Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation

Rakshan tej Baggam

January 2024

1 Introduction

This is based on the understanding from the paper **Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation.**
The research addresses common challenges in unpaired text style transfer, particularly the difficulty in separating content and style in the latent space. Current neural models face two key issues:

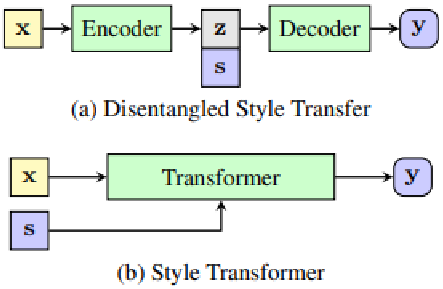
- Struggling to entirely **remove style information** from sentence semantics.
- **Ineffectively handling** long-term dependencies with the **recurrent neural network (RNN) based encoder and decoder**, leading to suboptimal preservation of non-stylistic semantic content.

The proposed Style Transformer takes a different approach by **not assuming any specific latent representation** for the source sentence. Instead, it leverages the **attention mechanism in the Transformer**, aiming to **improve both style transfer and content preservation**.

Text style transfer involves altering stylistic properties (e.g., sentiment) while maintaining style-independent content. The task is challenging due to the vague definition of text style, making it difficult to construct paired sentences with the same content but different styles. **Neural networks, particularly the "encoder-decoder" framework, have dominated recent**

approaches. However, **these methods often struggle with disentangling content and style in the latent space**. Adversarial loss is employed to discourage encoding style information, but concerns arise regarding the quality of disentanglement, the necessity of disentanglement, limited capacity of vector representation, and the weak ability of recurrent neural networks (RNNs) to capture long-range dependencies.

The proposed **Style Transformer addresses these concerns by leveraging the Transformer architecture**, known for its success in natural language processing tasks. Unlike RNNs, Transformer utilizes stacked self-attention and fully connected layers for both encoder and decoder. The **Transformer decoder fetches information from the encoder via attention mechanisms, allowing for better preservation of meaning during style transfer**. This departure from fixed-sized latent vectors enhances the model's ability to apply attention mechanisms directly, thus addressing the limitations associated with existing approaches. The **use of Transformer proves advantageous in capturing rich semantic information, especially in long texts, leading to improved style transfer while preserving content meaning**.
The key distinctions between our model and previous models are illustrated in Figure 1.



(a) Disentangled Style Transfer

(b) Style Transformer

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

Our contributions can be summarized as follows:

Novel Training Algorithm: We introduce a groundbreaking training algorithm that does not assume disentangled latent representations for input sentences. This innovation allows the model to leverage attention mechanisms, leading to further performance improvements.

Pioneering Use of Transformer Architecture: To the best of our knowledge, our work represents the first application of the Transformer architecture to the style transfer task. This departure from traditional approaches brings a fresh perspective and potential advantages.

Experimental Superiority: Through comprehensive experiments, our proposed approach consistently outperforms existing methods on two style transfer datasets. Notably, in terms of content preservation, Style Transformer demonstrates the highest performance with a substantial improvement over other models.

2 Style Transformer Problem

In the research paper, the **style transfer problem is defined as follows:**

- We have a bunch of datasets, where each dataset contains a number of language sentences. All these datasets share a single common characteristic i.e., all these datasets are the positive reviews for a specific product. This shared characteristic is called the style of these sentences. The style can also be defined by the distribution of a dataset.

- Suppose we have K different datasets, then we can have k different styles. The goal of the style transfer is that: given an arbitrary natural language sentence and a desired style, we need to be able to rewrite the sentence in the new style while preserving the information in the original sentence as much as possible.

2.1 Tackling the Style Transformer Problem

- In addressing the style transfer problem defined earlier, our objective is to learn a mapping function denoted as $f_\theta(x, s)$ where x represents a natural language sentence, and s is a style control variable. The function's output, denoted as $\hat{x}$, is the transferred sentence corresponding to the input sentence x.

- A significant challenge in text style transfer arises from the absence of access to parallel corpora. Consequently, direct supervision for training our transfer model is unavailable. In Section 3.4, we navigate this challenge by employing two discriminator-based approaches to create supervision using non-parallel corpora.
- Finally, in Section 3.5, the Style Transformer network and discriminator network are integrated through a comprehensive learning algorithm. This holistic approach is designed to effectively train our style transfer system despite the absence of parallel corpora.

2.2 Style Transformer Network

Generally, Transformer follows the standard encoder-decoder architecture. Explicitly, for a input sentence x = (x1, x2, ..., xn), the Transformer encoder Enc(x; Θ_E) maps inputs to a sequence of continuous representations z = (z1, z2, ..., zn). And the Transformer decoder Dec(z; Θ_D) estimates the conditional probability for the output sentence y = (y1, y2, ..., yn) by auto-regressively factorized its as:

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

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

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

where $\mathbf{o}_t$ is logic 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(x, s; Θ_E). Therefore the network can compute the probability of the output condition both on the input sentence x and the style control variable s. Formally, this can be expressed as:

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

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

2.3 Discriminator network

Suppose we use x and s to denote the sentence and its style from the dataset D. Because of the absence of the parallel corpora, we can't directly obtain the supervision for the case $f_\Theta(x, \hat{s})$ where s $\neq \hat{s}$. The discriminator network is used to learn this supervision from the nonparallel copora.

- For the content preservation, we train the network to reconstruct original input sentence x when we feed transferred sentence $\hat{b} = f_\Theta(x, \hat{s})$ to the Style Transformer network with the original style label s.
- 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.

There are 2 kinds of discriminators used in the research paper.

Conditional Discriminator :

- Explicitly, a sentence x and a proposal style s are feed into discriminator $d_\phi(x)$ and the discriminator is asked to answer whether the input sentence has the corresponding style.
- In discriminator training stage, the real sentence from datasets x, and the reconstructed sentence y = $f_\Theta(x, s)$ are labeled as positive, and the transferred sentences $\hat{b} = f_\Theta(x, \hat{s})$ where s $\neq \hat{s}$, are labeled as negative.
- In Style Transformer network training stage, the network f_Θ is trained to maximize the probability of positive when feed $f_\Theta(x, \hat{s})$ and $\hat{s}$ to the discriminator.

Multi-class Discriminator :

- In this case, only one sentence is feed into discriminator $d_\phi(x)$, 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 $f_{\Theta}(x, \hat{s})$, which is also often referred as fake sample
- In discriminator training stage, we label the real sentences x and reconstructed sentences y = $f_{\Theta}(x, s)$ to the label of the corresponding style. And for the transferred sentence $\hat{b} = f_{\Theta}(x, \hat{s})$ where s $\neq$ $\hat{s}$, is labeled as the class 0
- In Style Transformer network learning stage, we train the network $f_{\Theta}(x, \hat{s})$ to maximize the probability of the class which is stand for style $\hat{s}$.

2.4 Learning Algorithm

The training algorithm of our model can be divided into 2 parts: **the discriminator learning** and **Style Transformer Network Learning**.

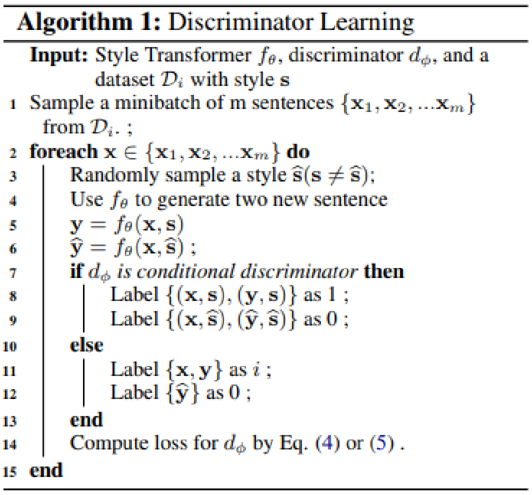
2.4.1 Discriminator Learning

In the discriminator Learning, we train our discriminator to **distinguish between the original sentence x and the reconstructed sentence** y = $f_{\Theta}(x, s)$ from the transferred sentence $\hat{y} = f_{\Theta}(x, \hat{s})$
The **loss function** for the discriminator is the **cross-entropy loss** of the classification problem.
For the **conditional discriminator**:

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

For the **multi-class discriminator**:

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



```
Algorithm 1: Discriminator Learning
Input: Style Transformer $f_\theta$, discriminator $d_\phi$, and a
       dataset $\mathcal{D}_i$ with style $\mathbf{s}$
Sample a minibatch of m sentences $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$
  from $\mathcal{D}_i$. ;
foreach $\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$ do
    Randomly sample a style $\widehat{\mathbf{s}}(\mathbf{s} \neq \widehat{\mathbf{s}})$;
    Use $f_\theta$ to generate two new sentence
    $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$
    $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ ;
    if $d_\phi$ is conditional discriminator then
        Label $\{(\mathbf{x}, \mathbf{s}), (\mathbf{y}, \mathbf{s})\}$ as 1 ;
        Label $\{(\mathbf{x}, \widehat{\mathbf{s}}), (\widehat{\mathbf{y}}, \widehat{\mathbf{s}})\}$ as 0 ;
    else
        Label $\{\mathbf{x}, \mathbf{y}\}$ as $i$ ;
        Label $\{\widehat{\mathbf{y}}\}$ as 0 ;
    end
    Compute loss for $d_\phi$ by Eq. (4) or (5) .
end
```

Figure 2: Algorithm for Discriminator Learning

2.4.2 Style Transformer Learning

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

- **Self Reconstruction:** For the case $s = \hat{s}$ or the case $f_\theta(x, s)$, The input sentence **x** and the input style **s** comes from the same dataset. Hence, we can train our transformer by minimizing the negative-log likelihood.

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

  For the case s $\neq \hat{s}$, we **cannot** obtain **direct superivision** from our training dataset. We use **2 different training loss** to create **supervision indirectly**.

- **Cycle Reconstruction:** For $s \neq \hat{s}$, Generating the sentence while preserving the information of the input sentence x, we feed the **generated sentence $\hat{y} = f_\theta(x, \hat{s})$ to the style transformer with the style of x** and train our network **to reconstruct original input sentence** by minimizing the negative-log likelihood

  $$\mathcal{L}_{self}(\theta) = -p_\theta(y = x|f_\theta(x, \hat{s}), s)$$

- **Cycle Reconstruction:** If we train our transformer to generate the reconstructed sentence same as the original sentence, we will only learn to copy the original sentence. Therefore, we add an additional loss called style controlling loss $\mathcal{L}_{style}$ for the generated sentence.

- **Style Controlling:** The generated sentence $\hat{y}$ is feed into the discriminator to maximize the probability of style $\hat{s}$.

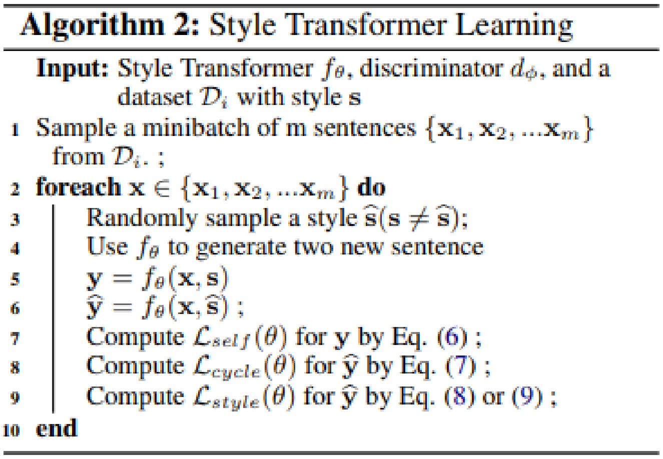
  There are 2 cases for the style transformer based on the type of discriminator.

  For the case of **conditional discriminator**, the Style Transformer aims to minimize the negative likelihood of class 1 when feed to the discriminator with the style label $\hat{s}$.

  $$\mathcal{L}_{style}(\theta) = -p_{\phi}(c = 1 | f_{\theta}(x, \hat{s}), \hat{s})$$

  In the case of multi-class discriminator, the style Transformer is trained to minimize the negative log-likelihood of the corresponding class of style $\hat{s}$

  $$\mathcal{L}_{style}(\theta) = -p_{\phi}(c = \hat{s} | f_{\theta}(x, \hat{s}))$$



```
Algorithm 2: Style Transformer Learning
Input: Style Transformer $f_\theta$, discriminator $d_\phi$, and a
       dataset $\mathcal{D}_i$ with style s
Sample a minibatch of m sentences $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$
  from $\mathcal{D}_i$. ;
foreach $\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$ do
    Randomly sample a style $\widehat{\mathbf{s}}(\mathbf{s} \neq \widehat{\mathbf{s}})$;
    Use $f_\theta$ to generate two new sentence
    $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$
    $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ ;
    Compute $\mathcal{L}_{self}(\theta)$ for $\mathbf{y}$ by Eq. (6) ;
    Compute $\mathcal{L}_{cycle}(\theta)$ for $\widehat{\mathbf{y}}$ by Eq. (7) ;
    Compute $\mathcal{L}_{style}(\theta)$ for $\widehat{\mathbf{y}}$ by Eq. (8) or (9) ;
end
```

Figure 3: Algorithm for Style Transformer Learning

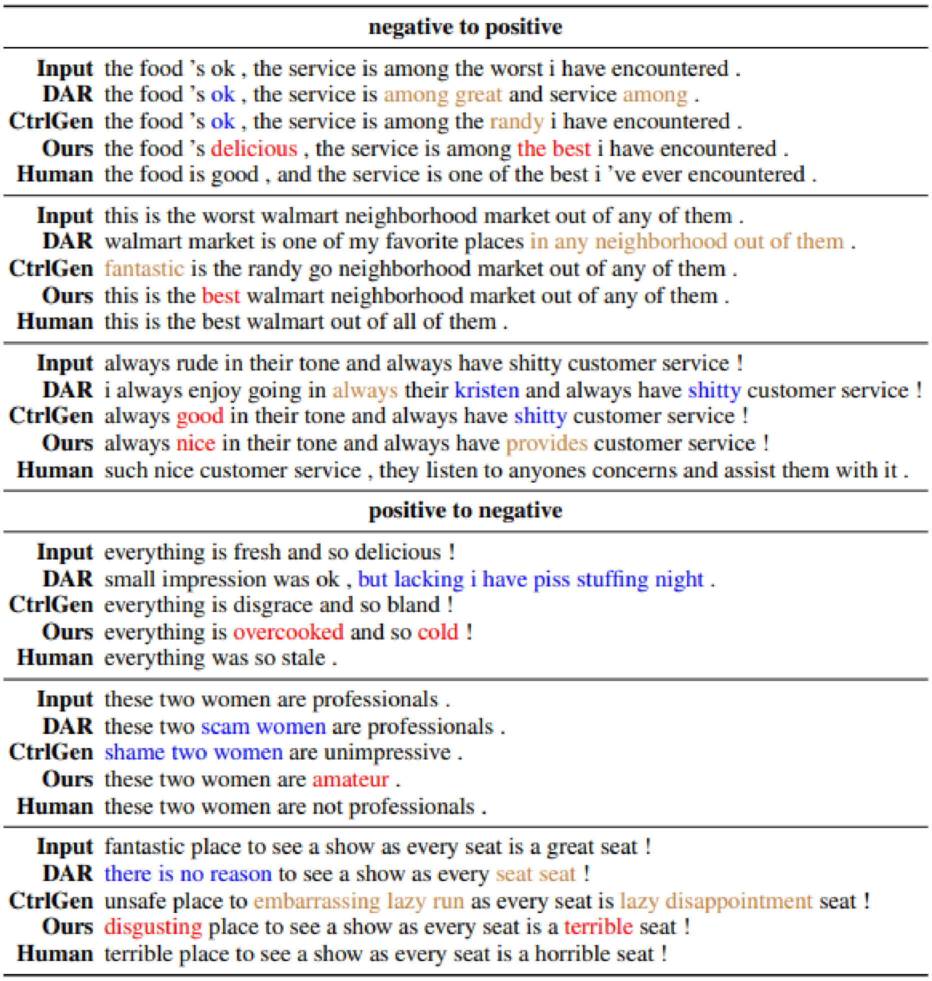
3 Experimental observations

- The models were implemented on 2 datasets, Yelp Review Dataset and IMDB Movie Review Dataset.
- Evaluation is done based on the three dimensions of generated samples: **1) Style Control, 2) Content preservation, 3) Fluency**
- To check the **Content preservation, BLEU score** was calculated using the NLTK Library, which is a measure of the content preservation.
- If a **human reference** is present, BLEU scores between **transferred sentence and corresponding reference** is also calculated. One is called **self-BLEU** and the other one is called **ref-BLEU**.
- Fluency is measured by the perplexity of the transferred sentence, and we trained a 5-gram model on the training dataset using KenLM.
- Due to the lack of parallel data in style transfer, automatic metrics are insufficient to evaluate the quality of the transferred sentence.
- For this, we used **human evaluation**. 100 sentences were taken(50 for each sentiment) for each test set for human evaluation.
- Following questions were asked
  1) Which sentence has the most opposite sentiment toward the source sentence?
  2) Which sentence retains most content from the source sentence?
  3) Which sentence is the most fluent one?

  To avoid interference from similar or same generated sentences, ”no preference.” is also an option

4 Conclusions

In this paper, we proposed the Style Transformer with a novel training algorithm for text style transfer task. Our model has demonstrated competitive or superior performance compared to previous state-of-the-art methods in text style transfer datasets, according to experimental results. Notably, our proposed approach stands out as it doesn't rely on assuming a disentangled latent representation to manipulate sentence styles. This unique feature allows our model to excel in preserving content on both datasets.



	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 !

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