Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation

Ning Dai, Jianze Liang, Xipeng Qiu[*]**, Xuanjing Huang**

Shanghai Key Laboratory of Intelligent Information Processing, Fudan University

School of Computer Science, Fudan University

825 Zhangheng Road, Shanghai, China

{ndai16,jzliang18,xpqiu,xjhuang}@fudan.edu.cn

Abstract

Disentangling the content and style in the latent space is prevalent in unpaired text style transfer. However, two major issues exist in most of the current neural models. 1) It is difficult to completely strip the style information from the semantics for a sentence. 2) The recurrent neural network (RNN) based encoder and decoder, mediated by the latent representation, cannot well deal with the issue of the long-term dependency, resulting in poor preservation of non-stylistic semantic content. In this paper, we propose the Style Transformer, which makes no assumption about the latent representation of source sentence and equips the power of attention mechanism in Transformer to achieve better style transfer and better content preservation. Source code will be available on Github[1].

1 Introduction

Text style transfer is the task of changing the stylistic properties (e.g., sentiment) of the text while retaining the style-independent content within the context. Since the definition of the text style is vague, it is difficult to construct paired sentences with the same content and differing styles. Therefore, the studies of text style transfer focus on the unpaired transfer.

Recently, neural networks have become the dominant methods in text style transfer. Most of the previous methods (Hu et al., 2017; Shen et al., 2017; Fu et al., 2018; Carlson et al., 2017; Zhang et al., 2018b,a; Prabhumoye et al., 2018; Jin et al., 2019; Melnyk et al., 2017; dos Santos et al., 2018) formulate the style transfer problem into the "encoder-decoder" framework. The encoder maps the text into a style-independent latent representation (vector representation), and the decoder generates a new text with the same content but a different style from the disentangled latent representation plus a style variable.

These methods focus on how to disentangle the content and style in the latent space. The latent representation needs better preserve the meaning of the text while reducing its stylistic properties. Due to lacking paired sentence, an adversarial loss (Goodfellow et al., 2014) is used in the latent space to discourage encoding style information in the latent representation. Although the disentangled latent representation brings better interpretability, in this paper, we address the following concerns for these models.

1) It is difficult to judge the quality of disentanglement. As reported in (Elazar and Goldberg, 2018; Lample et al., 2019), the style information can be still recovered from the latent representation even the model has trained adversarially. Therefore, it is not easy to disentangle the stylistic property from the semantics of a sentence.

2) Disentanglement is also unnecessary. Lample et al. (2019) reported that a good decoder can generate the text with the desired style from an entangled latent representation by "overwriting" the original style.

3) Due to the limited capacity of vector representation, the latent representation is hard to capture the rich semantic information, especially for the long text. The recent progress of neural machine translation also proves that it is hard to recover the target sentence from the latent representation without referring to the original sentence.

4) To disentangle the content and style information in the latent space, all of the existing approaches have to assume the input sentence is encoded by a fix-sized latent vector. As a result, these approaches can not directly apply the attention mechanism to enhance the ability to preserve

---

[*]Corresponding author

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

the information in the input sentence.

5) Most of these models adopt recurrent neural networks (RNNs) as encoder and decoder, which has a weak ability to capture the long-range dependencies between words in a sentence. Besides, without referring the original text, RNN-based decoder is also hard to preserve the content. The generation quality for long text is also uncontrollable.

In this paper, we address the above concerns of disentangled models for style transfer. Different from them, we propose Style Transformer, which takes Transformer (Vaswani et al., 2017) as the basic block. Transformer is a fully-connected self-attention neural architecture, which has achieved many exciting results on natural language processing (NLP) tasks, such as machine translation (Vaswani et al., 2017), language modeling (Dai et al., 2019), text classification (Devlin et al., 2018). Different from RNNs, Transformer uses stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. Moreover, Transformer decoder fetches the information from the encoder part via attention mechanism, compared to a fixed size vector used by RNNs. With the strong ability of Transformer, our model can transfer the style of a sentence while better preserving its meaning. The difference between our model and the previous model is shown in Figure 1.

Our contributions are summarized as follows:

- We introduce a novel training algorithm which makes no assumptions about the disentangled latent representations of the input sentences, and thus the model can employ attention mechanisms to improve its performance further.
- To the best of our knowledge, this is the first work that applies the Transformer architecture to style transfer task.
- Experimental results show that our proposed approach generally outperforms the other approaches on two style transfer datasets. Specifically, to the content preservation, Style Transformer achieves the best performance with a significant improvement.

2   Related Work

Recently, many text style transfer approaches have been proposed. Among these approaches, there is a line of works aims to infer a latent representation for the input sentence, and manipulate

the style of the generated sentence based on this learned latent representation. Shen et al. (2017) propose a cross-aligned auto-encoder with adversarial training to learn a shared latent content distribution and a separated latent style distribution. Hu et al. (2017) propose a new neural generative model which combines variational auto-encoders and holistic attribute discriminators for the effective imposition of semantic structures. Following their work, many methods (Fu et al., 2018; John et al., 2018; Zhang et al., 2018a,b) has been proposed based on standard encoder-decoder architecture.

Although, learning a latent representation will make the model more interpretable and easy to manipulate, the model which is assumed a fixed size latent representation cannot utilize the information from the source sentence anymore.

On the other hand, there are also some approaches without manipulating latent representation are proposed recently. Xu et al. (2018) propose a cycled reinforcement learning method for unpaired sentiment-to-sentiment translation task. Li et al. (2018) propose a three-stage method. Their model first extracts content words by deleting phrases a strong attribute value, then retrieves new phrases associated with the target attribute, and finally uses a neural model to combine these into a final output. Lample et al. (2019) reduce text style transfer to unsupervised machine translation problem (Lample et al., 2018). They employ Denoising Auto-encoders (Vincent et al., 2008) and back-translation (Sennrich et al., 2016) to build a translation style between different styles.

However, both lines of the previous models make few attempts to utilize the attention mechanism to refer the long-term history or the source sentence, except Lample et al. (2019). In many NLP tasks, especially for text generation, attention mechanism has been proved to be an essential technique to enable the model to capture the long-term dependency (Bahdanau et al., 2014; Luong et al., 2015; Vaswani et al., 2017).

In this paper, we follow the second line of work and propose a novel method which makes no assumption about the latent representation of source sentence and takes the proven self-attention network, Transformer, as a basic module to train a style transfer system.
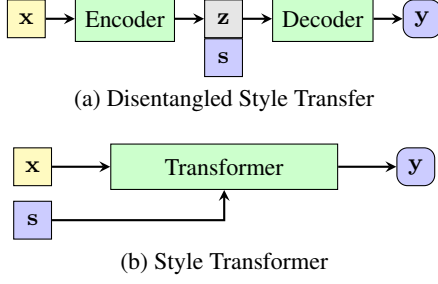
(a) Disentangled Style Transfer

(b) Style Transformer

Figure 1: General illustration of previous models and our model. $\mathbf{z}$ denotes style-independent content vector and $\mathbf{s}$ denotes the style variable.

3 Style Transformer

To make our discussion more clearly, in this section, we will first give a brief introduction to the style transfer task, and then start to discuss our proposed model based on our problem definition.

3.1 Problem Formalization

In this paper, we define the style transfer problem as follows: Considering a bunch of datasets $\{\mathcal{D}_i\}_{i=1}^K$, and each dataset $\mathcal{D}_i$ is composed of many natural language sentences. For all of the sentences in a single dataset $\mathcal{D}_i$, they share some specific characteristic (e.g. they are all the positive reviews for a specific product), and we refer this shared characteristic as the *style* of these sentences. In other words, a style is defined by the distribution of a dataset. Suppose we have K different datasets $\mathcal{D}_i$, then we can define K different styles, and we denote each style by the symbol $\mathbf{s}^{(i)}$. The goal of style transfer is that: given a arbitrary natural language sentence $\mathbf{x}$ and a desired style $\widehat{\mathbf{s}} \in \{\mathbf{s}^{(i)}\}_{i=1}^K$, rewrite this sentence to a new one $\widehat{\mathbf{x}}$ which has the style $\widehat{\mathbf{s}}$ and preserve the information in original sentence $\mathbf{x}$ as much as possible.

3.2 Model Overview

To tackle the style transfer problem we defined above, our goal is to learn a mapping function $f_\theta(\mathbf{x}, \mathbf{s})$ where $\mathbf{x}$ is a natural language sentence and $\mathbf{s}$ is a style control variable. The output of this function is the transferred sentence $\widehat{\mathbf{x}}$ for the input sentence $\mathbf{x}$.

A big challenge in the text style transfer is that we have no access to the parallel corpora. Thus we can't directly obtain supervision to train our transfer model. In section 3.4, we employ two discriminator-based approaches to create supervision from non-parallel corpora.

Finally, we will combine the Style Transformer network and discriminator network via an overall learning algorithm in section 3.5 to train our style transfer system.

3.3 Style Transformer Network

Generally, Transformer follows the standard encoder-decoder architecture. Explicitly, for a input sentence $\mathbf{x} = (x_1, x_2, ..., x_n)$, the Transformer encoder $Enc(\mathbf{x}; \theta_E)$ maps inputs to a sequence of continuous representations $\mathbf{z} = (z_1, z_2, ..., z_n)$. And the Transformer decoder $Dec(\mathbf{z}; \theta_D)$ estimates the conditional probability for the output sentence $\mathbf{y} = (y_1, y_2, ..., y_n)$ by auto-regressively factorized its as:

$$p_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^m p_\theta(y_t|\mathbf{z}, y_1, ..., y_{t-1}). \quad (1)$$

At each time step t, the probability of the next token is computed by a softmax classifier:

$$p_\theta(y_t|\mathbf{z}, y_1, ..., y_{t-1}) = \text{softmax}(\mathbf{o}_t), \quad (2)$$

where $\mathbf{o}_t$ is logit vector outputted by decoder network.

To enable style control in the standard Transformer framework, we add a extra style embedding as input to the Transformer encoder $Enc(\mathbf{x}, \mathbf{s}; \theta_E)$. Therefore the network can compute the probability of the output condition both on the input sentence $\mathbf{x}$ and the style control variable $\mathbf{s}$. Formally, this can be expressed as:

$$p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{s}) = \prod_{t=1}^m p_\theta(y_t|\mathbf{z}, y_1, ..., y_{t-1}), \quad (3)$$

and we denote the predicted output sentence of this network by $f_\theta(\mathbf{x}, \mathbf{s})$.

3.4 Discriminator Network

Suppose we use $\mathbf{x}$ and $\mathbf{s}$ to denote the sentence and its style from the dataset $\mathcal{D}$. Because of the absence of the parallel corpora, we can't directly obtain the supervision for the case $f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ where $\mathbf{s} \neq \widehat{\mathbf{s}}$. Therefore, we introduce a discriminator network to learn this supervision from the non-parallel copora.

The intuition behind the training of discriminator is based on the assumption below: As we mentioned above, we only have the supervision for the case $f_\theta(\mathbf{x}, \mathbf{s})$. In this case, because of the input sentence $\mathbf{x}$ and chosen style $\mathbf{s}$ are both come from

the same dataset $\mathcal{D}$, one of the optimum solutions, in this case, is to reproduce the input sentence. Thus, we can train our network to reconstruct the input in this case. In the case of $f_\theta(\mathbf{x}, \mathbf{s})$ where $\mathbf{s} \neq \widehat{\mathbf{s}}$, we construct supervision from two ways. 1) For the content preservation, we train the network to reconstruct original input sentence $\mathbf{x}$ when we feed transferred sentence $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ to the Style Transformer network with the original style label $\mathbf{s}$. 2) For the style controlling, we train a discriminator network to assist the Style Transformer network to better control the style of the generated sentence.

In short, the discriminator network is another Transformer encoder, which learns to distinguish the style of different sentences. And the Style Transformer network receives style supervision from this discriminator. To achieve this goal, we experiment with two different discriminator architectures.

Conditional Discriminator In a setting similar to Conditional GANs (Mirza and Osindero, 2014), discriminator makes decision condition on a input style. Explicitly, a sentence $\mathbf{x}$ and a proposal style $\mathbf{s}$ are feed into discriminator $d_\phi(\mathbf{x}, \mathbf{s})$, and the discriminator is asked to answer whether the input sentence has the corresponding style. In discriminator training stage, the real sentence from datasets $\mathbf{x}$, and the reconstructed sentence $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$ are labeled as *positive*, and the transferred sentences $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ where $\mathbf{s} \neq \widehat{\mathbf{s}}$, are labeled as *negative*. In Style Transformer network training stage, the network f_θ is trained to maximize the probability of *positive* when feed $f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ and $\widehat{\mathbf{s}}$ to the discriminator.

Multi-class Discriminator Different from the previous one, in this case, only one sentence is feed into discriminator $d_\phi(\mathbf{x})$, and the discriminator aims to answer the style of this sentence. More concretely, the discriminator is a classifier with $K + 1$ classes. The first K classes represent K different styles, and the last class is stand for the generated data from $f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$, which is also often referred as fake sample. In discriminator training stage, we label the real sentences $\mathbf{x}$ and reconstructed sentences $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$ to the label of the corresponding style. And for the transferred sentence $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ where $\mathbf{s} \neq \widehat{\mathbf{s}}$, is labeled as the class 0. In Style Transformer network learning stage, we train the network $f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ to maximize
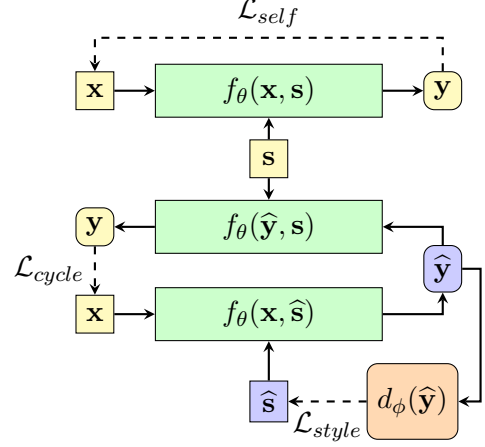

Figure 2: The training process for Style Transformer network. The input sentence $\mathbf{x}$ and input style $\mathbf{s}(\widehat{\mathbf{s}})$ is feed into Transformer network f_θ. If the input style $\mathbf{s}$ is the same as the style of sentence $\mathbf{x}$, generated sentence $\mathbf{y}$ will be trained to reconstruct $\mathbf{x}$. Otherwise, the generated sentence $\widehat{\mathbf{y}}$ will be feed into Transformer f_θ and discriminator d_ϕ to reconstruct input sentence $\mathbf{x}$ and input style $\widehat{\mathbf{s}}$ respectively.

the probability of the class which is stand for style $\widehat{\mathbf{s}}$.

3.5 Learning Algorithm

In this section, we will discuss how to train these two networks. And the training algorithm of our model can be divided into two parts: the discriminator learning and Style Transformer network learning. The brief illustration is shown in Figure 2.

3.5.1 Discriminator Learning

Loosely speaking, in the discriminator training stage, we train our discriminator to distinguish between the real sentence $\mathbf{x}$ and reconstructed sentence $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$ from the transferred sentence $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$. The loss function for the discriminator is simply the cross-entropy loss of the classification problem.

For the conditional discriminator:

$$\mathcal{L}_{discriminator}(\phi) = -p_\phi(\mathbf{c}|\mathbf{x}, \mathbf{s}). \quad (4)$$

And for the multi-class discriminator:

$$\mathcal{L}_{discriminator}(\phi) = -p_\phi(\mathbf{c}|\mathbf{x}). \quad (5)$$

According to the difference of discriminator architecture, there is a different protocol for how to label these sentences, and the details can be found in Algorithm 1.

Algorithm 1: Discriminator Learning

Input: Style Transformer f_θ, discriminator d_ϕ, and a dataset $\mathcal{D}_i$ with style $\mathbf{s}$

1   Sample a minibatch of m sentences $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$ from $\mathcal{D}_i$. ;
2   **foreach** $\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$ **do**
3   Randomly sample a style $\widehat{\mathbf{s}}(\mathbf{s} \neq \widehat{\mathbf{s}})$;
4   Use f_θ to generate two new sentence
5     $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$
6     $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$;
7   **if** d_ϕ *is conditional discriminator* **then**
8   Label $\{(\mathbf{x}, \mathbf{s}), (\mathbf{y}, \mathbf{s})\}$ as 1 ;
9   Label $\{(\mathbf{x}, \widehat{\mathbf{s}}), (\widehat{\mathbf{y}}, \widehat{\mathbf{s}})\}$ as 0 ;
10   **else**
11   Label $\{\mathbf{x}, \mathbf{y}\}$ as i ;
12   Label $\{\widehat{\mathbf{y}}\}$ as 0 ;
13   **end**
14   Compute loss for d_ϕ by Eq. (4) or (5) .
15   **end**

3.5.2   Style Transformer Learning

The training of Style Transformer is developed according to the different cases of $f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ where $\mathbf{s} = \widehat{\mathbf{s}}$ or $\mathbf{s} \neq \widehat{\mathbf{s}}$.

Self Reconstruction For the case $\mathbf{s} = \widehat{\mathbf{s}}$, or equivalently, the case $f_\theta(\mathbf{x}, \mathbf{s})$. As we discussed before, the input sentence $\mathbf{x}$ and the input style $\mathbf{s}$ comes from the same dataset , we can simply train our Style Transformer to reconstruct the input sentence by minimizing negative log-likelihood:

$$\mathcal{L}_{self}(\theta) = -p_\theta(\mathbf{y} = \mathbf{x}|\mathbf{x}, \mathbf{s}). \quad (6)$$

For the case $\mathbf{s} \neq \widehat{\mathbf{s}}$, we can't obtain direct supervision from our training set. So, we introduce two different training loss to create supervision indirectly.

Cycle Reconstruction To encourage generated sentence preserving the information in the input sentence $\mathbf{x}$, we feed the generated sentence $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ to the Style Transformer with the style of $\mathbf{x}$ and training our network to reconstruct original input sentence by minimizing negative log-likelihood:

$$\mathcal{L}_{cycle}(\theta) = -p_\theta(\mathbf{y} = \mathbf{x}|f_\theta(\mathbf{x}, \widehat{\mathbf{s}}), \mathbf{s}). \quad (7)$$

Style Controlling If we only train our Style Transformer to reconstruct the input sentence $\mathbf{x}$ from transferred sentence $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$, the network can only learn to copy the input to the output. To handle this degeneration problem, we further add a style controlling loss for the generated sentence. Namely, the network generated sentence

$\widehat{\mathbf{y}}$ is feed into discriminator to maximize the probability of style $\widehat{\mathbf{s}}$.

For the conditional discriminator, the Style Transformer aims to minimize the negative log-likelihood of class 1 when feed to the discriminator with the style label $\widehat{\mathbf{s}}$:

$$\mathcal{L}_{style}(\theta) = -p_\phi(\mathbf{c} = 1|f_\theta(\mathbf{x}, \widehat{\mathbf{s}}), \widehat{\mathbf{s}}). \quad (8)$$

And in the case of the multi-class discriminator, the Style Transformer is trained to minimize the the negative log-likelihood of the corresponding class of style $\widehat{\mathbf{s}}$:

$$\mathcal{L}_{style}(\theta) = -p_\phi(\mathbf{c} = \widehat{\mathbf{s}}|f_\theta(\mathbf{x}, \widehat{\mathbf{s}})). \quad (9)$$

Combining the loss function we discussed above, the training procedure of the Style Transformer is summarized in Algorithm 2.

Algorithm 2: Style Transformer Learning

Input: Style Transformer f_θ, discriminator d_ϕ, and a dataset $\mathcal{D}_i$ with style $\mathbf{s}$

1   Sample a minibatch of m sentences $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$ from $\mathcal{D}_i$. ;
2   **foreach** $\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$ **do**
3   Randomly sample a style $\widehat{\mathbf{s}}(\mathbf{s} \neq \widehat{\mathbf{s}})$;
4   Use f_θ to generate two new sentence
5     $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$
6     $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$;
7   Compute $\mathcal{L}_{self}(\theta)$ for $\mathbf{y}$ by Eq. (6) ;
8   Compute $\mathcal{L}_{cycle}(\theta)$ for $\widehat{\mathbf{y}}$ by Eq. (7) ;
9   Compute $\mathcal{L}_{style}(\theta)$ for $\widehat{\mathbf{y}}$ by Eq. (8) or (9) ;
10   **end**

3.5.3   Summarization and Discussion

Finally, we can construct our final training algorithm based on discriminator learning and Style Transformer learning steps. Similar to the training process of GANs (Goodfellow et al., 2014), in each training iteration, we first perform n_d steps discriminator learning to get a better discriminator, and then train our Style Transformer n_f steps to improve its performance. The training process is summarized in Algorithm 3.

Before finishing this section, we finally discuss a problem which we will be faced with in the training process. Because of the discrete nature of the natural language, for the generated sentence $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$, we can't directly propagate gradients from the discriminator through the discrete samples. To handle this problem, one can use REINFORCE (Williams, 1992) or the Gumbel-Softmax trick (Kusner and Hernández-Lobato, 2016) to estimates gradients from the discriminator. However, these two approaches are faced with high

Algorithm 3: Training Algorithm

Input: A bunch of datasets $\{\mathcal{D}_i\}_{i=1}^K$, and each represent a different style $\mathbf{s}^{(i)}$

1 Initialize the Style Transformer network f_θ, and the discriminator network d_ϕ with random weights θ, ϕ ;
2 **repeat**
3   **for** n_d *step* **do**
4   **foreach** *dataset* $\mathcal{D}_i$ **do**
5   Accumulate loss by Algorithm 1
6   **end**
7   Perform gradient decent to update d_ϕ.
8   **end**
9   **for** n_f *step* **do**
10   **foreach** *dataset* $\mathcal{D}_i$ **do**
11   Accumulate loss by Algorithm 2
12   **end**
13   Perform gradient decent to update f_θ.
14   **end**
15 **until** *network* $f_\theta(\mathbf{x}, \mathbf{s})$ *converges*;

Dataset	Yelp		IMDb	
	Positive	Negative	Positive	Negative
Train	266,041	177,218	178,869	187,597
Dev	2,000	2,000	2,000	2,000
Test	500	500	1,000	1,000
Avg. Len.	8.9		18.5	

Table 1: Datasets statistic.

variance problem, which will make the model hard to converge. In our experiment, we also observed that the Gumbel-Softmax trick would slow down the model converging, and didn't bring much performance improvement to the model. For the reasons above, empirically, we view the softmax distribution generated by f_θ as a "soft" generated sentence and feed this distribution to the downstream network to keep the continuity of the whole training process. When this approximation is used, we also switch our decoder network from greedy decoding to continuous decoding. Which is to say, at every time step, instead of feed the token that has maximum probability in previous prediction step to the network, we feed the whole softmax distribution (Eq. (2)) to the network. And the decoder uses this distribution to compute a weighted average embedding from embedding matrix for the input.

4 Experiment

4.1 Datasets

We evaluated and compared our approach with several state-of-the-art systems on two review datasets, Yelp Review Dataset (Yelp) and IMDb Movie Review Dataset (IMDb). The statistics of the two datasets are shown in Table 1.

Yelp Review Dataset (Yelp) The Yelp dataset is provided by the Yelp Dataset Challenge, consisting of restaurants and business reviews with sentiment labels (negative or positive). Following previous work, we use the possessed dataset provided by Li et al. (2018). Additionally, it also provides human reference sentences for the test set.

IMDb Movie Review Dataset (IMDb) The IMDb dataset[2] consists of movie reviews written by online users. To get a high quality dataset, we use the highly polar movie reviews provided by Maas et al. (2011). Based on this dataset, we construct a highly polar sentence-level style transfer dataset by the following steps: 1) fine tune a BERT (Devlin et al., 2018) classifier on original training set, which achieves 95% accuracy on test set; 2) split each review in the original dataset into several sentences; 3) filter out sentences with confidence threshold below 0.9 by our fine-tuned BERT classifier; 4) remove sentences with uncommon words. Finally, this dataset contains 366K, 4k, 2k sentences for training, validation, and testing, respectively.

4.2 Evaluation

A goal transferred sentence should be a fluent, content-complete one with target style. To evaluate the performance of the different model, following previous works, we compared three different dimensions of generated samples: 1) Style control, 2) Content preservation and 3) Fluency.

4.2.1 Automatic Evaluation

Style Control We measure style control automatically by evaluating the target sentiment accuracy of transferred sentences. For an accurate evaluation of style control, we trained two sentiment classifiers on the training set of Yelp and IMDb using fastText (Joulin et al., 2017).

Content Preservation To measure content preservation, we calculate the BLEU score (Papineni et al., 2002) between the transferred sentence and its source input using NLTK. A higher BLEU score indicates the transferred sentence can achieve better content preservation by retaining more words from the source sentence. If a human reference is available, we will calculate the BLEU

---

[2]https://github.com/fastnlp/nlp-dataset

Model	Yelp				IMDb		
	ACC	*ref*-BLEU	*self*-BLEU	PPL	ACC	*self*-BLEU	PPL
Input Copy	3.8	23	100	41	5.1	100	58
RetrieveOnly (Li et al., 2018)	92.6	0.4	0.7	**7**	N/A	N/A	N/A
TemplateBased (Li et al., 2018)	84.3	13.7	44.1	117	N/A	N/A	N/A
DeleteOnly (Li et al., 2018)	85.7	9.7	28.6	72	N/A	N/A	N/A
DeleteAndRetrieve (Li et al., 2018)	87.7	10.4	29.1	60	58.8	55.4	**57**
ControlledGen (Hu et al., 2017)	88.8	14.3	45.7	219	94.1	62.1	143
CrossAlignment (Shen et al., 2017)	76.3	4.3	13.2	53	N/A	N/A	N/A
MultiDecoder (Fu et al., 2018)	49.8	9.2	37.9	90	N/A	N/A	N/A
CycleRL(Xu et al., 2018)	88.0	2.8	7.2	107	**97.8**	4.9	177
Ours (Conditional)	**93.7**	17.1	45.3	90	86.6	66.2	107
Ours (Multi-Class)	87.7	**20.3**	**54.9**	73	80.3	**70.5**	105

Table 2: Automatic evaluation results on Yelp and IMDb datset

score between the transferred sentence and corresponding reference as well. Two BLEU score metrics are referred to as *self*-BLEU and *ref*-BLEU respectively.

Fluency Fluency is measured by the perplexity of the transferred sentence, and we trained a 5-gram language model on the training set of two datasets using KenLM (Heafield, 2011).

4.2.2 Human Evaluation

Due to the lack of parallel data in style transfer area, automatic metrics are insufficient to evaluate the quality of the transferred sentence. Therefore we also conduct human evaluation experiments on two datasets.

We randomly select 100 source sentences (50 for each sentiment) from each test set for human evaluation. For each review, one source input and three anonymous transferred samples are shown to a reviewer. And the reviewer is asked to choose the best sentence for style control, content preservation, and fluency respectively.

- Which sentence has the most opposite sentiment toward the source sentence?

- Which sentence retains most content from the source sentence?

- Which sentence is the most fluent one?

To avoid interference from similar or same generated sentences, "no preference." is also an option answer to these questions.

4.3 Training Details

In all of the experiment, for the encoder, decoder, and discriminator, we all use 4-layer Transformer with four attention heads in each layer. The hidden size, embedding size, and positional encoding size in Transformer are all 256 dimensions. Another embedding matrix with 256 hidden units is used to represent different style, which is feed into encoder as an extra token of the input sentence. And the positional encoding isn't used for the style token. For the discriminator, similar to Radford et al. (2018) and Devlin et al. (2018), we further add a `<cls>` token to the input, and the output vector of the corresponding position is feed into a softmax classifier which represents the output of discriminator.

In the experiment, we also found that preforming random word dropout for the input sentence when computing the self reconstruction loss (Eq. (6)) can help model more easily to converge to a reasonable performance. On the other hand, by adding a temperature parameter to the softmax layer (Eq. (2)) and using a sophisticated temperature decay schedule can also help the model to get a better result in some case.

4.4 Experimental Results

Results using **automatic metrics** are presented in Table 2. Comparing to previous approaches, our models achieve competitive performance overall and get better content preservation at all of two datasets. Our conditional model can achieve a better style controlling compared to the multi-class model. Both our models are able to generate sentences with relatively low perplexity. For those previous models performing the best on a single metric, an obvious drawback can always be found on another metric.

For the **human evaluation**, we choose two

Model	Yelp			IMDb		
	Style	Content	Fluency	Style	Content	Fluency
CtrlGen	16.8	23.6	17.7	**30.0**	19.5	22.0
DAR	13.6	15.5	21.4	21.0	27.0	25.0
Ours	**48.6**	**36.8**	**41.4**	29.5	**35.0**	**31.5**
No Preference	20.9	24.1	19.5	19.5	18.5	21.5

Table 3: Human evaluation results on two datasets. Each cell indicates the proportion of being preferred.

of the most well-performed models according to the automatic evaluation results as competitors: DeleteAndRetrieve (DAR) (Li et al., 2018) and Controlled Generation (CtrlGen) (Hu et al., 2017). And the generated outputs from multi-class discriminator model is used as our final model. We have performed over 400 human evaluation reviews. Results are presented in Table 3. The human evaluation results are mainly conformed with our automatic evaluation results. And it also shows that our models are better in content preservation, compared to two competitor model.

Finally, to better understand the characteristic of different models, we sampled several output sentences from the Yelp dataset, which are shown in Table 4.

4.5 Ablation Study

To study the impact of different components on overall performance, we further did an ablation study for our model on Yelp dataset, and results are reported in Table 5.

For better understanding the role of different loss functions, we disable each loss function by turns and retrain our model with the same setting for the rest of hyperparameters. After we disable self-reconstruction loss (Eq. (6)), our model failed to learn a meaningful output and only learned to generate a single word for any combination of input sentence and style. However, when we don't use cycle reconstruction loss (Eq. (7)), it's also possible to train the model successfully, and both of two models converge to reasonable performance. And comparing to the full model, there is a small improvement in style accuracy, but a significant drop in BLEU score. As our expected, the cycle reconstruction loss is able to encourage the model to preserve the information from the input sentence. At last, when the discriminator loss (Eq. (8) and (9)) is not used, the model quickly degenerates to a model which is only copying the

input sentence to output without any style modification. This behaviour also conforms with our intuition. If the model is only asked to minimize the self-reconstruction loss and cycle reconstruction loss, directly copying input is one of the optimum solutions which is the easiest to achieve. In summary, each of these loss plays an important role in the Style Transformer training stage: 1) the self-reconstruction loss guides the model to generate readable natural language sentence. 2) the cycle reconstruction loss encourages the model to preserve the information in the source sentence. 3) the discriminator provides style supervision to help the model control the style of generated sentences.

Another group of study is focused on the different type of samples used in the discriminator training step. In Algorithm 1, we used a mixture of real sentence $\mathbf{x}$ and generated sentence $\mathbf{y}$ as the positive training samples for the discriminator. By contrast, in the ablation study, we trained our model with only one of them. As the result shows, the generated sentence is the key component in discriminator training. When we remove the real sentence from the training data of discriminator, our model can also achieve a competitive result as the full model with only a small performance drop. However, if we only use the real sentence the model will lose a significant part of the ability to control the style of the generated sentence, and thus yields a bad performance in style accuracy. However, the model can still perform a style control far better than the input copy model discussed in the previous part. For the reasons above, we used a mixture of real sample and generated sample in our final version.

5 Conclusions and Future Work

In this paper, we proposed the Style Transformer with a novel training algorithm for text style transfer task. Experimental results on two text style transfer datasets have shown that our model achieved a competitive or better performance compared to previous state-of-the-art approaches. Especially, because our proposed approach doesn't assume a disentangled latent representation for manipulating the sentence style, our model can get better content preservation on both of two datasets.

In the future, we are planning to adapt our Style Transformer to the multiple-attribute setting like Lample et al. (2019). On the other hand, the back-

	negative to positive
Input	the food 's ok , the service is among the worst i have encountered .
DAR	the food 's ok , the service is among great and service among .
CtrlGen	the food 's ok , the service is among the randy i have encountered .
Ours	the food 's delicious , the service is among the best i have encountered .
Human	the food is good , and the service is one of the best i 've ever encountered .
Input	this is the worst walmart neighborhood market out of any of them .
DAR	walmart market is one of my favorite places in any neighborhood out of them .
CtrlGen	fantastic is the randy go neighborhood market out of any of them .
Ours	this is the best walmart neighborhood market out of any of them .
Human	this is the best walmart out of all of them .
Input	always rude in their tone and always have shitty customer service !
DAR	i always enjoy going in always their kristen and always have shitty customer service !
CtrlGen	always good in their tone and always have shitty customer service !
Ours	always nice in their tone and always have provides customer service !
Human	such nice customer service , they listen to anyones concerns and assist them with it .
	positive to negative
Input	everything is fresh and so delicious !
DAR	small impression was ok , but lacking i have piss stuffing night .
CtrlGen	everything is disgrace and so bland !
Ours	everything is overcooked and so cold !
Human	everything was so stale .
Input	these two women are professionals .
DAR	these two scam women are professionals .
CtrlGen	shame two women are unimpressive .
Ours	these two women are amateur .
Human	these two women are not professionals .
Input	fantastic place to see a show as every seat is a great seat !
DAR	there is no reason to see a show as every seat seat !
CtrlGen	unsafe place to embarrassing lazy run as every seat is lazy disappointment seat !
Ours	disgusting place to see a show as every seat is a terrible seat !
Human	terrible place to see a show as every seat is a horrible seat !

Table 4: Case study from Yelp dataset. The red words indicate good transfer; the blue words indicate bad transfer; the brown words indicate grammar error.

	Conditional			Multi-class		
Model	ACC	BLEU	PPL	ACC	BLEU	PPL
Style Transformer	93.6	17.1	78	87.6	20.3	50
- self reconstruction	50.0	0	N/A	20.7	0	N/A
- cycle reconstruction	94.2	8.6	56	93.2	8.7	40
- discriminator	3.3	22.9	11	3.3	22.9	11
- real sample	89.7	17.4	75	83.8	19.4	55
- generated sample	46.3	21.6	34	35.6	22.0	33

Table 5: Model ablation study results on Yelp dataset

translation technique developed in Lample et al. (2019) can also be adapted to the training process of Style Transformer. How to combine the back-translation with our training algorithm is also a good research direction that is worth to explore.

Acknowledgment

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Keith Carlson, Allen Riddell, and Daniel N. Rockmore. 2017. Zero-shot style transfer in text using recurrent neural networks. *CoRR*, abs/1711.04731.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 11–21.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. 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*, pages 1587–1596. JMLR. org.

Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. 2019. Unsupervised Text Style Transfer via Iterative Matching and Translation. *arXiv e-prints*, page arXiv:1901.11333.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for text style transfer. *arXiv preprint arXiv:1808.04339*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Matt J. Kusner and José Miguel Hernández-Lobato. 2016. GANS for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Igor Melnyk, Cícero Nogueira dos Santos, Kahini Wadhawan, Inkit Padhi, and Abhishek Kumar. 2017. Improved neural text attribute transfer with non-parallel data. *CoRR*, abs/1711.09395.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

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 on association for computational linguistics*, pages 311–318. 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 866–876.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Cícero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 189–194.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pages 1096–1103.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Jingjing Xu, SUN Xu, 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)*, volume 1, pages 979–988.

Ye Zhang, Nan Ding, and Radu Soricut. 2018a. SHAPED: shared-private encoder-decoder for text style adaptation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1528–1538.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018b. Style transfer as unsupervised machine translation. *CoRR*, abs/1808.07894.