# Fine-grained Sentiment Controlled Text Generation Approach Based on Pre-trained Language Model

Anonymous ACL submission

## Abstract

Sentiment-controlled text generation aims to generate texts according to the given sentiment. However, most of the existing studies focus only on document- or sentence-level sentiment control, leaving a gap for finer-grained control over the content of generated results. Some previous works attempted to generate reviews conditioned on the aspect-level sentiments, but they usually suffer from low adaptability and the lack of annotated dataset. To alleviate these problems, we propose a pre-trained model extended generative model together with an auxiliary classifier to perform training on both annotated and unannotated datasets. We also propose a query-hint mechanism to further guide the generation process towards 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.

## 1 Introduction

In recent years, the Transformer-based pre-trained language models (LMs) have greatly improved the state-of-the-art on natural language processing tasks as well as natural language generation (NLG). Large-scale autoregressive Transformer models (Vaswani et al., 2017) that leverage large amounts of unannotated data and a simple log-likelihood training objective has achieved remarkable results in many text generation tasks such as machine translation, text summarization, 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 (Keskar et al., 2019; Ziegler et al., 2019). Therefore, some approaches like PPLM (Dathathri et al., 2019), controls 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 sentiment (*e.g.*, aspect-opinion, aspect-sentiment et.) as input. Zang and Wan (2017) first introduced the aspect-sentiment information to perform aspect-level sentiment-controllable review generation. They conducted a 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. (2021) proposed a mutual learning framework leveraging large unlabeled data through interactive learning between generator and 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 and proposed a conditional generative model based on a pre-trained language model together with an auxiliary fine-grained sentiment classifier.

Our aspect-opinion conditioned generating task aims to generate a review text $X$ that correctly contains the sentiment information from $n$ non-repeated aspect-opinion pairs $\langle \text{a, o} \rangle_{1:n}$. In the generator, we incorporate a GPT-2 345M model (Radford et al., 2019) 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, with the assistance of a classifier, we leveraged a large unlabeled dataset to train the generator.

**Our Contributions:** (1) 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 generation at a finer-grained level. (2) We introduce the aspect-opinion pair as the fine-grained sentiment unit into controlling the constrained text generation. (3) 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.

## 2 Related Work

### 2.1 Controlled Text Generation

Recently, there is a bunch of works that aims 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 (Kikuchi et al., 2016; Ficler and Goldberg, 2017), while fine-tuning pre-trained models with Reinforcement Learning (RL) (Ziegler et al., 2019) and training a Generative Adversarial Network (Yu et al., 2016) have also shown inspiring results. CTRL (Keskar et al., 2019) is a recent approach that trains a language model conditioned on a variety of control codes, 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 (Dathathri et al., 2019) composed a pre-trained LM with attribute controllers guiding text generation towards 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. CoCon (Chan et al., 2020) incorporated a pre-trained GPT-2 model with a Content-Conditioner to control the generated text under the guidance of target text content. Different from our "fine-grained sentiment text generation", these works focus on sentence-based sentiment and topic control in text generating. In the "fine-grained sentiment text generation" task, the text generation process is controlled by a series of fine-grained sentiments (*e.g.*, aspect-opinion or aspect-sentiment et.).

### 2.2 Review Generation

Review generation (Dong et al., 2017; Lipton et al., 2015), 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 (Dong et al., 2017; Sharma et al., 2018) 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. (2020) 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 the sentence-level sentiments and ignore the aspect-level sentiment control and very few researchers studied generating reviews from fine-grained sentiments due to the lack of announced data. Zang and Wan (2017) 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. (2021) proposed a mutual learning framework that enhanced the generation results with the assistance of a classifier.

## 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 trained with both labeled and unlabeled data to learn a fine-grained sentiment review generator with the assistance of a classifier.

Firstly, we give the formalization of our fine-grained sentiment controllable text generation task. Formally, giving a list of review aspect-opinion phrase pairs $s = \{\langle a_1, o_1\rangle, \langle a_2, o_2\rangle, \ldots, \langle a_n, o_n\rangle\}$, the task aims to generate a review text $X$ comprising of $m$ words ($X = \{x_1, x_2, \ldots, x_m\}$), which presents each aspect phrase $a_i$ and its corresponding opinion phrase $o_i$ ($i \in \{1, 2, ..., n\}$) properly.

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

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

### 3.1 Main Framework

To make full use of both limited labeled dataset and large unlabeled dataset, inspired by Chen et al. (2021), our proposed method in the basic of a text generator $G$ additionally employ a sentiment classifier $C$. The generator $G$ generates a review text according to a series of given attributes including a prompt text together with a list of pairs each composed of one aspect phrase and one opinion phrase, representing the fine-grained sentiment. The classifier $C$ is incorporated to extract all the fine-grained sentiments consisting of aspect and opinion phrases 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.

Specifically, following Chen et al. (2021), we adopt three steps to make full use of the large unlabeled dataset:

**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 get 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.

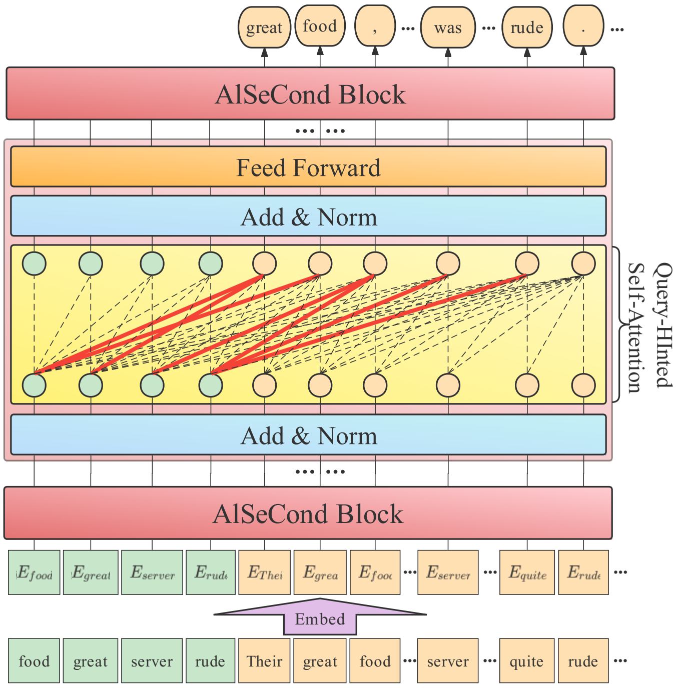


Figure 1: Architecture of the Generator.

### 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(Manning and Schütze, 1999; Bengio et al., 2000) where $x_i$ is the next token, $x_1{:}x_{i-1}$ is the previous tokens including prompt text and generated text. 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 the attribute $a$ into the unconditional distribution $p(x_i|x_1{:}x_{i-1})$. What's more, the pre-trained Transformer-based language model GPT-2 (Radford et al., 2019) in recent years has demonstrated remarkable natural text generation in the auto-regressive manner. Thereby, to improve the generated texts' quality, our generative model incorporate 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 (Ba et al., 2016), multi-head self-attention (Vaswani et al., 2017), and position-wise feed-forward operations. Therefore, our AlSeCond's block 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 inner the AlSeCond's blocks. Specifically, the target fine-grained sentiment pairs **s0** are appended 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's blocks, thus $h_t^1$ perform its self-attention with the hidden states of 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.

Our AlSeCond's block (detailed in the pink block in Figure 1) is a special Transformer-Decoder block that incorporates our proposed query-hint mechanism to guide the controlled generation process. Specifically, for a fine-grained sentiments appended hidden states $h = [h^0; h^1]$, 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:

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

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

where $M_h \in R^{t\times n}$ 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 pair, $l_a$ ($a \in \{1, 2, \dots, n\}$) 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.

### 3.3 Loss functions

**Generation loss function:** Through a 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. Specifically, for every aspect phrase $a$ and opinion phrase $o$ presented in the source text, the training loss is defined as:

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

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 $l_t * 1$ ($l_t$ is the length of the target text). $\odot$ and $\oplus$ represent element-wise product and XOR, respectively.

As a result, our final loss function comprehensively consider 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.

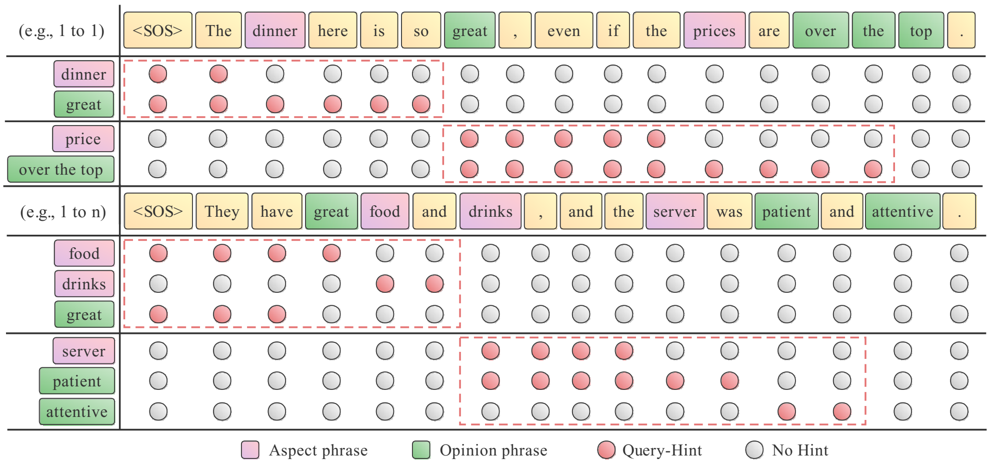

Figure 2: Query-hint strategy

### 3.4 Hint-strategy

As mentioned in 3.2, we introduce a query-hint mechanism to further guide the generation towards sentiment inclusion. The strategy of query-hint is slightly different between the process of generating and training. 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. During 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.

Specifically, for each casual sentiment pair, its aspect and opinion phrases have their own corresponding subsequence to provide query-hints. As shown in Figure 2 (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 2).

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 to one "hint-unit". As shown in Figure 2 (e.g.,1 to n), inner the "hint-unit", each aspect/opinion phrase gives the query-hint sequentially.

### 3.5 Classifier

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

The task of AOPE aims to extract aspect terms

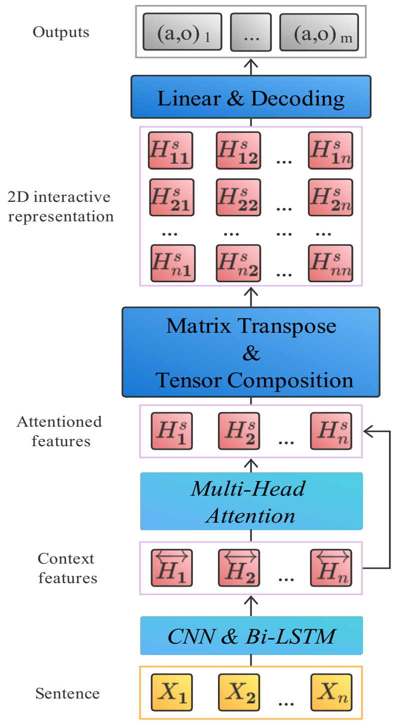

Figure 3: Architecture of the Classifier.

and their corresponding opinion terms as pairs (Zhao et al., 2020; Chen et al., 2020). This task can be defined as follows: Given a sentence with $m$ words $X = \{x_1, x_2, ..., x_m\}$, 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.

The overall architecture of our classifier: two-dimensional interaction-based multi-task learning framework (2D-IMLF) is shown in Figure 3. Given an input sentence, two high-related work 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 at the end obtains the final results by applying a decoding algorithm (Wu et al., 2020).

As shown in Figure 3. We first use a group of CNN layers to encode the input sentence:

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

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

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

Afterwards, we concatenate the hidden state $H_c$ with their transferring state $H_c^T$ to get a gird-formed features. 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 (Wu et al., 2020), we can easily obtain a serious of aspect-opinion pairs.

# 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

### 4.1.1 Labeled dataset

We conduct experiments of aspect-opinion and aspect-polarity pairs conditioned controllable text generation on English restaurant review with ASTE-Data-V2 from Xu et al. (2020) and MAMS-ASTA from Jiang et al. (2019), respectively.

**ASTE-Data-V2:** ASTE-Data-V2[1] from Xu et al. (2020), is originally come from SemEval Challenges (Pontiki et al., 2014, 2015, 2016), and contain both aspect and opinion labels in each review data. Specifically, we union the 14Rest, 15Rest, and 16Rest included in the ASTE-Data-V2 as our labeled dataset. The statistics of the dataset are reported in Table 1.

**MAMS-ASTA:** From MAMS[2] (Multi-Aspect Multi-Sentiment) (Jiang et al., 2019) is an aspect-level sentiment labeled dataset. Wherein, each data 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 the training data in the related review domain, we use the Yelp's review dataset[3] as the unlabeled dataset and filtered 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[4] model extended from a pre-trained GPT-2 medium 345M model (Radford et al., 2019). The AlSeCond's blocks clones the GPT-2 Transformer blocks' parameters and settings. To ensure that the generator can generate any string, we apply Byte Pair Encoding (BPE) (Sennrich et al., 2015) for the inputs. The max generating length is set to 32. We tune the $\lambda_G$ together with $\lambda_{senti}$ to 1 and 8, respectively. Adam (Kingma and Ba, 2014) is used for optimization, the batch size is set to 16, and the learning rate is set to 5e-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 G1, the generator is fine-tuned with the labeled dataset for 24 epochs. The above steps are trained on a RTX A4000 GPU for 24 hours. We ran our model and baselines 5 times to average the scores.

**Classifier:** Following GTS (Wu et al., 2020), we combine a 300-dimension domain-general embedding from pre-trained GloVe (Pennington et al., 2014) and a 100-dimension domain-specific embedding trained with fastText (Bojanowski et al.,

[1] https://github.com/xuuuluuu/SemEval-Triplet-data

[2] https://github.com/siat-nlp/MAMS-for-ABSA

[3] https://www.kaggle.com/yelp-dataset/yelp-dataset

[4] Codes available at: https://github.com/ashooha0/Alsecond

| 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 | - | 1160546 | - | - | - | - |

Table 1: Statistics of the labeled and unlabeled datasets. Sentence in the ASTE-Data-V2-Rest is labeled with aspect, opinion, and polarity, while in the MAMS-ASTA labeled with only aspect and polarity.

2016) to initialize double word embeddings. We use Adam as optimizer and the learning rate is set to 5e-4. The batch size and dropout rate are set to 32 and 0.5, respectively. The number of hidden units in BiLSTM is set to 128.

### 4.2 Baselines

We compare with 5 baselines. **PPLM** (Dathathri et al., 2019) incorporates an attribute model BoW (bag of words) to steer a pre-trained GPT2 model towards increasing the generating probability of the target words. In this baseline, the BoW is formed with the words contained in the target sentiment pairs. Through prepending the task description before the input text, the state-of-the-art text-to-text model **T5** (Liu et al., 2019) is pre-trained with a multitask 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 finetune **UniLM** (Dong et al., 2019), **UniLM-v2** (Piao et al., 2020) and **BERT-Gen** (Piao et al., 2020) in a similar sequence-to-sequence fashion with both the large unlabeled dataset and the limited labeled dataset.

### 4.3 Generated Quality Evaluation

#### 4.3.1 Fluency and Diversity Evaluation:

We conduct fluency evaluation on the generated texts with automatic metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Lavie and Agarwal, 2007) which compare the similarity between the generated text and ground truth based on n-gram matching. Besides, the diversity of generations is also an important indicator. We measure diversity for the generated results with Dist-1,-2,-3 (Brockett et al., 2015) scores and Self-Bleu (Zhu et al., 2018).

#### 4.3.2 Sentiment Evaluation:

As to measure the quality of sentiment containment in the generated sentence, we employ two

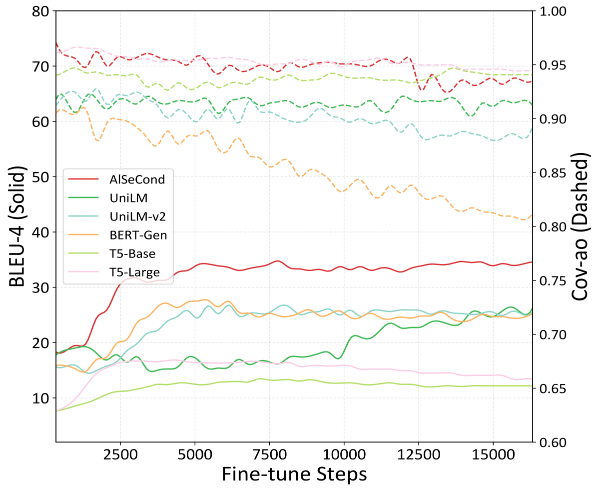


Figure 4: Learning curves for fine-tuning models with the labeled dataset.Note that the solid curves and the dotted curves are for the BLEU-4 and the Cov-ao changing with the number of fine-tune steps, respectively.

metrics indicating whether the input sentiments are correctly expressed in the generated text.

**Coverage** (Cov): Just like in Lin et al. (2019), 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.): We use the external sentiment classifier (Jiang et al., 2019) trained on MAMS-ASTA to evaluate the rate about how many sentiment pairs are correctly expressed in the generated texts as the sentiment accuracy.

Table 2 shows the fluency and diversity evaluation results. From the results we can observe that: (1) Comparing with baseline models, our AlSeCond model extends from the GPT-2 achieves better performance in fluency evaluations. (2) Comparing results in diversity metrics, it can be observed that our AlSeCond model perform much better than the rest of 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 3 shows the results of sentiment coverage and accuracy for generated texts. It is worth not-

| Dataset | Models | BLEU-3(↑) | BLEU-4(↑) | METETOR(↑) | ROUGE-L(↑) | Self-Bleu-4(↓) | Dist-1(↑) | Dist-2(↑) | Dist-3(↑) |
|---|---|---|---|---|---|---|---|---|---|
| | PPLM | 0.196 | 0.032 | 14.078 | 13.827 | **7.939** | 0.0841 | 0.4102 | 0.7180 |
| ASTE-Data-V2 | 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** |
| MAMS-ASTA | 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** |

Table 2: Results for the fluency and diversity evaluation.

| Dataset | Models | Cov-a | Cov-o | Cov-ao | Acc. |
|---|---|---|---|---|---|
| | PPLM | 0.3597 | 0.3642 | 0.1094 | 0.1761 |
| ASTE | T5-base | 0.9563 | 0.9745 | 0.9400 | 0.7812 |
| | T5-large | 0.9668 | **0.9831** | 0.9549 | **0.7948** |
| | UniLM | 0.9513 | 0.9568 | 0.9182 | 0.7450 |
| | BERT-Gen | 0.9390 | 0.9363 | 0.8932 | 0.7521 |
| | UniLM-v2 | 0.9478 | 0.9463 | 0.9100 | 0.7475 |
| | AlSeCond | **0.9719** | 0.9824 | **0.9614** | 0.7688 |
| | ⌊ w/o sentiment loss | 0.9633 | 0.9649 | 0.9468 | 0.7683 |
| | ⌊ w/o query-hint | 0.9412 | 0.9313 | 0.8966 | 0.7443 |
| MAMS | T5-base | 0.9619 | 0.9128 | 0.9032 | 0.5734 |
| | T5-large | 0.9733 | 0.9459 | 0.9422 | 0.5698 |
| | UniLM | 0.9297 | 0.7818 | 0.7624 | 0.5883 |
| | BERT-Gen | 0.9431 | 0.7778 | 0.7561 | 0.6048 |
| | UniLM-v2 | 0.9389 | 0.7532 | 0.7332 | **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 |

Table 3: Results for the sentiment evaluation. Note that Acc. is automatically evaluated by a external classifier.

ing 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's 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.

Comparing the above metrics results for all models on different datasets, we can observe that our model has stable advantages on both ASTE-Data-V2 and MAMS-ASTA, which indicates that our AlSeCond model has stronger adaptability.

### 4.4 Case Study

Figure 5 presents some generated cases from AlSeCond, T5, UniLM, BERT-Gen, and UniLM-v2. From the cases, we found that: AlSeCond tends to generate more linguistically complicated sentences. While other baselines are more likely to focus on generating review texts that correctly 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.
**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.
**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.
**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

Figure 5: Generated samples from the generative models.

## 5 Conclusion 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 which 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 works, we will explore the controllable text generation for implicitly expressed fine-grained sentiments, since the query-hint mechanism proposed in this paper is only effective for explicitly expressed fine-grained sentiments.

# References


Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv: Machine Learning*.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *neural information processing systems*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*.

Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, and Jiwei Li. 2015. A diversity-promoting objective function for neural conversation models. *north american chapter of the association for computational linguistics*.

Alvin T. S. Chan, Yew-Soon Ong, Bill Tuck Weng Pung, Aston Zhang, and Jie Fu. 2020. Cocon: A self-supervised approach for controlled text generation. *Learning*.

Huimin Chen, Yankai Lin, Fanchao Qi, Hu Jinyi, Peng Li, Jie Zhou, and Maosong Sun. 2021. Aspect-level sentiment-controllable review generation with mutual learning framework. *national conference on artificial intelligence*.

Shaowei Chen, Jie Liu, Yu Wang, Wenzheng Zhang, and Ziming Chi. 2020. Synchronous double-channel recurrent network for aspect-opinion pair extraction. *meeting of the association for computational linguistics*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *Learning*.

Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. *conference of the european chapter of the association for computational linguistics*.

Li Dong, Yu Wang, Furu Wei, Ming Zhou, Nan Yang, Jianfeng Gao, Hsiao-Wuen Hon, Xiaodong Liu, and Wenhui Wang. 2019. Unified language model pre-training for natural language understanding and generation. *neural information processing systems*.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. *arXiv: Computation and Language*.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. *empirical methods in natural language processing*.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv: Computation and Language*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. *empirical methods in natural language processing*.

Jihyeok Kim, Seungtaek Choi, Reinald Kim Amplayo, and Seung won Hwang. 2020. Retrieval-augmented controllable review generation. *international conference on computational linguistics*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv: Learning*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. *workshop on statistical machine translation*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2019. Commongen: A constrained text generation challenge for generative commonsense reasoning. *automated knowledge base construction*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *meeting of the association for computational linguistics*.

Zachary C. Lipton, Sharad Vikram, and Julian McAuley. 2015. Generative concatenative nets jointly learn to write and classify reviews. *arXiv: Computation and Language*.

Peter J. Liu, Michael Matena, Katherine Lee, Adam Roberts, Yanqi Zhou, Noam Shazeer, Colin Raffel, Sharan Narang, and Wei Li. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Christopher D. Manning and Hinrich Schütze. 1999. Foundations of statistical natural language processing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *meeting of the association for computational linguistics*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. *empirical methods in natural language processing*.

Songhao Piao, Li Dong, Yu Wang, Furu Wei, Ming Zhou, Nan Yang, Jianfeng Gao, Hsiao-Wuen Hon, Hangbo Bao, Xiaodong Liu, and Wenhui Wang. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *international conference on machine learning*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5 : aspect based sentiment analysis. *north american chapter of the association for computational linguistics*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. *north american chapter of the association for computational linguistics*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *international conference on computational linguistics*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *meeting of the association for computational linguistics*.

Vasu Sharma, Harsh Sharma, Ankita Bishnu, and Labhesh Patel. 2018. Cyclegen: Cyclic consistency based product review generator from attributes. *international conference on natural language generation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *neural information processing systems*.

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: Computation and Language*.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-aware tagging for aspect sentiment triplet extraction. *empirical methods in natural language processing*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2016. Seqgan: Sequence generative adversarial nets with policy gradient. *national conference on artificial intelligence*.

Hongyu Zang and Xiaojun Wan. 2017. Towards automatic generation of product reviews from aspect-sentiment scores. *international conference on natural language generation*.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, and Hui Xue. 2020. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. *meeting of the association for computational linguistics*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *international acm sigir conference on research and development in information retrieval*.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv: Computation and Language*.