Learning to Generate Multiple Style Transfer Outputs for an Input Sentence

Kevin Lin[♣], **Ming-Yu Liu**[♦], **Ming-Ting Sun**[♣], **Jan Kautz**[♦]
[♣]University of Washington [♦]NVIDIA
{kvlin,mts}@uw.edu, {mingyul,jkautz}@nvidia.com

Abstract

Text style transfer refers to the task of rephrasing a given text in a different style. While various methods have been proposed to advance the state of the art, they often assume the transfer output follows a delta distribution, and thus their models cannot generate different style transfer results for a given input text. To address the limitation, we propose a one-to-many text style transfer framework. In contrast to prior works that learn a one-to-one mapping that converts an input sentence to one output sentence, our approach learns a one-to-many mapping that can convert an input sentence to multiple different output sentences, while preserving the input content. This is achieved by applying adversarial training with a latent decomposition scheme. Specifically, we decompose the latent representation of the input sentence to a style code that captures the language style variation and a content code that encodes the language style-independent content. We then combine the content code with the style code for generating a style transfer output. By combining the same content code with a different style code, we generate a different style transfer output. Extensive experimental results with comparisons to several text style transfer approaches on multiple public datasets using a diverse set of performance metrics validate effectiveness of the proposed approach.

1 Introduction

Text style transfer aims at changing the language style of an input sentence to a target style with the constraint that the style-independent content should remain the same across the transfer. While several methods are proposed for the task (John et al., 2019; Smith et al., 2019; Jhamtani et al., 2017; Kerpedjiev, 1992; Xu et al., 2012; Shen et al., 2017; Subramanian et al., 2018; Xu et al., 2018), they commonly model the distribution of the transfer outputs as a delta distribution, which implies a one-to-one mapping mechanism that converts an input sentence in one language style to a *single* corresponding sentence in the target language style.

We argue a multimodal mapping is better suited for the text style transfer task. For examples, the following two reviews:

1. *"This lightweight vacuum is simply effective.",*
2. *"This easy-to-carry vacuum picks up dust and trash amazingly well."*

would both be considered correct negative-to-positive transfer results for the input sentence, "*This heavy vacuum sucks*". Furthermore, a one-to-many mapping allows a user to pick the preferred text style transfer outputs in the inference time.

In this paper, we propose a one-to-many text style transfer framework that can be trained using non-parallel text. That is, we assume the training data consists of two corpora of different styles, and no paired input and output sentences are available. The core of our framework is a latent decomposition scheme learned via adversarial training. We decompose the latent representation of a sentence into two parts where one encodes the style of a sentence, while the other encodes the style-independent content of the sentence. In the test time, for changing the style of an input sentence, we first extract its content code. We then sample a sentence from the training dataset of the target style corpus and extract its style code. The two codes are combined to generate an output sentence, which would carry the same content but in the target style. As sampling a different style sentence, we have a different style code and have a different style transfer output. We conduct experiments with comparison to several state-of-the-art approaches on multiple public datasets, including Yelp (yel) and Amazon (He and McAuley, 2016). The results, evaluated using various performance metrics, including content preservation, style accuracy, output diversity, and user preference, show that the model trained with our framework performs consistently better than the competing approaches.

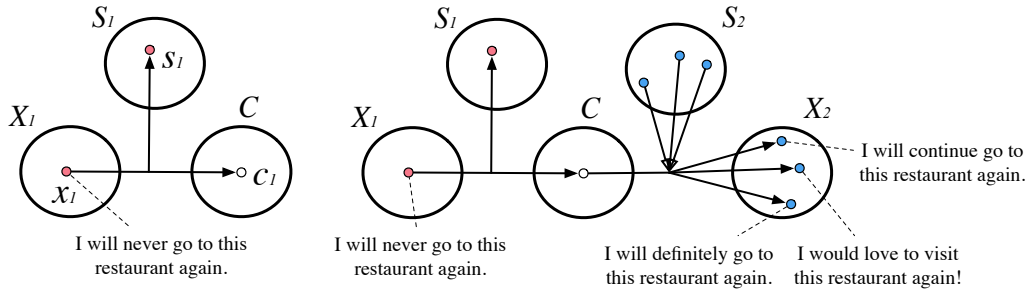


Figure 1: We formulate text style transfer as a one-to-many mapping function. Left: We decompose the sentence x_1 to a content code c_1 that controls the sentence meaning, and a style code s_1 that captures the stylistic properties of the input x_1. Right: One-to-many style transfer is achieved by fusing the content code c_1 and a style code s_2 randomly sampled from the target style space S_2.

2 Methodology

Let X_1 and X_2 be two spaces of sentences of two different language styles. Let Z_1 and Z_2 be their corresponding latent spaces. We further assume Z_1 and Z_2 can be decomposed into two latent spaces $Z_1 = S_1 \times C_1$ and $Z_2 = S_2 \times C_2$ where S_1 and S_2 are the latent spaces that control the style variations in X_1 and X_2 and C_1 and C_2 are the latent spaces that control the style-independent content information. Since C_1 and C_2 are style-independent content representation, we have $C \equiv C_1 \equiv C_2$. For example, X_1 and X_2 may denote the spaces of negative and positive product reviews where the elements in C encode the product and its features reviewed in a sentence, the elements in S_1 represent variations in negative styles such as the degree of preferences and the exact phrasing, and the elements in S_2 represent the corresponding variations in positive styles. The above modeling implies

1. A sentence $x_1 \in X_1$ can be decomposed to a content code $c_1 \in C$ and a style code $s_1 \in S_1$.
2. A sentence $x_1 \in X_1$ can be reconstructed by fusing its content code c_1 and its style code s_1.
3. To transfer a sentence in X_1 to a corresponding sentence in X_2, one can simply fuse the content code c_1 with a style code s_2 where $s_2 \in S_2$.

Figure 1 provides a visualization of the modeling.

Under this formulation, the text style transfer mechanism is given by a conditional distribution $p(x_{1\to2}|x_1)$, where $x_{1\to2}$ is the sentence generated by transferring sentence x_1 to the target domain X_2. Note that existing works (Fu et al., 2018; Shen et al., 2017) formulate the text style transfer mechanism to be a one-to-one mapping that converts an input sentence to only a single corresponding output sentence. That is $p(x_{1\to2}|x_1) = \delta(x_1)$ where δ is the Dirac delta function. As a results, they can not be used to generate multiple style transfer outputs for an input sentence.

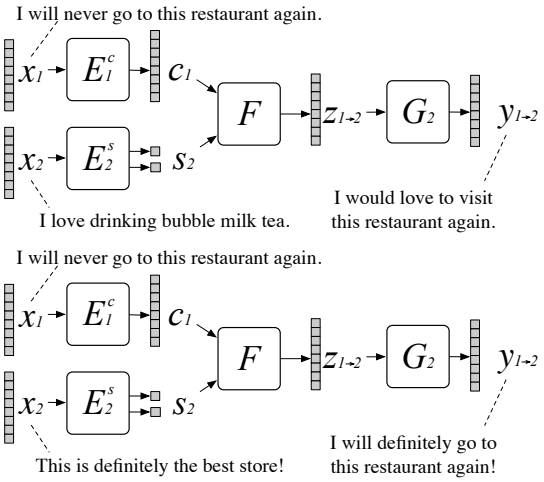


Figure 2: Overview of the proposed one-to-many style transfer approach. We show an example of transferring a negative restaurant review sentence x_1 to multiple different positive ones $y_{1\to2}$. To transfer the sentence, we first randomly sample a sentence x_2 from the space of positive reviews X_2 and extract its style code s_2 using E_2^s. We then compute $z_{1\to2}$ by combining c_1 with s_2 and convert it to the transfer output $y_{1\to2}$ using G_2. We note that by sampling a different x_2 and hence a different s_2, we have a different style transfer output $y_{1\to2}$.

One-to-Many Style Transfer. To model the transfer function, we use a framework consists of a set of networks as visualized in Figure 2. It has a content encoder E_i^c, a style encoder E_i^s, and a decoder G_i for each domain X_i. In the following, we will explain the framework in details using the task of transferring from X_1 to X_2. The task of transferring from X_2 to X_1 follows the same pattern.

The content encoder E_1^c takes the sequence $x_1 = \{x_1^1, x_1^2, \ldots, x_1^{m_1(x_1)}\}$ of $m_1(x_1)$ elements as input and computes a content code $c_1 \equiv \{c_1^1, c_1^2, \ldots, c_1^{m_1(x_1)}\} = E_1^c(x_1)$, which is a se-

quence of vectors describing the sentence's style-independent content. The style encoder E_2^S converts x_2 to a style code $s_2 \equiv (s_{2,\mu}, s_{2,\sigma}) = E_2^s(x_2)$, which is a pair of vectors. Note that we will use $s_{2,\mu}$ and $s_{2,\sigma}$ as the new mean and standard deviation of the feature activation of the input x_1 for the style transfer task of converting a sentence in X_1 to a corresponding sentence in X_2. Specifically, we combine the content code c_1 and the style code s_2 using a composition function F, which will be discussed momentarily, to obtain $z_{1\to2} = \{z_{1\to2}^1, z_{1\to2}^2, \ldots, z_{1\to2}^{m_1(x_1)}\}$. Then, we use the decoder G_2 to map the representation $z_{1\to2}$ to the output sequence $y_{1\to2}$. Note that s_2 is extracted from a randomly sampled $x_2 \in X_2$, and by sampling a different sentence, say $x_2' \in X_2$ where $x_2' \neq x_2$, we have $s_2' \neq s_2$ and hence a different style transfer output. By treating style variations as sample-able quantities, we achieve one-to-many style transfer output capability.

The combination function is given by

$$F(c_i^k, s_j) = s_{j,\sigma} \otimes (c_i^k - \mu(c_i)) \oslash \sigma(c_i) + s_{j,\mu}, \quad (1)$$

where $\otimes$ denotes element-wise product, $\oslash$ denotes element-wise division, $\mu(\cdot)$ and $\sigma(\cdot)$ indicate the operation of computing mean and standard derivation for the content latent code by treating each vector in c_i as an independent realization of a random variable. In other words, the latent representation $z_{i\to j}^k = F(c_i^k, s_j)$ is constructed by first normalizing the content code c_i in the latent space and then applying the non-linear transformation whose parameters are provided from a sentence of target style. Since F contains no learnable parameters, we consider F as part of the decoder. This design draws inspirations from image style transfer works (Huang and Belongie, 2017; Dumoulin et al., 2016), which show that image style transfer can be achieved by controlling the mean and variance of the feature activations in the neural networks. We hypothesize this is the same case for the text style transfer task and apply it to achieve the one-to-many style transfer capability.

Network Design. We realize the content encoder E_i^c using a convolutional network. To ensure the length of the output sequence c is equal to the length of the input sentence, we pad the input by $m-1$ zero vectors on both left and right side, where m is the length of the input sequence as discussed in (Gehring et al., 2017). For the convolution operation, we do not include any stride convolution. We also realize the style encoder E_i^s using a convolutional network. To extract the style code, after several convolution layers, we apply global average pooling and then project the results to $s_{i,\mu}$ and $s_{i,\sigma}$ using a two-layer multi-layer perceptron. We apply the log-exponential nonlinearity to compute $s_{i,\sigma}$ to ensure the outputs are strictly positive, required for modeling the deviations. The decoder G_i is realized using a convolutional network with an attention mechanism followed by a convolutional sequence-to-sequence network (ConvS2S) (Gehring et al., 2017). We realized our method based on ConvS2S, but it can be extended to work with transformer models (Vaswani et al., 2017; Devlin et al., 2018; Radford et al., 2019). Further details are given in the appendix.

2.1 Learning Objective

We train our one-to-many text style transfer model by minimizing multiple loss terms.

Reconstruction loss. We use reconstruction loss to regularize the text style transfer learning. Specifically, we assume the pair of content encoder E_i^c and style encoder E_i^s and the decoder G_i form an auto-encoder. We train them by minimizing the negative log likelihood of the training corpus:

$$\mathcal{L}_{rec}^i = \mathbb{E}_{x_i}[-\log P(y_i^k | x_i^k; \theta_{E_i^c}, \theta_{E_i^s}, \theta_{G_i})] \quad (2)$$

where $\theta_{E_i^c}$, $\theta_{E_i^s}$ and θ_{G_i} denote the parameters of E_i^c, E_i^s, and G_i respectively.

For each training sentence, G_i synthesizes the output sequence by predicting the most possible token y^t based on the latent representation $z_i \equiv \{z_i^1, z_i^2, ..., z_i^m\}$ and the previous output predictions $\{y^1, y^2, \ldots, y^{t-1}\}$, so that the probability of a sentence can be calculated by

$$P(y|x;\theta_{E_i^c}, \theta_{E_i^s}, \theta_{G_i}) = \prod_{t=1}^{T} p(y^t | z_i, y^1, y^2, \ldots, y^{t-1}; \theta_{G_i}), \quad (3)$$

where t denotes the token index and T is the sentence length. Following (Gehring et al., 2017), the probability of a token is computed by the linear projection of the decoder output using softmax.

Back-translation loss. Inspired by recent studies (Prabhumoye et al., 2018; Sennrich et al., 2015; Brislin, 1970) that show that back-translation loss, which is closely related to the cycle-consistency

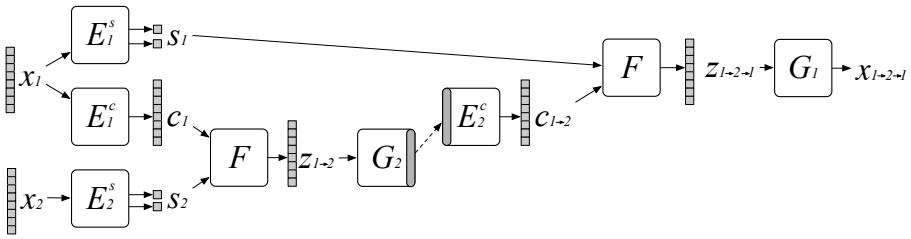


Figure 3: Illustration of the back-translation loss. We transfer x_1 to the domain of X_2 and then transfer it back to the domain of X_1 using its original style code s_1. The resultant sentence $x_{1\to2\to1}$ should be as similar as possible to x_1 if the content code is preserved across transfer. To tackle the non-differentiable of the sentence decoding mechanism (beam search), we replace the hard decoding of $x_{1\to2}$ by a learned non-linear projections between the decoder G_2 and the content encoder E_2^c.

loss (Zhu et al., 2017a) used in computer vision, is helpful for preserving the content of the input, we adopt a back-translation loss to regularize the learning. To achieve the goal, as shown in Figure 3, we transfer the input x_1 to the other style domain X_2. We then transfer it back to the original domain X_1 by using its original style code s_1. By doing so, the resulting sentence $x_{1\to2\to1}$ should be as similar as possible to the original input x_1. In other words, we minimize the discrepancy between x_1 and $x_{1\to2\to1}$ given by

$$\mathcal{L}^1_{back} = \mathbb{E}_{x_1,x_2}[-\log P(y_1^k|x^k_{1\to2\to1};\underline{\theta})] \quad (4)$$

where $\underline{\theta} = \{\theta_{E_1^c}, \theta_{E_1^s}, \theta_{G_1}, \theta_{E_2}, \theta_{E_2^s}, \theta_{G_2}\}$. We also define $\mathcal{L}^2_{back}$ in a similar way.

To avoid the non-differentiability of the beam search (Och and Ney, 2004; Sutskever et al., 2014), we substitute the hard decoding of $x_{1\to2}$ by using a set of differentiable non-linear transformations between the decoder G_2 and the content encoder E_1^c when minimizing the back-translation loss. The non-linear transformations project the feature activation of the second last layer of the decoder G_2 to the second layer of the content encoder E_1^c. These non-linear projections are learned by the multilayer perceptron (MLP), which are trained jointly with the text style transfer task. We also apply the same mechanism to compute $x_{2\to1}$. This way, our model can be trained purely using back-propagation.

To ensure the MLP correctly project the feature activation to the second layer of E_2^c, we enforce the output of the MLP to be as similar as possible to the feature activation of the second layer of E_1^c. This is based on the idea that x_1 and $x_{1\to2}$ should have the same content code across transfer, and their feature activation in the content encoder should also be the same. Accordingly, we apply Mean Square Error (MSE) loss function to achieve this objective:

$$\mathcal{L}^1_{mse} = \mathbb{E}_{x_1,x_2}[||E_2^{c,h}(x_{1\to2}) - E_1^{c,h}(x_1)||_2^2] \quad (5)$$

where $E_1^{c,h}$ and $E_2^{c,h}$ denote the function for computing feature activation of the second layer of E_1^c and E_2^c, respectively. The loss $\mathcal{L}^2_{mse}$ for the other domain is defined in a similar way.

Style classification loss. During learning, we enforce a style classification loss on the style code $s_i = E_i^s(x_i)$ with the standard cross-entropy loss $\mathcal{L}^i_{cls}$. This encourages the style code s_i to capture the stylistic properties of the input sentences.

Adversarial loss. We use GANs (Goodfellow et al., 2014) for matching the distribution of the input latent code to the decoder from the reconstruction streams to the distribution of the input latent code to the decoder from the translation stream. That is (1) we match the distribution of $z_{1\to2}$ to the distribution of z_2, and (2) we match the distribution of $z_{2\to1}$ to the distribution of z_1. This way we ensure distribution of the transfer outputs matches distribution of the target style sentences since they use the same decoder. As we apply adversarial training to the latent representation, we also avoid dealing with the non-differentiability of beam search. The adversarial loss for the second domain is given by

$$\begin{aligned}\mathcal{L}^2_{adv} = &\mathbb{E}_{x_1,x_2}\left[\log(1 - D_2(z_{1\to2}))\right] \\ &+ \mathbb{E}_{x_2}\left[\log(D_2(z_2))\right],\end{aligned} \quad (6)$$

where D_2 is the discriminator which aims at distinguishing the latent representation of the sentence $z_{1\to2}$ from $z_2 = \mathcal{C}_z(c_2, s_2)$. The adversarial loss $\mathcal{L}^1_{adv}$ is defined in a similar manner.

Overall learning objective. We then learn a one-

to-many text style transfer model by solving

$$\min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \sum_{i=1}^{2} (\mathcal{L}^i_{rec} + \mathcal{L}^i_{back} + \mathcal{L}^i_{mse} + \mathcal{L}^i_{cls} + \mathcal{L}^i_{adv}). \quad (7)$$

3 Experiments

In the following, we first introduce the datasets and evaluation metrics and then present the experiment results with comparison to the competing methods.

Datasets. We use the following datasets.

- **Amazon product reviews (Amazon)** (He and McAuley, 2016) contains 277, 228 positive and 277, 769 negative review sentences for training, and 500 positive and 500 negative review sentences for testing. The length of a sentence ranges from 8 to 25 words. We use this dataset for converting a negative product review to a positive one, and vice versa. Our evaluation follows the protocol described in Li et al. (2018).
- **Yelp restaurant reviews (Yelp)** (yel) contains a training set of 267, 314 positive and 176, 787 negative sentences, and a test set of 76, 392 positive and 50, 278 negative testing sentences. The length of a sentence ranges from 1 to 15 words. We use this dataset for converting a negative restaurant review to a positive one, and vice versa. We use two evaluation settings: Yelp500 and Yelp25000. Yelp500 is proposed by (Li et al., 2018), which includes randomly sampled 500 positive and 500 negative sentences from the test set, while Yelp25000 includes randomly sampled 25000 positive and 25000 negative sentences from the test set.

Evaluation metrics. We evaluate a text style transfer model on several aspects. Firstly, the transfer output should carry the target style (style score). Secondly, the style-independent content should be preserved (content preservation score). We also measure the diversity of the style transfer outputs for an input sentence (diversity score).

- **Style score.** We use a classifier to evaluate the fidelity of the style transfer results(Fu et al., 2018; Shen et al., 2017). Specifically, we apply the Byte-mLSTM (Radford et al., 2017) to classify the output sentence generated by a text style transfer model. As transferring a negative sentence to a positive one, we expect a good transfer model should be able to generate a sentence that is classified positive by the classifier. The overall style transfer performance of a model is then given by the average accuracy on the test set measured by the classifier.
- **Content score.** We build a style-independent distance metric that can quantify content similarity between two sentences, by comparing embeddings of the sentences after removing their style words. Specifically, we compute embedding of each non-style word in the sentence using the word2vec (Mikolov et al., 2013). Next, we compute the average embedding, which serves as the content representation of the sentence. The content similarity between two sentences is given by the cosine distance of their average embeddings. We compute the relative n-gram frequency to determine which word is a style word based on the observation that the language style is largely encoded in the n-gram distribution (Xu et al., 2012). This is in spirit similar to the term frequency-inverse document frequency analysis (Sparck Jones, 1972). Let D_1 and D_2 be the n-gram frequencies of two corpora of different styles. The style magnitude of an n-gram u in style domain i is given by

$$s_i(u) = \frac{D_i(u) + \lambda}{\sum_{j \neq i} D_j(u) + \lambda} \quad (8)$$

  where λ is a small constant. We use 1-gram. A word is considered a style word if $\min_{k \in \{i,j\}} s_k(u)$ is greater than a threshold.
- **Diversity score.** To quantify the diversity of the style transfer outputs, we resort to the self-BLEU score proposed by Zhu et al. (2018). Given an input sentence, we apply the style transfer model 5 times to obtain 5 outputs. We then compute self-BLEU scores between any two generated sentences (10 pairs). We apply this procedure to all the sentences in the test set and compute the average self-BLEU score v. After that, we define the diversity score as $100 - v$. A model with a higher diversity score means that the model is better in generating diverse outputs. In the experiments, we denote Diversity-K as the diversity score computed by using self-BLEU-K.

Implementation. We use the convolutional sequence-to-sequence model (Gehring et al., 2017). Our content and style encoder consist of 3 convolution layers, respectively. The decoder has 4 convolution layers. The content and style codes are 256 dimensional. We use the `pytorch` (Paszke

et al., 2017) and `fairseq` (Ott et al., 2019) libraries and train our model using a single GeForce GTX 1080 Ti GPU. We use the SGD algorithm with the learning rate set to 0.1. Once the content and style scores converge, we reduce the learning rate by an order of magnitude after every epoch until it reaches 0.0001. Detail model parameters are given in the appendix.

Baselines. We compare the proposed approach to the following competing methods.

- **CAE** (Shen et al., 2017) is based on auto-encoder and is trained using a GAN framework. It assumes a shared content latent space between different domains and computes the content code by using a content encoder. The output is generated with a pre-defined binary style code.
- **MD** (Fu et al., 2018) extends the CAE to work with multiple style-specific decoders. It learns style-independent representation by adversarial training and generates output sentences by using style-specific decoders.
- **BTS** (Prabhumoye et al., 2018) learns style-independent representations by using back-translation techniques. BTS assumes the latent representation of the sentence preserves the meaning after machine translation.
- **DR** (Li et al., 2018) employs retrieval techniques to find similar sentences with desired style. They use neural networks to fuse the input and the retrieved sentences for generating the output.
- **CopyPast** simply uses the input as the output, which serves as a reference for evaluation.

3.1 Results on One-to-Many Style Transfer

Our model can generate different text style transfer outputs for an input sentence. To generate multiple outputs for an input, we randomly sample a style code from the target style training dataset during testing. Since the **CAE** (Shen et al., 2017) and **BTS** (Prabhumoye et al., 2018) are not designed for the one-to-many style transfer, we extend their methods to achieve this capability by injecting random noise, termed **CAE+noise** and **BTS+noise**. Specifically, we add random Gaussian noise to the latent code of their models during training, which is based on the intuition that the randomness would result in different activations in the networks, leading to different outputs. Table 1 shows the average diversity scores achieved by the competing methods over 5 runs. We find that our method performs favorably against others.

Amazon	Diversity-4	Diversity-3	Diversity-2
CAE	2.60	2.15	1.64
CAE+noise	33.01	29.33	24.66
BTS+noise	39.22	35.46	30.48
Ours	**46.31**	**41.69**	**36.01**

Yelp	Diversity-4	Diversity-3	Diversity-2
CAE	1.03	0.80	0.60
CAE+noise	16.91	14.63	11.73
BTS+noise	48.36	43.69	37.38
Ours	**58.29**	**50.90**	**42.34**

Table 1: One-to-many text style transfer results.

Method	Diversity	Fluency	Overall
CAE+noise	13.13	11.62	12.12
No Pref.	35.35	16.16	36.87
Ours	**51.52**	**72.22**	**51.01**
BTS+noise	13.13	11.11	16.16
No Pref.	42.93	22.22	40.40
Ours	**43.94**	**66.67**	**43.43**

Table 2: User study results on one-to-many text style transfer. The numbers are the user preference score of competing methods.

User Study. We conduct a user study to evaluate one-to-many style transfer performance using the Amazon Mechanical Turk (AMT) platform. We set up the pairwise comparison following Prabhumoye et al. (2018). Given an input sentence and two sets of model-generated sentences (5 sentences per set), the workers are asked to choose which set has more diverse sentences with the same meaning, and which set provides more desirable sentences considering both content preservation and style transfer. These are denoted as *Diversity*, and *Overall* in Table 2. The workers are also asked to compare the transfer quality in terms of grammatically and fluency, which is denoted as *Fluency*. For each comparison, a third option *No Preference* is given for cases that both are equally good or bad.

We randomly sampled 250 sentences from Yelp500 test set for the user study. Each comparison is evaluated by at least three different workers. We received more than 3,600 responses from the AMT, and the results are summarized in Table 2. Our method outperforms the competing methods by a large margin in terms of diversity, fluency, and overall quality. In the appendix, we present further details of the comparisons with different variants of **CAE+noise** and **BTS+noise**. Our method achieves

Input: I will never go to this restaurant again. **Output A:** I will definitely go to this restaurant again. **Output B:** I will continue go to this restaurant again. **Output C:** I will definitely go to this place again.
Input: It was just a crappy experience over all. **Output A:** It was just a wonderful experience at all. **Output B:** Great place just a full experience over all. **Output C:** It was such a good experience as all.
Lyrics input: My friends they told me you change like the weather; From one love to another you would go; But when I first met you your love was like the summer; Love I never dreamed of turning cold **Romantic style:** My friends they told me you change like the light; From one love to another you would go; But when I first met you your love was like the sun; Love I never dreamed of turning cold **Romantic style:** My lips they told me you change like the light; From one love to find you would go; But when I am you your love was like the mountain; Love I never wanted of me before

Table 3: One-to-many style transfer results computed by the proposed algorithm.

Method	Style	Content	Fluency	Overall
CAE	30.56	36.81	23.26	30.56
No Pref.	31.60	**39.93**	**51.74**	**37.50**
Ours	**37.85**	23.26	25.00	31.94
MD	29.26	27.56	27.35	28.41
No Pref.	21.88	**52.84**	**47.01**	29.83
Ours	**48.86**	19.60	25.64	**41.76**
BTS	30.66	**40.88**	16.79	30.29
No Pref.	31.75	22.99	**56.93**	32.12
Ours	**37.59**	36.13	26.28	**37.59**
DR	26.30	18.09	22.61	24.96
No Pref.	11.89	**69.01**	**57.96**	21.61
Ours	**61.81**	12.90	19.43	**53.43**

Table 4: User study results. The numbers are the user preference scores of the competing methods.

Model	Style Score	Content Score
sharing-decoder	63.75	42.54
sharing-encoders	81.41	81.48
full	**82.64**	**83.11**

Table 5: Comparison of different design choices of the proposed framework.

significantly better performance. Table 3 shows the qualitative results of the proposed method. Our proposed method generates multiple different style transfer outputs for restaurant reviews and lyrics[1].

3.2 More Results and Ablation Study

In addition to generating multiple style transfer outputs, our model can also generate high-quality style transfer outputs. In Figure 4, we compare the quality of our style transfer outputs with those from the competing methods. We show the performance of our model using the style–content curve where each point in the curve is the achieved style score and the content score at different training iterations. In Figure 4a, given a fixed content preservation score, our method achieves a better style score on Amazon dataset. Similarly, given a fixed style score, our model achieves a better content preservation score. The results on Yelp500 and Yelp25000 datasets also demonstrate a similar trend as shown in Figure 4b and Figure 4c, respectively.

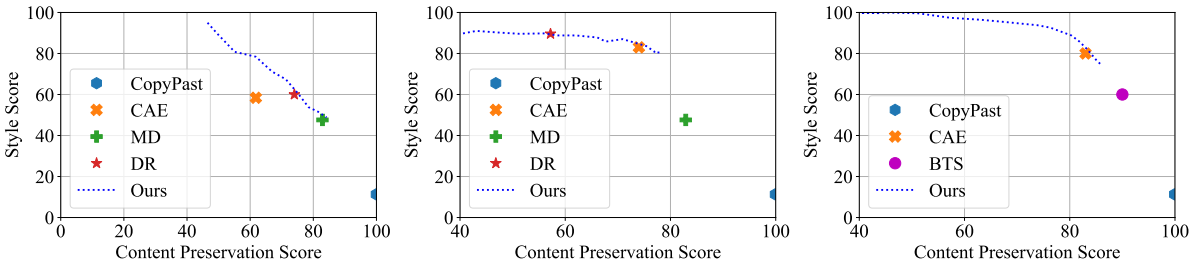
The style–content curve also depicts the behavior of the proposed model during the entire learning process. As visualized in Figure 5, we find that our model achieves a high style score but a low content score in the early training stage. With more iterations, our model improves the content score with the expense of a reduced style score. To strike a balance between the two scores, we decrease the learning rate when the model reaches a similar number for the two scores.

User Study. We also conduct a user study on the transfer output quality. Given an input sentence with two generated style transferred sentences from two different models[2], workers are asked to compare the transferred quality of the two generated sentences in terms of content preservation, style transfer, fluency, and overall performance, respectively. We received more than 2500 responses from AMT platform, and the results are summarized in Table 4. We observe *No Preference* was chosen more often than others, which shows exiting methods may not fully satisfy human expectation. However, our method achieves comparable or better performance than the prior works.

Ablation Study. We conduct a study where we consider three different designs of the proposed models. (1) *full*: This is the full version of the proposed model; (2) *sharing-encoders*: In this case, we have a content encoder and a style encoder that are shared by the two domains; (3) *sharing-decoder*: In this case, we have a decoder that is shared by the two domains. Through this study, we aim for studying if regularization via weight-sharing is beneficial to our approach.

[1]We use the country song lyrics and romance novel collections, which are available in the Stylish descriptions dataset (Chen et al., 2019).

[2]The sentences generated by other methods have been made publicly available by (Li et al., 2018).



(a) Amazon (b) Yelp500 (c) Yelp25000

Figure 4: Comparison to different style transfer algorithms on output quality.

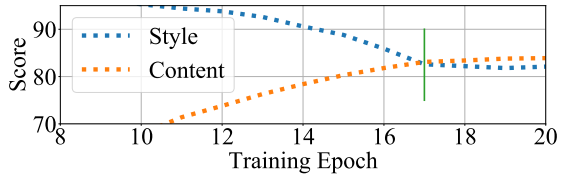


Figure 5: Style–content trade-off curves. The vertical line indicates the iteration at which the learning rate is decreased.

Table 5 shows the comparison of our method using different designs. The *sharing-encoders* baseline performs much better than the *sharing-decoder* baseline, and our *full* method performs the best. The results show that the style-specific decoder is more effective for generating target-style outputs. On the other hand, the style-specific encoder extracts more domain-specific style codes from the inputs. Weight-sharing schemes do not lead to a better performance.

Impact of the loss terms. In the appendix, we present an ablation study on the loss terms, which shows that all the terms in our objective function are important.

4 Related Works

Language modeling is a core problem in natural language processing. It has a wide range of applications including machine translation (Johnson et al., 2017; Wu et al., 2016), image captioning (Vinyals et al., 2015), and dialogue systems (Li et al., 2016a,b). Recent studies (Devlin et al., 2018; Gehring et al., 2017; Graves, 2013; Johnson et al., 2017; Radford et al., 2019; Wu et al., 2016) proposed to train deep neural networks using maximum-likelihood estimation (MLE) for computing the lexical translation probabilities in parallel corpus. Though effective, acquiring parallel corpus is difficult for many language tasks.

Text style transfer has a longstanding history (Kerpedjiev, 1992). Early studies utilize strongly supervision on parallel corpus (Rao and Tetreault, 2018; Xu, 2017; Xu et al., 2012). However, the lack of parallel training data renders existing methods non-applicable to many text style transfer tasks. Instead of training with paired sentences, recent studies (Fu et al., 2018; Hu et al., 2017; Prabhumoye et al., 2018; Shen et al., 2017; Li et al., 2019) addressed this problem by using adversarial learning techniques. In this paper, we argue while the existing methods address the parallel data acquisition difficulty, they do not address the diversity problem in the translated outputs. We address the issue by formulating text style transfer as a one-to-many mapping problem and demonstrate one-to-many style transfer results.

Generative adversarial network (GANs) (Arjovsky et al., 2017; Goodfellow et al., 2014; Salimans et al., 2016; Zhu et al., 2017a) have achieved great success on image generation (Huang et al., 2018; Zhu et al., 2017b). Several attempts are made to applying GAN for the text generation task (Guo et al., 2018; Lin et al., 2017; Yu et al., 2017; Zhang et al., 2017). However, these methods are based on unconditional GANs and tend to generate context-free sentences. Our method is different in that our model is conditioned on the content and style codes, and our method allows a more controllable style transfer.

5 Conclusion

We have presented a novel framework for generating different style transfer outputs for an input sentence. This was achieved by modeling the style transfer as a one-to-many mapping problem with a novel latent decomposition scheme. Experimental results showed that the proposed method achieves better performance than the baselines in terms of

the diversity and the overall quality.

References

Yelp Dataset Challenge. https://www.yelp.com/dataset/challenge.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Richard W Brislin. 1970. Back-translation for cross-cultural research. *Journal of cross-cultural psychology*, 1(3):185–216.

Cheng Kuan Chen, Zhu Feng Pan, Min Sun, and Ming-Yu Liu. 2019. Unsupervised stylish image description generation via domain layer norm. In *Proc. AAAI*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2016. A learned representation for artistic style. In *Proc. ICLR*.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proc. AAAI*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. ICML*.

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

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Prof. AAAI*.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proc. WWW*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proc. ICML*.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. ICCV*.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *Proc. ECCV*.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proc. EMNLP Workshop on Stylistic Variation*.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proc. ACL*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Googles multilingual neural machine translation system: Enabling zero-shot translation. *TACL*.

Stephan M. Kerpedjiev. 1992. Generation of informative texts with style. In *Proc. COLING*.

Dianqi Li Li, Yizhe Zhang, Zhe Gan Gan, Yu Cheng, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2019. Domain adaptive text style transfer. In *Proc. EMNLP*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. In *Proc. ACL*.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

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

Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial ranking for language generation. In *Proc. NeurIPS*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. NeurIPS*.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational linguistics*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. NAACL Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. ACL*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *Proc. NeurIPS Autodiff Workshop*.

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

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Tech Report*.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the yafc corpus: Corpus, benchmarks and metrics for formality style transfer. In *Proc. NAACL*.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. In *Proc. NeurIPS*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proc. NeruIPS*.

Eric Michael Smith, Diana Gonzalez-Rico, Emily Dinan, and Y-Lan Boureau. 2019. Zero-shot fine-grained style transfer: Leveraging distributed continuous style representations to transfer to unseen styles. *arXiv preprint arXiv:1911.03914*.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text style transfer. *arXiv preprint arXiv:1811.00552*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NeurIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NeurIPS*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proc. CVPR*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jingjing Xu, SUN Xu, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proc. ACL*.

Wei Xu. 2017. From shakespeare to twitter: What are language styles all about? In *Proc. EMNLP Workshop on Stylistic Variation*.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. *Proc. COLING*.

L Yu, W Zhang, J Wang, and Y Yu. 2017. Seqgan: sequence generative adversarial nets with policy gradient. In *Proc. AAAI*.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. *arXiv preprint arXiv:1706.03850*.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV*.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. 2017b. Toward multimodal image-to-image translation. In *Proc. NeurIPS*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *Proc. SIGIR*.

Method	Style	Content	Fluency	Diversity	Overall
Ours	36.36	42.42	**51.52**	**72.22**	**51.01**
No Pref.	28.79	**43.43**	35.35	16.16	36.87
$\text{CAE}_{\sigma=0.001}$	**34.85**	14.14	13.13	11.62	12.12
Ours	30.30	42.93	41.41	**72.73**	**51.01**
No Pref.	**35.35**	**43.94**	**44.95**	14.65	33.84
$\text{CAE}_{\sigma=0.01}$	34.34	13.13	13.64	12.63	15.15
Ours	34.34	**43.94**	**48.48**	**60.10**	**47.47**
No Pref.	29.29	42.93	42.42	28.79	39.39
$\text{CAE}_{\sigma=0.1}$	**36.36**	13.13	9.09	11.11	13.13
Ours	24.24	**48.48**	**41.41**	**56.57**	**50.00**
No Pref.	36.87	35.86	37.88	28.28	33.33
$\text{CAE}_{\sigma=1}$	**38.89**	15.66	20.71	15.15	16.67
Ours	30.81	**44.95**	37.88	**48.99**	**44.44**
No Pref.	**35.86**	41.41	**41.41**	39.39	34.34
$\text{CAE}_{\sigma=10}$	33.33	13.64	20.71	11.62	21.21
Ours	33.84	42.42	**46.97**	**68.18**	**43.94**
No Pref.	29.80	**48.80**	37.88	17.17	41.92
$\text{CAE}_{k=\{1\}}$	**36.36**	9.09	15.15	14.65	14.14
Ours	**36.87**	43.43	**45.96**	**76.77**	**48.48**
No Pref.	26.26	**45.96**	40.91	11.11	37.88
$\text{CAE}_{k=\{1,5\}}$	**36.87**	10.61	13.13	12.12	13.64
Ours	32.32	41.92	**44.44**	**71.72**	**46.46**
No Pref.	26.77	**46.46**	41.41	15.15	38.38
$\text{CAE}_{k=\{1,5,10\}}$	**40.91**	11.62	14.14	13.13	15.15
Ours	31.31	**43.94**	**50.51**	**73.74**	**47.47**
No Pref.	33.33	43.43	36.87	12.63	13.64
$\text{CAE}_{k=\{1,5,10,15\}}$	**35.35**	12.63	12.63	13.64	15.66

Table 6: Human preference comparison with the **CAE** on one-to-many style transfer results. The numbers are the user preference score of competing methods.

Method	Style	Content	Fluency	Diversity	Overall
Ours	34.34	37.88	**43.94**	**66.67**	**43.43**
No Pref.	30.30	**47.47**	42.93	22.22	40.40
$\text{BTS}_{\sigma=0.001}$	**35.35**	14.65	13.13	11.11	16.16
Ours	37.88	38.38	**44.95**	**54.55**	**46.46**
No Pref.	22.73	**45.45**	34.85	32.32	34.34
$\text{BTS}_{\sigma=0.01}$	**39.39**	16.16	20.20	13.13	19.19
Ours	29.80	**42.42**	**45.96**	**50.51**	**50.51**
No Pref.	29.29	41.92	36.36	35.86	35.35
$\text{BTS}_{\sigma=0.1}$	**40.91**	15.66	17.68	13.64	14.14
Ours	33.33	**42.93**	**46.97**	38.89	**52.53**
No Pref.	31.31	40.40	33.33	**46.97**	24.24
$\text{BTS}_{\sigma=1}$	**35.35**	16.67	19.70	14.14	23.23
Ours	34.34	**50.51**	**55.56**	**63.64**	**59.60**
No Pref.	30.30	33.84	25.25	25.25	18.69
$\text{BTS}_{\sigma=10}$	**35.35**	15.66	19.19	11.11	21.72
Ours	31.31	**45.96**	41.41	**72.22**	**56.06**
No Pref.	28.28	44.44	**42.93**	15.15	32.83
$\text{BTS}_{k=\{1\}}$	**40.40**	9.6	15.66	12.63	11.11
Ours	37.88	39.39	**48.48**	**71.72**	**48.99**
No Pref.	19.70	**48.99**	37.37	14.14	35.86
$\text{BTS}_{k=\{1,5\}}$	**42.42**	11.62	14.14	14.14	15.15
Ours	**37.88**	36.87	**44.95**	**71.72**	**47.47**
No Pref.	25.25	**47.47**	38.89	13.64	35.35
$\text{BTS}_{k=\{1,5,10\}}$	36.87	15.66	16.16	14.65	17.17
Ours	**36.36**	44.95	41.92	**72.73**	**56.57**
No Pref.	27.78	**46.46**	**45.96**	11.62	31.31
$\text{BTS}_{k=\{1,5,10,15\}}$	35.86	8.59	12.12	15.66	12.12

Table 7: Human preference comparison with the **BTS** on one-to-many style transfer results. The numbers are the user preference score of competing methods.

Appendix

A User Study

To control the quality of human evaluation, we conduct pilot study to design and improve our evaluation questionnaire. We invite 23 participants who are native or proficient English speakers to evaluate the sentences generated by different methods. For each participant, we randomly present 10 sentences from Yelp500 test set, and the corresponding style transferred sentences generated by different models. We ask the participants to vote the transferred sentence which they think the sentence meaning is closely related to the original sentence with an opposite sentiment. However, we find that it may be difficult to interpret the evaluation results in terms of transfer quality in details.

Therefore, instead of asking the participants to directly vote one sentence, we switch the task to evaluating the sentences in terms of four different aspects including style transfer, content preservation, fluency and grammatically, and overall performance. Following the literature (Prabhumoye et al., 2018), for each pairwise comparison, a third option *No Preference* is given for cases that both are equally good or bad. Figure 8 and Figure 9 show the instructions and the guidelines of our questionnaire for human evaluation on Amazon Mechanical Turk platform. We refer the reader to Section 3 in the main paper for the details of the human evaluation results.

To evaluate the performance of one-to-many style transfer, we extend the pair-wise comparison to set-wise comparison. Given an input sentence and two sets of model-generated sentences (5 sentences per set), the workers are asked to choose which set has more diverse sentences with the same meaning, and which set provides more desirable sentences considering both content preservation and style transfer. We also ask the workers to compare the transfer quality in terms of content preservation, style transfer, grammatically and fluency.

B Diversity Baselines

We report further comparisons with different variants of **CAE** and **BTS**. We added random Gaussian noise to the style code of **CAE** and **BTS**, respectively. Specifically, we randomly sample the noise

Recon. Loss	Back-Trans. Loss	Style Cls. Loss	Style Score	Content Preservation Score	BLEU
✗	✓	✗	31.20	45.89	0.00 (66.2/0.2/0.0/0.0)
✗	✗	✓	75.30	20.65	0.00 (0.0/0.0/0.0/0.0)
✗	✓	✓	100.00	42.46	0.00 (35.8/0.1/0.1/0.0)
✓	✗	✗	57.40	90.10	41.28 (69.7/48.2/34.8/25.2)
✓	✓	✗	61.38	90.14	39.72 (70.6/46.5/32.9/23.3)
✓	✗	✓	90.05	74.84	13.87 (48.0/20.4/9.6/4.1)
✓	✓	✓	82.64	83.11	24.5 (59.0/31.8/18.7/10.7)

Table 8: Empiricial analysis of the impact of each term in the proposed objective function for the proposed one-to-many style transfer task.

from the Gaussian distribution with $\mu = 0$ and $\sigma \in \{0.001, 0.01, 0.1, 1, 10\}$, respectively. We empirically found that the generations will be of poor quality when $\sigma > 10$. Thus, we evaluated the baselines with $\sigma \leq 10$ in the experiments. On the other hand, we also explored different extensions to enhance the diversity of sequence generation of the baselines. For example, we expanded the generations by randomly select a beam search size $k \in \{1, 5, 10, 15\}$ per generation.

C Additional One-to-Many Style Transfer User Study Results

We report the human evaluation with comparisons to different variants of the **CAE** and **BTS**. Similar to the human study presented in the main paper, we conduct evaluation using Amazon Mechanical Turk. We randomly sampled 200 sentences from Yelp test set for user study. Each comparison is evaluated by at least three experts whose HIT Approval Rate is greater than 90%. We received more than 3600 responses, and the results are summarized in Table 6 and Table 7. We observed previous models achieve higher style scores, but their output sentences are often in a generic format and may not preserve the content with correct grammar. In contrast, our method achieves significantly better performance than the baselines in terms of diversity, fluency, and overall quality.

D Ablation Study on Objective Function

The proposed objective function consists of five different learning objectives. We conduct ablation study to understand which loss function contributes to the performance. Since adversarial loss is essential for domain alignment, we evaluate loss functions by iterating different combination of the reconstruction loss, the back-translation loss (together with the mean square loss), and the style loss.

We report the style score and the content preservation score in this experiment. We additionally present the BLEU score (Papineni et al., 2002), which is a common metric for evaluating the performance of machine translation. A model with a higher BLEU score means that the model is better in translating reasonable sentences. As shown in Table 8, we find that training without reconstruction loss may not produce reasonable sentences according to the BLEU score. Training with reconstruction loss works well for content preservation yet it performs less favorably for style transfer. Back-translation loss is able to improve style and content preservation scores since it encourage content and style representations to be disentangle. When training with the style loss, our model improves the style accuracy, yet performs worse on content preservation. Overall, we observe that training with all the objective terms achieves a balanced performance in terms of different evaluation scores. The results show that the reconstruction loss, the back-translation loss, and the style loss are important for style transfer.

E Style Code Sampling Scheme

We design a sampling scheme that can lead to a more accurate style transfer. During inference, our network takes the input sentence as a query, and retrieves a pool of target style sentences whose content information is similar to the query. We measure the similarity by estimating the cosine similarity between the sentence embeddings. Next, we randomly sample a target style code from the retrieved pool, and generate the output sentence. The test-time sampling scheme improves the content preservation score from 83.11 to 83.41, and achieves similar style score from 82.64 to 82.66 on Yelp25000 test set. The results show that it is possible to improve the content preservation by using the top ranked target style sentences.

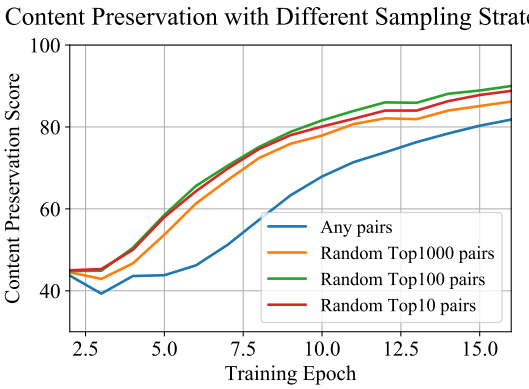


Figure 6: Performance comparison of our model using different sampling schemes.

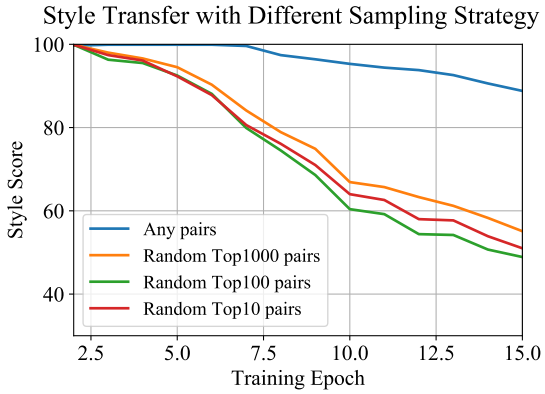


Figure 7: Performance comparison of our model using different sampling schemes.

We provide further analysis on the sampling scheme for the training phase. Specifically, during training, we sample the target style code from the pool of top ranked sentences in the target style domain. Figure 6 shows the content preservation scores of our method using different sampling schemes. The results suggest we can improve the content preservation by learning with the style codes extracted from the top ranked sentences in the target style domain. However, we noticed that this sampling scheme actually reduces the number of training data. It becomes more challenging for the model to learn the style transfer function as shown in Figure 7. The results suggest that it is more suitable to apply the sampling scheme in the inference phase.

F Additional Implementation Details

We use 256 hidden units for the content encoder, the style encoder, and the decoder. All embeddings in our model have dimensionality 256. We use the same dimensionalities for linear layers mapping between the hidden and embedding sizes. Additionally, we modify the convolution block in the style encoder E_i^s to have max pooling layers for capturing the activation of the style words. On the other hand, we also modify the convolution block of the content encoder E_i^c to have average pooling layers for computing the average activation of the input. During inference, the decoder generates the output sentence with the multi-step attention mechanism (Gehring et al., 2017).

Input: I stayed here but was disappointed as its air conditioner does not work properly. **Output:** I love here but was but as well feel's me work too.
Input: I might as well been at a street fest it was so crowded everywhere. **Output:** I well as well a at a reasonable price it was so pleasant.
Input: Free cheese puff - but had rye in it (I hate rye!). **Output:** It's not gourmet but it definitely satisfies my taste for good Mexican food.

Table 9: Example failure cases generated by the proposed method.

G Failure Cases

Although our approach performs more favorably against the previous methods, our model still fails in a couple of situations. Table 9 shows the common failure example generated by our model. We observe that it is challenging to preserve the content when the inputs are the lengthy sentences. It is also challenging to transfer the style if the sentence contains novel symbols or complicated structure.

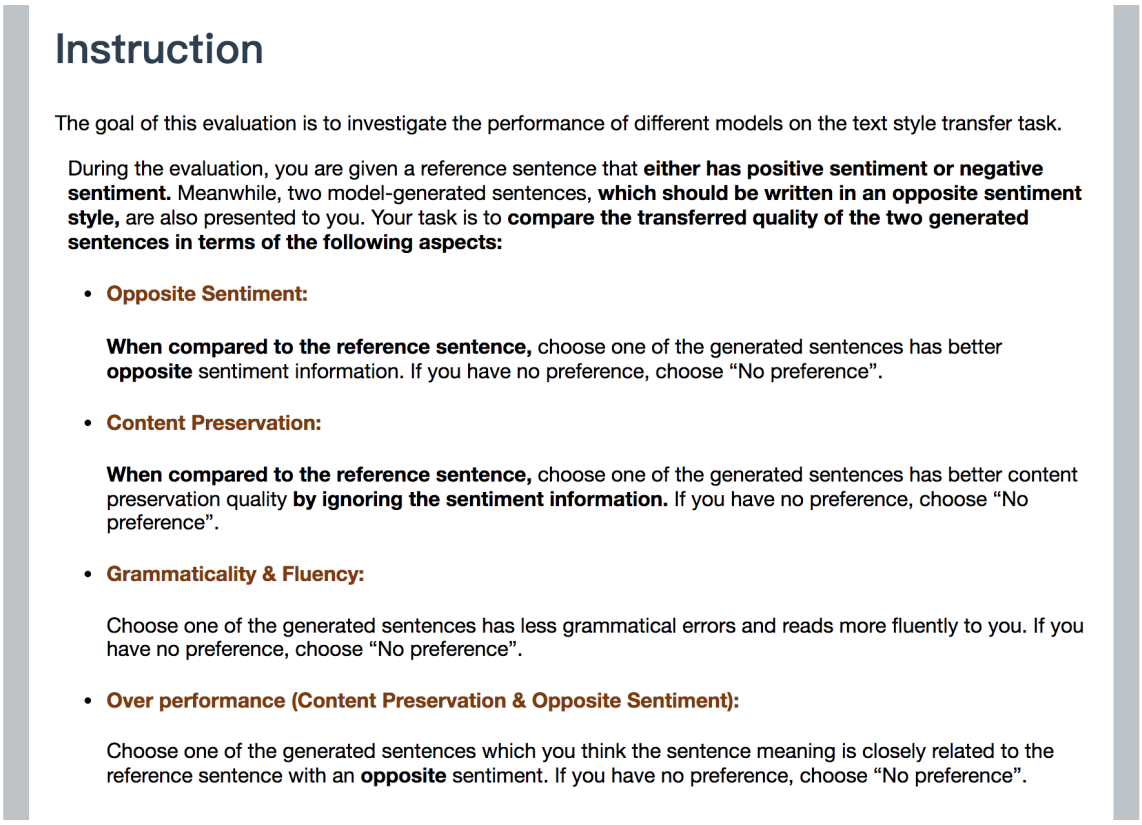


Figure 8: Instruction of our questionnaire on Amazon Mechanical Turk platform.

Give the reference sentence:

"This is a great movie!"

- **Opposite Sentiment:**

  You should choose these sentences that have the **opposite** sentiment:

  "This is not a good movie.";
  "This is a bad actor";
  "This pizza is disgusting!"

  You should not choose these sentences that have the same sentiment:

  "This is a good movie.";
  "This is a fantastic journey!";
  "I love this restaurant."

- **Content Preservation (by ignoring the sentiment information):**

  You should choose these sentences that have similar content:

  "This is a good movie.";
  "This is very terrible movie";

  You should not choose these sentences that have a completely different content:

  "I did not like the salad";
  "This pizza is tasty";
  "What a terrible experience!"

- **Overall Performance (Content Preservation & Opposite Sentiment):**

  You should choose these sentences that have similar content with the **opposite** sentiment:

  "This is not a good movie.";
  "This is very terrible movie";
  "I don't like this movie";

  You should not choose these sentences that have unrelated content or don't have opposite sentiment:

  "This is an awesome movie.";
  "This pizza is tasty";
  "What a terrible experience!"

Figure 9: Example and guideline of our questionnaire on Amazon Mechanical Turk platform.