Text Style Transfer via Learning Style Instance Supported Latent Space

Xiaoyuan Yi[1,2,3], **Zhenghao Liu**[1,2,3], **Wenhao Li**[1] and **Maosong Sun**[1,2,4*]
[1]Department of Computer Science and Technology, Tsinghua University
[2]Institute for Artificial Intelligence, Tsinghua University
[3]State Key Lab on Intelligent Technology and Systems, Tsinghua University
[4]Jiangsu Collaborative Innovation Center for Language Ability, Jiangsu Normal University
{yi-xy16, liu-zh16, liwh16}@mails.tsinghua.edu.cn, sms@tsinghua.edu.cn

Abstract

Text style transfer pursues altering the style of a sentence while remaining its main content unchanged. Due to the lack of parallel corpora, most recent work focuses on unsupervised methods and has achieved noticeable progress. Nonetheless, the intractability of completely disentangling content from style for text leads to a contradiction of content preservation and style transfer accuracy. To address this problem, we propose a style instance supported method, *StyIns*. Instead of representing styles with embeddings or latent variables learned from single sentences, our model leverages the *generative flow* technique to extract underlying stylistic properties from multiple instances of each style, which form a more discriminative and expressive latent style space. By combining such a space with the attention-based structure, our model can better maintain the content and simultaneously achieve high transfer accuracy. Furthermore, the proposed method can be flexibly extended to semi-supervised learning so as to utilize available limited paired data. Experiments on three transfer tasks, sentiment modification, formality rephrasing, and poeticness generation, show that StyIns obtains a better balance between content and style, outperforming several recent baselines.

1 Introduction

Text style transfer aims to endow a sentence with a different style and meanwhile keep its main semantic content unaltered, which could benefit various downstream applications such as text polishing [Rao and Tetreault, 2018], poetic writing [Yi *et al.*, 2018] and dialog generation [Zhou *et al.*, 2018].

Owing to the lack of large parallel corpora, recent work mainly pays attention to unsupervised transfer and generally achieves this goal by fusing content and representations of target styles. However, both literary theory [Embler, 1967] and machine learning study [Lample *et al.*, 2019] manifest that style is coupled with content to some degree, causing a *contradiction* of content preservation and style accuracy.

In this work, we mainly study two research paradigms of text style transfer and discuss their influence on performance. One paradigm is a typical *disentanglement approach*, which explicitly strips style from content, and then incorporates a separated target-style representation [Shen *et al.*, 2017; Fu *et al.*, 2018; John *et al.*, 2019]. Because of the intractability of disentanglement, specifying a target style often brings some unexpected content attached to the source style. As a result, this approach usually obtains high style transfer accuracy but fails to preserve full source content.

To improve content preservation, another paradigm adopts attention-based structures [Bahdanau *et al.*, 2015; Vaswani *et al.*, 2017] to maintain all word-level source information. Instead of disentangling, this method forces the model to focus on style-independent words by cycle reconstruction and uses style embeddings to encourage the fusion of style-related phrases [Lample *et al.*, 2019; Dai *et al.*, 2019]. Nevertheless, for text, style is a highly complex concept involving various linguistic features and individualities [Crystal, 1970]. It is hard to learn expressive and flexible style embeddings to represent such a concept. Consequently, these models tend to overemphasize content preservation and evade the difficult of embedding learning, incurring unsatisfactory style accuracy.

Related linguistic research demonstrates that stylistic syndromes can be better observed in multiple instances by making broader comparisons [Ide, 2004]. Inspired by this idea, we propose a style instance supported method, called *StyIns*, to alleviate the contradiction mentioned above. When transferring each sentence, StyIns adopts the attention mechanism to preserve complete source information. Then instead of using simple style embeddings, our model incorporates a set of instances sharing the same style, and learns to extract underlying stylistic properties with the powerful *generative flow* technique [Rezende and Mohamed, 2015] to form a more discriminative latent space. Samples drawn from this space are fed to the decoder to strengthen style signals, yielding a better balance between content preservation and style accuracy. Besides, StyIns can be extended to a semi-supervised version to utilize limited parallel data for further improvement.

In summary, our contributions are as follows:

- We propose a style instance supported method to learn a more discriminative and expressive latent space, which enhances style signals and makes a better balance between style transfer accuracy and content preservation.

* Corresponding author

- Our model can flexibly switch to a semi-supervision manner to take advantage of limited parallel data, without extra parameters or structures.
- On three text transfer tasks, sentiment, formality and poeticness, both automatic and human evaluations demonstrate that our model achieves better general performance, against several recent baselines[1].

2 Related Work

Style transfer has been widely explored in Computer Vision (CV) filed [Zhu *et al.*, 2017] but remained challenging for text due to the discrete nature and vague style definition of language. Without sufficient parallel text data, recent research interests mainly concentrate on unsupervised transfer methods. According to the way of representing content and style, we can categorize most existing models into four paradigms.

The first paradigm explicitly disentangles text as separated content and style representations, respectively, then combines the content with a target style to achieve transfer. Shen et al. [2017] take a pair of adversarial discriminators to align the source and transferred content distributions. Fu et al. [2018] concatenate the extracted content with a learned target-style embedding. These methods are also improved by utilizing locally-normalized language models as discriminators [Yang *et al.*, 2018]. More recently, John et al. [2019] design multiple losses to pack a sentence into a latent space, which is then split into sub-spaces of content and style. Since complete disentanglement is impracticable, this paradigm usually results in satisfactory style accuracy but poor content preservation.

The second paradigm takes multi-generator structures and generates sentences in each style with a corresponding generator. Namely, each style is implicitly represented as the generator parameters. Fu et al. [2018] make an attempt on this paradigm and adversarially train one encoder to disentangle content. Prabhumoye et al. [2018] utilize a back-translation technique to translate a sentence to another language to corrupt its stylistic properties. Then adversarially trained decoders are used to create transferred sentences in the original language. Based on Reinforcement Learning (RL), Gong et al. [2019] represent each transfer direction between two styles as one encoder-decoder model, and Luo et al. [2019] pair the two transfer directions with a dual learning schema for further improvement. Generally, this paradigm is effective but also resource-consuming since each style or transfer direction needs to be modelled by a separated generator.

The third paradigm is a locate-and-replace schema, which locates style-dependent words and then replaces them with the target-style ones. We can consider the content and style to be represented as corresponding sets of words. Li et al. [2018] design a delete-and-retrieve method to combine content words in a source sentence with stylistic words in a retrieved semantically similar sentence. Wu et al. [2019b] mask all sentimental tokens in a source sentence, then use a pre-trained BERT [Devlin *et al.*, 2019] to infill target-sentiment ones. Wu et al. [2019a] take a hierarchical RL method, which uses two agents to locate style-related words and alter the sentence, respectively. To sum up, this paradigm is more accurate since it maintains all word-level information, but the lack of stylistic vocabularies limits its locating performance. Moreover, it doesn't apply to the scenarios that styles are expressed beyond word level, *e.g.*, poeticness.

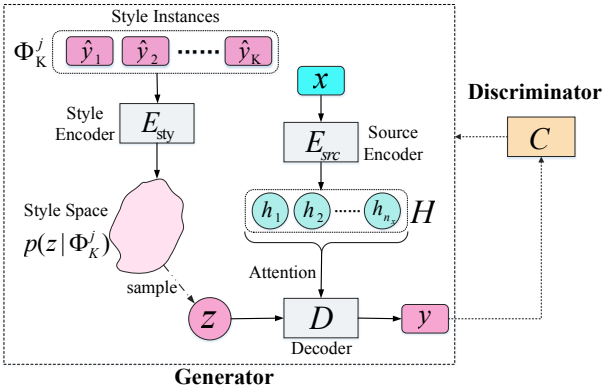


Figure 1: A graphical illustration of the proposed model.

The last paradigm adopts one single attention-based encoder-decoder model [Bahdanau *et al.*, 2015] and feeds a style embedding to the decoder to provide style signals, without explicit disentanglement. Lample et al. [2019] take this paradigm and try to control multiple attributes of text with an attention-based LSTM model. Instead of LSTM, Dai et al. [2019] use the more powerful Transformer [Vaswani *et al.*, 2017]. Such a design helps better preserve source information, avoids structural redundancy of paradigm 2, and could cover broader cases compared to paradigm 3. Nevertheless, since the learned embedding is not expressive enough to model the highly complex concept of style, this paradigm usually causes unsatisfactory style transfer accuracy.

In addition, Shang et al. [2019] devise a semi-supervised method that projects the latent spaces of different styles. Despite achieving further improvement, this model is sensitive to parallel data size and not suitable for unsupervised cases.

Absorbing advantages of paradigm 1 & 4, StyIns learns a more discriminative latent style space to better balance style and content, and it could flexibly switch to a semi-supervised version, compatible with a broader range of scenarios.

3 Methodology

3.1 Formalization and Overview

We first formalize the unsupervised text style transfer task. Suppose there are M datasets $\{\mathcal{D}_i\}_{i=1}^{M}$, and sentences in $\mathcal{D}_i$ share the same style s_i. Given an arbitrary sentence x with a source style s_i, our goal is to rephrase x to a new one y with a target style $s_j (j \neq i)$ while preserving its main content.

As discussed before, to produce strong style signals we provide a set of sentences, $\Phi_K^j = \{\hat{y}_k\}_{k=1}^{K} \subset \mathcal{D}_j$, called *style instances*, to represent an empirical distribution of style s_j, which helps the model better learn underlying stylistic properties. For this sake, we incorporate a latent variable z constructed by these instances to represent the complex concept of style. Since sentences of the same style are conditionally

[1] Our source code is available at github.com/XiaoyuanYi/StyIns.

Algorithm 1 Training Process

for number of iterations **do**
  Sample a source style s_i and a target style s_j;
  Sample instances Φ_K^i from $\mathcal{D}_i$ and Φ_K^j from $\mathcal{D}_j$;
  Sample x from $\mathcal{D}_i$, $x \notin \Phi_K^i$;
  Accumulate $\mathcal{L}_{recon}$, $\mathcal{L}_{cycle}$, $\mathcal{L}_{style}$;
  if y^* exits **then**
    Accumulate $\mathcal{L}_{super}$;
  end if
  Update the parameters of G;
  for n_C steps **do**
    Update the parameters of C with $\mathcal{L}_C$;
  end for
end for

independent *w.r.t.* z, we can derive a new parametric form of text style transfer:

$$\begin{aligned} p(y|x,\Phi_K^j) &= \int p(y,z|x,\Phi_K^j)dz \\ &= \int p(y|x,z) * p(z|\Phi_K^j)dz \\ &= \mathbb{E}_{z\sim p(z|\Phi_K^j)}[p(y|x,z)]. \end{aligned} \tag{1}$$

Eq. (1) differs from previous work [Shen *et al.*, 2017; Yang *et al.*, 2018] and suggests the architecture of our StyIns, as presented in Figure 1.

Define $E_{src}(x)$ as a bidirectional LSTM encoder, called *source encoder*, $E_{sty}(\Phi_K^j)$ as a *style encoder* to model the distribution $p(z|\Phi_K^j)$, and $D(H,z)$ as a decoder with the attention mechanism [Bahdanau *et al.*, 2015]. The source encoder maps a given sentence x to a sequence of hidden states H. Then the decoder generates a transferred sentence y with H and z as inputs, where z is sampled from $p(z|\Phi_K^j)$. These three components together form our generator $G(x,\Phi_K^j)$.

3.2 Learning Latent Style Space

The style encoder $E_{sty}(\Phi_K^j)$ takes style instances as inputs, constructs a latent style space, and then outputs a sampled style representation z for the decoder to guide stylistic generation. Previous work [John *et al.*, 2019] usually adopts the Variational Auto-Encoder (VAE) [Kingma and Welling, 2014] to build latent spaces. On the basis of the mean-field approximation, VAE assumes the independence of sentences and allocates each a corresponding isotropic Gaussian latent space. Despite the tractability of computation, this approach is implausible. For one thing, the dimension-independent Gaussian distribution is not expressive enough, which has been explored in various work [Atanov *et al.*, 2019]. For another, sentences with the same style are not dependent but connected by sharing one global style space.

Generative Flow

To avert these problems, we resort to the *generative flow* (GF) [Rezende and Mohamed, 2015], a potent technique to construct sophisticated distributions. Put simply, GF maps a simple initial latent variable z_0 to a complex one z_T by applying a chain of parameterized mapping functions f_t:

$$z_t = f_t(z_{t-1},c), z_0 \sim p(z_0|c), t \in \{1,2,\ldots,T\}, \tag{2}$$

where c is a given condition and T is the length of the chain. GF requires each f_t to be invertible and its Jacobian determinant to be computable. Then we can get the probability density of the final distribution by:

$$\log p(z_T|c) = \log p(z_0|c) - \sum_{t=1}^{T} \log\det\left|\frac{dz_t}{dz_{t-1}}\right|. \tag{3}$$

Various choices of f_t have been proposed these years. We use a simple but effective one here, the Inverse Autoregressive Flow (IAF) [Kingma *et al.*, 2016]. More concretely, we have:

$$[m_t, o_t] \leftarrow g_t(z_{t-1},c), \sigma_t = sigmoid(o_t), \tag{4}$$

$$z_t = \sigma_t \odot z_{t-1} + (1-\sigma_t) \odot m_t, \tag{5}$$

where $\odot$ is element-wise multiplication. g_t is an autoregressive network, in which the i-th element of output vectors is calculated with the first $i-1$ elements of z_{t-1}. We use the structure proposed in [Germain *et al.*, 2015] as g_t.

Style Instance Supported Latent Space

As mentioned in Sec. 3.1, to construct a more expressive latent space, we discard the mean-filed assumption by utilizing K style instances $\Phi_K^j = \{\hat{y}_k\}_{k=1}^K$ rather than only one single sentence. In detail, we feed each $\hat{y}_k$ to another bidirectional LSTM, and represent it as v_k, the concatenated final hidden state. Then we assume the initial latent variable z_0 in Eq. (2) follows the isotropic Gaussian distribution:

$$z_0 \sim p(z_0|\Phi_K^j) = \mathcal{N}(\mu_0, \sigma_0^2 \boldsymbol{I}), \tag{6}$$

$$\mu_0 \approx \frac{1}{K}\sum_{k=1}^{K} v_k, \sigma_0^2 \approx \frac{1}{K-1}\sum_{k=1}^{K}(v_k-\mu_0)^2, \tag{7}$$

$$c = MLP(\mu_0), \tag{8}$$

where the mean of z_0 is approximated by Maximum Likelihood Estimation and we use the unbiased estimator for variance. c is a global representation of Φ_K^j which is computed by a Multi-Layer Perceptron (MLP) and used in Eq. (4).

With the modules introduced above, we can get an output z of the style encoder $E_{sty}(\Phi_K^j)$ by sampling z_0 with Eq. (6) and mapping it with Eq. (2). Then the sampled z is concatenated with the embedded word and fed to the decoder at each time step. We will show that such a learned latent space is highly discriminative in Sec. 4.

3.3 Unsupervised Training

Given a source sentence x, two sets of style instances, Φ_K^i ($x \notin \Phi_K^i$) and Φ_K^j, with the source style s_i and the target style s_j , we adopt the following losses to create indirect signals.

Reconstruction Loss. This loss is used by different paradigms of model [Shen *et al.*, 2017; Luo *et al.*, 2019; Wu *et al.*, 2019b], which requires the model to reconstruct the given sentence with source-style signals:

$$\mathcal{L}_{recon} = -\log p_G(x|x,\Phi_K^i). \tag{9}$$

Cycle Consistency Loss. Cycle consistency is first applied to image style transfer [Zhu *et al.*, 2017] to strengthen content preservation and then also adopted for text [Dai *et al.*, 2019; Lample *et al.*, 2019]. We transfer a source sentence in two directions with the support of instances in different styles:

$$\mathcal{L}_{cycle} = -\log p_G(x|y, \Phi_K^i), \quad y \leftarrow G(x, \Phi_K^j). \tag{10}$$

Note that in each iteration, we provide different sampled instances to help StyIns better generalize stylistic properties.

Adversarial Style Loss. Without any parallel corpus, adversarial training [Goodfellow *et al.*, 2014] is utilized to build style supervision. Following [Dai *et al.*, 2019], we use a classifier with M+1 classes as the discriminator C to tell the style of an input sentence (M+1-th class indicates a generated fake). The generator is expected to fool the discriminator by:

$$\mathcal{L}_{style} = -\log p_C(j|y), \tag{11}$$

and the discriminator is alternately optimized by:

$$\mathcal{L}_C = -[\log p_C(i|x) + \log p_C(i|\hat{x}) + \log p_C(M+1|y)], \tag{12}$$

where $\hat{x} \leftarrow G(x, \Phi_K^i)$.

3.4 Semi-Supervised Training

Our method can be regarded as the construction of target-style information with the support of style instances. When the ground truth $y^* \notin \Phi_K^j$ is available, we can create supervision by maximizing $\log p(y^*|x, \Phi_K^j)$. We drive a lower bound as:

$$\begin{aligned} \log p(y^*|x, \Phi_K^j) \geq\ & \mathbb{E}_{q(z|y^*, \Phi_K^j)}[\log p(y^*|z, x)] \\ & - KL[q(z|y^*, \Phi_K^j)||p(z|\Phi_K^j)]. \end{aligned} \tag{13}$$

Based on this lower bound, we get the final supervised loss:

$$\begin{aligned} \mathcal{L}_{super} = & -\alpha * \mathbb{E}_{q(z|y^*, \Phi_K^j)}[\log p(y^*|z, x) + \log p(z|\Phi_K^j) \\ & - \log q(z|y^*, \Phi_K^j)] + \beta * \mathbb{E}_{q(z|\Phi_K^j)}[-\log p(y^*|z, x)], \end{aligned} \tag{14}$$

where α and β are hyper parameters to re-scale the loss.

By optimizing Eq. (14), we simultaneously maximize a lower bound of $\log p(y^*|x, \Phi_K^j)$ and minimize an upper bound of $-\log p(y^*|x, \Phi_K^j)$ (the second term in Eq. (14)). The KL term, which is approximately estimated with Eq. (3), could benefit alignment of latent style distributions learned with/without ground truth, and thus helps better extract common stylistic properties. We describe the complete training process in Algorithm 1.

4 Experiments

4.1 Datasets

We conduct experiments on three text style transfer tasks.

Sentiment Modification. We use the **Yelp** dataset processed by [Li *et al.*, 2018], which consists of restaurant reviews with two sentiments, namely *negative* and *positive*.

Dataset	Styles	Paired			Unpaired	
		Train	Valid	Test	Train	Valid
Yelp	Neg.	N/A	N/A	500	180k	2,000
	Pos.			500	270k	2,000
GYAFC	Inf.	52k	2,788	1,332	N/A	N/A
	For.	52k	2,247	1,019		
CPVT	Ver.	4k	1,000	2,000	200k	10k
	Poe.	4k	1,000	2,000	200k	10k

Table 1: Data Statistics.

Formality Rephrasing. The recently released dataset **GYAFC** [Rao and Tetreault, 2018] contains paired *formal* and *informal* sentences in two domains. We use the Family & Relationships domain.

Poeticness Generation. We also consider Chinese poeticness generation, as in [Shang *et al.*, 2019], which seeks to transfer a *vernacular* sentence to a classical *poetic* one. As Chinese vernacular text and classical poetry share similar vocabulary, differences between them, *e.g.*, grammar and syntax, lie beyond simple word usage. Hence, this task is more challenging than the above two. We build a corpus called **Chinese Poetic and Vernacular Text (CPVT)** with vernacular sentences from Chinese prose and poetic sentences from classical poems. Besides, we collect 7,000 pairs of human-authored sentences for testing and semi-supervised training. We only evaluate the transfer direction from vernacular to poetic. Since the opposite direction is more difficult requiring more sophisticated structures, we leave it for future work.

We use Yelp and GYAFC for unsupervised transfer; CPVT and GYAFC for semi-supervised transfer. English sentences are tokenized with the NLTK tool, and Chinese sentences are segmented as Chinese characters. All digits are replaced with a <NUM> symbol. Table 1 presents detailed data statistics.

4.2 Setups

We set word embedding size, hidden state size, the number of style instances K and the length of generative flow chain T to 256, 512, 10 and 6 respectively. The encoder and decoder share the same word embedding. The prior and posteriori distributions of z in Eq. (13) share parameters to reduce model size. The discriminator is a Convolutional Neural Network (CNN) based classifier with Spectral Normalization [Miyato *et al.*, 2018]. To handle the discrete nature of sentences, as in [Dai *et al.*, 2019], we multiply the softmax distribution by the word embedding matrix, to get a soft generated word and feed this weighted embedding to the discriminator. Adam with mini-batches (batch size=64) is used for optimization.

4.3 Baselines

We conduct comprehensive comparisons with several state-of-the-art style transfer models. For unsupervised transfer, we consider CrossAlign [Shen *et al.*, 2017], MultiDec [Fu *et al.*, 2018], DelRetri [Li *et al.*, 2018], Template [Li *et al.*, 2018], Disentangled [John *et al.*, 2019] and StyleTransformer [Dai *et al.*, 2019]. These models cover the four paradigms described in Sec. 2. We emphasize Disentangled and StyleTransformer (abbreviated as **Disent.** and **StyleTr.**) as the representatives

Models	Acc↑	BLEU↑	Cos↑	PPL↓	GM↑	Acc↑	BLEU↑	Cos↑	PPL↓	GM↑
	Yelp (*Unsupervised*)					GYAFC (*Unsupervised*)				
MultiDec [Fu *et al.*, 2018]	46.0	15.09	91	175	10.52	24.9	11.53	91	97	8.69
CorssAlign [Shen *et al.*, 2017]	73.2	9.41	90	76	10.94	66.8	3.18	88	35	8.52
DelRetri [Li *et al.*, 2018]	88.5	16.61	93	136	12.92	61.1	21.20	91	110	12.58
Template [Li *et al.*, 2018]	81.6	22.62	92	296	13.14	49.2	34.75	94	249	13.06
Disentangled [John *et al.*, 2019]	**91.7**	6.71	89	**26**	11.39	67.5	8.16	90	**24**	11.18
StyleTransformer [Dai *et al.*, 2019]	86.2	**27.45**	**96**	231	14.29	63.1	40.91	95	180	14.74
StyIns (Ours)	90.8	26.03	**96**	109	**14.83**	**67.8**	**46.73**	**96**	92	**16.11**

Models	CPVT (*Semi-Supervised 1k*)					GYAFC (*Semi-Supervised 2.5k*)				
CPLS [Shang *et al.*, 2019]	**99.0**	0.77	89	**329**	5.85	**71.2**	36.99	93	**41**	16.03
StyIns (Ours)	97.4	**3.74**	**95**	443	**8.68**	68.0	**47.30**	**96**	93	**16.16**
Models	CPVT (*Semi-Supervised 4k*)					GYAFC (*Semi-Supervised 10k*)				
CPLS [Shang *et al.*, 2019]	**98.3**	3.13	90	**283**	8.37	**71.4**	39.25	94	**44**	16.24
StyIns (Ours)	97.5	**4.00**	**95**	410	**8.86**	70.6	**47.81**	**96**	92	**16.36**

Table 2: Automatic evaluation results of unsupervised transfer and semi-supervised transfer with different numbers of paired data.

Models	Yelp			GYAFC		
	Sty.	Con.	Flu.	Sty.	Con.	Flu.
DelRetri	3.26	3.24	3.46	2.31	2.37	2.39
Template	3.03	3.34	3.12	2.16	3.56	2.97
Disent.	3.95	3.20	**4.46**	2.84	1.85	3.79
StyleTr.	3.51	4.35	3.78	3.03	3.27	3.14
StyIns	**4.52***	**4.41***	4.41	**3.97***	**4.41***	**4.48***

(a) Unsupervised transfer of sentiment and formality.

Models	Sty.	Con.	Flu
CPLS	**3.13**	2.18	2.41
StyIns	2.82	**3.67***	**2.91***

(b) Semi-supervised transfer of poeticness with 4k paired data.

Table 3: Human evaluation results. The Krippen-dorff's alpha of human rating is 0.64, indicating acceptable inter-annotator agreement. The diacritic $*$ ($p < 0.01$) represents that StyIns significantly outperforms baseline models.

of paradigms 1 & 4 respectively. For semi-supervised transfer, we compare CPLS [Shang *et al.*, 2019], which is the only one semi-supervised transfer model to our best knowledge.

4.4 Metrics

We consider three criteria: Style Transfer Accuracy (**Sty.**), Content Preservation (**Con.**) and Fluency (**Flu.**).

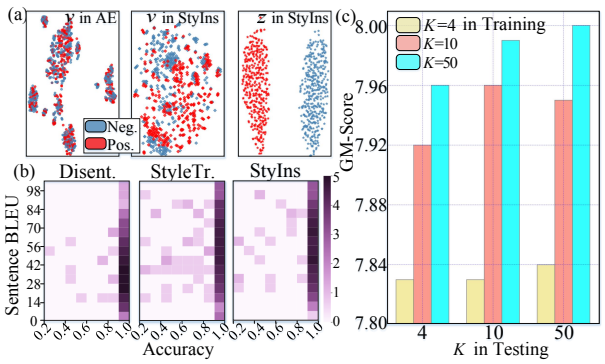
Automatic Evaluation. Following previous work [Fu *et al.*, 2018; Luo *et al.*, 2019; John *et al.*, 2019; Dai *et al.*, 2019], we use a classifier's accuracy (**Acc**) to measure style accuracy. For Yelp and GYAFC, we fine-tuned a pre-trained BERT [Devlin *et al.*, 2019] with each dataset. For CPVT, we train a CNN-based classifier. The three classifiers achieve 98%, 88% and 98% accuracy, respectively. The **BLEU** score between transferred sentences and human-authored references, and the cosine distance (**Cos**) between the source and transferred embeddings [Fu *et al.*, 2018], are utilized to measure content preservation. Cos is multiplied by 100 to match the scale of other metrics. We train a 5-gram language model KenLM [Heafield, 2011] with sentences of each style, and measure fluency by perplexity (**PPL**) of transferred sentences. We also report the geometric mean (**GM**) of Acc, BLEU, Cos and $\frac{1}{\log \text{PPL}}$ as the overall performance.

Human Evaluation. We conduct human rating on our StyIns and four baselines with the highest GM scores under automatic metrics. Due to the limitation of manual labour involved, we access unsupervised results of Yelp and GYAFC, and semi-supervised results of CPVT. For each model with each transfer direction, we sample 50 sentences and get 1,100 generated sentences in total. We invite three annotators to evaluate in a blind review manner. Each of the three criteria is scored on a 5-point scale ranging from 1 (worst) to 5 (best).

4.5 Experimental Results

As shown in Table 2 (upper part), our model achieves the best overall performance (GM). Disentangled gets satisfactory accuracy and PPL on both Yelp and GYAFC datasets, but performs worse for other metrics. These models belonging to paradigm 1 (*e.g.*, Disentangled and CrossAlign) try to separate content and style. However, as discussed in Sec. 1 & 2, due to the intractability of disentanglement, specifying one style may also drag some attached content out from the content space, resulting in fluent transferred sentences with the desired style but irrelevant phrases. On the contrary, StyleTransformer is better at content preservation, benefiting from its powerful attention structures. Nevertheless, the simple style embedding hinders this model from higher transfer accuracy. Moreover, we can found our StyIns also outperforms StyleTransformer on BLEU and Cos for GYAFC. With less data, the complicated Transformer can't be adequately trained, while our model is relatively insensitive to data size.

Table 2 (lower part) gives the results of semi-supervised transfer. Our model gets better overall results and excels at content preservation, while CPLS performs better in style control. CPLS also achieves lower PPL. The reason lies in that CPLS adopts multiple decoders, but our model only contains one decoder. With more paired data, both CPLS and StyIns obtain further improvement. Besides, CPLS is more sensitive to the size of parallel data on BLEU but not on accuracy, opposite to our model. Take poeticness transfer as an example. When the number of paired data increases from



(d)	negative → positive	[Yelp]	positive → negative
Input	do not go to this practice they are horrible!		they were extremely friendly and reasonably priced.
Disent.	love this practice.		they were extremely rude and extremely slow.
StyleTr.	do not go to this practice they are amazing!		they were extremely rude and reasonably priced.
StyIns	i always go to this practice they are great!		they were extremely rude and over priced.
	informal → formal	[GYAFC]	formal → informal
Input	u 'll find a man who really deserves u one day).		that is strange but perhaps. i do not know. i will speak to my friend later next to his locker.
Disent.	you should take a chance to find a person.		this is a friend that i know i like him and i know that.
StyleTr.	perhaps you can find a man who really deserves one day because.		that 's strange but u.
StyIns	you will find a man who really deserves you one day .		:) strange but maybe . . do n't know . . . get to my friend later next to his locker . . . ! !
		[CVPT]	
Input	敬仰你的高尚品德，别后会日夜把你思念。		
	Due to (my) respect for your noble character, I will miss you day and night after our separation.		
CPLS	喜君最名事，当日夜歌致。		
	I heard of your well-known fame, and thus, the very night I sang to you to express my ecstasy.		
StyIns	敬仰君高德，日夜把君念。		
	Due to (my) respect for your noble character, I will miss you day and night.		

Figure 2: (a) Visualization of data points and samples from the latent style space on Yelp with t-SNE. (b) The logarithm of the number of transferred sentences in different ranges of style accuracy and sentence-BLEU. We present those with accuracy$\geq$0.2 on Yelp. (c) The geometric mean of Acc, BLEU and $\frac{1}{\log \text{PPL}}$ on Yelp with different numbers of style instances. (d) Transferred samples from the three datasets. Phrases with different styles are marked in blue and red. Brown words are content irrelevant to source sentences.

1k to 4k, CPLS quintuples its BLEU score, but our model gets limited improvement. Please note that our model is suitable for both unsupervised and semi-supervised cases, while CPLS can be applied to semi-supervised transfer only.

Table 3 presents human evaluation results. Again, StyleTransformer gets worse style accuracy but better content preservation than Disentangled. In addition, StyIns achieves comparable or even better fluency compared to baselines under human rating. This result indicates that overly low PPL may be obtained by ignoring the required content, and a moderate PPL value is enough to reflect acceptable fluency.

4.6 Further Analysis

In Figure 2 (a), we vectorize sentences by a pre-trained AutoEncoder and the LSTM in our style encoder (E_{sty}), respectively. We can observe that the former fails to distinguish different sentiments while our style encoder can separate these data points to some extent. We also visualize samples from the latent style space of StyIns. Compared to original sentence representations, this space is much more discriminative, which could produce flexible and strong style signals.

In Figure 2 (b), we plot the accuracy and sentence-BLEU (calculated with NLTK) of sentences transferred by different models. We can see for Disentangled, sentences fall in a low-BLEU and high-Acc area. For StyleTransformer, more sentences spread in the lower-Acc region compared to StyIns. Such results manifest that StyIns makes a better balance between style accuracy and content preservation.

In Figure 2 (c), we investigate the effect of different numbers of style instances K. We found when we use a small K (*e.g.*, 4) in the training phase, setting different K in the testing phase makes a negligible difference. When we use a larger K in training, it's a better choice to take the same one for testing. In general, increasing K could facilitate the learning of latent style space and hence leads to better performance, which could support our claim that the independent assumption of sentences mentioned in Sec. 3.2 is implausible. However, larger K also requires more resources and slows down the training. As a compromise, we set K=10.

Figure 2 (d) shows some transferred samples from different datasets. We can see that Disentangled creates quite fluent sentences in apparent target style, but often loses source information, as discussed before. On Yelp, StyleTransformer can copy most style-independent source phrases but sometimes fails to generate required stylistic words. On the contrary, our StyIns makes a better balance on the two criteria. On CPVT, we can observe more interesting results. As discussed in Sec. 4.1, expressed beyond the use of stylistic words, poeticness is a more complex style than sentiment. CPLS chooses to sidestep this obstacle at times by generating fluent but irrelevant phrases. Our model, by contrast, learns to delete some vernacular words and reorder the remaining ones to better meet the syntactic requirements of classical poetry.

5 Conclusion and Future Work

In this work, we propose a style instance supported method, *StyIns*, to alleviate the contradiction of content preservation and style accuracy in text style transfer tasks. StyIns adopts the generative flow technique to construct a more discriminative and expressive latent style space with the support of multiple style instances, which provides strong style signals to an attention-based decoder. Besides, our model can be flexibly extended to the semi-supervised version to utilize limited parallel data for further improvement. Experiments on three transfer tasks show that our model achieves a better balance between content and style, against several state-of-the-arts.

We plan to explore few-instance text style transfer, in which case a new style and a few instances of it are available only in the testing phase. Without explicitly defined style categories, our model possesses the potential to achieve such transfer. We highlight this task and leave it for future work.

Acknowledgments

We would like to thank anonymous reviewers for their insightful comments. This work is supported by the Major Program of the National Social Science Fund of China (Grant No. 18ZDA238), as well as the NSFC project (Grant No. 61661146007) and the NExT++ project, the National Research Foundation, Prime Minister's Office, Singapore under its IRC@Singapore Funding Initiative.

References

[Atanov *et al.*, 2019] Andrei Atanov, Arsenii Ashukha, Kirill Struminsky, Dmitry Vetrov, and Max Welling. The deep weight prior. In *ICLR*, 2019.

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[Crystal, 1970] David Crystal. New perspectives for language study. 1:stylistics. *English Language Teaching*, 24(2):99–106, 1970.

[Dai *et al.*, 2019] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In *ACL*, pages 5997–6007, 2019.

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

[Embler, 1967] Weller Embler. Style is as style does. *ETC: A Review of General Semantics*, 24(4):447–454, 1967.

[Fu *et al.*, 2018] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In *AAAI*, pages 663–670, 2018.

[Germain *et al.*, 2015] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *ICML*, 2015.

[Gong *et al.*, 2019] Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. Reinforcement learning based text style transfer without parallel training corpus. In *NAACL-HLT*, pages 3168–3180, 2019.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[Heafield, 2011] Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 2011.

[Ide, 2004] Nancy Ide. Preparation and analysis of linguistic corpora. *A Companion to digital Humanities*, 27:278–305, 2004.

[John *et al.*, 2019] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *ACL*, pages 424–434, 2019.

[Kingma and Welling, 2014] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Kingma *et al.*, 2016] Durk P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *NeurIPS*, 2016.

[Lample *et al.*, 2019] Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. Multiple-attribute text rewriting. In *ICLR*, 2019.

[Li *et al.*, 2018] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *NAACL-HLT*, pages 1865–1874, 2018.

[Luo *et al.*, 2019] Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*, pages 5116–5122, 2019.

[Miyato *et al.*, 2018] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshid. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[Prabhumoye *et al.*, 2018] Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. Style transfer through back-translation. In *ACL*, 2018.

[Rao and Tetreault, 2018] Sudha Rao and Joel Tetreault. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. In *NAACL-HLT*, pages 129–140, 2018.

[Rezende and Mohamed, 2015] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, pages 1530–1538, 2015.

[Shang *et al.*, 2019] Mingyue Shang, Piji Li, Zhenxin Fu, Lidong Bing, Dongyan Zhao, Shuming Shi, and Rui Yan. Semi-supervised text style transfer: Cross projection in latent space. In *EMNLP-IJCNLP*, pages 4936–4945, 2019.

[Shen *et al.*, 2017] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *NeurIPS*, 2017.

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

[Wu *et al.*, 2019a] Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *ACL*, pages 4873–4883, 2019.

[Wu *et al.*, 2019b] Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. Mask and infill: Applying masked language model to sentiment transfer. In *IJCAI*, pages 5271–5277, 2019.

[Yang *et al.*, 2018] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In *NeurIPS*, 2018.

[Yi *et al.*, 2018] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. Automatic poetry generation with mutual reinforcement learning. In *EMNLP*, pages 3143–3153, 2018.

[Zhou *et al.*, 2018] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internaland external memory. In *AAAI*, pages 730–738, 2018.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translationusing cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017.