three real-world datasets, two labeled and one unlabeled; the statistics of the datasets are reported in Table 1. Moreover, the experimental settings are also listed in this subsection.

**Table 1.** Statistics of the labeled and unlabeled datasets. Note that "Val" is short for "Validation", the ASTE-Data-V2-Rest is labeled with aspect, opinion, and polarity, while the MAMS-ASTA is labeled with only aspect and polarity.

| Dataset | | #Instance | #Positive | #Neutral | #Negative | Sentiment Form |
|---|---|---|---|---|---|---|
| ASTE-Data-V2-Rest | Train | 2728 | 3490 | 241 | 1014 | Aspect-Opinion-Polarity |
| | Val | 668 | 841 | 76 | 248 | |
| | Test | 1140 | 1497 | 120 | 376 | |
| MAMS-ASTA | Train | 4297 | 3380 | 5042 | 2764 | Aspect-Polarity |
| | Val | 500 | 403 | 604 | 325 | |
| | Test | 500 | 400 | 607 | 329 | |
| Yelp | - | 1,160,546 | - | - | - | - |

#### 4.1.1. Labeled Dataset

We conduct experiments of aspect-opinion and aspect-polarity pairs of conditioned controllable text generation on English restaurant reviews with ASTE-Data-V2 from Xu et al. [29] and MAMS-ASTA from Jiang et al. [30], respectively.

**ASTE-Data-V2** (https://github.com/xuuuluuu/SemEval-Triplet-data, accessed on accessed on 18 May 2022): From Xu et al. [29], is originally from SemEval Challenges [31–33], and contains both aspect and opinion labels in each review datum. Specifically, we union the 14Rest, 15Rest, and 16Rest included in the ASTE-Data-V2 as our labeled dataset.

**MAMS-ASTA:** From MAMS (https://github.com/siat-nlp/MAMS-for-ABSA, accessed on accessed on 14 May 2022) (Multi-Aspect Multi-Sentiment), ref. [30] is an aspect-level sentiment-labeled dataset. Wherein, each datum instance in MAMS-ASTA is labeled with at least two aspects and different sentiment polarities, while no opinion term is labeled. Therefore, by using our classifier to retrieve opinion phrases according to the original pairs of aspect-polarity, we also conduct aspect-level sentiment-controllable text generation on MAMS-ASTA.

#### 4.1.2. Unlabeled Dataset

To ensure that the training data are in the relevant review domain, we use Yelp's review dataset (https://www.kaggle.com/yelp-dataset/yelp-dataset, accessed on accessed on 18 May 2022) as the unlabeled dataset and filter out the sentences with a length greater than 150. Unlike the labeled datasets, the Yelp dataset did not contain fine-grained sentiment labels. Therefore, we only use the sentences in the unlabeled data and discard other items, including user information.

#### 4.1.3. Experimental Settings

**Generator:** In the experiment, we train our AlSeCond model that extends from a pre-trained GPT-2 medium 345M model [9]. The AlSeCond's blocks clone the GPT-2 Transformer blocks' parameters and settings. To ensure the generator can compute the probability of (and also generate) any string, we apply Byte Pair Encoding (BPE) [34] for the inputs. The max generating length was set to 32. We tune the $\lambda_G$ together with $\lambda_{senti}$ to 1 and 8, respectively. Adam [35] is used for optimization, while the batch size is set to 16, and the learning rate is set to $5 \times 10^{-5}$. During the period of G0, the generator is trained with the labeled and pseudo-labeled dataset for 4 and 2 epochs, respectively. In the following G1, the generator is fine-tuned with the labeled dataset for 24 epochs. We apply the above steps to train our model on an RTX A4000 GPU for 20 h. Furthermore, the above steps are also applied to train other baseline models. We ran our model and all baselines five times to average the scores.

**Classifier:** Following GTS [28], we combine a 300-dimension domain-general embedding from pre-trained GloVe [36] and a 100-dimension domain-specific embedding trained with fastText [37] to initialize double word embeddings. We use Adam as the optimizer, and the learning rate is $5 \times 10^{-4}$. The batch size and dropout rate are set to 32 and 0.5, respectively. The number of hidden units in Bi-LSTM is set to 128.

### *4.2. Baselines*

We compare with six baselines. **PPLM** [4] incorporates an attribute model BoW (bag of words) to steer a pre-trained GPT-2 model toward increasing the generating probability of the target words. In this baseline, the BoW is formed with the words contained in the target sentiment pairs. For **HTT** [18], we omit the process of opinion phrase generation and only use its results (i.e., sentiment pairs) to compose the review. Through prepending the task description before the input text, the state-of-the-art text-to-text model **T5** [38] is pre-trained with a multi-task objective. Following this schema, we append the sentiment pairs into the prompt, thus forming: "*generate a sentence with* $a_1$ *is* $o_1, \ldots,$ $a_n$ *is* $o_n$.", and fine-tune the model with the target sentence. Its coverage of the input sentiment pairs in the baselines serves as an upper bound. Moreover, we also fine-tune **UniLM** [39], **UniLM-v2** [40], and

**BERT-Gen** [40] in a similar sequence-to-sequence fashion with both the large unlabeled dataset and the limited labeled dataset.

### *4.3. Generated Quality Evaluation*

To study the performance of these models in a diversified manner, we conduct evaluations on both the quality and sentiment coverage of the generated text.

#### 4.3.1. Fluency and Diversity Evaluation

We conduct a fluency evaluation on the generated texts with some automatic metrics: BLEU [41], ROUGE [42], and METEOR [43], which compare the similarity between the generated text and ground truth based on n-gram matching. Moreover, the diversity of generations is also an important indicator. We measure diversity for the generated results with Dist-1,-2,-3 [44] scores and Self-Bleu [45].

Table 2 shows the fluency and diversity evaluation results by the automatic evaluations. From the results, we can observe that: (1) Compared with baseline models, our AlSeCond model extended from the GPT-2 achieves better performance in fluency evaluations. (2) Comparing results in diversity metrics, it can be observed that our AlSeCond model performs much better than the rest of the baselines in the MAMS-ASTA dataset, which means the results generated by our model are less like the template-generated text than that generated by other models.

**Table 2.** Results for the fluency and diversity evaluation. Note that "↑" means the higher the better, "↓" means the lower the better, "w/o" means "no".

| Dataset | Models | BLEU-3 (↑) | BLEU-4 (↑) | METETOR (↑) | ROUGE-L (↑) | Self-Bleu-4(↓) | Dist-1 (↑) | Dist-2 (↑) | Dist-3 (↑) |
|---|---|---|---|---|---|---|---|---|---|
| ASTE-Data-V2 | PPLM | 0.196 | 0.032 | 14.078 | 13.827 | **7.939** | 0.0841 | 0.4102 | 0.7180 |
| | HTT | 13.100 | 7.656 | 34.899 | 42.544 | 42.664 | 0.0525 | 0.2356 | 0.4113 |
| | T5-base | 21.246 | 13.216 | 29.007 | 41.092 | 22.580 | 0.1621 | 0.4725 | 0.6101 |
| | T5-large | 24.747 | 16.462 | 29.986 | 43.614 | 23.045 | **0.1721** | 0.4658 | 0.5934 |
| | UniLM | 33.093 | 27.486 | 46.808 | 52.582 | 20.334 | 0.1489 | 0.4961 | 0.6663 |
| | BERT-Gen | 32.693 | 28.050 | 45.223 | 45.162 | 24.149 | 0.1450 | 0.4957 | 0.6411 |
| | UniLM-v2 | 32.159 | 27.525 | 45.107 | 44.514 | 22.830 | 0.1451 | 0.5060 | 0.6553 |
| | AlSeCond | **40.453** | **34.611** | 55.127 | **63.720** | 15.972 | 0.1610 | **0.5439** | 0.7073 |
| | ⌊ w/o sentiment loss | 37.961 | 32.190 | **55.699** | 62.911 | 16.195 | 0.1552 | 0.5301 | 0.7028 |
| | ⌊ w/o query-hint | 34.305 | 29.080 | 55.391 | 61.237 | 14.442 | 0.1551 | 0.5431 | **0.7264** |
| | ⌊ w/o unlabeled dataset | 29.085 | 26.387 | 42.601 | 48.213 | 21.727 | 0.1444 | 0.4942 | 0.6628 |
| MAMS-ASTA | HTT | 2.279 | 0.412 | 17.193 | 23.197 | 51.373 | 0.0602 | 0.2271 | 0.4003 |
| | T5-base | 3.653 | 1.479 | 14.400 | 24.181 | 27.671 | 0.1299 | 0.3761 | 0.5541 |
| | T5-large | 4.212 | 1.767 | 15.180 | 25.828 | 27.626 | 0.1418 | 0.3761 | 0.5591 |
| | UniLM | 3.178 | 1.251 | 18.833 | 23.872 | 37.890 | 0.1032 | 0.3211 | 0.4878 |
| | BERT-Gen | 4.003 | 1.605 | 17.751 | 24.162 | 28.284 | 0.1284 | 0.4024 | 0.5778 |
| | UniLM-v2 | 3.898 | 1.559 | 17.757 | 23.999 | 27.858 | 0.1255 | 0.3989 | 0.5796 |
| | AlSeCond | **5.159** | **2.113** | 19.736 | **31.738** | 13.714 | **0.1627** | 0.5085 | 0.6811 |
| | ⌊ w/o sentiment loss | 4.944 | 1.999 | **23.734** | 31.302 | 14.112 | 0.1477 | 0.4978 | 0.7171 |
| | ⌊ w/o query-hint | 4.208 | 1.635 | 23.661 | 29.497 | **10.835** | 0.1604 | **0.5538** | **0.7653** |
| | ⌊ w/o unlabeled dataset | 3.458 | 1.026 | 20.761 | 28.924 | 15.787 | 0.1478 | 0.4728 | 0.6627 |

#### 4.3.2. Sentiment Evaluation

As to measure the quality of sentiment containment in the generated sentence and indicate whether the input sentiments are correctly expressed in the generated text, we employ two metrics: **Coverage** (Cov.), just like in Lin et al. [46], which is the average rate of input sentiment pairs presented in the generated texts. This metric includes Cov-a, Cov-o, and Cov-ao, representing the presenting rate of aspect, opinion, and aspect-opinion pairs, respectively. **Accuracy** (Acc.) is a rate indicating how many fine-grained sentiments are accurately expressed in the sentence, and it is evaluated by the external sentiment classifier [30] trained on MAMS-ASTA.

Table 3 shows the results of sentiment coverage and accuracy for generated texts. It is worth noting that for a linguistically complicated sentence, its aspect-level sentiments are more difficult to be correctly predicted by the external classifier than a relatively simple sentence, so its sentiment accuracy may be lower than the actual situation. What is more, T5's original seq2seq architecture allows it to generate texts that highly correspond to

the input sequences. Hence its coverage and accuracy scores serve as an upper bound, although its generated results' syntax is relatively simple and repetitive.

**Table 3.** Results for the sentiment evaluation. Note that Accuracy (Acc.) is a rate indicating how many fine-grained sentiments are accurately expressed in the sentence, and it is automatically evaluated by an external classifier.

| Dataset | Models | Cov-a | Cov-o | Cov-ao | Acc. |
|---|---|---|---|---|---|
| ASTE-Data-V2 | PPLM | 0.3597 | 0.3642 | 0.1094 | 0.1761 |
| | HTT | 0.7689 | 0.7773 | 0.6050 | 0.6328 |
| | T5-base | 0.9563 | 0.9764 | 0.9403 | 0.7812 |
| | T5-large | 0.9633 | 0.9839 | 0.9508 | **0.7948** |
| | UniLM | 0.9513 | 0.9568 | 0.9182 | 0.7450 |
| | BERT-Gen | 0.9352 | 0.9343 | 0.8886 | 0.7521 |
| | UniLM-v2 | 0.9438 | 0.9488 | 0.9087 | 0.7475 |
| | AlSeCond | **0.9824** | **0.9849** | **0.9734** | 0.7771 |
| | ⌊ w/o sentiment loss | 0.9633 | 0.9649 | 0.9468 | 0.7683 |
| | ⌊ w/o query-hint | 0.9412 | 0.9313 | 0.8966 | 0.7443 |
| | ⌊ w/o unlabeled dataset | 0.8158 | 0.8841 | 0.7556 | 0.6306 |
| MAMS-ASTA | HTT | 0.7203 | 0.5123 | 0.3800 | 0.4532 |
| | T5-base | 0.9610 | 0.9147 | 0.9042 | 0.5734 |
| | T5-large | 0.9738 | 0.9453 | 0.9416 | 0.5698 |
| | UniLM | 0.9251 | 0.7821 | 0.7590 | 0.5883 |
| | BERT-Gen | 0.9438 | 0.8009 | 0.7807 | 0.6048 |
| | UniLM-v2 | 0.9341 | 0.7515 | 0.7305 | **0.6310** |
| | AlSeCond | **0.9798** | **0.9588** | **0.9558** | 0.6267 |
| | ⌊ w/o sentiment loss | 0.9318 | 0.8952 | 0.8825 | 0.6050 |
| | ⌊ w/o query-hint | 0.8338 | 0.6811 | 0.6257 | 0.5447 |
| | ⌊ w/o unlabeled dataset | 0.7829 | 0.7095 | 0.6325 | 0.5157 |

Comparing the above metrics results for all models on different datasets, we can observe that our model has stable advantages over both ASTE-Data-V2 and MAMS-ASTA, which indicates that our AlSeCond model has stronger adaptability. Additionally, Figure 7 presents the learning curves for fine-tuning all models with the labeled dataset, which also demonstrates the strong capabilities of our model compared to baselines.

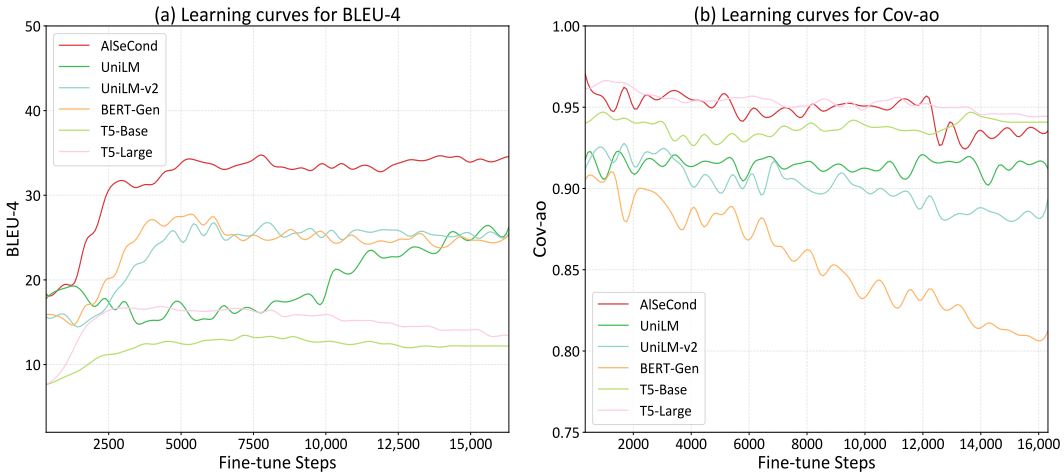


**Figure 7.** Learning curves for fine-tuning models with the labeled dataset. (**a**) illustrated the learning curves for BLEU-4 changing with fine-tuning steps. (**b**) illustrated the learning curves for Cov-ao changing with fine-tuning steps.

### *4.4. Case Study*

Figure 8 presents some generated cases from AlSeCond, HTT, T5, UniLM, BERT-Gen, and UniLM-v2. From the cases, we found that: AlSeCond tends to generate more linguistically complicated sentences, while the other baselines are more likely to focus on generating review texts that simply express the input information and less on the complexity of the expressions and the syntaxes.

Aspect-level Sentiments: {wait staff - friendly, meal - great}
**AlSeCond:** the wait staff is very friendly and will take great care of you, if you end up getting a great meal, they 'll even throw in some dessert.
**HTT:** the wait staff is very friendly and you get a great deal for a great meal at a great price and a great value.
**T5-Large:** wait staff was friendly and the meal was great.
**UniLM:** The wait staff is friendly and you always have a great meal and always leave feeling satisfied.
**BERT-Gen:** the wait staff is very friendly and always has a great meal.
**UniLM-v2:** wait staff is friendly and we have always had a great meal!
Aspect-level Sentiments: {hostess - kind, hostess - gracious}
**AlSeCond:** It's always a delight to have greeted by a kind and gracious hostess.
**HTT:** the hostess was very kind, gracious, and made us feel at home with a smile.
**T5-Large:** the hostess was kind and gracious.
**UniLM:** The hostess was very kind and gracious.
**BERT-Gen:** the hostess is very kind and gracious.
**UniLM-v2:** our hostess and all of the people helping her were kind and gracious.
Aspect-level Sentiments: {atmosphere - cozy, service - horrible}
**AlSeCond:** When I sat down at the bar the atmosphere was cozy but service was horrible.
**HTT:** the atmosphere is very cozy, but the service is horrible, i would never go there again.
**T5-Large:** the atmosphere is cozy, but the service is horrible.
**UniLM:** The atmosphere is very cozy but the service is horrible.
**BERT-Gen:** cozy atmosphere and horrible service.
**UniLM-v2:** cozy atmosphere but horrible service
Aspect-level Sentiments: {place - hidden away, place - worth}
**AlSeCond:** The place is a little hidden away, but it is worth finding.
**HTT:** this place is a must if you want to try it, it's a must and worth it.
**T5-Large:** hidden away and worth every penny.
**UniLM:** This place is so hidden away but is def worth an hour drive.
**BERT-Gen:** this place is hidden away but is def worth it.
**UniLM-v2:** this place is a bit hidden away but is worth it.

**Figure 8.** Generated samples from the generative models. Red phrases represent the aspect-level sentiment formed by aspect-opinion pairs.

## 5. Conclusions and Future Work

In this paper, we propose a fine-grained sentiment-controllable text-generation method based on the pre-trained language model and the auxiliary sentiment classifier that utilizes both the labeled and unlabeled dataset to reach the aspect-level sentiment control in text generation. Our proposed query-hint mechanism and fine-grained sentiment control loss function have greatly enhanced the generator in controlling the sentiment during the text-generating process. Experiments on real-world datasets have demonstrated our generator's ability to generate aspect-level sentiment-controllable review statements with high quality and diverse syntax.

For future work, we will explore the controllable text generation for implicitly expressed fine-grained sentiments (e.g., in this sentence: *"We had to constantly ask the waiter to top up water glasses."*, the reviewer had a negative opinion of the waiter although there is no related opinion phrase in the sentence.), since the query-hint mechanism proposed in this paper is only effective for explicitly expressed fine-grained sentiments.

**Author Contributions:** Conceptualization, L.Z. and Y.X.; methodology, Y.X.; software, Y.X.; validation, Y.X. and Z.Z.; formal analysis, Y.B.; investigation, L.Z. and Y.B.; resources, Y.X. and X.K.; data curation, Y.X. and X.K.; writing—original draft preparation, Y.X. and Y.B.; writing—review and editing, Y.X. and Y.B.; visualization, Y.X. and Z.Z.; supervision, L.Z. and X.K.; project administration, Y.X.; funding acquisition, L.Z. and X.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the National Natural Science Foundation of China (No. 62176234, 62072409).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The source code used to generate the results shown in this paper is available at https://github.com/ashooha0/Alsecond, accessed on 1 November 2022. The dataset, attached with pseudo labels by our classifier, is available at: https://drive.google.com/file/d/1HjyTLBBlyAOn_pphWC6VjgWQ2HPZglAp/view?usp=share_link, accessed on 15 November 2022. The ASTE-data-V2 dataset is available at https://github.com/xuuuluuu/SemEval-Triplet-data, accessed on accessed on 18 May 2022, and the MAMS dataset is available at https://github.com/siat-nlp/MAMS-for-ABSA, accessed on accessed on 14 May 2022.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| LMs | Language Models |
| NLG | Natural Language Generation |
| GPT | Generative Pre-Training |
| PPLM | Plug and Play Language Model |
| CTRL | Conditional-Transformer-Language |
| CoCon | Content-Conditioner |
| FSCTG | Fine-grained Sentiment-Controlled Text Generation |
| A2T | Attribute-to-Text |
| AT2T | Attribute-matched-Text-to-Text |
| HTT | Hierarchical Template-Transformer |
| AlSeCond | Aspect-level Sentiment Conditioner |
| AOPE | Aspect Opinion Pair Extraction |
| 2D-IMLF | Two-Dimensional Interaction-Based Multi-task Learning Framework |
| CNN | Convolutional Neural Networks |
| Bi-LSTM | Bidirectional Long Short-Term Memory |
| Val | Validation |
| AST | Aspect Sentiment Triplet |
| ASTE | Aspect Sentiment Triplet Extraction |
| MAMS-ASTA | Multi-Aspect Multi-Sentiment Aspect-Term Sentiment Analysis |
| BPE | Byte Pair Encoding |
| GTS | Grid Tagging Scheme |
| GloVe | Global Vectors |
| UniLM | Unified Language Model |
| BERT | Bidirectional Encoder Representations from Transformer |
| BLEU | Bilingual Evaluation Understudy |
| ROUGE | Recall-Oriented Understudy for Gisting Evaluation |
| METEOR | Metric for Evaluation of Translation with Explicit Ordering |
| Dist | Distinct |
| Cov | Coverage |
| Acc. | Accuracy |

## References


1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
2. Keskar, N.S.; McCann, B.; Varshney, L.R.; Xiong, C.; Socher, R. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv* **2019**, arXiv:1909.05858.
3. Ziegler, D.M.; Stiennon, N.; Wu, J.; Brown, T.B.; Radford, A.; Amodei, D.; Christiano, P.F.; Irving, G. Fine-Tuning Language Models from Human Preferences. *arXiv* **2019**, arXiv:1909.08593.
4. Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; Liu, R. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. *arXiv* **2019**, arXiv:1912.02164.
5. Zang, H.; Wan, X. Towards Automatic Generation of Product Reviews from Aspect-Sentiment Scores. In Proceedings of the International Conference on Natural Language Generation, Santiago de Compostela, Spain, 4–7 September 2017.
6. Chen, H.; Lin, Y.; Qi, F.; Hu, J.; Li, P.; Zhou, J.; Sun, M. Aspect-Level Sentiment-Controllable Review Generation with Mutual Learning Framework. In Proceedings of the National Conference on Artificial Intelligence, Online, 2–9 February 2021.
7. Dong, L.; Huang, S.; Wei, F.; Lapata, M.; Zhou, M.; Xu, K. Learning to Generate Product Reviews from Attributes. In Proceedings of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017.
8. Sharma, V.; Sharma, H.; Bishnu, A.; Patel, L. Cyclegen: Cyclic consistency based product review generator from attributes. In Proceedings of the International Conference on Natural Language Generation, Tilburg, The Netherlands, 5–8 November 2018.
9. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
10. Kikuchi, Y.; Neubig, G.; Sasano, R.; Takamura, H.; Okumura, M. Controlling Output Length in Neural Encoder-Decoders. *arXiv* **2016**, arXiv:1609.09552.
11. Ficler, J.; Goldberg, Y. Controlling Linguistic Style Aspects in Neural Language Generation. *arXiv* **2017**, arXiv:1707.02633.
12. Yu, L.; Zhang, W.; Wang, J.; Yu, Y. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In Proceedings of the National Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016.
13. Chan, A.T.S.; Ong, Y.S.; Pung, B.T.W.; Zhang, A.; Fu, J. CoCon: A Self-Supervised Approach for Controlled Text Generation. *arXiv* **2020**, arXiv:2006.03535.
14. Yu, D.; Yu, Z.; Sagae, K. Attribute Alignment: Controlling Text Generation from Pre-trained Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Online, 7–11 November 2021; pp. 2251–2268.
15. Lipton, Z.C.; Vikram, S.; McAuley, J. Generative Concatenative Nets Jointly Learn to Write and Classify Reviews. *arXiv* **2015**, arXiv:1511.03683.
16. Kim, J.; Choi, S.; Amplayo, R.K.; Hwang, S-w. Retrieval-Augmented Controllable Review Generation. In Proceedings of the International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020.
17. Fei, H.; Li, C.; Ji, D.; Li, F. Mutual disentanglement learning for joint fine-grained sentiment classification and controllable text generation. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 1555–1565.
18. Yuan, L.; Zhang, X.; Yu, L.C. Hierarchical template transformer for fine-grained sentiment controllable generation. *Inf. Process. Manag.* **2022**, *59*, 103048. [CrossRef]
19. Peng, H.; Xu, L.; Bing, L.; Huang, F.; Lu, W.; Si, L. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8600–8607.
20. Manning, C.D.; Schütze, H. *Foundations of Statistical Natural Language Processing*; MIT Press: Cambridge, MA, USA, 1999.
21. Bengio, Y.; Ducharme, R.; Vincent, P. A Neural Probabilistic Language Model. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 29 November–4 December 2000.
22. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450.
23. Zou, X.; Yin, D.; Zhong, Q.; Ding, M.; Yang, Z.; Tang, J. Controllable Generation from Pre-trained Language Models via Inverse Prompting. In Proceedings of the Knowledge Discovery and Data Mining, Virtual, 14–18 August 2021.
24. Xia, F.; Wang, L.; Tang, T.; Chen, X.; Kong, X.; Oatley, G.; King, I. CenGCN: Centralized Convolutional Networks with Vertex Imbalance for Scale-Free Graphs. *IEEE Trans. Knowl. Data Eng.* **2022**. [CrossRef]
25. Zhao, H.; Huang, L.; Zhang, R.; Lu, Q.; Xue, H. SpanMlt: A Span-based Multi-Task Learning Framework for Pair-wise Aspect and Opinion Terms Extraction. In Proceedings of the Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
26. Chen, S.; Liu, J.; Wang, Y.; Zhang, W.; Chi, Z. Synchronous Double-channel Recurrent Network for Aspect-Opinion Pair Extraction. In Proceedings of the Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
27. Zhu, L.; Xu, M.; Bao, Y.; Xu, Y.; Kong, X. Deep learning for aspect-based sentiment analysis: A review. *PeerJ Comput. Sci.* **2022**, *8*, e1044. [CrossRef] [PubMed]
28. Wu, Z.; Ying, C.; Zhao, F.; Fan, Z.; Dai, X.; Xia, R. Grid Tagging Scheme for End-to-End Fine-grained Opinion Extraction. In Proceedings of the EMNLP (Findings), Online, 16–20 November 2020.
29. Xu, L.; Li, H.; Lu, W.; Bing, L. Position-Aware Tagging for Aspect Sentiment Triplet Extraction. *arXiv* **2020**, arXiv:2010.02609.

30. Jiang, Q.; Chen, L.; Xu, R.; Ao, X.; Yang, M. A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis. In Proceedings of the Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019.
31. Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the International Conference on Computational Linguistics, Dublin, Ireland, 23–29 August 2014.
32. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; Androutsopoulos, I. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Denver, CO, USA, 31 May–5 June 2015.
33. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; Clercq, O.D.; et al. SemEval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the North American Chapter of the Association for Computational Linguistics, San Diego, CA, USA, 12–17 June 2016.
34. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015.
35. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
36. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014.
37. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
38. Liu, P.J.; Matena, M.; Lee, K.; Roberts, A.; Zhou, Y.; Shazeer, N.; Raffel, C.; Narang, S.; Li, W. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
39. Dong, L.; Wang, Y.; Wei, F.; Zhou, M.; Yang, N.; Gao, J.; Hon, H.W.; Liu, X.; Wang, W. Unified Language Model Pre-training for Natural Language Understanding and Generation. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
40. Piao, S.; Dong, L.; Wang, Y.; Wei, F.; Zhou, M.; Yang, N.; Gao, J.; Hon, H.W.; Bao, H.; Liu, X.; et al. UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020.
41. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002.
42. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the Meeting of the Association for Computational Linguistics, Barcelona, Spain, 21–26 July 2004.
43. Lavie, A.; Agarwal, A. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In Proceedings of the Workshop on Statistical Machine Translation, Prague, Czech Republic, 23 June 2007.
44. Brockett, C.; Dolan, B.; Galley, M.; Gao, J.; Li, J. A Diversity-Promoting Objective Function for Neural Conversation Models. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Denver, CO, USA, 31 May–5 June 2015.
45. Zhu, Y.; Lu, S.; Zheng, L.; Guo, J.; Zhang, W.; Wang, J.; Yu, Y. Texygen: A Benchmarking Platform for Text Generation Models. In Proceedings of the International Acm Sigir Conference on Research and Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018.
46. Lin, B.Y.; Zhou, W.; Shen, M.; Zhou, P.; Bhagavatula, C.; Choi, Y.; Ren, X. CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning. *arXiv* **2019**, arXiv:1911.03705.