Enhancing Content Preservation in Text Style Transfer Using Reverse Attention and Conditional Layer Normalization

Dongkyu Lee **Zhiliang Tian** **Lanqing Xue** **Nevin L. Zhang**
Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology
{dleear, ztianac, lxueaa, lzhang}@cse.ust.hk

Abstract

Text style transfer aims to alter the style (e.g., sentiment) of a sentence while preserving its content. A common approach is to map a given sentence to content representation that is free of style, and the content representation is fed to a decoder with a target style. Previous methods in filtering style completely remove tokens with style at the token level, which incurs the loss of content information. In this paper, we propose to enhance content preservation by implicitly removing the style information of each token with reverse attention, and thereby retain the content. Furthermore, we fuse content information when building the target style representation, making it dynamic with respect to the content. Our method creates not only style-independent content representation, but also content-dependent style representation in transferring style. Empirical results show that our method outperforms the state-of-the-art baselines by a large margin in terms of content preservation. In addition, it is also competitive in terms of style transfer accuracy and fluency.

1 Introduction

Style transfer is a popular task in computer vision and natural language processing. It aims to convert an input with a certain style (e.g., sentiment, formality) into a different style while preserving the original content.

One mainstream approach is to separate style from content, and to generate a transferred sentence conditioned on the content information and a target style. Recently, several models (Li et al., 2018; Xu et al., 2018; Wu et al., 2019) have proposed removing style information at the token level by filtering out tokens with style information, which are identified using either attention-based methods (Bahdanau et al., 2015) or frequency-ratio based methods (Wu et al., 2019). This line of work is built upon the assumption that style is *localized* to certain tokens in a sentence, and a token has *either* content or style information, but *not both*. Thus by utilizing a style marking module, the models filter out the style tokens entirely when constructing a style-independent content representation of the input sentence. The drawback with the filtering method is that one needs to manually set a threshold to decide whether a token is stylistic or content-related. Previous studies address this issue by using the average attention score as a threshold (Li et al., 2018; Xu et al., 2018; Wu et al., 2019). A major shortcoming of this approach is the incapability of handling flat attention distribution. When the distribution is flat, in which similar attention scores are assigned to tokens, the style marking module would remove/mask out more tokens than necessary. This incurs information loss in content as depicted in Figure 1.

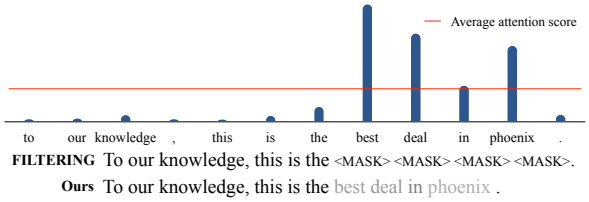


Figure 1: Illustration of difference between our method and filtering method in handling flat attention distribution. Each bar indicates attention score of the corresponding word.

In this paper, we propose a novel method for text style transfer. A key idea is to exploit the fact that a token often posses both style and content information. For example, the word "delicious" is a token with strong style information, but it also implies the subject is food. Such words play a pivotal role in representing style (e.g., positive sentiment) as well as presenting a hint at the subject matter/content (e.g., food). The complete removal of such tokens leads to the loss of content information.

For the sake of enhancing content preservation,

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 93–102
August 1–6, 2021.

we propose a method to *implicitly* remove style at the token level using *reverse attention*. We utilize knowledge attained from attention networks (Bahdanau et al., 2015) to estimate style information of a token, and suppress such signal to take out style. Attention mechanism is known to attend to interdependent representations given a query. In style classification task, an attention score could be interpreted as to what extent a token has style attribute. If we can identify which tokens reveal stylistic property and to what extent, it is then possible to take the negation and to approximate the amount of content attribute within a token. In this paper, we call it reverse attention. We utilize such score to suppress the stylistic attribute of tokens, fully capturing content property.

This paper further enhances content preservation by fusing content information in creating target style representation. Despite of extensive efforts in creating content representation, the previous work has overlooked building content-dependent style representations. The common approach is to project the target style onto an embedding space, and share the style embedding among the same style as an input to the decoder. However, our work sheds light on building *content-related style* by utilizing conditional layer normalization (CLN). This module of ours takes in content representations, and creates content-dependent style representation by shaping the content variable to fit in the distribution of target style. This way, our style representation varies according to the content of the input sequence even with the same target style.

Our method is based on two techniques, Reverse Attention and Conditional Layer Normalization, thus we call it RACoLN. In empirical evaluation, RACoLN achieves the state-of-the-art performance in terms of content preservation, outperforming the previous state-of-the-art by a large margin, and shows competency in style transfer accuracy and fluency. The contributions are as follows:

- We introduce reverse attention as a way to suppress style information while preserving content information when building a content representation of an input.
- Aside from building style-independent content representation, our approach utilizes conditional layer normalization to construct content-dependent style representation.
- Our model achieves state-of-the-art performance in terms of content preservation, outperforming current state-of-the-art by more than 4 BLEU score on Yelp dataset, and shows competency in other metrics as well.

2 Related Work

In recent years, text style transfer in unsupervised learning environment has been studied and explored extensively. Text style transfer task views a sentence as being comprised of content and style. Thus, there have been attempts to disentangle the components (Shen et al., 2017; Li et al., 2018; Xu et al., 2018; Wu et al., 2019). Shen et al. (2017) map a sentence to a shared content space among styles to create style-independent content variable. Some studies view style as localized feature of sentences. Xu et al. (2018) propose to identify style tokens with attention mechanism, and filter out such tokens. Frequency-based is proposed to enhance the filtering process (Wu et al., 2019). This stream of work is similar to our work in that the objective is to take out style at the token level, but different since ours does not remove tokens completely.

Instead of disentangling content and style, other papers focus on revising an entangled representation of an input. A few previous studies utilize a pre-trained classifier and edit entangled latent variable until it contains target style using the gradient-based optimization (Wang et al., 2019; Liu et al., 2020). He et al. (2020) view each domain of data as a partially observable variable, and transfer sentence using amortized variational inference. Dai et al. (2019) use the transformer architecture and rewrite style in the entangled representation at the decoder. We consider this model as the strongest baseline model in terms of content preservation.

In the domain of computer vision, it is a prevalent practice to exploit variants of normalization to transfer style (Dumoulin et al., 2017; Ulyanov et al., 2016). Dumoulin et al. (2017) proposed *conditional instance normalization* (CIN) in which each style is assigned with separate instance normalization parameter, in other words, a model learns separate gain and bias parameters of instance normalization for each style.

Our work differs in several ways. Style transfer in image views style transfer as changing the "texture" of an image. Therefore, Dumoulin et al. (2017) place CIN module following every convolution layer, "painting" with style-specific parameters on the content representation. Therefore, the

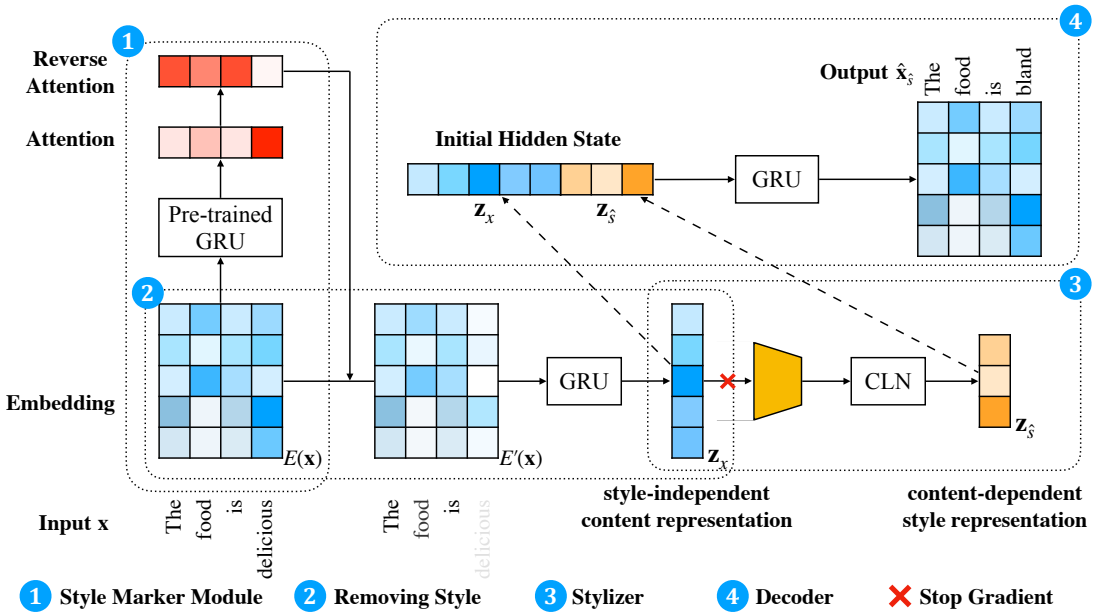


Figure 2: Input $\mathbf{x}$ first passes style marker module for computing reverse attention. The reverse attention score is then applied to token embeddings, implicitly removing style. The content representation from the encoder is fed to stylizer, in which style representation is made from the content. The decoder generates transferred output by conditioning on the two representations.

network passes on entangled representation of an image. Our work is different in that we disentangle content and style, thus we do not overwrite content with style-specific parameters. In addition, we apply CLN only once before passing it to decoder.

3 Approach

3.1 Task Definition

Let $\mathcal{D} = \{(\mathbf{x}_i, s_i)_{i=1}^N\}$ be a training corpus, where each $\mathbf{x}_i$ is a sentence, and s_i is its style label. Our experiments were carried on a sentiment analysis task, where there are two style labels, namely "positive" and "negative."

The task is to learn from $\mathcal{D}$ a model $\hat{\mathbf{x}}_{\hat{s}} = f_\theta(\mathbf{x}, \hat{s})$, with parameters θ, that takes an input sentence $\mathbf{x}$ and a target style $\hat{s}$ as inputs, and outputs a new sentence $\hat{\mathbf{x}}_{\hat{s}}$ that is in the target style and retains the content information of $\mathbf{x}$.

3.2 Model Overview

We conduct this task in an unsupervised environment in which ground truth sentence $\mathbf{x}_{\hat{s}}$ is not provided. To achieve our goal, we employ a style classifier $s = C(\mathbf{x})$ that takes a sentence $\mathbf{x}$ as input and returns its style label. We pre-train such model on $\mathcal{D}$ and keep it frozen in the process of learning f_θ.

Given the style classifier $C(\mathbf{x})$, our task becomes to learn a model $\hat{\mathbf{x}}_{\hat{s}} = f_\theta(\mathbf{x}, \hat{s})$ such that $C(\hat{\mathbf{x}}_{\hat{s}}) = \hat{s}$. As such, the task is conceptually similar to adversarial attack: The input $\mathbf{x}$ is from the style class s, and we want to modify it so that it will be classified into the target style class $\hat{s}$.

The architecture of our model f_θ is shown in Figure 2, which will some times referred to as the generator network. It consists of an encoder, a stylizer and a decoder. The encoder maps an input sequence $\mathbf{x}$ into a style-independent representation $\mathbf{z}_\mathbf{x}$. Particularly, the encoder has a style marker module that computes attention scores of input tokens, and it "reverses" them to estimate the content information. The reversed attention scores are applied to the token embedding $E(\mathbf{x})$ and the results $E'(\mathbf{x})$ are fed to bidirectional GRU to produce $\mathbf{z}_\mathbf{x}$.

The stylizer takes a target style $\hat{s}$ and the content representation $\mathbf{z}_\mathbf{x}$ as inputs, and produces a content-related style representation $\mathbf{z}_{\hat{s}}$. Finally, the decoder takes the content representation $\mathbf{z}_\mathbf{x}$ and style representation $\mathbf{z}_{\hat{s}}$ as inputs, and generates a new sequence $\hat{\mathbf{x}}_{\hat{s}}$.

3.3 Encoder

3.3.1 Style Marker Module

Let $\mathbf{x} = [x_1, x_2, \ldots, x_T]$ be a length T sequence of input with a style s. The style marker module is pre-trained in order to calculate the amount of style information in each token in a given input. We use one layer of bidirectional GRU with attention

(Yang et al., 2016). Specifically,

$$\mathbf{v}_t = tanh(\mathbf{W}_w \mathbf{h}_t + \mathbf{b}_w) \tag{1}$$

$$\alpha_t = \frac{exp(\mathbf{v}_t^\top \mathbf{u}/\tau)}{\sum_{t=1}^{T} exp(\mathbf{v}_t^\top \mathbf{u}/\tau)} \tag{2}$$

where $\mathbf{h}_t$ is the hidden representation from the bidirectional GRU at time step t. $\mathbf{u}$ is learnable parameters initialized with random weights, and τ denotes the temperature in softmax. When pre-training the style marker module, we construct a sentence representation by taking the weighted sum of the token representations with the weights being the attention scores, and feed the context vector to a fully-connected layer.

$$\mathbf{o} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t \tag{3}$$

$$\mathbf{p} = softmax(\mathbf{W}_c \mathbf{o} + \mathbf{b}_c) \tag{4}$$

The cross-entropy loss is used to learn the parameters of the style marker module. The attention scores in the style marker indicate what tokens are important to style classification, and to what extent. Those scores will be "reversed" in the next section to reveal the content information. The fully-connected layer of the style marker module is no longer needed once the style marker module is trained. It is hence removed.

3.3.2 Reverse Attention

Using attention score from the pre-trained style marker module, we propose to implicitly remove the style information in each token. We negate the extent of style information in each token to estimate the extent of content information, namely reverse attention.

$$\tilde{\alpha}_t = 1 - \alpha_t, \quad \sum_{t=1}^{T} \alpha_t = 1 \tag{5}$$

where α_t is an attention value from style marker module, and $\tilde{\alpha}_t$ is the corresponding reverse attention score. We multiply the reverse attention scores to the embedding vectors of tokens.

$$\tilde{\mathbf{e}}_t = \tilde{\alpha}_t \mathbf{e}_t, \quad \mathbf{e}_t = E(\mathbf{x}_t) \tag{6}$$

Intuitively, this can be viewed as implicitly removing the stylistic attribute of tokens, suppressing the norm of a token embedding respect to corresponding reverse attention score. The representations finally flow into a bidirectional GRU

$$\mathbf{z}_\mathbf{x} = bidirectionalGRU(\tilde{\mathbf{e}}) \tag{7}$$

to produce a content representation $\mathbf{z}_\mathbf{x}$, which is the last hidden state of the bidirectional GRU. By utilizing reverse attention, we map a sentence to style-independent content representation.

3.4 Stylizer

The goal of the stylizer is to create a content-related style representation. We do this by applying conditional layer normalization on the content representation $\mathbf{z}_\mathbf{x}$ from encoder as input to this module.

Layer normalization requires the number of gain and bias parameters to match the size of input representation. Therefore, mainly for the purpose of shrinking the size, we perform affine transformation on the content variable.

$$\tilde{\mathbf{z}}_\mathbf{x} = \mathbf{W}_z \mathbf{z}_\mathbf{x} + \mathbf{b}_z \tag{8}$$

The representation is then fed to conditional layer normalization so that the representation falls into target style distribution in style space. Specifically,

$$\mathbf{z}_{\hat{s}} = CLN(\tilde{\mathbf{z}}_\mathbf{x}; \hat{s}) = \gamma^{\hat{s}} \odot N(\tilde{\mathbf{z}}_\mathbf{x}) + \beta^{\hat{s}} \tag{9}$$

$$N(\tilde{\mathbf{z}}_\mathbf{x}) = \frac{\tilde{\mathbf{z}}_\mathbf{x} - \mu}{\sigma} \tag{10}$$

where μ and σ are mean and standard deviation of input vector respectively, and $\hat{s}$ is target style. Our model learns separate γ^s (gain) and β^s (bias) parameters for different styles.

Normalization method is commonly used to change feature values in common scale, but known to implicitly keep the features. Therefore, we argue that the normalized content feature values retain content information of the content variable. By passing through conditional layer normalization module, the content latent vector is scaled and shifted with style-specific gain and bias parameter, falling into target style distribution. Thus, unlike previous attempts in text style transfer, the style representation is dynamic respect to the content, being content-dependent embedding.

In order to block backpropagation signal related to style flowing into $\mathbf{z}_\mathbf{x}$, we apply stop gradient on $\mathbf{z}_\mathbf{x}$ before feeding it to stylizer.

3.5 Decoder

The decoder generates a sentence with the target style conditioned on content-related style representation and content representation. We construct our decoder using one single layer of GRU.

$$\hat{\mathbf{x}}_{\hat{s}} \sim Dec_\theta(\mathbf{z}_\mathbf{x}, \mathbf{z}_{\hat{s}}) = p_D(\hat{\mathbf{x}}_{\hat{s}}|\mathbf{z}_\mathbf{x}, \mathbf{z}_{\hat{s}}) \quad (11)$$

As briefly discussed in Section 3.2, the outputs from our generator are further passed on for different loss functions. However, sampling process or greedy decoding does not allow gradient to flow, because the methods are not differentiable. Therefore, we use soft sampling to keep the gradient flow. Specifically, when the gradient flow is required through the outputs, we take the product of probability distribution of each time step and the weight of embedding layer to project the outputs onto word embedding space. We empirically found that soft sampling is more suitable in our environment than gumbel-softmax (Jang et al., 2017).

3.6 Pre-trained Style Classifier

Due to the lack of parallel corpus, we cannot train generator network with maximum likelihood estimation on style transfer ability. Therefore, this paper employs a pre-trained classifier $C(\mathbf{x})$ to train our generator on transferring style. Our classifier network has the same structure as style marker module with fully-connected layer appended, nonetheless, it is a separate model obtained from a different set of initial model parameters. We use the cross-entropy loss for training:

$$\mathcal{L}_{pre} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}[\log p_C(s|\mathbf{x}_s)] \quad (12)$$

We freeze the weights of this network after it has been fully trained.

3.7 The Loss Function

As shown in Figure 3, our loss function consists of four parts: a self reconstruction loss $\mathcal{L}_{self}$, a cycle reconstruction loss $\mathcal{L}_{cycle}$, a content loss $\mathcal{L}_{content}$, and a style transfer loss $\mathcal{L}_{style}$.

3.7.1 Self Reconstruction Loss

Let $(\mathbf{x}, s) \in \mathcal{D}$ be a training example. If we ask our model to $f_\theta(\mathbf{x}, \hat{s})$ to "transfer" the input into its original style, i.e., $\hat{s} = s$, we would expect it to reconstruct the input.

$$\mathcal{L}_{self} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}[\log p_D(\mathbf{x}|\mathbf{z}_\mathbf{x}, \mathbf{z}_s)] \quad (13)$$

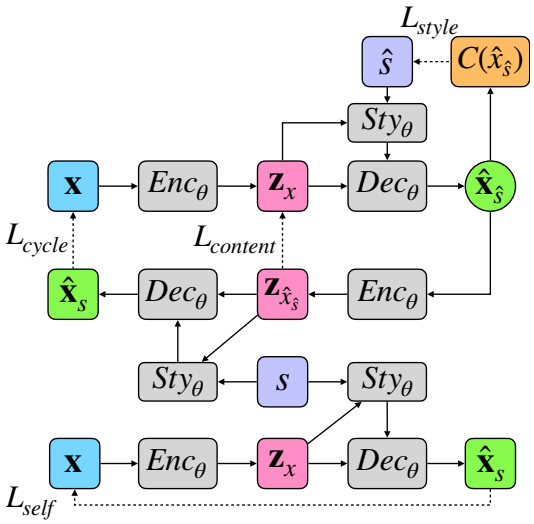


Figure 3: Illustration of loss functions in training phase. Enc_θ, Sty_θ, and Dec_θ denote the encoder, the stylizer, and the decoder respectively. The circle figure denotes a generated sentence with soft sampling. As illustrated, $\mathcal{L}_{cycle}$, $\mathcal{L}_{style}$ and $\mathcal{L}_{content}$ require soft sampling to keep the gradient flow.

where $\mathbf{z}_\mathbf{x}$ is the content representation of the input $\mathbf{x}$, $\mathbf{z}_s$ is the representation of the style s, and p_D is the conditional distribution over sequences defined by the decoder.

3.7.2 Cycle Reconstruction Loss

Suppose we first transfer a sequence $\mathbf{x}$ into another style $\hat{s}$ to get $\hat{\mathbf{x}}_{\hat{s}}$ using soft sampling, and then transfer $\hat{\mathbf{x}}_{\hat{s}}$ back to the original style s. We would expect to reconstruct the input $\mathbf{x}$. Hence we have the following cycle construction loss:

$$\mathcal{L}_{cycle} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}[\log p_D(\mathbf{x}|\mathbf{z}_{\hat{\mathbf{x}}_{\hat{s}}}, \mathbf{z}_s)] \quad (14)$$

where $\mathbf{z}_{\hat{\mathbf{x}}_{\hat{s}}}$ is the content representation of the transferred sequence $\hat{\mathbf{x}}_{\hat{s}}$.[1]

3.7.3 Content Loss

In the aforementioned cycle reconstruction process, we obtain a content representation $\mathbf{z}_\mathbf{x}$ of the input $\mathbf{x}$ and a content representation $\mathbf{z}_{\hat{\mathbf{x}}_{\hat{s}}}$ of the transferred sequence $\hat{\mathbf{x}}_{\hat{s}}$. As the two transfer steps presumably involve only style but not content, the two content representations should be similar. Hence we have the following content loss:

$$\mathcal{L}_{content} = \mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}||\mathbf{z}_\mathbf{x} - \mathbf{z}_{\hat{\mathbf{x}}_{\hat{s}}}||_2^2 \quad (15)$$

[1] Strictly speaking, the quantity is not well-defined because there is no description of how the target style $\hat{s}$ is picked. In our experiments, we use data with two styles. So, the target style just means the other style. To apply the method to problems with multiple styles, random sampling of different style should be added. This remark applies also to the two loss terms to be introduced below.

3.7.4 Style Transfer Loss

We would like the transferred sequence $\hat{\mathbf{x}}_{\hat{s}}$ to be of style $\hat{s}$. Hence we have the following style transfer loss:

$$\mathcal{L}_{style} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}[\log p_C(\hat{s}|\hat{\mathbf{x}}_{\hat{s}})] \quad (16)$$

where p_C is the conditional distribution over styles defined by the style classifier $C(\mathbf{x})$. As mentioned in Section 3.5, $\hat{\mathbf{x}}_{\hat{s}}$ was generated with soft sampling.

3.7.5 Total Loss

In summary, we balance the four loss functions to train our model.

$$\mathcal{L} = \lambda_1\mathcal{L}_{self} + \lambda_2\mathcal{L}_{cycle} + \lambda_3\mathcal{L}_{content} + \lambda_4\mathcal{L}_{style} \quad (17)$$

where λ_i is balancing parameter.

4 Experiment

4.1 Datasets

Following prior work on text style transfer, we use two common datasets: Yelp and IMDB review.

4.1.1 Yelp Review

Our study uses Yelp review dataset (Li et al., 2018) which contains 266K positive and 177K negative reviews. Test set contains a total of 1000 sentences, 500 positive and 500 negative, and human-annotated sentences are provided which are used in measuring content preservation.

4.1.2 IMDB Movie Review

Another dataset we test is IMDB movie review dataset (Dai et al., 2019). This dataset is comprised of 17.9K positive and 18.8K negative reviews for training corpus, and 2K sentences are used for testing.

4.2 Automatic Evaluation

4.2.1 Style Transfer Accuracy

Style transfer accuracy (S-ACC) measures whether the generated sentences reveal target style property. We have mentioned a style classifier before: $C(\mathbf{x})$ which is used in the loss function. To evaluate transfer accuracy, we train another style classifier $C_{eval}(\mathbf{x})$. It has the identical architecture as before and trained on the same data, except from a different set of initial model parameters. We utilize such structure due to its superior performance compared to that of commonly used CNN-based classifier (Kim, 2014). Our evaluation classifier achieves accuracy of 97.8% on Yelp and 98.9% on IMDB, which are higher than that of CNN-based.

4.2.2 Content Preservation

A well-transferred sentence must maintain its content. In this paper, content preservation was evaluated with two BLEU scores (Papineni et al., 2002), one between generated sentence and input sentence (self-BLEU), and the other with human-generated sentence (ref-BLEU). With this metric, one can evaluate how a sentence maintains its content throughout inference.

4.2.3 Fluency

A natural language generation task aims to output a sentence, which is not only task-specific, but also fluent. This study measures perplexity (PPL) of generated sentences in order to measure fluency. Following (Dai et al., 2019), we use 5-gram KenLM (Heafield, 2011) trained on the two training datasets. A lower PPL score indicates a transferred sentence is more fluent.

4.2.4 BERT Score

Zhang et al. (2020) proposed BERT score which computes contextual similarity of two sentences. Previous methods, such as BLEU score, compute n-gram matching score, while BERT score evaluates the contextual embedding of the tokens obtained from pre-trained BERT (Devlin et al., 2019). This evaluation metric has been shown to correlate with human judgement, thus our paper includes BERT score between model generated output and the human reference sentences. We report precision, recall, and F1 score.

4.3 Human Evaluation

In addition to automatic evaluation, we validate the generated outputs with human evaluation. With each model, we randomly sample 150 outputs from each of the two datasets, total of 300 outputs per model. Given the target style and the original sentence, the annotators are asked to evaluate the model generated sentence with a score range from 1 (Very Bad) to 5 (Very Good) on content preservation, style transfer accuracy, and fluency. We report the average scores from the 4 hired annotators in Table 3.

4.4 Implementation Details

In this paper, we set the embedding size to 128 dimension and hidden representation dimension of

Table 1: Automatic evaluation result on Yelp dataset. Bold numbers indicate best performance. G-Score denotes geometric mean of self-BLEU and S-ACC, and BERT-P, BERT-R, and BERT-F1 are BERT score precision, recall and F1 respectively. All the baseline model outputs and codes were used from their official repositories if provided to the public.

	Yelp							
	S-ACC	ref-BLEU	self-BLEU	PPL	G-score	BERT-P	BERT-R	BERT-F1
Cross-Alignment (Shen et al., 2017)[2]	74.2	4.2	13.2	53.1	32.0	87.8	86.2	87.0
ControlledGen (Hu et al., 2017)[3]	83.7	16.1	50.5	146.3	65.0	90.6	89.0	89.8
Style Transformer (Dai et al., 2019)[4]	87.3	19.8	55.2	73.8	69.4	91.6	89.9	90.7
Deep Latent (He et al., 2020)[5]	85.2	15.1	40.7	**36.7**	58.9	89.8	88.6	89.2
RACoLN (Ours)	**91.3**	**20.0**	**59.4**	60.1	**73.6**	**91.8**	**90.3**	**91.0**

Table 2: Automatic evaluation result on IMDB dataset. Bold numbers indicate best performance. As for IMDB Dataset, in the absence of human reference, BERT score and reference BLEU are not reported.

	IMDB			
	S-ACC	self-BLEU	PPL	G-score
Cross-Alignment	63.9	1.1	**29.9**	8.4
ControlledGen	81.2	63.8	119.7	71.2
Style Transformer	74.0	70.4	71.2	72.2
Deep Latent	59.3	64.0	41.1	61.6
RACoLN (Ours)	**83.1**	**70.9**	45.3	**76.8**

Table 3: Human evaluation result. Each score indicates the average score from the hired annotators. The inter-annotator agreement, Krippendorff's alpha, is 0.729.

	YELP			IMDB		
	Style	Content	Fluency	Style	Content	Fluency
Cross-Alignment	2.6	2.4	3.3	2.2	2.1	2.3
ControlledGen	3.3	4.0	3.7	3.3	3.8	3.6
Style Transformer	3.7	4.3	4.0	3.3	4.0	3.8
Deep Latent	3.5	3.6	**4.3**	2.7	3.7	**4.2**
RACoLN (Ours)	**4.0**	**4.5**	4.2	**3.6**	**4.1**	4.1

encoder to 500. The size of bias and gain parameters of conditional layer norm is 200, and the size of hidden representation for decoder is set to 700 to condition on both content and style representation. Adam optimizer (Kingma and Ba, 2015) was used to update parameter with learning rate set to 0.0005. For balancing parameters of total loss function, we set to 0.5 for λ_1 and λ_2, and 1 for the rest.

4.5 Experimental Result & Analysis

We compare our model with the baseline models, and the automatic evaluation result is presented in Table 1. Our model outperforms the baseline models in terms of content preservation on both of the datasets. Especially, on Yelp dataset, our model achieves 59.4 self-BLEU score, surpassing the previous state-of-the-art model by more than 4 points. Furthermore, our model also achieves the state-of-the-art result in content preservation on IMDB dataset, which is comprised of longer sequences than those of Yelp.

In terms of style transfer accuracy and fluency, our model is highly competitive. Our model achieves the highest score in style transfer accuracy on both of the datasets (91.3 on Yelp and 83.1 on IMDB). Additionally, our model shows the ability to produce fluent sentences as shown in the perplexity score. In terms of the BERT scores, the proposed model performs the best, having the highest contextual similarity with the human reference among the style transfer models.

With the automatic evaluation result, we see a trend of trade-off. Most of the baseline models are good at particular metric, but show room for improvement on other metrics. For example, Deep Latent and Cross-Alignment constantly perform well in terms of perplexity, but their ability to transfer style and preserving content needs improvement. Style Transformer achieves comparable performance across all evaluation metrics, but our model outperforms the model on every metric on both of the datasets. Therefore, the result shows that our model is well-balanced but also strong in every aspect in text style transfer task.

As for the human evaluation, we observe that the result mainly conform with the automatic evaluation. Our model received the highest score on the style and content evaluation metric on both of the datasets by a large margin compared to the other baselines. Moreover, the fluency score is comparable with that of Deep Latent model, showing its competency in creating a fluent output. Both automatic and human evaluation depict the strength of

[2] https://github.com/shentianxiao/language-style-transfer

[3] https://github.com/asyml/texar/tree/master/examples/text_style_transfer

[4] https://github.com/fastnlp/style-transformer

[5] https://github.com/cindyxinyiwang/deep-latent-sequence-model

Table 4: Sample outputs generated by the baseline models and our approach on Yelp and IMDB dataset. Bold words indicate successful transfer in style without grammatical error.

	YELP
Original Input	Everyone is always super friendly and helpful .
Cross-Alignment	Everyone is always super friendly and helpful and inattentive .
ControlledGen	Tonight selection of meats and cheeses .
Deep Latent	Now i 'm not sure how to be .
Style Transformer	Which is **n't** super friendly .
RACoLN (Ours)	Everyone is always super **rude** and **unprofessional** .
Original Input	I love this place , the service is always great !
Cross-Alignment	I know this place , the food is just a horrible !
ControlledGen	I **avoid** this place , the service is nasty depressing vomit
Deep Latent	I do n't know why the service is always great !
Style Transformer	I **do n't** recommend this place , the service is n't !
RACoLN (Ours)	I **avoid** this place , the service is always **horrible** !
	IMDB
Original Input	I actually disliked the leading characters so much that their antics were never funny but pathetic .
Cross-Alignment	I have never get a good movie , i have never have seen in this movie .
ControlledGen	I actually anticipated the leading characters so much that their antics were never funny but timeless .
Deep Latent	I actually disliked the leading characters so much that their antics were never funny but **incredible** .
Style Transformer	I actually disliked the leading characters so much that their antics were never funny but vhs .
RACoLN (Ours)	I actually **liked** the leading characters so much that their antics were **never corny but appropriate** .
Original Input	The plot is clumsy and has holes in it .
Cross-Alignment	The worst film is one of the worst movies i 've ever seen .
ControlledGen	The plot is **top-notch** and has one-liners in it .
Deep Latent	The plot is tight and has found it in a very well done .
Style Transformer	The plot is joys and has flynn in it .
RACoLN (Ours)	The plot is **incredible** and has **twists** in it .

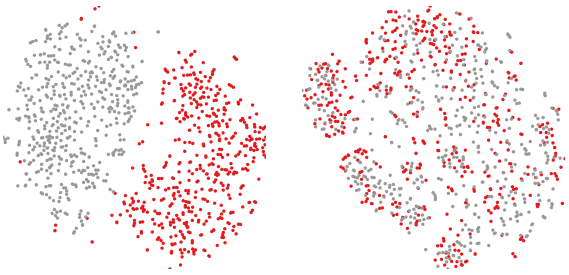
the proposed model not only in preserving content, but also on other metrics.

4.5.1 Style and Content Space

We visualize the test dataset of Yelp projected on content and style space using t-SNE in Figure 4. It is clearly observed that the content representations ($\mathbf{z_x}$) are spread across content space, showing that the representations are independent of style. After the content representations go through the stylizer module, there is a clear distinction between different styles representations ($\mathbf{z}_{\hat{s}}$) in style space. This is in sharp contrast to the corresponding distributions of the style-independent content representations shown on the right of the figure. The figure clearly depicts how style-specific parameters in the stylizer module shape the content representations to fall in the target style distribution. This figure illustrates how our model successfully removes style at the encoder, and constructs content-related style at the stylizer module.

4.5.2 Ablation Study

In order to validate the proposed modules, we conduct ablation study on Yelp dataset which is presented in Table 5. We observe a significant drop across all aspects without the reverse attention module. In other case, where we remove the stylizer module and use style embedding as in the previous papers, the model loses the ability to retain content, drop of around 6 score on self-BLEU. We find that the two core components are interdependent in successfully transferring style in text. Lastly, as for the loss functions, incorporating $\mathcal{L}_{content}$ brings a meaningful increase in content preservation.[6]



Style Space Content Space

Figure 4: Visualization of Yelp test dataset on content and style space using t-SNE. Gray dots denote sentences with negative style transferred to positive sentiment, while red dots are sentences with positive style transferred to negative sentiment.

Table 5: Ablation study on the proposed model. (-) indicates removing the corresponding component from the proposed model.

	S-ACC	ref-BLEU	self-BLEU	PPL
Input Copy	2.2	22.7	100.0	41.2
Proposed Model	91.3	20.0	59.4	60.1
(-) Reverse Attention	84.0	16.6	47.2	60.5
(-) Stylizer	91.8	19.1	53.0	59.0
(-) $\mathcal{L}_{content}$	87.2	19.5	54.8	62

5 Conclusion

In this paper, we introduce a way to implicitly remove style at the token level using reverse attention, and fuse content information to style representation using conditional layer normalization. With the two core components, our model is able to enhance content preservation while keeping the outputs fluent with target style. Both automatic and human evaluation shows that our model has the best ability in preserving content and is strong in other metrics as well. In the future, we plan to study problems with more than two styles and apply multiple attribute

[6]Other loss functions were not included, since the loss functions have been extensively tested and explored in previous papers (Prabhumoye et al., 2018; Dai et al., 2019).

style transfer, where the target style is comprised of multiple styles.

Acknowledgement

Research on this paper was supported by Hong Kong Research Grants Council under grant 16204920 and Tencent AI Lab Rhino-Bird Focused Research Program (No. GF202035).

Ethical Considerations

A text style transfer model is a conditional generative model, in which the condition is the target style. This makes a wide range of applications possible, since a style can be defined as any common feature in a corpus, such as formality, tense, sentiment, etc.

However, at the same time, due to its inherent functionality, a text style transfer model can pose potential harm when used with a malicious intention. It can lead to a situation where one deliberately distorts a sentence for his or her own benefit. To give an example in a political context, political stance can be viewed a style in political slant dataset (Voigt et al., 2018) as in (Prabhumoye et al., 2018). If one intentionally changes the style (political stance) of a person with the proposed model structure, the generated output can be exploited to create fake news or misinformation. One possible remedy for such potentially problematic situation is to employ fact checking system as a safety measure (Nadeem et al., 2019). We are fully aware that fact checking is not the fundamental solution to the potential harm that text style transfer models possess. Nevertheless, one can filter out misleading information using the system in certain domains (i.e., politics), lowering the level of the danger that can be otherwise posed by style transfer. In conclusion, such problem is shared among conditional generative models in general, and future studies on how to mitigate this problem are in crucial need.

Our work validates the proposed model and the baseline models on human evaluation, in which manual work was involved. Thus, we disclose the compensation level given to the hired annotators. The average lengths of the two corpora tested are 10.3 words for Yelp and 15.5 words for IMDB. In addition, the annotation was performed on sentence-level, in which the annotators were asked to score a model generated sentence. Considering the length and the difficulty, the expected annotations per hour was 100 sentences. The hourly pay was set to 100 Hong Kong dollars (HK$), which is higher than Hong Kong's statutory minimum wage. The annotators evaluated 1,500 sentences in total (750 sentences per dataset), thus each annotator was compensated with the total amount of HK$1,500.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.

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

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2017. A learned representation for artistic style. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th*

International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 2020. Revision in continuous space: Unsupervised text style transfer without adversarial learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8376–8383. AAAI Press.

Moin Nadeem, Wei Fang, Brian Xu, Mitra Mohtarami, and James Glass. 2019. FAKTA: An automatic end-to-end fact checking system. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 78–83, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6830–6841. Curran Associates, Inc.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022.

Rob Voigt, David Jurgens, Vinodkumar Prabhakaran, Dan Jurafsky, and Yulia Tsvetkov. 2018. RtGender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Advances in Neural Information Processing Systems 32*, pages 11036–11046. Curran Associates, Inc.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5271–5277. International Joint Conferences on Artificial Intelligence Organization.

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, Melbourne, Australia. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.