StyleBART: Decorate Pretrained Model with Style Adapters for Unsupervised Stylistic Headline Generation

Hanqing Wang1∗ Yajing Luo1∗ Boya Xiong1 Guanhua Chen2 Yun Chen1†
1Shanghai University of Finance and Economics
2Southern University of Science and Technology
{whq, luoyajing, xiongboya}@163.sufe.edu.cn
chengh3@sustech.edu.cn yunchen@sufe.edu.cn

Abstract

Stylistic headline generation is the task of generate a headline that not only summarizes the content of a news, but also reflects a desired style that attracts users. As style-specific news-headline pairs are scarce, previous research has focused on unsupervised approaches using a standard headline generation dataset and monostyle corpora. In this work, we follow this line and propose StyleBART, an unsupervised approach for stylistic headline generation. Our method decorates the pretrained BART model with adapters that are responsible for different styles and allows the generation of headlines with diverse styles by simply switching the adapters. Different from previous works, StyleBART separates the task of style learning and headline generation, making it possible to freely combine the base model and the style adapters during inference. We further propose an inverse paraphrasing task to enhance the style adapters. Extensive automatic and human evaluations show that StyleBART achieves new state-of-the-art performance in the unsupervised stylistic headline generation task, producing high-quality headlines with the desired style. Code is available at https://github.com/sufenlp/StyleBART.

1 Introduction

The sequence-to-sequence-based neural headline generation (HG) model (Cao et al., 2018; Lin et al., 2018; Song et al., 2019) has demonstrated its ability to generate factual, concise, and fluent headlines (Chopra et al., 2016). Yet, the headlines are also expected to have stylistic attributes to draw more attention of the audiences. To address the problem, Jin et al. (2020) propose the task of Stylistic Headline Generation (SHG), which aims to generate a headline with a specified style such as humorous, romantic and clickbaity. However, acquiring enough parallel data for SHG is almost impossible, as the creation of headlines with specific styles demands creativity and often consumes significant effort. Hence, researchers (Jin et al., 2020; Zhan et al., 2022) in turn explore unsupervised approaches which only require a standard headline generation dataset and non-parallel stylistic text corpora.

Some existing works propose a pipeline method (Sudhakar et al., 2019; Dai et al., 2019; Krishna et al., 2020) which firstly generates a plain headline and then introduces the pre-specified style with a style transfer model. However, this pipeline method require two models during inference, which brings additional latency and storage cost. Some other works (Jin et al., 2020; Zhan et al., 2022) jointly learn a plain headline generation model on news-headline pairs, and a denoising autoencoder on the stylistic text corpus. However, these approaches require to carefully design the scheduling
and parameter sharing mechanisms between tasks. Moreover, as the task of headline generation and style learning are entangled, the entire model has to be rebuilt when facing a new style or task.

In this paper, we propose StyleBART, an unsupervised SHG model based on the BART model (Lewis et al., 2020). Instead of using a multi-task learning framework, StyleBART disentangles the style and the specific task (e.g. headline generation) by the design of model architecture and training strategy. Intuitively, this enables StyleBART to separately train the style and task modules, then combine them as required during inference (Figure 1). Hence, StyleBART is more flexible and training-efficient compared with baselines (Jin et al., 2020; Zhan et al., 2022) which require training models for all style and task combinations.

Specifically, StyleBART utilizes style adapters as plug-and-play modules for the style control. The style adapters are learned at the pretraining stage through the inverse paraphrasing task on the style corpus, while the base HG model is learned at the fine-tuning stage on news-headline pairs. During inference, we achieve unsupervised SHG by switching to the target style adapter. In this way, StyleBART has the same high decoding efficiency as Jin et al. (2020), while overcoming its problems in task scheduling and inefficient training.

We evaluate StyleBART using the same three SHG tasks described in Jin et al. (2020). Both automatic and human evaluations demonstrate the superiority of our method over baselines.

2 Approach

2.1 Problem Formulation

We denote the parallel news-headline dataset as \( D = \{\langle x, y \rangle \} \), where each pair \( \langle x, y \rangle \) consists of a news article \( x \) and its corresponding plain headline \( y \). The corpus for the \( i \)th style \( s_i \) is denoted as \( T^{s_i} = \{t^{s_i} \} \). Our goal is to generate stylistic headline \( y^{s_i} \) with style \( s_i \) given the news article \( x \). Note that this is an unsupervised setup, as no news article and stylistic headline pairs are available.

2.2 StyleBART Architecture

As shown in Figure 2, our model consists of two parts: the BART model and the adapter modules. We use the BART as the base model for StyleBART. Adapters (Rebuffi et al., 2018) are light-weight bottleneck layers inserted into a base model. It is designed as a parameter-efficient method to fine-tune the base model for a new task (Houlsby et al., 2019), language (Pfeiffer et al., 2020) or domain (Bapna et al., 2019). In StyleBART, we instead use the adapters to control the style of the model output.

Formally, following Houlsby et al. (2019), an adapter module \( A \) is composed of layer normalization (LN) of the input \( z \in R^b \), down-projection \( W_{\text{down}} \in R^{b \times b} \) with the bottleneck dimension \( b \), non-linear function (ReLU), up-projection \( W_{\text{up}} \in R^{b \times k} \), and a residual connection with the input \( z \):

\[
A(z) = W_{\text{up}}^T \text{ReLU}(W_{\text{down}}^T \text{LN}(z)) + z. \quad (1)
\]

We insert the adapters into each decoder layer of the base model. For each style, we have a set of style adapters. In the following, we use \( \theta \) to denote the set of parameters of the BART model, and \( \theta^{s_i} \) to denote that of the style adapters for style \( s_i \).

2.3 Training and Inference

We divide the training process of StyleBART into three steps (Figure 3): (1) Style adapter pretraining, which learns the style adapters to control the style of the model output by pretraining on the style dataset \( T^{s_i} \); (2) Headline generation fine-tuning, which optimizes the base model to generate plain headline on the headline dataset \( D \); (3) Stylistic headline generation, which generates stylistic headline by switching to the target style adapters during inference.

Step 1: Style Adapter Pretraining. The style adapters can be trained with the style dataset \( T^{s_i} \) by the denoising auto-encoding task, which reconstructs the sentence \( t^{s_i} \) from \( g_n(t^{s_i}) \). Here \( g_n \) is a noise function that generates a perturbed version
of the input, such as token masking and token deletion used during BART pretraining (Lewis et al., 2020) and our baselines (Jin et al., 2020; Zhan et al., 2022). However, this training method is suboptimal.

Following Krishna et al. (2020), the style can be loosely defined as the common patterns of lexical choice and syntactic constructions that are distinct from the content of a sentence. Considering how the noise function $g_n$ works, we argue that the corrupted text $g_n(t_{si})$ still contains the stylistic information. As a result, the model may learn to guide the style of its output with the style of its input, which deviates from our goal to control the output style with style adapters. We call this undesirable spurious correlation (Gu et al., 2019). Instead, we propose to address this issue with inverse paraphrasing method.

Inspired by Krishna et al. (2020), we replace $g_n(t_{si})$ with $g_p(t_{si})$, which is generated by feeding $t_{si}$ to a paraphrase model trained to maximize diversity. Table 1 displays examples of stylistic sentences and their perturbed versions with the noise function $g_n$ and the paraphrase model $g_p$. Intuitively, the paraphrase model can better strip away information from $t_{si}$ that is predictive of its original style. As a result, our model has to rely on the style adapters to control the output style, improving the adapters’ ability for style control.

Specifically, to train the adapters for style $s_i$, we first feed each sentence $t_{si}$ to the paraphrase model and get the perturbed sentence $g_p(t_{si})$. Then we train the corresponding style adapters while freezing all other parameters with inverse paraphrasing:

$$\hat{\theta}^s_i = \arg\max_{\theta^s_i} \sum_{t_{si} \in T_{si}} \log p(t_{si} \mid g_p(t_{si}); \theta^s_i, \theta^b).$$

Since the input is style-agnostic and the other parameters are freezing, the model has to rely on the style adapters to control its output style. We also learn a style-less adapter $\theta^s_0$ in the same way except replacing $T_{si}$ with a dataset $T_{s0}$, which consists of plain text from BART pretraining.\footnote{We use the STRAP model (Krishna et al., 2020).}

**Step 2: Headline Generation Fine-tuning.** In this step, StyleBART learns headline generation on the headline dataset $D = \{(x, y)\}$. Since the headline is style-less, we insert the style-less adapters into StyleBART and finetune it on the headline generation task. This step is required to force the model to learn the task of headline generation. During finetuning, we only update the parameters of the encoder while freezing the style-less adapters and all other model parameters:

$$\hat{\theta}^b_{enc} = \arg\max_{\theta^b_{enc}} \sum_{(x, y) \in D} \log p(y \mid x; \hat{\theta}^{s0}, \theta^b).$$

In this way, we limit the computational cost, and more importantly, mitigate the catastrophic forgetting problem in style control, thus facilitating the switching of different style adapters in Step 3.

**Step 3: Stylistic Headline Generating.** To generate a headline in a given style, we can achieve this by replacing the style-less adapters of StyleBART with the corresponding style-specific adapters.

\footnotetext[2]{We choose plain text from BART pretraining instead of the headline generation dataset in order to separate the style and downstream task at the data level.}
Table 1: Examples of sentences and their perturbed versions in three styles: humor, romance and clickbait. The
noised text indicates the output after the noise function \(g_s\), and the paraphrased text indicates the output after
passing through the paraphrase model \(g_p\).

<table>
<thead>
<tr>
<th>Style (s_i)</th>
<th>Stylistic Sentence (t^{s_i})</th>
<th>Noised Text (g_s(t^{s_i}))</th>
<th>Paraphrased Text (g_p(t^{s_i}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Humor</td>
<td>There are few things more awkward on a blind date than looking up from your phone to realize she’s left. She obviously wasn’t blind at all.</td>
<td>She [MASK] blind at all. There are few things [MASK] on a blind date than [MASK] looking up from your phone to realize she’s left.</td>
<td>The fact that she left, and the fact that she’s obviously not blind, is much more awkward.</td>
</tr>
<tr>
<td>Romance</td>
<td>He confessed to her that she was his most precious jewel, and that she would always be the rarest gem in his eyes.</td>
<td>[MASK] to her that she was his most [MASK] jewel, and that she would [MASK] rarest gem in his eyes.</td>
<td>He confessed that she was the most valuable thing in his eyes, and he would never see another one.</td>
</tr>
<tr>
<td>Clickbait</td>
<td>Unbelievable meeting Bob Dylan in Europe for the third time twenty years.</td>
<td>Unbelievable meeting Bob Dylan in [MASK] for the third time twenty years.</td>
<td>The extraordinary meeting with Bob Dylan in Europe for three times.</td>
</tr>
</tbody>
</table>

StyleBART has already learned the task of headline generation. Therefore, by switching the style-less adapters to the style-specific adapters, we can obtain a style-specific headline generation model. We use \(\theta^0\), \(\theta^{w_s}\) and \(\theta^{s_i}\) to denote the model parameters of the base model, the style-less adapters and the style adapters after Step 2. Then given a news article, StyleBART outputs its style-specific headline \(y^{s_i}\) with

\[
y^{s_i} = \arg \max_y p(y \mid x; \hat{\theta}^{s_i}, \hat{\theta}^b).
\] (4)

3 Experiments

3.1 Datasets
Following Jin et al. (2020), the experiment datasets consist of a headline generation dataset CNN-NYT, and three stylistic text datasets for humorous, romantic and clickbaity. The CNN-NYT dataset consists of 146K news-headline pairs from two sources: the New York Times (Sandhaus, 2008) and the CNN dataset (Hermann et al., 2015). We use the same dataset split as Jin et al. (2020). Each style dataset contains 500K style-specific sentences. The humor and romance datasets are collected from the novels in the corresponding genres of BookCorpus (Zhu et al., 2015). The clickbait dataset is obtained from The Examiner-SpamClickBait News dataset.3 To train the style-less adapters, we also build a style-less dataset, which consists of 500K sentences randomly sampled from BookCorpus, one of the datasets used in BART pretraining. For style datasets, we use the same split as Jin et al. (2020). For the style-less dataset, we randomly sample 3,000 sentences for both the validation and test set, leaving the rest as the training set.

3.2 Baselines
We compare StyleBART with the following baselines:
- **Neural Headline Generation (NHG)**: Finetuning all parameters of BART on the plain headline generation dataset.
- **Two-Step Decoding(TSD)**: A two-step decoding method that first generates a plain headline and then injects style with an unsupervised style transfer model (Dai et al., 2019).
- **Multitask**: A multitask framework that jointly trains on plain headline generation and stylistic text denoising with the BART model.
- **TitleStylist (Jin et al., 2020)**: An approach similar to Multitask, but with carefully designed parameter sharing and switching between tasks.
- **S-SHG (Zhan et al., 2022)\(^4\)**: An approach that first constructs the stem and syntax of the headline similarly to TitleStylist, and then populates that with substantive context.

3.3 Evaluation Metrics

**Automatic Evaluation.** We use automatic evaluation to measure the quality and style of the gener-

---

3https://www.kaggle.com/datasets/therohk/examine-the-examiner

4Our reimplemented S-SHG performs slightly worse than the original paper. However, the paper provides no open-source code or implementation details.
<table>
<thead>
<tr>
<th>Style</th>
<th>Method</th>
<th>Generation Quality</th>
<th>Style Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R1(↑) R2(↑) RL(↑)</td>
<td>BERT(↑) PPL(↓)</td>
</tr>
<tr>
<td>Style-less</td>
<td>NHG</td>
<td>29.7 10.9 26.3</td>
<td>88.2 52.6</td>
</tr>
<tr>
<td></td>
<td>StyleBART-N</td>
<td>29.4 10.7 26.2</td>
<td>88.1 55.6</td>
</tr>
<tr>
<td>Humor</td>
<td>TSD*</td>
<td>20.3 5.5 18.5</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>MultiTask</td>
<td>30.0 11.1 26.6</td>
<td>87.9 52.8</td>
</tr>
<tr>
<td></td>
<td>TitleStylist</td>
<td>27.7 10.0 24.5</td>
<td>87.5 39.4</td>
</tr>
<tr>
<td></td>
<td>S-SHG</td>
<td>26.2 8.2 22.5</td>
<td>87.4 107.4</td>
</tr>
<tr>
<td></td>
<td>StyleBART</td>
<td>28.2 9.9 25.0</td>
<td>87.6 42.5</td>
</tr>
<tr>
<td>Romance</td>
<td>TSD*</td>
<td>20.5 5.8 18.7</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>MultiTask</td>
<td>30.0 11.4 26.8</td>
<td>88.1 53.2</td>
</tr>
<tr>
<td></td>
<td>TitleStylist</td>
<td>27.6 9.9 24.5</td>
<td>87.5 39.5</td>
</tr>
<tr>
<td></td>
<td>S-SHG</td>
<td>26.2 8.6 22.5</td>
<td>87.4 101.0</td>
</tr>
<tr>
<td></td>
<td>StyleBART</td>
<td>27.4 9.3 24.3</td>
<td>87.4 36.7</td>
</tr>
<tr>
<td>Clickbait</td>
<td>TSD*</td>
<td>20.3 7.1 23.2</td>
<td>- -</td>
</tr>
<tr>
<td></td>
<td>MultiTask</td>
<td>30.3 11.3 27.0</td>
<td>88.3 52.2</td>
</tr>
<tr>
<td></td>
<td>TitleStylist</td>
<td>28.2 10.1 25.2</td>
<td>87.8 39.1</td>
</tr>
<tr>
<td></td>
<td>S-SHG</td>
<td>27.0 8.8 23.5</td>
<td>87.6 67.9</td>
</tr>
<tr>
<td></td>
<td>StyleBART</td>
<td>25.3 7.7 23.0</td>
<td>87.5 50.9</td>
</tr>
</tbody>
</table>

Table 2: Automatic evaluation results of ROUGE, BERTScore, PPL and PPL-S. TSD* result is directly cited from Li et al. (2022), while all other methods are implemented by ourselves or with the open-sourced code. StyleBART-N: StyleBART with the style-less adapters.

For the quality, we use ROUGE (Lin, 2004) and BERTScore (Zhang* et al., 2020) to measure the relevance of the generated headlines to the reference headline and PPL to evaluate its language fluency. The PPL is calculated with OpenAI GPT2 (Radford et al., 2019) finetuned on the plain headlines following Zhan et al. (2022). To evaluate the style, we use PPL-S, which computes the PPL score of the generated headline using GPT2 finetuned on the corresponding style corpus.

### Human Evaluation
We conducted a human evaluation to more comprehensively assess our model. We randomly sampled 50 news articles from each test set and asked three judges to rate the outputs from TitleStylist, S-SHG and our StyleBART.\(^3\) To assess generation quality, we ask the judges to rate from 1 to 10 (integer values) from three aspects: 1) Relevance—how semantically relevant the headline is to the news article. 2) Attractiveness—how appealing they feel the headline is. 3) Fluency—how comprehensive and easy-to-read the headline is. We then report the average score for each aspect across all test samples and judges. For the style evaluation, we ask the judges to choose the best headline of each style from an entire set of TitleStylist, S-SHG and StyleBART’s outputs. We then report the percentage of the times each model is chosen as the best.

### 3.4 Model Configuration
We implemented StyleBART with the pretrained BART model using Hugging Face. We set the bottleneck dimension \(b\) of the adapters to be 64. To pretrain the adapters, we use the AdamW optimizer (Kingma and Ba, 2014) with \(\beta_1 = 0.9\) and \(\beta_2 = 0.999\). The learning rate is \(5 \times 10^{-5}\). For fine-tuning on the headline generation dataset, we use the same AdamW optimizer and learning rate. For all steps of training, we use a batch size of 8. At inference time, we use beam search with a beam size of 4. All experiments are run on a single NVIDIA RTX 2080ti GPU, except that LLaMA2-InsTuning is run on a single NVIDIA A100 40G GPU.

### 4 Results and Discussion

#### 4.1 Automatic Evaluation Results
Table 2 shows the automatic evaluation results of our proposed StyleBART and all baselines.

### Content Relevance
We use ROUGE (R1,R2, and RL) and BERTScore (BERT) to measure the relevance of the generated headlines to the reference headline. As can be seen, NHG, StyleBART-N, and MultiTask achieve the highest relevance score. This can be explained by that more stylistic headlines would lose some relevance as 1) the reference headlines are style-less; 2) the stylistic head-
Table 3: Human evaluation results on three aspects.

<table>
<thead>
<tr>
<th>Style</th>
<th>Method</th>
<th>Relevance</th>
<th>Attraction</th>
<th>Fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Humor</td>
<td>TitleStylist</td>
<td>5.82</td>
<td>5.60</td>
<td>7.40</td>
</tr>
<tr>
<td></td>
<td>S-SHG</td>
<td>6.13</td>
<td>7.12</td>
<td>8.23</td>
</tr>
<tr>
<td></td>
<td>StyleBART</td>
<td>7.04</td>
<td>7.58</td>
<td>8.42</td>
</tr>
<tr>
<td>Romance</td>
<td>TitleStylist</td>
<td>6.12</td>
<td>6.06</td>
<td>7.78</td>
</tr>
<tr>
<td></td>
<td>S-SHG</td>
<td>6.37</td>
<td>6.04</td>
<td>7.63</td>
</tr>
<tr>
<td></td>
<td>StyleBART</td>
<td>6.70</td>
<td>7.30</td>
<td>8.40</td>
</tr>
<tr>
<td>Clickbait</td>
<td>TitleStylist</td>
<td>6.00</td>
<td>6.16</td>
<td>8.00</td>
</tr>
<tr>
<td></td>
<td>S-SHG</td>
<td>6.17</td>
<td>6.65</td>
<td>8.17</td>
</tr>
<tr>
<td></td>
<td>StyleBART</td>
<td>6.76</td>
<td>6.96</td>
<td>8.26</td>
</tr>
</tbody>
</table>

**Table 4**: The percentage of choices (%) for the most humorous, romantic or clickbaity headlines among TitleStylist, S-SHG, and StyleBART.

<table>
<thead>
<tr>
<th>Style</th>
<th>TitleStylist</th>
<th>S-SHG</th>
<th>StyleBART</th>
</tr>
</thead>
<tbody>
<tr>
<td>Humor</td>
<td>20%</td>
<td>34%</td>
<td>46%</td>
</tr>
<tr>
<td>Romance</td>
<td>22%</td>
<td>28%</td>
<td>50%</td>
</tr>
<tr>
<td>Clickbait</td>
<td>32%</td>
<td>26%</td>
<td>42%</td>
</tr>
</tbody>
</table>

**Generation Quality.** We assess generation quality in relevance, attraction, and fluency, as shown in Table 3. StyleBART performs the best in all three aspects compared to baselines. The results are slightly inconsistent with automatic quality evaluation. This may be explained by that automatic quality evaluation metrics favor less stylistic headlines, while humans do not have such bias.

**Style Strength.** We measure the style strength in Table 4. As can be seen, StyleBART surpasses all baselines by a significant margin except that it is slightly worse than TitleStylist. This may be because the headlines from StyleBART have stronger style and more stylistic headlines would also lose some PPL.

**Language Perplexity.** We use PPL on the GPT2 fine-tuned on plain headlines to measure the fluency of the generated headline. As can be seen, StyleBART surpasses all baselines by a significant margin except that it is slightly worse than TitleStylist. This may be because the headlines from StyleBART have stronger style and more stylistic headlines would also lose some PPL.

**Style Strength.** We use PPL-S to measure the style strength of the generated headline. As can be seen, StyleBART generates headlines with the strongest style across all stylistic headline generation tasks. Compared with the baseline TitleStylist (resp. S-SHG), StyleBART obtains 119.5 (resp. 114.2) lower averaged PPL-S score. The baseline MultiTask performs the worst in style control. Moreover, StyleBART disentangles style learning and headline generation learning, thus only trains once on the headline generation dataset to support all three styles. In contrast, all baselines except TSD require training three times on the headline generation dataset for the three stylistic generation tasks.

### 4.2 Human Evaluation Results

Table 3 and Table 4 present the human evaluation results. The baselines can also support multiple styles within one model by jointly training on a headline generation dataset and all style corpora. However, the performance will decrease according to Jin et al. (2020).

### 4.3 Comparison with LLMs

In this part, we compare StyleBART with methods using Large Language Models (LLMs), as presented in Table 5. For GPT3.5-prompting, we perform few-shot prompting with the gpt-3.5-turbo API. We provide stylistic sentences and news-plain headline pairs in the prompt and query the model to generate stylistic headlines for the input news. For LLaMA2-InsTuning, we conduct instruction tuning with LoRA on LLaMA2 (Touvron et al., 2023) for both the inverse paraphrasing task and the news headline generation task. Then we perform stylistic headline generation during testing. Appendix A provides more details. We find that StyleBART overall generates headlines with the best content relevance and the strongest style, demonstrating the superiority of StyleBART even in the era of large language models.

### 4.4 Ablation Study

**Design Choices of Style Adapter Pretraining.** Table 6 shows the effect of different design choices at the style adapter pretraining step. StyleBART—para represents training the adapters

---

6 The baselines can also support multiple styles within one model by jointly training on a headline generation dataset and all style corpora. However, the performance will decrease according to Jin et al. (2020).

7 https://platform.openai.com/docs/models/gpt-3-5

8 Note that StyleBART and these two LLM-based methods are not strictly comparable due to the difference in model size and training data. We include the comparison as a reference.
with the denoising task instead of the inverse paraphrasing task. StyleBART − $s_0$ adapters fine-tune the BART model without style-less adapters at step 2, thus without pretraining the style-less adapters. As can be seen, all methods perform similarly in style-less headline generation. When it comes to stylistic headline generation, StyleBART − $s_0$ adapters decrease dramatically in both the summarization quality and the style control. The inverse paraphrasing task is crucial for style control, while slightly decreasing the relevance of the generated headline to the reference headline. This may be because the reference headlines are style-less, while the headlines produced by using the inverse paraphrasing task have a stronger style.

### Trainable Parameters of Headline Generation Fine-tuning

We compare different choices of trainable parameters at the headline generation fine-tuning step, as illustrated in Table 7. As can be seen, all fine-tuning methods get similar scores when generating style-less headlines, indicating that fine-tuning only partial parameters is enough for learning the headline generation task. When comparing on the SHG task, only updating the encoder can best control the style of the generated headlines among all fine-tuning methods. As the style adapters are inserted at the decoder, freezing the decoder and cross-attention can better maintain their style control ability obtained during pretraining.

#### 4.5 Case Study

In this section, we present examples of generated stylistic headlines, as shown in Table 8. Again, we concentrate on StyleBART, TitleStylist, and S-SHG, as they outperform the other methods.

From this example as well as others, we find that all three methods generate relevant and fluent headlines. However, StyleBART is better at style control. For example, the phrase "won’t last forever"
Cyber Monday has been the biggest single shopping day of the year for online retailers. But retailers are spreading their online sales throughout the thanksgiving holiday. As a result, cyber Monday’s growth is flattening. Analyst: cyber Monday will phase out eventually.

<table>
<thead>
<tr>
<th>Style</th>
<th>Method</th>
<th>BERT</th>
<th>PPL</th>
<th>PPL-S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Style-less</td>
<td>NST</td>
<td>88.5</td>
<td>4.4</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>StyleBART-N</td>
<td>88.5</td>
<td>4.7</td>
<td>-</td>
</tr>
<tr>
<td>Humor</td>
<td>-0.5</td>
<td>+0.4</td>
<td>-17.0</td>
<td></td>
</tr>
<tr>
<td>Romance</td>
<td>StyleBART</td>
<td>-0.4</td>
<td>+0.4</td>
<td>-14.4</td>
</tr>
<tr>
<td>Clickbait</td>
<td>-1.6</td>
<td>+23.1</td>
<td>-968.7</td>
<td></td>
</tr>
</tbody>
</table>

Table 9: Automatic evaluation results of stylistic story telling. NST: fine-tune all BART parameters on the plain story telling data. For StyleBART, we report the score changes with respect to the StyleBART-N model.
and mono-style corpora. They design a multitask learning framework to jointly learn both plain headline generation and stylistic text denoising with carefully designed parameter sharing and switching strategy. Zhan et al. (2022) further extend by decomposing the headline into style and content. They define stem and syntax as the style and generate the style first using a similar multitask framework as Jin et al. (2020). After that, substantive context is populated into that style with the conditional masked language modelling task. However, these works jointly learn style and headline generation, thus cannot support freely combination of styles and generation tasks at inference time.

5.2 Unsupervised Text Style Transfer

Text style transfer is to generate a sentence consistent with desired style while preserving the content of the source sentence (Shen et al., 2017). Due to the scarcity of style parallel corpora, unsupervised text style transfer is widely explored in previous works. Wu et al. (2019) regard text style transfer as a cloze task to accomplish sentiment style conversion of sentences. Dai et al. (2019) incorporates the sentiment style information through a reconstruction task. Krishna et al. (2020) redefine the style transfer problem as a paraphrase generation problem and propose a method based on reverse paraphrasing to generate the desired style. Lai et al. (2021) utilize large pretrained models such as BART and GPT-2, using BLEU scores and style classification scores as rewards, to achieve significant improvements in the transformation of formal and informal text. These models can be combined with plain headline generation model to achieve stylistic headline generation. However, they require two-steps decoding at inference time, while StyleBART generates stylistic headlines in one decoding step.

6 Conclusion

We propose StyleBART, an unsupervised stylistic generation method, which enables to freely combine the downstream tasks and styles by disentangling style learning and downstream task learning. Experimental results show that our model can generate content-relevant and style-intensive headlines, and can be extended to other stylistic generation tasks.

Limitations

This work mainly explores the stylistic headline generation task. We leave the exploration of more combinations of tasks and styles such as stylistic machine translation, stylistic document summarization as future work. At the same time, our current method achieves stylistic generation by simply switching the adapters, which cannot provide fine-grained control of the style. This hinders StyleBART from meeting the diverse user demands related to style control.

Ethics Statement

We present a training-efficient approach to build an unsupervised stylistic headline generation model which disentangles the headline generation learning and style learning. Despite the strong performance of style control, StyleBART inherits the societal impacts including some negative ones of the original BART model, such as societal biases (Milios and BehnamGhader, 2022) and misuse of language models (Tamkin et al., 2021). The implicit biases are expected to be removed by debiasing either the dataset or the model (Meade et al., 2022; Zhou et al., 2022). StyleBART makes it possible to generate text in various (e.g. clickbait) style which can be used to propagate the malicious or offensive content (Welbl et al., 2021). Future explorations are needed to mitigate the misuse of StyleBART model.

Acknowledgements

This project was supported by National Natural Science Foundation of China (No. 62106138, No. 62306132) and Shanghai Sailing Program (No. 21YF1412100). We thank the anonymous reviewers for their insightful feedbacks on this work.

References


Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In thirty-second AAAI conference on artificial intelligence.


Zhengpeng Li, Jian Wu, Jiawei Miao, and Xinmiao Yu. 2022. News headline generation based on improved decoder from transformer. Scientific Reports, 12.


Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. "mask and infill": Applying masked language model to sentiment transfer.

Task: News Style Headline Generation

Stage 1: Learning Text Styles
Please read the following representative text with a particular style and understand its stylistic features.
Style Text: [STYLE TEXT]

Stage 2: Learning Headline Generation
Here are some examples of news and corresponding headlines to learn how to generate news headlines:
News: [NEWS]
Headline: [HEADLINE]

Stage 3: Generating Stylized Headlines
Generate a news headline with the previously learned style based on the news text given above:
News: [NEWS]
Style headline:

Table 10: The data template used for GPT3.5-prompting.

A Appendix

A.1 GPT3.5-prompting Details
We perform few-shot prompting with the gpt-3.5-turbo API. We follow the unsupervised setup of StyleBART which assumes paired news-stylistic headlines are unavailable. As a results, we use three non-parallel stylistic sentences, five news-plain headline pairs in our data template. Table 10 shows the details. We set the the temperature to 0.5.

A.2 LLaMA2-InsTuning Details
We sample 10,000 data points per task\(^9\) and construct the corresponding instruction data. Table 11 shows our data template in training and inference stages. We utilize the 40,000 instruction data jointly to fine-tune LLaMA2-7B with LoRA. The hyperparameters for the LoRA module are set as follows: \(lora\_rank = 256\), \(lora\_alpha = 256\), and \(lora\_dropout = 0.05\). The training process employs the AdamW optimizer with parameters \(\beta_1 = 0.9\), \(\beta_2 = 0.999\). We adopt a learning rate of \(2e - 5\). The training is executed with a batch size of 80 for one epoch, amounting to a total of 500 steps. After our training stage, we directly use the inference template in Table 11 to evaluate the fine-tuned model.

\(^9\)Our training loss shows no significant reductions after 150 steps when using a batch size of 80. Therefore, we do not use the full dataset as larger dataset requires more computation and only brings marginal performance improvement.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverse Paraphrasing Task</td>
<td>Instruction: Convert the following sentence to its [STYLE] version.</td>
</tr>
<tr>
<td></td>
<td>Input: [PARAPHRASED TEXT]</td>
</tr>
<tr>
<td>Output: [STYLISTIC TEXT]</td>
<td></td>
</tr>
</tbody>
</table>

| Headline Generation Task | Instruction: Read the following news abstract and write a headline for it. |
| | Input: News: [NEWS] |
| Output: Headline: [HEADLINE] |

| Stylistic Headline Generation | Instruction: Read the following news abstract and write a [STYLE] version headline of the news. |
| | Input: News: [NEWS] |
| Output: [STYLE] version headline: |

Table 11: The data template used for LLaMA2-InsTuning training and inference stage.

A.3 Comparison with TitleStylist-BART
TitleStylist is initialized using the MASS model, a pretrained model with an encoder-decoder structure similar to BART. We adopt the methodology employed in TitleStylist and apply it to the BART model. We execute both TitleStylist-MASS (with their open-source code) and TitleStylist-BART (with our reimplemented code). As shown in Table 12, We can find that these two variations yield similar ROUGE and BERTScore results while TitleStylist-MASS gets better PPL and PPL-S scores. Therefore, we present the results for TitleStylist-MASS in this paper, as it demonstrates better performance and is consistent with the original paper.
<table>
<thead>
<tr>
<th>Style</th>
<th>Method</th>
<th>Generation Quality</th>
<th>Style Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R1(↑)</td>
<td>R2(↑)</td>
</tr>
<tr>
<td>Humor</td>
<td>TitleStylist-BART</td>
<td>28.2</td>
<td>10.0</td>
</tr>
<tr>
<td></td>
<td>TitleStylist-MASS</td>
<td>27.7</td>
<td>10.0</td>
</tr>
<tr>
<td>Romance</td>
<td>TitleStylist-BART</td>
<td>28.4</td>
<td>10.1</td>
</tr>
<tr>
<td></td>
<td>TitleStylist-MASS</td>
<td>27.6</td>
<td>9.9</td>
</tr>
<tr>
<td>Clickbait</td>
<td>TitleStylist-BART</td>
<td>28.2</td>
<td>9.8</td>
</tr>
<tr>
<td></td>
<td>TitleStylist-MASS</td>
<td>28.2</td>
<td>10.1</td>
</tr>
</tbody>
</table>

Table 12: Comparison results of TitleStylist-BART and TitleStylist-MASS.