Text Style Transfer with Contrastive Transfer Pattern Mining

Jingxuan Han[1], **Quan Wang**[2], **Licheng Zhang**[1], **Weidong Chen**[1]
Yan Song[1] and **Zhendong Mao**[1]*
[1]University of Science and Technology of China, Hefei, China
[2]MOE Key Laboratory of Trustworthy Distributed Computing and Service,
Beijing University of Posts and Telecommunications, Beijing, China
hjx999222@mail.ustc.edu.cn, zdmao@ustc.edu.cn

Abstract

Text style transfer (TST) is an important task in natural language generation, which aims to alter the stylistic attributes (e.g., sentiment) of a sentence and keep its semantic meaning unchanged. Most existing studies mainly focus on the transformation between styles, yet ignore that this transformation can be actually carried out via different hidden transfer patterns. To address this problem, we propose a novel approach, contrastive transfer pattern mining (CTPM), which automatically mines and utilizes inherent latent transfer patterns to improve the performance of TST. Specifically, we design an adaptive clustering module to automatically discover hidden transfer patterns from the data, and introduce contrastive learning based on the discovered patterns to obtain more accurate sentence representations, and thereby benefit the TST task. To the best of our knowledge, this is the first work that proposes the concept of transfer patterns in TST, and our approach can be applied in a plug-and-play manner to enhance other TST methods to further improve their performance. Extensive experiments on benchmark datasets verify the effectiveness and generality of our approach.[1]

1 Introduction

Text style transfer (TST), an important task in natural language generation, aims to change the stylistic properties while preserving the style-independent content within the context. Stylistic properties include but are not limited to sentiment, politeness, formality, and humor. This task has wide applications, such as neutralizing offensive remarks (Cicero et al., 2018), data augmentation (Xu et al., 2019), and human-computer interaction (Li et al., 2016). The main difficulty in this task is the lack of parallel datasets, making most previous works develop their methods in an unsupervised manner and can be roughly divided into two groups.

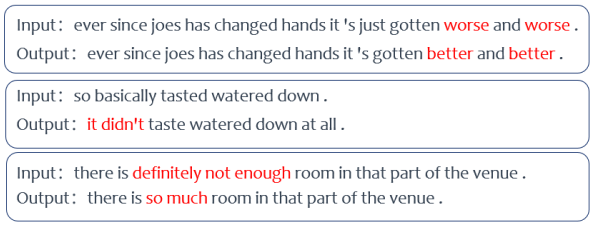

Figure 1: Three cases from the test set of sentiment transfer dataset Yelp, which can reflect different transfer patterns.

Methods in the first group (Lee et al., 2021; Reid and Zhong, 2021) separate style-independent sentence representations and revise them with style attributes, while methods in the second group (Dai et al., 2019; Kashyap et al., 2022) directly revise an entangled representation of an input by using an extra style embedding. However, the two groups of existing methods only focus on the transformation between styles and do not take into account that this transformation might be achieved via different latent transfer patterns.

As a matter of fact, there are lots of transfer patterns in the TST task. Figure 1 shows three cases from the test set of sentiment transfer dataset Yelp (Li et al., 2018). In the first case, the transfer pattern is extracting key emotional words and taking their antonyms. The transfer pattern of the second case is to change the affirmative polarity to negative polarity while the last case is to change the negative polarity to affirmative polarity. Such latent patterns naturally exist for various source texts and imply different specialized ways to solve the transfer task.

Intuitively, it would be dramatically informative and helpful to mine and exploit latent transfer patterns as prior knowledge. However, one key obstacle lies ahead we have no access to such golden annotation of a systematic taxonomy for all available patterns, nor does it cost-effective to manually conclude and annotate one.

In this work, we propose a novel **contrastive**

*Corresponding author: Zhendong Mao.
[1]Our code and data will be released at GitHub.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7914–7927

transfer pattern mining (CTPM) method, which can automatically detect different transfer patterns and use contrastive learning on this basis. In this way, we can obtain more accurate sentence representations, which thereby help us to achieve better text style transfer. Figure 2 gives an overview of our approach. We first use a clustering module to automatically mine latent transfer patterns and obtain transfer pattern labels. Then we simultaneously exploit intra-style contrastive learning with transfer pattern labels and inter-style contrastive learning with style labels to enhance text representations, boosting the performance of text style transfer. It is worth mentioning that our method can be combined plug-and-play with both the above two groups of mainstream TST methods to improve their performance.

In summary, our contributions are as follows:

- We propose the concept of latent transfer patterns in the TST task for the first time, and design a clustering module to mine and distinguish such patterns.
- Based on the mined transfer patterns, we introduce intra-style and inter-style contrastive learning to obtain more accurate sentence representations.
- We combine our method with two typical basic TST models belonging to the above two mainstream groups and conduct extensive experiments on two benchmarking datasets. The results demonstrate the effectiveness and generality of our approach.

2 Approach

The overall architecture of our method is depicted in Figure 2, which consists of an **adaptive clustering module** and a **contrastive learning module**. The adaptive clustering module automatically mines latent transfer patterns in each style. Based on the mined transfer patterns, the contrastive learning module adopts intra-style and inter-style contrastive learning losses to learn more precise sentence representations. The two losses are applied to a basic TST model to further improve its performance. We will first give a brief introduction to the TST task, and then elaborate on the two modules.

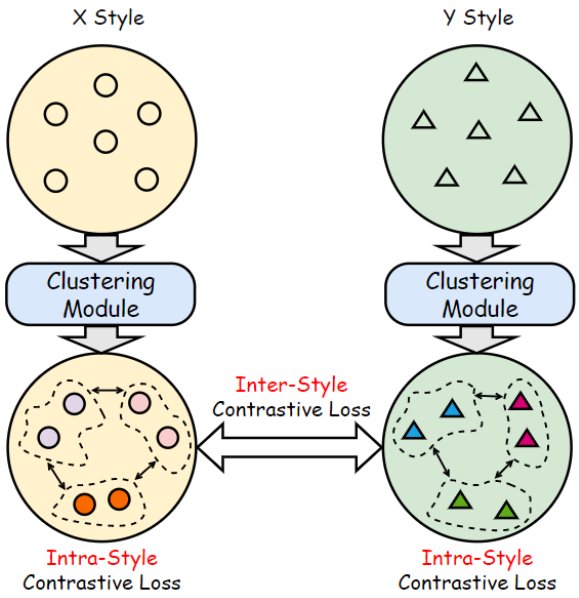

Figure 2: General architecture of our method. Without loss of generality, we take the TST task with two styles for example. The clustering module mines and distinguishes latent transfer patterns within each style, and the contrastive learning module further adopts intra-style and inter-style contrastive learning on the basis of the mined patterns.

2.1 Problem Formulation

Considering a training corpus $\mathcal{D} = \left\{(\boldsymbol{x}_i, s_i)_{i=1}^{T}\right\}$, $\boldsymbol{x}_i$ is an input sentence and s_i is its style label. TST aims to learn a model $\hat{\boldsymbol{x}}_{\hat{s}} = f_\theta(\boldsymbol{x}, \hat{s})$, which takes an arbitrary natural language sentence $\boldsymbol{x}$ and a desired style $\hat{s} \in \{s_j\}_{j=1}^{H}$ as input, and then generates a sentence $\hat{\boldsymbol{x}}_{\hat{s}}$ with style $\hat{s}$, while preserving as much information in original sentence $\boldsymbol{x}$ as possible. Take the classic sentiment transfer task as an example. Two styles are considered there namely the positive and negative sentiments, and the task is to transfer the sentence between the two styles while maintaining content information.

2.2 Adaptive Clustering Module

We propose an adaptive clustering module to mine latent transfer patterns and cluster the sentences with the same style into different transfer patterns. Concretely, we adopt an entropy-based method to find an optimal number of clusters for each style automatically, and then design a neural clustering algorithm to cluster the sentences into different transfer patterns. The clustering result could be regarded as some kind of supervision for obtaining more accurate sentence representations.

Cluster Number Calculation Generally, the clustering effect is strongly influenced by the number of clusters. Inspired by U-k-means (Sinaga and Yang, 2020), we create a learning schema to find the optimal cluster number. For a style set, we obtain a d-dimension vector $\boldsymbol{r}_j \in \mathbb{R}^d$ as the representation of each sentence $\boldsymbol{x}_j$ using a pre-trained language model BERT (Devlin et al., 2019). We initialized the cluster number by the sentence number and gradually cluster sentence representations to decrease the cluster number during iterations. By minimizing the entropy of the probability of one sentence belonging to its corresponding clusters, an optimal cluster number k can be automatically found according to the data structure.

Neural Clustering Network Inspired by K-means (MacQueen, 1967), We propose a neural clustering network to mine latent transfer patterns. In our method, the sentence representations are classified according to their distance to cluster centers (centroids), and then each centroid is calculated by a weighted sum of all sentence representations. We designed a clustering loss to optimize the network to divide sentences into different clusters.

In detail, our clustering module takes the sentence representations $\{\boldsymbol{r}_j\}_{j=1}^N$ and centroid hidden states $\{\boldsymbol{c}_i\}_{i=1}^k$ as input, where N is the batch size and k is the cluster number for each style. $\boldsymbol{c}_i \in \mathbb{R}^d$ is initialized randomly. We can obtain the distance matrix M between sentence representations and centroids, defined as:

$$M_{ij} = \frac{\exp\left(\phi\left(\boldsymbol{c}_i, \boldsymbol{r}_j U^\theta\right)\right)}{\sum_{j=1}^N \exp\left(\phi\left(\boldsymbol{c}_i, \boldsymbol{r}_j U^\theta\right)\right)} \quad 1 \le i \le k, \quad 1 \le j \le N \tag{1}$$

where $M_{ij} \in [0, 1]$ is the normalized distance value between the i-th centroid vector $\boldsymbol{c}_i$ and j-th sentence representation $\boldsymbol{r}_j$, which implies the negative degree of the $\boldsymbol{r}_j$ belonging to the centroid $\boldsymbol{c}_i$. $U^\theta \in \mathbb{R}^{d \times d}$ is a learnable parameter matrix of a MLP, $\phi(\cdot)$ is a distance measure function.

Then we can classify the sentence representations according to the distance matrix M:

$$I_j = \operatorname{argmin}\left(M_{:,j}\right) \quad 1 \le j \le N \tag{2}$$

where $M_{:,j}$ represents the negative degree of the sentence representation $\boldsymbol{r}_j$ belonging to each centroid and function argmin assigns sentence representations to the associated cluster according to the minimum distance value. As a consequence, $I \in \mathbb{R}^N$ implies the clustered index of the sentence representations.

To train the clustering module to learn the optimal clustering formula, we propose a clustering loss that minimizes the distance between sentence representations and their belonging centroids. The clustering loss aims to find the optimal U^θ and thereby calculate the optimal $\boldsymbol{c}$, so that after the sentence representations go through the clustering module, the distance to the corresponding centroid is lower, and to other centroids is higher.

$$\mathcal{L}_{clu} = \frac{1}{N} \sum_{j=1}^{N} \phi\left(\boldsymbol{r}_j U^\theta, \boldsymbol{c}_{I_j}\right) \tag{3}$$

After that, we calculate the new centroids with the weighted sum of all sentence representations, where weight is related to the distance matrix.

$$\boldsymbol{c}_i = \sum_{j=1}^{N} (1 - M_{ij}) \boldsymbol{r}_j U^\theta \quad 1 \le i \le k \tag{4}$$

When the network is adequately trained, we are able to obtain the labels of transfer patterns for each sentence using Eq.2.

2.3 Contrastive Learning Module

We introduce supervised contrastive learning (Khosla et al., 2020) to regularize the latent space, so that two sentences with the same transfer pattern label or style label (positive pairs) will lie close together, and otherwise far apart, which finally makes the sentence representations more distinguishable. In detail, we design intra-contrastive loss based on the transfer pattern labels within the same style, and design inter-contrastive loss based on the style labels between different styles. Both two losses will be applied to a basic TST model eventually.

Intra-style Contrastive Learning Considering a sentence $\boldsymbol{x}_i \in \mathcal{D}$ in a batch B, $P_t(i)$ is a positive sentence set in which sentences share the same transfer patterns with $\boldsymbol{x}_i$, and $A_t(i)$ is $B \setminus P_t(i)$ which means negative sentence set. Intra-style contrastive loss of sentence $\boldsymbol{x}_i$ is defined as follows[2]:

$$\mathcal{L}_{intra} = \frac{-1}{|P_t(i)|} \sum_{p \in P_t(i)} \log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau)}{\sum_{a \in A_t(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau)} \tag{5}$$

[2]In fact, we will add up the loss of all sentences $\boldsymbol{x}_i$ in a batch as $\mathcal{L}_{intra}$. In order to simplify the notation, we have omitted the expression of summation.

where $\boldsymbol{z}$ is the representation obtained from the basic TST model, τ is temperature. By minimizing $\mathcal{L}_{intra}$, sentences with the same transfer patterns will lie close together, and otherwise far apart.

Inter-style Contrastive Learning Considering a sentence $\boldsymbol{x}_i \in \mathcal{D}$ in a batch B, $P_s(i)$ is a positive sentence set in which sentences share the same styles with $\boldsymbol{x}_i$, and $A_s(i)$ is $B \setminus P_s(i)$ which means negative sentence set. Inter-style contrastive loss of sentence $\boldsymbol{x}_i$ is defined as follows[3]:

$$\mathcal{L}_{inter} = \frac{-1}{|P_s(i)|} \sum_{p \in P_s(i)} \log \frac{\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau)}{\sum_{a \in A_s(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau)} \quad (6)$$

where $\boldsymbol{z}$ and τ are consistent with Eq.5. By minimizing $\mathcal{L}_{inter}$, sentences with the same style will lie close together, and otherwise far apart.

There is an intra-style loss $\mathcal{L}_{intra}$ for each style and an inter-style loss $\mathcal{L}_{inter}$ between styles in latent space. Thus, for the H styles, the general formulation of total contrastive loss is:

$$\mathcal{L}_{con} = \frac{1}{H+1}(\mathcal{L}^1_{intra} + \cdots + \mathcal{L}^H_{intra} + \mathcal{L}_{inter}) \quad (7)$$

For example, we have $H = 2$ for the sentiment transfer task, as Figure 2 depicts.

2.4 Training and Inference

Our training consists of two stages. In the first stage, we perform the training of an independent clustering module with $\mathcal{L}_{clu}$ to obtain the transfer pattern labels. In the next stage, based on the labels of transfer patterns and styles, $\mathcal{L}_{con}$ is calculated by contrastive learning module with Eq.7. We retain the loss of basic TST models denoted as $\mathcal{L}_{base}$, and our model is trained jointly with $\mathcal{L}_{base}$ and $\mathcal{L}_{con}$. We select two typical methods (Dai et al., 2019; Lee et al., 2021) as basic TST models corresponding to two mainstream groups mentioned in Section 1. A brief introduction is as follows, more details can be found in Appendix A.

RACoLN (Lee et al., 2021) belongs to the first group which separates style-independent sentence representations and revises them with style attributes. It takes an encoder, a stylizer and a decoder as the basic block. The encoder maps an input sequence $\boldsymbol{x}$ into a style-independent representation $\boldsymbol{z}_x$. The stylize takes the content representation $\boldsymbol{z}_x$ and a target style $\hat{s}$ as inputs, and produces a content-related style representation $\boldsymbol{z}_{\hat{s}}$. The decoder takes $\boldsymbol{z}_x$ and $\boldsymbol{z}_{\hat{s}}$ as inputs, and generates a new sequence $\hat{\boldsymbol{x}}_{\hat{s}}$. We regard $[\boldsymbol{z}_x, \boldsymbol{z}_{\hat{s}}]$ as $\boldsymbol{z}$ in Eq.5 to apply our method.

Style Transformer (Dai et al., 2019) belongs to the second group which directly revises an entangled representation of an input by using an extra style embedding. It takes the Transformer (Vaswani et al., 2017) as the basic block. It adds an extra style embedding to the standard Transformer architecture to conduct style control which maps style s into a style representation $\boldsymbol{e}_s$. The encoder maps an sentence $\boldsymbol{x}$ and $\boldsymbol{e}_s$ into a continuous representations $\boldsymbol{z}_{xs}$. The decoder takes $\boldsymbol{z}_{xs}$ as input and computes the output corresponding to both $\boldsymbol{x}$ and s. We apply our method to its encoder and regard $\boldsymbol{z}_{xs}$ as $\boldsymbol{z}$ in Eq.5.

We train our model with $\mathcal{L}_{con}$ and $\mathcal{L}_{base}$. The total loss of the training steps is defined as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{base} + \lambda \mathcal{L}_{con} \quad (8)$$

where λ is a balancing parameter to balance $\mathcal{L}_{con}$ with $\mathcal{L}_{base}$.

Our inference process is the same as the basic TST model due to its unmodified structure, and the clustering module is not required anymore.

3 Experimental Settings

3.1 Datasets

Following prior work on text style transfer, we use two common 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) Yelp is provided by the Yelp Dataset Challenge (Li et al., 2018) [4], consisting of restaurants and business reviews with negative and positive sentiment labels. Besides, it also provides human-annotated sentences which are used in measuring content preservation.

IMDb Movie Review Dataset (IMDb) IMDb is provided by the Style Transformer (Dai et al., 2019)[5], consisting of movie reviews with negative and positive sentiment labels written by online users. However, it does not provide human-annotated sentences.

3.2 Automatic Evaluation

We adopt transfer accuracy and content preservation to evaluate our method which are currently the

[3] Same process as $\mathcal{L}_{intra}$.

[4] https://github.com/lijuncen/Sentiment-and-Style-Transfer

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

Dataset	Yelp		IMDb	
	Positive	Negative	Positive	Negative
Train	266041	177218	178869	187597
Dev	2000	2000	2000	2000
Test	500	500	1000	1000
Avg.Len	8.9		18.5	
Vocab	10K		30K	

Table 1: Statistics of Yelp and IMDb.

most important aspects in evaluating style transfer models (Xiao et al., 2021; Huang et al., 2021).[6]

Style Transfer Accuracy To measure whether generated sentences reveal the target style property, we evaluate the target sentiment accuracy (**S-ACC**) of transferred sentences. For an accurate evaluation of style transfer accuracy, we trained two sentiment classifiers on the training set of Yelp and IMDb.

Content Preservation The Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002) can measure the similarity between two sentences at the lexical level. With this metric, one can evaluate how a sentence maintains its content throughout inference. Following the recent studies (Shuo, 2022; Lample et al., 2018), two BLEU scores are computed by the Natural Language Toolkit (NLTK) (Bird et al., 2009) in our work: **self-BLEU**, which is the BLEU score between the output and input, and **ref-BLEU**, which is the BLEU score between the output and human reference sentences.

G-score G-score denotes the geometric mean of the self-BLEU and S-ACC, which is a comprehensive metric to imply the quality of both style controlling and content preservation.

3.3 Human Evaluation

In addition to automatic evaluation, we also conduct human evaluation experiments on generated outputs. We randomly select 200 outputs (100 outputs per style) from each of the two datasets, a total number of 400 outputs per model. Given the target style and original sentence, three annotators are asked to evaluate the generated sentence with a score range from 1 (Very Bad) to 5 (Very Good) on **style controlling (S)**, **content preservation (C)**, and **fluency (F)**. we adopt the average score of three annotators eventually.

3.4 Implementation Details

We implement the cluster number calculation based on the code of U-k-means[7](Sinaga and Yang, 2020). For the Yelp dataset, the optimal cluster number is 4 for the positive style and 4 for the negative style. For the IMDb dataset, the corresponding optimal cluster numbers are 5 and 7.

The hyperparameters of the Style Transformer (Dai et al., 2019) and RACoLN (Lee et al., 2021)[8] are kept unchanged. The $\phi(\cdot)$ is Euclidean distance, d, N and τ are respectively 768, 256 and 0.5. Adam optimizer (Kingma and Ba, 2015) is used to update the parameter of the cluster module with a learning rate set to 0.0001. For balancing parameters of the new loss function, we set λ to 0.6. We train our method on one machine with 1 NVIDIA 3090 GPU. The method based on Style Transformer takes about 24 hours and the method based on RACoLN takes 6-7 hours.

4 Results and Analysis

4.1 Automatic Evaluation Result

We choose two basic TST models and some other baselines for comparison and cite all metrics from the paper of RACoLN (Lee et al., 2021). Results using the automatic evaluation are presented in Table 2. Since our method is used plug-and-play to improve the basic TST model, we pay more attention to the comparison with the basic TST model. The automatic evaluation results show the strong ability of our method to achieve style control and preserve the content information.

Our method achieves significant performance in style control. Concretely, on the Yelp dataset, CTPM increases the StyleTrans and RACoLN by **3.7** and 2.7 S-ACC score respectively. In addition, on the IMDb dataset, CTPM improves the StyleTrans and RACoLN by **7.6** and 3.4 S-ACC score.

Our method also does well in content preservation. Especially, on the Yelp dataset, CTPM raises the StyleTrans by 2.4 self-BLEU score which is the previous stat-of-the-art model and RACoLN by 2.2 self-BLEU score. Moreover, on the IMDB dataset, CTPM enhances the StyleTrans by **4.3** self-BLEU

[6]We don't adopt PPL to evaluate fluency. Recently some scholars have shown that the PPL referee is unqualified and it cannot evaluate the generated text fairly for the following reasons (Wang et al., 2022): (i) The PPL of short text is larger than long text, which goes against common sense, (ii) The repeated text span could damage the performance of PPL, and (iii) The punctuation marks could affect the performance of PPL heavily.

[7]https://github.com/kpnaga08/Unsupervised-k-means

[8]https://github.com/MovingKyu/RACoLN

Methods	Yelp				IMDb		
	S-ACC↑	ref-BLEU↑	self-BLEU↑	G-score↑	S-ACC↑	self-BLEU↑	G-score↑
Input Copy	2.1	22.8	100.0	14.5	4.4	100.0	21.0
CycleRl (Xu et al., 2018)	88.0	2.8	7.2	25.2	97.6	4.9	21.9
Deep Latent (He et al., 2019)	85.2	15.1	40.7	58.9	59.3	64.0	61.6
DIRR (Liu et al., 2021)	93.9	21.6	55.3	72.1	85.6	68.5	76.6
LEWIS (Reid and Zhong, 2021)	86.8	19.3	52.2	67.3	N/A	N/A	N/A
CRF (Shuo, 2022)	94.0	20.6	53.7	71.0	85.3	58.3	70.5
StyleTrans (Dai et al., 2019)	87.3	19.8	55.2	69.4	74.0	70.4	72.2
StyleTrans+CTPM(Ours)	**91.0**	**19.9**	**57.6**	**72.4**	**81.6**	**74.7**	**78.1**
RACoLN (Lee et al., 2021)	91.3	20.0	59.4	73.6	83.1	70.9	76.8
RACoLN+CTPM(Ours)	**94.0**	**20.8**	**61.6**	**76.1**	**86.5**	**72.2**	**77.3**

Table 2: Automatic evaluation results. The ref-BLEU for the IMDb dataset is not reported due to the absence of human references. Input Copy means an unmodified copy of the input sentence. The bold numbers indicate a better performance of baseline+CTPM than the corresponding baseline alone.

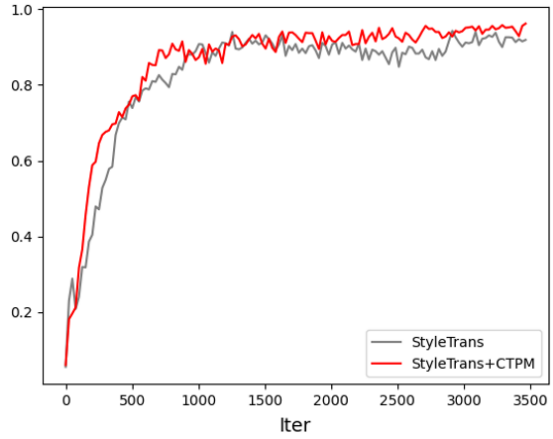

Figure 3: Visualization of S-ACC during training process on Yelp.

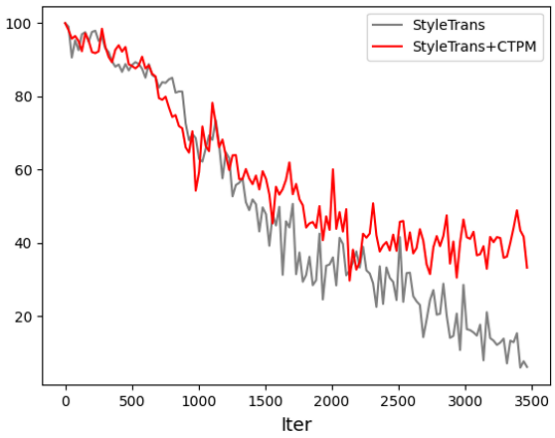

Figure 4: Visualization of self-BLEU during training process on Yelp.

score and RACoLN by 1.3 self-BLEU score. In addition, the ref-BLEU of our method is consistently higher than two basic TST methods.

During the training process, we did a visual analysis of the StyleTrans on the Yelp dataset depicted in Figure 3 and Figure 4. The figures show that with iteration steps increasing, the accuracy of our method is consistently better than the StyleTrans. In terms of content preservation, it can be observed in Table 2 that unmodified sentences will achieve content preservation best, so all methods will sacrifice the BLEU score for the performance of S-ACC in the training process. From Figure 3 and Figure 4, the self-BLEU of the StyleTrans decreases faster with no sign of convergence at a large number of iterations, which implies our ability for better content preservation.

4.2 Human Evaluation Result

Considering the limitation of manual labor involved, we conduct human rating on two of our models and the corresponding basic models. The results are shown in Table 3 and are generally consistent with the automatic evaluation. Compared with all the basic TST models, on two datasets, our method achieves higher scores on the style controlling and content preservation evaluation metric. Moreover, the fluency score also proves its ability to generate fluent outputs.

4.3 Ablation Study

In this section, we mainly validate the impact of different factors on overall performance. We further choose the Yelp dataset and the Style Transformer as the basic TST model to conduct our ablation study. The findings are still holding to the

Methods	Yelp			IMDb		
	S	C	F	S	C	F
StyleTrans	3.8	4.0	4.0	3.2	2.8	3.9
StyleTrans+CTPM	**4.0**	**4.2**	**4.1**	**3.6**	**3.4**	**4.0**
RACoLN	4.1	3.8	4.1	3.5	2.9	4.0
RACoLN+CTPM	**4.5**	**4.1**	**4.3**	**3.8**	**3.0**	**4.2**

Table 3: Human evaluation result. Each score represents the average score of three annotators.

RACoLN. Additional ablation experiments are presented in Appendix A.

Ablation of cluster numbers. We choose different cluster numbers and retrain with the same hyperparameters. The quantitative results are reported in Table 4 and simultaneously Figure 5 gives a clear visualization. We set the positive and negative cluster numbers equal to our optimal value. Table 4 and Figure 5 show that CTPM can always achieve better results than StyleTrans with the cluster number in an appropriate range.

Ablation of loss components. In order to explore the impact of each contrastive loss, we have considered an appropriate setting: only inter-style contrastive loss or intra-style contrastive loss, and retrain our method with the same hyperparameters. The results are shown in Table 5. It can be observed that removing any component will be worse than ours and better than StyleTrans, which tells us both two losses are effective.

Ablation of λ. We selected different values of λ and retrained the model with the same hyperparameters. The quantitative results are reported in Table 6. The results show that λ is insensitive and can bring stable improvement $(1.4 \sim 2.0)$ of G-score within a certain range $(0.4 \sim 0.8)$. By the way, the training procedure will encounter the difficulty of gradient explosion with λ (>0.9).

Ablation of batch size. The contrastive loss is closely linked to the size of the batch size, we also did an ablation experiment on the batch size as shown in Table 7. The batch size can bring stable improvement $(1.2 \sim 3.0)$ of G-score within a certain range $(128 \sim 512)$. We infer that this is because the small batch size will cause poor contrastive learning performance, while the large one is inappropriate for our small-scale datasets.

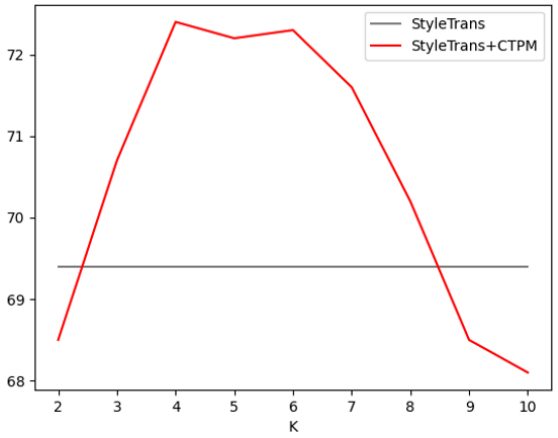

Figure 5: G-score of StyleTrans+CTPM with different cluster numbers on Yelp. The optimal result is obtained when $k = 4$.

Method	S-ACC↑	*ref*-BLEU↑	*self*-BLEU↑	G-score↑
StyleTrans	87.3	19.8	55.2	69.4
StyleTrans+CTPM				
$k_1 = k_2 = 2$	88.9	19.0	52.8	68.5
$k_1 = k_2 = 3$	88.9	19.4	56.3	70.7
$k_1 = k_2 = 4$	91.0	19.9	57.6	72.4
$k_1 = k_2 = 5$	88.0	20.1	59.3	72.2
$k_1 = k_2 = 6$	90.0	20.0	58.1	72.3
$k_1 = k_2 = 7$	87.9	19.6	58.3	71.6
$k_1 = k_2 = 8$	91.0	19.9	54.1	70.2
$k_1 = k_2 = 9$	87.9	18.9	53.4	68.5
$k_1 = k_2 = 10$	88.1	18.5	52.6	68.1

Table 4: Performance of StyleTrans+CTPM with different cluster numbers on Yelp. The optimal result is obtained when $k_1 = k_2 = 4$.

4.4 Case study

In terms of the StyleTrans and StyleTrans+CTPM, we randomly sampled 20000 sentences (10000 sentences per style) from Yelp and projected them in a two-dimensional space using t-SNE (van der Maaten and Hinton, 2008) as shown in Figure 6. It can be observed that for StyleTrans, although two colors that imply two styles are roughly separated, there is still a partial intersection between the two styles. However, after adding our method, there is no obvious intersection between the two styles, and each style is generally divided into four distinctive clusters, corresponding to four transfer patterns. Therefore, our method is able to obtain more meaningful and distinguishable sentence representations, and thereby benefits the TST task.

We further randomly selected some instances from the four clusters presented in Table 8, in which each cluster shows its own different transfer pattern, and there exists a clear distinction be-

Method	S-ACC↑	*ref*-BLEU↑	*self*-BLEU↑	G-score↑
StyleTrans	87.3	19.8	55.2	69.4
StyleTrans+CTPM	91.0	19.9	57.6	72.4
(-)$\mathcal{L}_{intra}$	88.4	20.1	56.0	70.4
(-)$\mathcal{L}_{inter}$	88.3	19.9	58.7	72.0

Table 5: Performance of StyleTrans+CTPM with different loss components on Yelp, where (-) indicates removing the corresponding component from the full model.

Method	S-ACC↑	*ref*-BLEU↑	*self*-BLEU↑	G-score↑
StyleTrans	87.3	19.8	55.2	69.4
StyleTrans+CTPM				
$\lambda = 0.2$	87.6	18.7	54.7	69.2
$\lambda = 0.4$	90.4	19.6	56.9	71.7
$\lambda = 0.6$	91.0	19.9	57.6	72.4
$\lambda = 0.8$	87.2	19.4	57.5	70.8

Table 6: Performance of StyleTrans+CTPM with different λ on Yelp.

tween different transfer patterns. Note that our transfer patterns are automatically learned in the latent space, and may not necessarily have a strict one-to-one correspondence with actual patterns.

5 Related Work

Text Style Transfer In recent years, style transfer has been widely explored in Computer Vision filed (Zhu et al., 2017) but remained challenging for text because of the vague style definition of language and the discrete nature. Most approaches focus on unsupervised methods owing to the difficulty of obtaining parallel data. Previous work can mainly be categorized into two families.

The text style transfer task considers a sentence as being formed of content and style. Therefore, the first family attempts to separate content and style attributes (Rao and Tetreault, 2018; Li et al., 2018; Wu et al., 2019; Malmi et al., 2020; Lee et al., 2021; Reid and Zhong, 2021). RACoLN (Lee et al., 2021) proposes a method to implicitly remove style at the token level using reverse attention and then fuse content information to style representation using conditional layer normalization. (Rao and Tetreault, 2018) perform the task of politeness transfer by first identifying words with stylistic attributes using TF-IDF and then training a model to replace or augment these stylistic words with ones associated with the target attribute.

Apart from distangling content and style attributes, the second family focuses on revising an entangled representation of input (Dai et al., 2019; Liu et al., 2020; Huang et al., 2020; Kashyap et al., 2022). ARAE (Kashyap et al., 2022) introduces two cooperative losses to the adversarially regularized autoencoder that further regularizes the latent space to maintain other desirable constraints while retaining content and changing the style of sentences. Style Transformer(Dai et al., 2019) uses the transformer architecture and rewrite style in the entangled representation at the decoder which we consider as a strong baseline model.

Method	S-ACC↑	*ref*-BLEU↑	*self*-BLEU↑	G-score↑
StyleTrans	87.3	19.8	55.2	69.4
StyleTrans+CTPM				
batch size = 64	87.2	18.7	51.3	66.9
batch size = 128	90.1	20.2	56.0	71.0
batch size = 256	91.0	19.9	57.6	72.4
batch size = 512	88.5	19.6	56.3	70.6
batch size = 1024	90.4	17.7	53.0	69.2

Table 7: Performance of StyleTrans+CTPM with different batch sizes on Yelp.

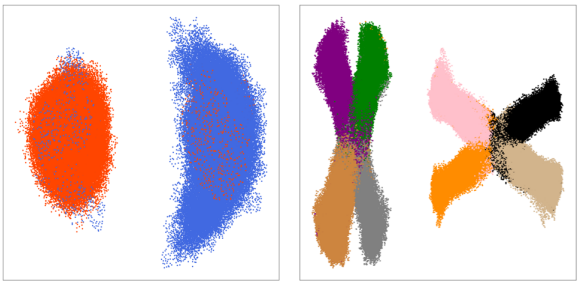

Figure 6: Visualization of Yelp dataset representations using t-SNE. The left figure depicts the sentence representations of StyleTrans, in which red dots represent sentences with a positive style, while blue dots denote sentences with a negative style. The right depicts the sentence representations of StyleTrans+CTPM, in which the left four colors imply four transfer patterns corresponding to the positive style sentences and the right four colors imply four transfer patterns corresponding to the negative style sentences.

Contrastive Learning In NLP, contrastive learning has made remarkable achievements at many tasks. For Sentence Embedding, NonLing-CSE (Jian et al., 2022b) clusters examples from a non-linguistic domain with a similar contrastive loss to obtain higher quality sentence embeddings. For Text Classification, DualCL (Chen et al., 2022) regards the parameters of the classifiers as augmented samples associating with different labels and then exploits the contrastive learning between the input samples and the augmented samples as to improve the classification accuracy. For Text Sum-

Cluster	Input Sentence	Output Sentence
1	oh i got my band geek back on now !	oh i don't got my band geek back on now !
	i love their star design collection .	i don't love their star design collection .
	i love their fresh juices as well .	i don't love their fresh juices either .
2	good drinks , and good company .	disappointing drinks , and disappointing company .
	they have delicious soups everyday .	they have awful soups everyday .
	the service has always been wonderful .	the service has always been bad .
3	definitely a place to keep in mind .	not a place to keep in mind at all .
	enjoyed the dolly a lot .	not enjoyed the dolly at all .
	well worth searching out this gem .	not worth searching out this gem at all .
4	steve was professional and found exactly the right unit to fit in our space .	steve was disorganized and not found exactly the like unit to fit in our space .
	i spent time with my best buds and excellent wine and food .	i waste time with my worst buds and not excellent wine and food at all .
	the outside seating is too packed, and happy hour never happens .	i love the outside seating and sad hour never happens .

Table 8: Randomly sampled cases from the four different clusters. Input sentences are with a positive style in Yelp, and output sentences are transferred from input sentences by our method. Sentences in cluster 1 are transferred through negative auxiliary verbs such as "don't". Sentences in cluster 2 are likely transferred by replacing their adjectives. Sentences in cluster 3 are usually verb phrases and can be modified by adding negation structures. Sentences in cluster 4 are compound sentences, which need to be modified in several places and even the structure may be changed.

marization, SeqCon (Xu et al., 2022) proposes a contrastive learning model to maximize the similarities between the gold summary and model-generated summaries for supervised abstractive text summarization. Existing works also have shown it to be beneficial to Machine Translation (Vamvas and Sennrich, 2021; Pan et al., 2021), Data Augmentation (Margatina et al., 2021; Qu et al., 2020), and Few-shot Learning (Das et al., 2022; Luo et al., 2021; Jian et al., 2022a) as well.

6 Conclusion

In this paper, we propose a novel approach named Contrastive Transfer Pattern Mining (CTPM) for text style transfer (TST) tasks. Different from the previous methods that mainly consider the differences between styles, we mine and exploit the latent transfer pattern in each style. Specifically, we design an adaptive clustering module to mine and exploit the latent transfer patterns, and then introduce a contrastive learning module, including an inter-style contrastive loss and an intra-style contrastive loss, to obtain more meaningful and distinguishable sentence representations, which could improve the performance of TST. Our approach does not depend on a specific network structure and can be widely applied to basic TST models. Experiments with two mainstream basic TST models and two widely used benchmark datasets have shown the effectiveness and generality of our proposed approach.

Limitations

Since methods based on pre-trained language models on text style transfer requires larger GPU resources and are not mainstream methods, we have not yet tested the effectiveness of our method on pre-trained language models. Moreover, since there is no multiple-attribute dataset in existing research, the applicability of our method on multiple-attribute TST tasks has also not been verified.

Ethics Statement

Text style transfer task is widely used in the field of controllable text generation. However, because of the diversified corpus of style, the model has the potential to be both used for good and used with malicious intent. For example, if one intentionally changes the style (news) in the news field, fake news may be generated. Moreover, in the realm of politics, the transformation can give rise to fabricated political statements, thereby engendering a climate of misinformation and deceit.

We hired human annotators to evaluate our method and two basic TST models. Here we show the details of the employed annotators. The annotators were asked to score the model-generated sentence. Considering the difference between the two datasets, annotators will get $0.1 for each sentence in the Yelp dataset and $0.2 for each sentence in the IMDb dataset. There are 1600 sentences evaluated (800 sentences per dataset), so each annotator was rewarded $240 in total.

Acknowledgements

We would like to express our sincere gratitude to the professional reviewers for their suggestions and comments. This work is supported by the National Key Research and Development Program of China under Grant 2020YFB1406603, the National Science Fund for Excellent Young Scholars under Grant 62222212 and the National Natural Science Foundation of China under Grant 61876223.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. " O'Reilly Media, Inc.".

Q. Chen, R. Zhang, Y. Zheng, and Y. Mao. 2022. Dual contrastive learning: Text classification via label-aware data augmentation. *arXiv preprint arXiv:2201.08702*.

Nogueira Dos Santos Cicero, I. Melnyk, and I. 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 (Volume 2: Short Papers)*, pages 189–194. arXiv.

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

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2022. Container: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

J. He, X. Wang, G. Neubig, and T. Berg-Kirkpatrick. 2019. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Fei Huang, Zikai Chen, Chen Henry Wu, Qihan Guo, Xiaoyan Zhu, and Minlie Huang. 2021. Nast: A non-autoregressive generator with word alignment for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1577–1590.

Y. Huang, W. Zhu, D. Xiong, Y. Zhang, C. Hu, and F. Xu. 2020. Cycle-consistent adversarial autoencoders for unsupervised text style transfer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2213–2223.

Y. Jian, C. Gao, and S. Vosoughi. 2022a. Contrastive learning for prompt-based few-shot language learners. *arXiv preprint arXiv:2205.01308*.

Yiren Jian, Chongyang Gao, and Soroush Vosoughi. 2022b. Non-linguistic supervision for contrastive learning of sentence embeddings. *arXiv preprint arXiv:2209.09433*.

A. R. Kashyap, D. Hazarika, M. Y. Kan, R. Zimmermann, and S. Poria. 2022. So different yet so alike! constrained unsupervised text style transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 416–431.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

D. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

D. Lee, Z. Tian, L. Xue, and N. L. Zhang. 2021. Enhancing content preservation in text style transfer using reverse attention and conditional layer normalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 93–102.

J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan. 2016. A persona-based neural conversation model. In *ACL (1)*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *NAACL-HLT*.

D. Liu, J. Fu, Y. Zhang, C. Pal, and J. Lv. 2020. Revision in continuous space: Unsupervised text style transfer without adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8376–8383.

Yixin Liu, Graham Neubig, and John Wieting. 2021. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4262–4273.

R. Luo, G. Huang, and X. Quan. 2021. Bi-granularity contrastive learning for post-training in few-shot scene. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1733–1742.

J MacQueen. 1967. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297. University of California Los Angeles LA USA.

E. Malmi, A. Severyn, and S. Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8671–8680.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258.

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

Y. Qu, D. Shen, Y. Shen, S. Sajeev, J. Han, and W. Chen. 2020. Coda: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. In *International Conference on Learning Representations*.

S. Rao and J. Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140.

Machel Reid and Victor Zhong. 2021. Lewis: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3932–3944.

Yang Shuo. 2022. Tagging without rewriting: A probabilistic model for unpaired sentiment and style transfer. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 293–303.

Kristina P. Sinaga and Miin-Shen Yang. 2020. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727.

Jannis Vamvas and Rico Sennrich. 2021. Contrastive conditioning for assessing disambiguation in mt: A case study of distilled bias. In *2021 Conference on Empirical Methods in Natural Language Processing*, pages 10246–10265. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yequan Wang, Jiawen Deng, Aixin Sun, and Xuying Meng. 2022. Perplexity from plm is unreliable for evaluating text quality. *arXiv preprint arXiv:2210.05892*.

X. Wu, T. Zhang, L. Zang, J. Han, and S. Hu. 2019. "mask and infill" : Applying masked language model to sentiment transfer. *arXiv preprint arXiv:1908.08039*.

F. Xiao, L. Pang, Y. Lan, Y. Wang, H. Shen, and X. Cheng. 2021. Transductive learning for unsupervised text style transfer. pages 2510–2521.

J. Xu, X. Sun, Q. Zeng, X. Ren, X. Zhang, H. Wang, and W. Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2022. Sequence level contrastive learning for text summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11556–11565.

Z. Xu, T. Chalasani, K. Ghosal, S. Lutz, and A. Smolic. 2019. Stada: Style transfer as data augmentation. pages arXiv–1909.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

A Example Appendix

A.1 Style Transformer

Style Transformer (Dai et al., 2019) takes Transformer (Vaswani et al., 2017) as the basic block. Their base loss consists of three losses.

Self reconstruction loss Considering the input sentence x and the input style s from the same dataset, they train the Style Transformer to reconstruct the input sentence by minimizing negative log-likelihood:

$$\mathcal{L}_{\text{self}}(\theta) = -p_\theta(y = x \mid x, s) \tag{9}$$

Cycle reconstruction loss In order to preserve the information in the input sentence x, they feed the generated sentence $\hat{y} = f_\theta(x, \hat{s})$ to the Style Transformer with the style of x and train the Style Transformer to reconstruct original input sentence by minimizing negative log-likelihood:

$$\mathcal{L}_{\text{cycle}}(\theta) = -p_\theta\left(y = x \mid f_\theta(x, \widehat{s}), \mathbf{s}\right) \tag{10}$$

Style Controlling loss The Style Transformer is also trained to minimize the negative log-likelihood of the corresponding class of style $\widehat{s}$ with a trained discriminator φ:

$$\mathcal{L}_{\text{style}}(\theta) = -p_\varphi\left(c = \widehat{s} \mid f_\theta(x, \widehat{s})\right) \tag{11}$$

Style Transformer + CTPM We apply our method to the encoder of the Style Transformer. Concretely, we regard the encoder output as z in Eq.5. Based on the supervised information acquired in 2.2 and z, we obtain the $\mathcal{L}_{\text{con}}$ with algorithm in 2.3. Finally the base loss is:

$$\mathcal{L}_{\text{base}} = \lambda_1\mathcal{L}_{\text{self}} + \lambda_2\mathcal{L}_{\text{cycle}} + \lambda_3\mathcal{L}_{\text{style}} \tag{12}$$

where λ_1, λ_2, λ_3 are 0.25, 0.5, 1, as same as the original paper. Finally the updated new loss can be obtained by Eq.8.

A.2 RACoLN

RACoLN (Lee et al., 2021) consists of an encoder, a stylizer and a decoder. The encoder maps an input sequence x into a style-independent representation z_x. The stylize takes the content representation z_x and a target style $\hat{s}$ as inputs, and produces a content-related style representation $z_{\hat{s}}$. Finally, the decoder takes the content representation z_x and style representation $z_{\hat{s}}$ as inputs, and generates a new sequence $\hat{x}_{\hat{s}}$. RACoLN also has the first three losses as Style Transformer. Besides, RACoLN propose an extra content loss.

Content loss They first obtain a content representation z_x of the input x and a content representation $z_{\hat{x}_{\hat{s}}}$ of the transferred sequence $\hat{x}_{\hat{s}}$ through encoder. The two content representations should be similar, hence the content loss is:

$$\mathcal{L}_{\text{content}} = \mathbb{E}_{(x,s)\sim\mathcal{D}} \left\| z_x - z_{\hat{x}_{\hat{s}}} \right\|_2^2 \tag{13}$$

RACoLN + CTPM We apply our method to the stylizer of the RACoLN. Concretely, we regard the stylizer output $[z_x, z_{\hat{x}_{\hat{s}}}]$ as z in Eq.5. Based on the supervised information acquired in 2.2 and z, we obtain the $\mathcal{L}_{\text{con}}$ with algorithm in 2.3. Finally the base loss is:

$$\mathcal{L}_{\text{base}} = \lambda_1\mathcal{L}_{\text{self}} + \lambda_2\mathcal{L}_{\text{cycle}} + \lambda_3\mathcal{L}_{\text{style}} + \lambda_4\mathcal{L}_{\text{content}} \tag{14}$$

where λ_1, λ_2, λ_3, λ_4 are 0.5, 0.5, 1, 1, as same as the original paper. The updated new loss can be obtained by Eq.8.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3

☑ B1. Did you cite the creators of artifacts you used?
Section 3

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We use dataset and code which are open-source on github.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We use the artifacts consistent with their intended use.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
There is no such information that names or uniquely identifies individual people or offensive content.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
If our work are received, we will report these on github with our source code.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Section 3

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
We gave an oral explanation.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section Ethics Statement

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
We gave an oral explanation.

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Section Ethics Statement

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
The annotators are all from China.