applied sciences

MDPI

Article

Mask Transformer: Unpaired Text Style Transfer Based on Masked Language

Chunhua Wu *, Xiaolong Chen * and Xingbiao Li

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; liqingbiao@bupt.edu.cn

* Correspondence: wuchunhua@bupt.edu.cn (C.W.); chenxiaolong@bupt.edu.cn (X.C.)

Received: 2 August 2020; Accepted: 1 September 2020; Published: 6 September 2020

Abstract: Currently, most text style transfer methods encode the text into a style-independent latent representation and decode it into new sentences with the target style. Due to the limitation of the latent representation, previous works can hardly get satisfactory target style sentence especially in terms of semantic remaining of the original sentence. We propose a "Mask and Generation" structure, which can obtain an explicit representation of the content of original sentence and generate the target sentence with a transformer. This explicit representation is a masked text that masks the words with the strong style attribute in the sentence. Therefore, it can preserve most of the semantic meaning of the original sentence. In addition, as it is the input of the generator, it also simplified this process compared to the current work who generate the target sentence from scratch. As the explicit representation is readable and the model has better interpretability, we can clearly know which words changed and why the words changed. We evaluate our model on two review datasets with quantitative, qualitative, and human evaluations. The experimental results show that our model generally outperform other methods in terms of transfer accuracy and content preservation.

Keywords: natural language process; mask language; transformer; style transfer

1. Introduction

The text style transfer task aims to change the stylistic attributes of sentences (e.g., emotions), while retaining the style-independent content of the context as much as possible. This method can be widely used to transfer review sentiment, rewrite news, change dialogue emotion, and so on. For example, a positive comment "The restaurant's dishes are very delicious, I highly recommend it!" can be transformed into a negative comment "The restaurant's dishes are a bit disappointing, and I will never come again!" Currently, there is no specific and common definition of text style, so we usually set the definition depending on the task. In addition, due to the high construction costs of parallel corpus, the current research is mainly conducted using nonparallel corpora [1].

Currently, a deep neural network model based on the seq2seq framework is the main text style transfer method. The first class of methods focuses on disentangling the content and style in the latent space and encodes sentences into latent representations in the semantic space and style space, respectively. After disentangling the content, the style-independent representation will be decoded into a sentence with the target style via a generative model [2,3]. Another class of methods attempts to learn the mixed content and style distribution in the latent space and directly map it in the latent space to complete the style transfer [4,5].

Both classes of methods have certain problems. In the first class of methods, the quality of the separated hidden vector is difficult to evaluate, and the latent representation has difficulties to keep rich semantic information of the original sentence due to its limited capabilities, especially for long text. Regarding the direct transfer method, because of the limitation of the latent space, it also cannot

keep the rich semantic information of the original sentence. In addition, because of its poor interpretability, it has difficulties dealing with some adaption problems or incorporating external information. In addition, most previous work used recurrent neural networks (RNNs) as encoders and decoders [6], which are limited by their weak abilities to capture long-term dependencies. In summary, previous works can hardly get satisfactory target style sentence especially in terms of semantic remaining of the original sentence.

In this paper, we propose a "Mask and Generation" structure, which can get an explicit representation of the content of original sentence and generate the target sentence with a transformer. The overview of our approach is shown in Figure 1. Our method consists of two parts. In the mask part, we locate the words with higher style attributes in the sentence and replace them with the mask symbols. We follow [7] to train a self-attention [8] style classifier [9] in which the learned attention weights can be used to analyze the style attribute of the words in the sentences. The larger the weight is, the stronger the style attribute. Using this attention mechanism and the style dictionary method, we can find the words with strong style attributes in the sentence. Then, we mask those words to turn the sentence into a neutral one. In the generation part, we take the sentence with the mask symbols as input and use the powerful self-attention mechanism of transformer to generate new sentence according to the target style. Different from [10], who filled [11] the mask position in the sentence, we generate a new sentence with the target style by training a transformer-based [12] generation model for the input masked sentence and the specified style. This approach finds an explicit neutral representation of the sentence via the mask, and then, it flexibly generates the sentence with the target style through the transformer. This not only ensures that the model can capture rich semantic information for the conversion but also maintains high style transfer accuracy.

Our contributions are summarized as follows:

1. We propose a "Mask and Generation" structure, which can get an explicit representation of the content of original sentence and generate the target sentence with a transformer. This explicit representation is a masked text that masks the words with the strong style attribute in the sentence. Therefore, it can preserve most of the semantic meaning of the original sentence and also simplify the generator process.
2. As the explicit representation is readable, the model has better interpretability, we can clearly know which words changed and why the words changed.
3. We use a self-attention mechanism and the style dictionary method to analyze the contributions of the style attributes for each component in the sentence.
4. We generate the target sentence by training a transformer-based generation model for the input masked sentence and the specified style.

The experimental results show that our method generally outperforms most models, especially in style accuracy and content preservation.

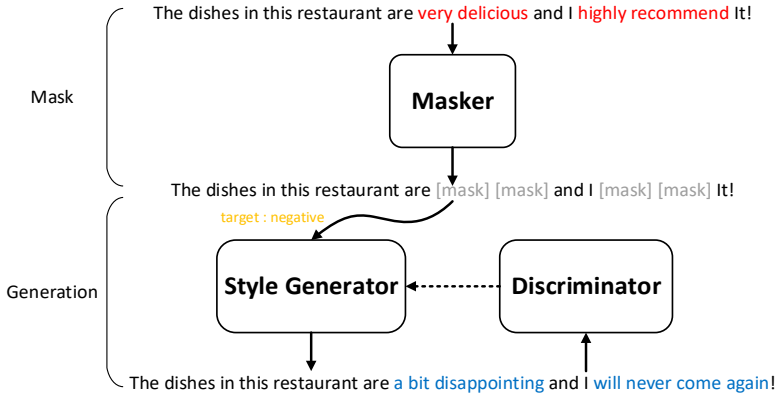

Figure 1. The overview of our approach.

2. Related Work

In the early stages of the research, text style transfer methods tended to find the style and content representations of sentences. Reference [13] propose a cross-aligned autoencoder with adversarial training [14] to learn a shared latent content distribution and a separated latent style distribution. Reference [15] focused on disentangling style and content representations. They designed a multitask and adversarial loss for a variational autoencoder to ensure the separation. In addition, there are some methods that implicitly find the neutral representation of a sentence. Reference [16] believe that text style transfer can be accomplished using a combination of delete, modify, and generate operations. This is done by deleting words with strong style attributes to find the style-independent part of the sentence and generating new sentences using modifying and generating operations. Reference [10] designed a two-step "Mask and Infill" approach by masking sentimental tokens and predicting words according to the target sentiment.

Although separating the style and content in a sentence will help us in the text style transfer task, it makes it hard to guarantee the quality of the separated style distribution and content distribution. In addition, the fixed-size latent representation limits the generation ability of the model, and it inevitably loses some content information from the original sentence. Therefore, some researchers have proposed some methods that do not need to separate the style and content. Reference [7] was inspired by the cycle style transfer method [17] in Computer Vision. They proposed a cycle reinforcement learning method for nonparallel sentiment transfer tasks. Reference [18] utilized the powerful attention mechanism in the transformer to implement the direct transfer of text style. Reference [19] propose a Context-Aware Style Transfer (CAST) model, which uses two separate encoders for each input sentence and its surrounding context. Reference [20] define a generative probabilistic model that treats a non-parallel corpus in two domains as a partially observed parallel corpus.

The direct transfer method also loses the semantic meaning of the original sentence for the limitation of the latent space. In addition, because of its poor interpretability, it has difficulties dealing with some adaption problems or incorporating external information. Besides, most previous work used recurrent neural networks (RNNs) as encoders and decoders [6], which are limited by their weak abilities to capture long-term dependencies. In summary, previous works can hardly get satisfactory target style sentence especially in terms of semantic remaining of the original sentence.

Based on previous work, our method explicitly obtains a sentence's neutral representation through masking the strong style attribute words in the sentence. This method greatly enhances the content remaining of the original sentence as well as the interpretability of style transfer. Different from the previous work that directly predicts the masked words, we use the powerful attention mechanism in the transformer to regenerate the neutral masked sentences according to the target style.

3. Approach

In this section, we will introduce our text style transfer method in detail. Section 3.1 is the basic definition of the problem, Section 3.2 is an overview of our model, and Sections 3.3–3.5 will show the details of the modules one by one.

3.1. Problem Formalization

In this paper, we define the text style transfer problem as follows. Consider a collection of datasets $\{D_i\}_{i=1}^{K}$, and each dataset D_i consists of many natural language sentences. For all sentences in a single dataset D_i, they have some common specific characteristics (for example, they are all positive reviews of a specific product), and we call such shared characteristics the styles of these sentences. In other words, the style is defined by the distribution of the dataset. Suppose we have K different datasets D_i. Then, we can define K different styles, and each style is represented by the symbol $s^{(i)}$. The goal of text style transfer is the following: given a sentence x of any style and the

target style $\hat{s} \in \{s^i\}_{i=1}^K$, rewrite x as a new sentence $\hat{x}$ with style $\hat{s}$ and retain the content as much as possible.

3.2. Model Overview

To solve the style transfer problem defined above, our goal is to learn a model with $(\hat{s}, x)$ as the input, where x is the sentence with the original style attribute, $\hat{s}$ is the target style, and the output of the model is sentence $\hat{x}$ with style $\hat{s}$.

Our method consists of three parts: a Masker module, a Style Generator module, and a discriminator module, as shown in Figure 2. In Section 3.3, the Masker module combines the advantages of the two methods by using the self-attention classification model and the auxiliary style dictionary to find the words with strong style attributes in the sentence. After that, it will perform the mask operation to obtain the masked sentence. In Section 3.4, we take the masked sentence and the specified target style as the input, and the Style Generator will regenerate a sentence with the target style. Unlike the method of filling the mask position, our method will regenerate a complete sentence based on the masked sentence, so that the generated sentence is more flexible in terms of structure and semantics. In addition, a major problem in text style transfer is that there are not enough parallel corpora. Therefore, we cannot directly train our style conversion model in a supervised manner. In Section 3.5, we will introduce a discriminator-based approach [21,22] to conducting supervised training using nonparallel corpora. Finally, we will combine these three parts to train our style transfer model through the learning algorithm in Section 3.6.

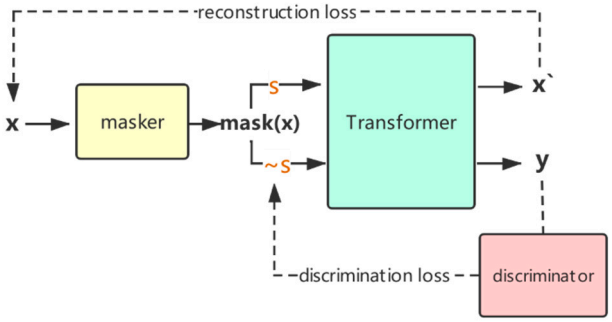

Figure 2. The architecture of the model.

3.3. Masker

We first introduce the method based on the self-attention classification model and then introduce an assisted style dictionary method. Finally, we propose a fusion method that combines the advantages of both methods.

3.3.1. Self-Attention Classifier-Based Method

For a sentence, each word in the sentence contributes differently to the sentence's style. If a word or phrase contributes more to the style of a sentence, it means that this component has a higher style attribute. In other words, if we can find the components with large style contributions in each sentence and mask them, we can obtain masked sentences that tend to be neutral. This is an approach to approximately obtain style-independent representations. For a sentence $x =< t_1, t_2, \ldots, t_N >$ with N words, we use bidirectional LSTM (long short-term memory) to encode the sentence and concatenate the forward hidden state and backward hidden state of each word to obtain the final hidden state:

$$H = (h_1, h_2, \ldots, h_N), \quad (1)$$

where N is the length of the given sentence. The self-attention mechanism calculates an attention weight vector a. In addition, the weighted hidden state vector c is obtained by multiplying the

hidden weight vector H by a. Finally, we convert c into a probability distribution y through the Softmax layer:

$$a = softmax(w \cdot \tanh(WH^T)), \quad (2)$$

$$y = softmax(W' \cdot c), \quad (3)$$

where w, W, and W' are the network parameters. W maps the hidden layer vector to a high-dimensional space to learn the impact of the input on the label. w is used to map the vector to a scalar, and the attention weight is obtained by Softmaxing the sequence. W' maps the output vector after the weighted summation to the category dimension and obtains the probability of each category through Softmax. After sufficient training, the classifier can achieve 97% accuracy. We use cls_ρ to represent the attention-based classification model, where ρ represents the model parameters.

Definitely, the attention value calculated during classification can be extracted and used to analyze the contribution of the style attributes for each component of the sentence. Considering that the length of each sentence may be different, using an average attention value or fixed value as a threshold [7] has certain limitations and cannot be adapted to each sentence. We propose a mask method based on the proportion of the sentence length. For example, the words of a sentence are sorted according to their attention values, and then the top 15% of the words will be masked. This strategy can adapt to various sentence lengths.

3.3.2. Style-Dictionary-Based Method

We define this method as follows. For any dataset $\{D_i\}_{i=1}^K$, let $count(u, D_i)$ denote the number of times the n-gram word u appears in dataset D_i. The smoothed frequency ratio represents the significance of u relative to the dataset D_i and is calculated as follows:

$$score(u,i) = \frac{count(u, D_i) + \lambda}{(\sum_{i^* \in K, i^* \neq i} count(u, D_{i^*})) + \lambda}, \quad (4)$$

where λ is the smoothing parameter. When $score(u,i)$ is greater than the threshold γ_{score}, the word u is considered to be a component with a large contribution to the style attribute. Finally, these words form the auxiliary style dictionary v_i.

3.3.3. Fusion Method

In this paper, we propose a fusion method that combines the advantages of these two approaches described above. The auxiliary-style-dictionary based method has the advantages of stability and scalability. When the threshold γ_{score} is larger, the words in style dictionary v_i will be more characteristic. The self-attention classifier-based method has the advantages of flexibility and self-adaptation. It can automatically analyze the style attributes of each component of the sentence and find some components with potential style attributes.

The specific process is as follows. First, calculate the style dictionary v_i through (4). For each sample, mask all the words appearing in the style dictionary v_i. If the words masked by the assisted style dictionary method do not reach the mask proportion, the self-attention classifier-based method will remask the sentences by calculating the attention value of each component. For sample x in dataset D, we use $\tilde{x}$ to represent the masked sentences—that is, $\tilde{x} = mask(x)$—and the dataset of masked sentences is denoted as D_{mask}.

3.4. Style Generator

In the Style Generator, we chose the standard Transformer model, following the classic encoder-decoder structure. For example, for the input $\tilde{x} = (x_1, x_2, \ldots, x_N)$, the transformer encoder $Encoder_\theta(\tilde{x})$ maps it to a latent continuous representation vector $c = (c_1, c_2, \ldots, c_N)$. Then, the transformer decoder $Decoder_\theta(c)$ generates the conditional probability of output $y = (y_1, y_2, \ldots, y_N)$ through an autoregressive calculation as follows:

$$P_\theta(y|\tilde{x}) = \prod_{t=1}^{m} P_\theta(y_t|c, y_1, \ldots, y_{t-1}), \quad (5)$$

For each time t, in the decoder, the probability of generating a word is calculated by a Softmax layer:

$$P_\theta(y_t|c, y_1, \ldots, y_{t-1}) = softmax(o_t), \quad (6)$$

where o_t is the logit vector output by the decoder.

To apply the target style control to the generation, we additionally add a mark of the target style before the input, similar to the <cls> mark for BERT. That is, $Encoder_\theta(s, \tilde{x})$. Therefore, the model can calculate the output probability under the conditions of input $\tilde{x}$ and target style s:

$$P_\theta(y|s, \tilde{x}) = \prod_{t=1}^{m} P_\theta(y_t|c, y_1, \ldots, y_{t-1}), \quad (7)$$

We denote the Style Generator model as f_θ, where θ represents the model parameters. Then, the predicted sentence calculated above is denoted by $f_\theta(s, \tilde{x})$.

3.5. Discriminator

The purpose of introducing the discriminator module is to solve the problem of nonparallel corpus training. When we take samples from the dataset D and obtain $(s, \tilde{x})$ through the masker module, but due to the lack of a parallel corpus, we cannot obtain the corresponding reference to sentence $f_\theta(\hat{s}, \tilde{x})$, while target style $\hat{s} \neq s$. Therefore, we introduce a discriminator module to learn via supervised training from nonparallel corpora.

For the data $(s, \tilde{x})$, we can intuitively restore $\tilde{x}$ to sentence x according to its original style s. Furthermore, we use its own supervised training to make the model have a certain style transfer ability. For the target style $\hat{s} \neq s$, we train a discriminator network to constrain the optimization direction of the generation module in order to better generate target style sentences.

The discriminator network we use includes a Transformer encoder, which is used to distinguish the styles of sentences. The style control information of the discriminator network will be passed to the generation module. Different from the traditional discriminator, in order to better guide the generation module during training, we refer to the discriminator training method of [18] and use two different discriminator structures. We denote the discriminant model as d_φ, where φ is the model parameter.

1. Conditional Discriminator

Similar to the discriminator in conditional GANs (Generative Adversarial Networks), the conditional discriminator makes decisions based on the input sentence and style. Specifically, the conditional discriminator d_φ needs to complete a binary classification task, and its inputs are the sentence x and the matching style s. The output of the discriminator $d_\varphi(s, x)$ determines whether the style of the input sentence x is s.

In the discriminator training process, for the style s, the positive sample is the real sentence x and the reconstructed sentence $f_\theta(s, \tilde{x})$, and the negative sample is the transfer sentence $f_\theta(\hat{s}, \tilde{x})$, while the target style $\hat{s} \neq s$. In the training process of the Style Generator, the goal of the generator f_θ is to maximize the probability that the discriminator determines that $d_\varphi(\hat{s}, f_\theta(\hat{s}, \tilde{x}))$ is true.

2. Multi-Class Discriminator

Compared to the former, the multiclass discriminator only uses one sentence as its input, and its goal is to judge the style of the sentence. Unlike traditional discriminators, for K-style tasks, multiclass discriminators need to perform $K + 1$ classification tasks. The first K categories are K styles, and the last category is the transfer sentence $f_\theta(\hat{s}, \tilde{x})$ of the target style $\hat{s} \neq s$. The purpose of this design is to help the generation module learn more accurate knowledge from the discriminator. As the transfer sentence $f_\theta(\hat{s}, \tilde{x})$ is usually poor at the beginning of training, setting these sentences as another class can make the generator closer to the distribution of real sentences during the iterative training process.

In the discriminator training process, the real sentence x and the reconstructed sentence $f_\theta(\hat{s}, \tilde{x})$ will be labeled as style s, and the transfer sentence $f_\theta(\hat{s}, \tilde{x})$ will be labeled as class 0. In the training

process of the Style Generator, the goal of f_θ is to maximize the probability that the discriminator determines that $f_\theta(\hat{s}, \tilde{x})$ has style $\hat{s}$.

3.6. Training Algorithm

This section will mainly introduce the training algorithm of each module.

1. Masker Training Algorithm

The training algorithm of the masker module mainly trains a classification model based on self-attention. Its goal is to determine the style category of each sentence to obtain the attention weight as the subsidiary product. This is the basis for analyzing and masking the sentence. The loss function for the Masker is the cross-entropy loss of the classification problem; that is,

$$\mathcal{L}_{masker}(\rho) = -\sum_{(s,x_i)\in D}^{n} logp(c|x_i), \quad (8)$$

where dataset D is the original training set.

The learning algorithm of the discriminator mainly trains a classification model based on the Transformer encoder. Its goal is to distinguish between the original sentence x, the reconstructed sentence $f_\theta(s, \tilde{x})$, and the transfer sentence $f_\theta(\hat{s}, \tilde{x})$. The loss function for the discriminator is the cross-entropy loss of the classification problem.

For the conditional discriminator,

$$\mathcal{L}_{discriminator}(\varphi) = -\sum_{(s,x_i)\in D_{dis}}^{n} logp(c|s, x_i), \quad (9)$$

and for multiclass discriminator,

$$\mathcal{L}_{discriminator}(\varphi) = -\sum_{(s,x_i)\in D_{dis}}^{n} logp(c|x_i), \quad (10)$$

where dataset D consists of x, $f_\theta(s, \tilde{x})$ and $f_\theta(\hat{s}, \tilde{x})$. The details are given in Algorithm 1.

Algorithm 1: The Training of the Discriminator

Input: dataset D_{mask}, discriminator d_φ, style generator f_θ
for each sentence $(s, x, \tilde{x})$ in dataset D_{mask} do
  Select a style $\hat{s}$ where $\hat{s} \neq s$;
  $y = f_\theta(s, \tilde{x})$;
  $\hat{y} = f_\theta(\hat{s}, \tilde{x})$;
  Append $(s, x, y, \hat{s}, \hat{y})$ to dataset D_{dis}
end for
for each iteration i = 1,2,…,m do
  Sample a minibatch of cases from D_{dis};
  for each case $(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{s}}, \hat{\boldsymbol{y}})$ in a batch do
    if discriminator is multi-class then
      Set $\{x, y\}$ as class $_S + 1$;
      Set $\{\hat{y}\}$ as class 0;
      Compute $\mathcal{L}_{discriminator}$ (10);
    else
      Set $\{(s, x), (s, y)\}$ as True;
      Set $\{(\hat{s}, x), (\hat{s}, \hat{y})\}$ as False;
      Compute $L_{discriminator}$ (9);
    end if
  end for
  Update the model parameters φ;
end for

2. Style Generator Training Algorithm

The training of the style generator is divided into two parts. One part is the case where the target style $\hat{s} = s$, and the other part is the case where the target style $\hat{s} \neq s$.

Sentence reconstruction: For the case when the target style $\hat{s} = s$, we can directly apply a training method that reconstructs the mask sentence $\tilde{x}$ into the original sentence x by using its own supervision information. Specifically, when using the style s and the masked sentence $\tilde{x}$ as input, the model output is as $f_\theta(s, \tilde{x})$ close as possible to the original sentence x. The training goal is to minimize the negative log-likelihood:

$$\mathcal{L}_{reconstruction}(\theta) = -\sum_{(s,\tilde{x}_i)\in D_{mask}}^{n} logp(y = x|s, \tilde{x}_i), \quad (11)$$

where the dataset D_{mask} is obtained by masking the sentences in the original training set D.

Style generation: For the case when the target style $\hat{s} \neq s$, we introduce a loss function to control the generation of the style, so that the transformed sentences are closer to the distribution of the real sentences and the reconstructed sentences.

Using the conditional discriminator, we can obtain the probability that $d_\varphi(\hat{s}, f_\theta(\hat{s}, \tilde{x})$ is true. The goal of the generator is to minimize the negative log-likelihood:

$$\mathcal{L}_{style}(\theta) = -\sum_{(s,\tilde{x}_i)\in D_{mask}}^{n} logp(c = 1|f_\theta(\hat{s}, \tilde{x}), \hat{s}), \quad (12)$$

Using the multiclass discriminator, we can calculate the probability that $d_\varphi(f_\theta(\hat{s}, \tilde{x}))$ has the style $\hat{s}$. The goal of the generator is to minimize the negative log-likelihood of the class probability:

$$\mathcal{L}_{style}(\theta) = -\sum_{(s,\tilde{x}_i)\in D_{mask}}^{n} logp(c = \hat{s}|f_\theta(\hat{s}, \tilde{x})), \quad (13)$$

Combining the loss functions described above, we can conduct training for the generator. Algorithm 2 shows the details of the training process.

Algorithm 2: The Training of the Style Generator

Input: dataset D_{mask}, discriminator d_φ, style generator f_θ
for each iteration i = 1,2,…,m do
  Sample a minibatch of cases from D_{mask};
  for each case $(s, x, \tilde{x})$ in a batch do
    Select a style $\hat{s}$ where $\hat{s} \neq s$;
    Reconstruct the sentence $y = f_\theta(s, \tilde{x})$;
    Generate a sentence $\hat{y} = f_\theta(\hat{s}, \tilde{x})$;
    Judge $\hat{y}$ by discriminator d_φ;
    Compute $L_{reconstruction}$ (11);
    Compute L_{style} (12 or 13);
  end for
  Update the model parameters θ;
end for

3. Summarization

By combining the training algorithms of the above modules, we can obtain the whole training process of the model. In the mask part, first, the self-attention model of the Masker is trained. After the model convergence is stable, the Masker masks the sentences in the original dataset D and obtains the dataset D_{mask}. In the style generation part, using the alternate training method of the generator and discriminator in GANs [14], we also alternately train the Style Generator and Discriminator. In each iterative round, we first train the Discriminator n_d steps to obtain an optimized discrimination module. Under the updated Discriminator, the Style Generator will be trained n_g steps to optimize the results of the generation. The iterative training continues until the model converges and stabilizes. The specific algorithm is given in Algorithm 3.

Algorithm 3: The Whole Training Algorithm

Input: dataset D, attention-based classifier cls_ρ
for each iteration $i = 1,2,\ldots,m$ do
  Sample a minibatch of sentences from D;
  Compute $\mathcal{L}_{masker}$ (8) for the batch;
  Update the model parameters ρ;
end for
Build the style dictionary by (5)
for each sentence in dataset Ddo
  Mask the sentence using the style dictionary;
  if (mask words/length of sentence) < masking percentage then
    Mask again according to the attention weight from cls_ρ;
  end if
  Append the case $(s, x, \tilde{x})$ into D_{mask};
end for
Initialize Style Generator network θ and Discriminator network φ;
for each iteration i = 1,2,…,m do
  for n_d steps do
    Algorithm 2;
  end for
  for n_g steps do
    Algorithm 3;
  end for
end for

The learning algorithm of the discriminator mainly trains a classification model based on the Transformer encoder. Its goal is to distinguish between the original sentence x, the reconstructed sentence $f_\theta(s, \tilde{x})$ and the transfer sentence $f_\theta(\hat{s}, \tilde{x})$. The loss function for the discriminator is the cross-entropy loss of the classification problem.

In addition, there is a problem in training that needs to be briefly explained. Due to the discrete nature of natural language, when we obtain the transfer sentence and input it into the discrimination module, the gradient calculated by the discrimination module cannot be propagated back to the generation module. To solve this problem, it is common to use the Gumbel–Softmax strategy or the reinforcement learning method to evaluate the gradient from the discriminator. However, both methods have the problem of high variance, which makes it difficult for the model to converge and stabilize. Therefore, we use the way that [18] deal with discrete sample problems. Instead of directly using the generated words as the input, we use the Softmax distribution generated by f_θ as the input. Similarly, for the decoder of the generator, the decoding method is also changed from greedy decoding to continuous decoding. Specifically, at each calculation time, instead of using the word with the highest probability predicted in the previous step, we use the probability distribution as the input. Regarding the input in the form of a probability distribution, the decoder will calculate the weighted average representation of the probability distribution through the embedding matrix.

4. Experiment

4.1. Datasets

We compared our method with several recent approaches on two review datasets: the Yelp Review dataset (Yelp) and the IMDb Movie Review dataset (IMDb). Table 1 shows the statistics of these two datasets.

Yelp Review dataset (Yelp): The Yelp dataset is provided by the Yelp dataset challenge. It consists of restaurant and business reviews with sentiment labels (negative or positive). Following

previous work, we used the dataset provided by [16]. In addition, it also provides artificial reference sentences for the test set

IMDb Movie Review dataset (IMDb): The IMDb dataset consists of movie reviews written by online users. To obtain a high-quality dataset, we used the highly polar movie reviews provided by [18]. Using the original dataset, they construct a highly polar sentence-level style transfer dataset via the following steps. (1) A BERT-based classifier is trained and fine-tuned on the original dataset. The accuracy is 95%. (2) The original comment in the dataset is split into several sentences. (3) Each sentence with a confidence threshold lower than 0.9 is filtered by the fine-tuned BERT-based classifier. (4) Sentences containing uncommon words are deleted. Finally, the highly polar dataset contains 366k training, 4k validation, and 2k test sentences.

Table 1. The statistics of the Yelp and IMDb datasets.

Dataset	Attributes	Train	Dev	Test	Avg Len
Yelp	Positive	266k	2000	500	8.9
	Negative	177k	2000	500	
IMDb	Positive	179k	2000	1000	18.5
	Negative	188k	2000	1000	

4.2. Evaluation

An ideal transfer sentence should be prominent, content-complete, and fluent. After referring to the evaluations in previous work, we mainly focus on the following three aspects of sentence generation: (1) the degree of style transfer, (2) content preservation, and (3) fluency. In terms of the current evaluation methods, we include two parts: automatic evaluation and manual evaluation.

4.2.1. Automatic Evaluation

Style transfer: To evaluate the style transfer, we use the accuracy of the generated sentences in the style classification as an automatic evaluation metric. Specifically, we refer to [9] and use fastText to train a style classifier based on the Yelp and IMDb datasets, respectively. For the transfer sentence $f_\theta(\hat{s}, \tilde{x})$, we use the classifier to evaluate the style accuracy.

Content preservation: To measure the content preservation, we adopt the BLEU (Bilingual Evaluation Understudy) score [23] as the evaluation metric. Specifically, we use the calculation tool provided by NLTK (Natural Language Toolkit) to calculate the BLEU score for the transfer sentence and the original sentence. A high BLEU score indicates that there is a high degree of similarity between the converted sentence and the original sentence on the word-level. This indicates that there is good content reservation. In addition, if the dataset provides artificial reference sentences, we will also calculate the BLEU score of the transfer sentence and the reference sentence. The two BLEU metrics are defined as self-BLEU and ref-BLEU.

Fluency: A common way to evaluate the fluency is calculating the perplexity of the transfer sentence. Specifically, we use KenLM to train a 5-g language model on the Yelp and IMDb datasets, and we use this model to calculate the perplexity of the sentence. The lower the perplexity is, the higher the generation probability and fluency of the sentence.

4.2.2. Manual Evaluation

For the manual evaluation, we chose the scoring method and hired three reviewers to score the output of our model and the best models of [18] (Style Transformer) and [16] (Delete and Retrieve). Scoring is similar to the automatic evaluation. We mainly evaluate three aspects: the style transfer, the content reservation, and the fluency of the output results. Our scoring scale has scores that range from 1 (very poor) to 5 (very good). For each dataset, we will sample 100 sentences for each target style in the evaluation.

4.3. Training Details

Mask part: For the self-attention classification model, the word embedding size is 256, the bidirectional LSTM hidden size is 256, and the number of hidden layers is two. For training, we use the Adam optimizer with a learning rate of 0.001 and train for five epochs. For the assisted style dictionary method, we use the statistical results of 1-g and 2-g, set the smoothing parameter λ to 1, and set the threshold γ_{score} to 0.75.

Generation part: In the generator and the discriminator, we use four layers of transformers, and each layer has a multi-head attention of eight. The word embedding, position encoding, and style embedding are all 256 in size. In the encoder, style embedding will be added to the head of the sentence as a mark, and the mark does not use position encoding information. For the discriminator, similar to the BERT [24], we add the $< cls >$ mark to the head of the input sentence, and the output corresponding to this mark will be passed to a Softmax layer to obtain the output of the discriminator. In terms of the training parameters, n_d is set to 10 and n_g is set to five. We use the Adam optimizer with a learning rate of 0.0001.

In the experiment, in order to improve the robustness of the model and converge to a more reasonable result, we use the practice of [18]. When the model calculates the reconstruction loss function (6), we conduct random word dropout on the input, which makes the model more robust. The experiments show that the random word dropout improves the transfer results in some cases.

4.4. Experimental Results

Table 2 shows the automatic evaluation results of the model on the two datasets. In the Yelp dataset, although RetrieveOnly achieved the highest accuracy rate of 92.6%, and the perplexity is the lowest of 7. While in terms of semantic retention, the self-BLEU of RetrieveOnly was only 0.7, and the ref-BLEU was only 0.4. Almost no information about the original sentence was retained. This is because RetrieveOnly uses a retrieve method to find the most suitable template in the target style dataset, and directly use the template as the generated sentence. Therefore, the style conversion degree of this method is very high and the fluency of the template sentence almost as natural language. However, since the content of the generated sentence is completely changed compared to the original sentence, this method has no practical value. In terms of IMDB, the CycleRL method suffers from the same problem. Although it achieves an accuracy of 97.8% in style conversion, it only has a self-BLEU of 4.9 and a perplexity of 177 in terms of semantic retention and language fluency. We can see that our model has achieved the best results in style accuracy and content reservation for both datasets, and it also has better perplexity.

Table 2. The automatic evaluation results of the Yelp and IMDb datasets.

	Yelp				**IMDb**		
	ACC	**self-BLEU**	**ref-BLEU**	**PPL**	**ACC**	**self-BLEU**	**PPL**
ControlledGen [21]	88.8	45.7	14.3	219	94.6	62.1	143
DeleteOnly [16]	85.7	28.6	9.7	72	N/A	N/A	N/A
RetrieveOnly [16]	92.6	0.7	0.4	7	N/A	N/A	N/A
TemplateBased [16]	84.3	44.1	13.7	117	N/A	N/A	N/A
DeleteAndRetrieve [16]	87.7	29.1	10.4	60	58.8	55.4	57
CycleRL [7]	88.0	7.2	2.8	107	97.8	4.9	177
StyleTransformer [18]	89.5	54.5	18.7	76	82.3	67.5	108
Conditional MaskTransformer	91.8	54.6	19.3	81	95.7	63.9	92
Multi-class MaskTransformer	88.3	55.4	20.1	75	91.0	66.2	86

ACC and PPL denote accuracy and perplexity, respectively.

For the manual evaluation, we select two representative models: Delete And Retrieve [16] and the Style Transformer [18]. For our method, we chose the model based on the multiclass discriminator. We randomly sample 100 sentences for each style in the dataset, and the transfer sentences were scored by three reviewers after being scrambled. The results are shown in Table 3. It can be seen that our model performs significantly better than other methods in style transfer accuracy as well as content remaining

and fluency of the target sentence. To better understand the characteristics of our model, we sampled some sentences in the Yelp dataset, as shown in Table 4.

Table 3. The manual evaluation results of the Yelp and IMDb datasets.

	Yelp			IMDb		
	Style	**Content**	**fluency**	**Style**	**Content**	**fluency**
DeleteAndRetrieve	3.67	2.61	2.84	3.75	2.33	2.26
StyleTransformer	3.41	3.34	3.64	3.36	3.58	3.52
Ours (conditional)	4.12	4.08	3.67	3.98	3.87	3.31

Table 4. Case studies from the Yelp dataset.

	Positive to Negative
Input	the salads were fresh and crispy.
Masked	the salads were [mask] and [mask].
ST	the salads were dry and crispy.
Ours	the salads were terrible and not worthy.
Reference	the salads were old and wilted.
Input	reasonable price, bottom line guaranteed.
Masked	[mask] price, bottom line guaranteed.
ST	reasonable price, bottom line fees.
Ours	overpriced and no bottom line fees.
Reference	the price isn't reasonable, bottom line isn't guaranteed.
	Negative to Positive
Input	i can't believe how inconsiderate this pharmacy is.
Masked	i ca [mask] believe how [mask] this pharmacy is.
ST	i can always believe how compassionate this pharmacy is.
Ours	i can always believe how impressed this pharmacy is.
Reference	this pharmacy is really considerate.
Input	the burgers were over cooked to the point the meat was crunchy.
Masked	the burgers were [mask] cooked to the point the meat was [mask].
ST	the burgers are while cooked to the point the meat loved crunchy!
Ours	the burgers were cooked to the point and the meat was delicious.
Reference	the burgers were cooked perfectly and the meat was juicy.

5. Conclusions

In this paper, we focus on solving the style transfer problem of nonparallel corpora and propose a style generation method based on a "Mask and Generation" structure, which can be trained using nonparallel corpora and has good interpretability. The experimental results on two review datasets show that our method outperforms previous approaches in terms of style conversion and content reservation. The masked sentences can keep the semantic meaning of the original sentence and also help us to understand the transfer process. In the future, we want to further explore the introduction of prior knowledge in the masker module and improve the model for more fine-grained tasks (multiple emotions).

Author Contributions: Conceptualization, C.W. and X.C.; data curation, C.W.; methodology, X.C. and C.W.; validation X.C. and X.L.; writing—original draft, X.C.; writing—review & editing, C.W.; project administration, C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by the National Key R&D Program of China under Grant No. 2017YFB0802803 and the Natural Science Foundation of China under Grant No. 61602052.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fu, Z., Tan, X., Peng, N., Zhao, D. and Yan, R. Style transfer in text: Exploration and evaluation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, February 2–7, 2018.
2. Zhang, Y.; Ding, N.; Soricut, R. 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, LA, USA, 1–6 June 2018; Volume 1, pp. 1528–1538.
3. Prabhumoye, S.; Tsvetkov, Y.; Salakhutdinov, R.; Black, A.W. Style transfer through back-translation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 866–876.
4. Santos, C.N.D.; Melnyk, I.; Padhi, I. 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, 15–20 July 2018; Volume 2, pp. 189–194.
5. Luo, F.; Li, P.; Zhou, J.; Yang, P.; Chang, B.; Sui, Z.; Sun, X. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer. *arXiv* **2019**, arXiv:1905.10060.
6. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
7. Xu, J.; Sun, X.; Zeng, Q.; Ren, X.; Zhang, X.; Wang, H.; Li, W. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 979–988.
8. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
9. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; Volume 2, pp. 427–431.
10. Wu, X.; Zhang, T.; Zang, L.; Han, J.; Hu, S. Mask and Infill: Applying Masked Language Model to Sentiment Transfer. *arXiv* **2019**, arXiv:1908.08039.
11. Zhu, W.; Hu, Z.; Xing, E. Text infilling. *arXiv* **2019**, arXiv:1901.00158.
12. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; MIT Press: Cambridge, MA, USA, 2017.
13. Shen, T.; Lei, T.; Barzilay, R.; Jaakkola, T. Style transfer from non-parallel text by cross-alignment In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017*; MIT Press: Cambridge, MA, USA, 2017.
14. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2014; Volume 2, pp. 2672–2680.
15. John, V.; Mou, L.; Bahuleyan, H.; Vechtomova, O. Disentangled representation learning for text style transfer. *arXiv* **2019**, arXiv:1808.04339.
16. Li, J.; Jia, R.; He, H.; Liang, P. 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, NAACL-HLT 2018, New Orleans, LA, USA, 1–6 June 2018; Volume 1, pp. 1865–1874.
17. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 2242–2251.
18. Dai, N.; Liang, J.; Qiu, X.; Huang, X. StyleTransformer: Unpaired Text Style Transfer without Disentangled Latent Representation. *arXiv* **2019**, arXiv:1905.05621.

19. Cheng, Y.; Gan, Z.; Zhang, Y.; Elachqar, O.; Li, D.; Liu, J. Contextual text style transfer. *arXiv* **2020**, arXiv:2005.00136.
20. He, J.; Wang, X.; Neubig, G.; Berg-Kirkpatrick, T. A Probabilistic Formulation of Unsupervised Text Style Transfer. *arXiv* **2020**, arXiv:2002.03912.
21. Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; Xing, E.P. Toward controlled generation of text. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1587–1596.
22. Yang, Z.; Hu, Z.; Dyer, C.; Xing, E.P.; Berg-Kirkpatrick, T. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems, Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, QC, Canada, 3–8 December 2018;* MIT Press: Cambridge, MA, USA, 2018; pp. 7298–7309.
23. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
24. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.