LANGUAGE STYLE TRANSFER FROM NON-PARALLEL TEXT WITH ARBITRARY STYLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Language style transfer is the problem of migrating the content of a source sentence to a target style. In many applications, parallel training data are not available and source sentences to be transferred may have arbitrary and unknown styles. In this paper, we present an encoder-decoder framework under this problem setting. Each sentence is encoded into its content and style latent representations. By recombining the content with the target style, we can decode a sentence aligned in the target domain. To adequately constrain the encoding and decoding functions, we couple them with two loss functions. The first is a style discrepancy loss, enforcing that the style representation accurately encodes the style information guided by the discrepancy between the sentence style and the target style. The second is a cycle consistency loss, which ensures that the transferred sentence should preserve the content of the original sentence disentangled from its style. We validate the effectiveness of our proposed model on two tasks: sentiment modification of restaurant reviews, and dialog response revision with a romantic style.

1 INTRODUCTION

Style transfer is a long-standing research problem that aims at migrating the content of a sample from a source style to a target style. Recently, great progress has been achieved by applying deep neural networks to redraw an image in a particular style (Kulkarni et al., 2015; Liu & Tuzel, 2016; Gatys et al., 2016; Zhu et al., 2017; Luan et al., 2017). However, until now very few approaches have been proposed for style transfer of natural language sentences, i.e., changing the style or genre of a sentence while preserving its semantic content. For example, we would like a system that can convert a given text piece in the language of Shakespeare (Mueller et al., 2017); or rewrite product reviews with a favored sentiment (Shen et al., 2017).

One important issue on language style transfer is that parallel data are unavailable. For instance, considering the task of rewriting a negative review of a product to its counterpart with a positive sentiment, we can hardly find paired data that describe the same content. Yet, many text generation frameworks require parallel data, such as the popular sequence-to-sequence model in machine translation and document summarization (Sutskever et al., 2014; Rush et al., 2015), and thus are not applicable under this scenario. A few recent approaches have been proposed for style transfer with non-parallel data (Hu et al., 2017; Shen et al., 2017). Their key idea is to learn a latent representation of the content disentangled from the source style, and then recombine it with the target style to generate the corresponding sentence.

All the above approaches assume that data have only two styles, and their task is to transfer sentences from one style to the other. However, in many practical settings, we may deal with sentences in more than two styles. Taking the review sentiment modification as an example again, some reviews may be neither positive nor negative, but in a neutral style. Moreover, even reviews considered negative can be categorized into more fine-grained sentiments, such as anger, sadness, boredom and other negative styles. It may be beneficial if such styles are treated differently. As another example, consider a chatbot with a coherent persona, which has a consistent language behavior and interaction style (Li et al., 2016). A simple framework for this task is to first use human dialog data to train a chatbot system, such as a retrieval-based dialog model (Lowe et al., 2015), and then transfer the output responses with a language style transfer model so that multi-round responses always have a consistent style. Note that the human dialog sentences are collected from different users, and users'

expressions of the content and tones may be in different personalized characteristics. Thus the output responses retrieved from the dialog model may have the language style of any user. Simply treating the responses with a single style and employing the existing style transfer models would lead to unsatisfactory results. Hence, in this paper, we study the setting of language style transfer in which the source data to be transferred can have various (and possibly unknown) styles.

Another challenging problem in language style transfer is that the transferred sentence should preserve the content of the original sentence disentangled from its style. To tackle this problem, Shen et al. (2017) assumed the source domain and the target domain share the same latent content space, and trained their model by aligning these two latent spaces. Hu et al. (2017) constrained that the latent content representation of the original sentence could be inferred from the transferred sentence. However, these attempts considered content modification in the latent content space but not the sentence space.

In this work, we develop an encoder-decoder framework that can transfer a sentence from a source domain to its counterpart in a target domain. The training data in the two domains are non-parallel, and sentences in the source domain can have arbitrary language styles but those in the target domain are with a consensus style. We encode each sentence into two latent representations, one for the content disentangled from the style, and the other for the style. Intuitively, if a source sentence is considered having the target style with a high probability, its style representation should be close to the target style representation. To make use of this idea, we enforce that the discrepancy between an arbitrary style representation and the target style representation should be consistent with the closeness of its sentence style to the target style. A cycle consistency loss is further introduced to avoid content change by directly considering the transferred sentence. Its idea is that the generated sentence, when put back into the encoder and recombined with its original style representation, can recover the original sentence. We evaluate the performance of our proposed model on two tasks. The first is the sentiment modification task with its source domain containing more than one sentiments, and the second is to transfer general dialog responses to a romantic style.

2 Related Work

Most style transfer approaches in the literatures focus on vision data, and some of them are also designed for the non-parallel data setting. Kulkarni et al. (2015) proposed to disentangle the content representations from image attributes, and control the image generation by manipulating the graphics code that encodes the attribute information. Gatys et al. (2016) used the Convolutional Neural Networks (CNNs) to learn separated representations of the image content and style, and then created the new image from their combination. Some approaches have been proposed to align the two data domains with the idea of the generative adversarial networks (GAN) (Goodfellow et al., 2014). Liu & Tuzel (2016) proposed the coupled GAN framework to learn a joint distribution of multi-domain data by the weight-sharing constraint. Zhu et al. (2017) introduced a cycle consistency loss, which minimizes the gap between the transferred images and the original ones. However, due to the discreteness of the natural language, this loss function cannot be directly applied on text data. In our work, we show how the idea of cycle consistency can be used on text data.

Only a small number of approaches have been proposed for language style transfer. To handle the non-parallel data problem, Mueller et al. (2017) revised the latent representation of a sentence in a certain direction guided by a classifier, so that the decoded sentence imitates those favored by the classifier. Ficler & Goldberg (2017) encoded textual property values with embedding vectors, and adopted a conditioned language model to generate sentences satisfying the specified content and style properties. Hu et al. (2017) used the variational auto-encoder (VAE) to encode the sentence into a latent content representation disentangled from the source style, and then recombine it with the target style to generate its counterpart, An additional distribution is added to enforce that the generated sentence and the original sentence share the same latent content representation. Shen et al. (2017) considered transferring between two styles simultaneously. Specifically, they utilized adversarial training in the Professor-Forcing framework (Lamb et al., 2016), to align the generated sentences from one style to the data domain of the other style. We also adopt similar adversarial training in our model. However, since we assume the source domain contains data with various and possibly unknown styles, we cannot align data from the target domain to the source domain as in Shen et al. (2017).

3 MODEL

3.1 FORMULATION

We now formally present our problem formulation. Suppose there are two data domains, one source domain $\mathcal{X}_1$ in which each sentence may have its own language style, and one target domain $\mathcal{X}_2$ consisting of data with the same language style. During training, we observe n samples from $\mathcal{X}_1$ and m samples from $\mathcal{X}_2$, denoted as $\mathbf{X}_1 = \{\mathbf{x}_1^{(1)}, \mathbf{x}_1^{(2)}, \cdots, \mathbf{x}_1^{(n)}\}$ and $\mathbf{X}_2 = \{\mathbf{x}_2^{(1)}, \mathbf{x}_2^{(2)}, \cdots, \mathbf{x}_2^{(m)}\}$. Note that we can hardly find a sentence pair $(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(j)})$ that describes the same content. Our task is to design a model to learn from these non-parallel training data such that for an unseen testing sentence $\mathbf{x} \in \mathcal{X}_1$, we can transfer it into its counterpart $\tilde{\mathbf{x}} \in \mathcal{X}_2$, where $\tilde{\mathbf{x}}$ should preserve the content of $\mathbf{x}$ but with the language style in $\mathcal{X}_2$.

3.2 ENCODER-DECODER FRAMEWORK

Similar to Shen et al. (2017); Hu et al. (2017), we assume each sentence $\mathbf{x}$ can be decomposed into two representations: one is the style representation $\mathbf{y} \in \mathcal{Y}$, and the other is the content representation $\mathbf{z} \in \mathcal{Z}$, which is disentangled from its style. Each sentence $\mathbf{x}_1^{(i)} \in \mathbf{X}_1$ has its individual style $\mathbf{y}_1^{(i)}$, while all the sentences $\mathbf{x}_2^{(i)} \in \mathbf{X}_2$ share the same style, denoted as $\mathbf{y}^*$. Our model is built upon the encoder-decoder framework. In the encoding module, we assume that $\mathbf{z}$ and $\mathbf{y}$ of a sentence $\mathbf{x}$ can be obtained through two encoding functions $\mathbf{E_z}(\mathbf{x})$ and $\mathbf{E_y}(\mathbf{x})$ respectively:

$$\mathbf{z}_1^{(i)} = \mathbf{E_z}(\mathbf{x}_1^{(i)}), \quad \mathbf{y}_1^{(i)} = \mathbf{E_y}(\mathbf{x}_1^{(i)}), \quad \forall i \in \{1, 2, \cdots, n\}, \tag{1}$$

$$\mathbf{z}_2^{(j)} = \mathbf{E_z}(\mathbf{x}_2^{(j)}), \quad \mathbf{y}^* = \mathbf{E_y}(\mathbf{x}_2^{(j)}), \quad \forall j \in \{1, 2, \cdots, m\}, \tag{2}$$

where $\mathbf{E_y}(\mathbf{x}) = \mathbb{1}_{\{\mathbf{x} \in \mathbf{X}_1\}} \cdot g(\mathbf{x}) + \mathbb{1}_{\{\mathbf{x} \in \mathbf{X}_2\}} \cdot \mathbf{y}^\star$, and $\mathbb{1}_{\{\cdot\}}$ is an indicator function. When a sentence $\mathbf{x}$ comes from source domain, we use a function $g(\mathbf{x})$ to encode its style representation. For $\mathbf{x}$ from target domain, a shared style representation $\mathbf{y}^\star$ is used. Both $\mathbf{y}^*$ and parameters in $g(\mathbf{x})$ are learnt jointly together with other parameters in our model.

For the decoding module, we first employ a reconstruction loss to encourage that the sentence from the decoding function given $\mathbf{z}$ and $\mathbf{y}$ of a sentence $\mathbf{x}$ can well reconstruct $\mathbf{x}$ itself. Here, we use a probabilistic generator $\mathbf{G}$ as the decoding function and the reconstruction loss is:

$$\mathcal{L}_{rec}(\boldsymbol{\theta}_{\mathbf{E_z}}, \boldsymbol{\theta}_{\mathbf{E_y}}, \boldsymbol{\theta}_{\mathbf{G}}) = \mathbb{E}_{\mathbf{x}_1 \sim \mathbf{X}_1}\left[-\log p_{\mathbf{G}}(\mathbf{x}_1 | \mathbf{y}_1, \mathbf{z}_1)\right] + \mathbb{E}_{\mathbf{x}_2 \sim \mathbf{X}_2}\left[-\log p_{\mathbf{G}}(\mathbf{x}_2 | \mathbf{y}^*, \mathbf{z}_2)\right], \tag{3}$$

where $\boldsymbol{\theta}$ denotes the parameter of the corresponding module.

To enable style transfer using non-parallel training data, we enforce that for a sample $\mathbf{x}_1 \in \mathbf{X}_1$, its decoded sequence using $\mathbf{G}$ given its content representation $\mathbf{z}$ and the target style $\mathbf{y}^*$ should be in the target domain $\mathcal{X}_2$. We use the idea of GAN (Goodfellow et al., 2014)) and introduce an adversarial loss to be minimized in decoding. The goal of the discriminator $\mathbf{D}$ is to distinguish between $\mathbf{G}(\mathbf{z}_1, \mathbf{y}^*)$ and $\mathbf{G}(\mathbf{z}_2, \mathbf{y}^*)$, while the generator tries to bewilder the discriminator:

$$\mathcal{L}_{adv}(\boldsymbol{\theta}_{\mathbf{D}}, \boldsymbol{\theta}_{\mathbf{G}}, \boldsymbol{\theta}_{\mathbf{E_z}}, \boldsymbol{\theta}_{\mathbf{E_y}}) = \mathbb{E}_{\mathbf{x}_1 \sim \mathbf{X}_1}[-\log(1 - \mathbf{D}(\mathbf{G}(\mathbf{z}_1, \mathbf{y}^*)))] + \mathbb{E}_{\mathbf{x}_2 \sim \mathbf{X}_2}[-\log \mathbf{D}(\mathbf{G}(\mathbf{z}_2, \mathbf{y}^*))]. \tag{4}$$

As discussed in Section 2, since our source domain $\mathcal{X}_1$ contains sentences with various unknown language styles but not a consistent style, it is impossible for us to apply a discriminator to determine whether a sentence transferred from $\mathcal{X}_2$ is aligned in the domain $\mathcal{X}_1$ as in Shen et al. (2017).

During optimization, we adopt the continuous approximation in (Hu et al., 2017) for gradients propagation in adversarial training over discrete sentences. That is, instead of feeding a single word as the input to the generator, we use the approximation averaging word embeddings by a multinomial distribution. This distribution is computed as $\text{softmax}(\mathbf{o}_t/\gamma)$, where $\mathbf{o}_t$ is the logit vector output by the generator at time step t, $\gamma > 0$ is a temperature parameter. Next, we follow the framework of Professor-Forcing (Lamb et al., 2016), which matches two sequences of output words using a discriminator $\mathbf{D}$. Specifically, we have one kind of sequences $\mathbf{G}(\mathbf{z}_2, \mathbf{y}^*)$ teacher-forced by the ground-truth sample $\mathbf{x}_2 \in \mathbf{X}_2$, and the other one $\mathbf{G}(\mathbf{z}_1, \mathbf{y}^*)$ with $\mathbf{z}_1$ obtained from samples in $\mathbf{X}_1$, in which the input at each time step is self-generated by the previous continuous approximation.

However, the above encoder-decoder framework is under-constrained. First, for a sample $\mathbf{x}_1 \in \mathbf{X}_1$, $\mathbf{y}_1$ can have an arbitrary value that minimizes the above losses in Equation 3 and 4, which may not

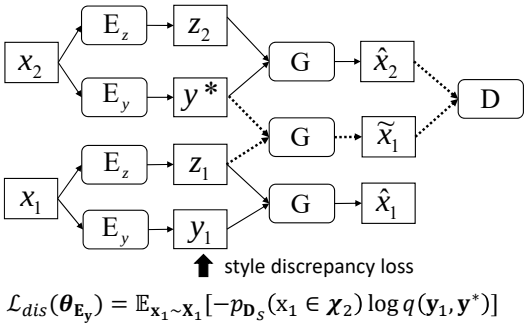

Figure 1: Basic model with the style discrepancy loss. Solid lines: encode and decode the sample itself; dash lines: transfer $\mathbf{x}_1 \in \mathbf{X}_1$ into $\mathcal{X}_2$.

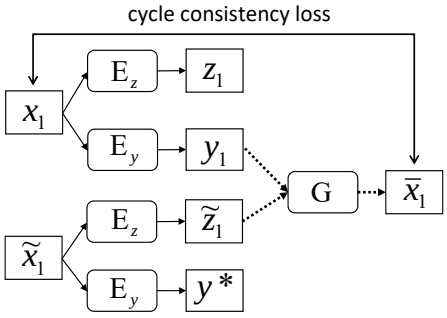

Figure 2: Proposed cycle consistency loss (can be applied for samples in $\mathcal{X}_2$ similarly).

necessarily capture the sentence style. This will affect the other decomposed part $\mathbf{z}$, making it not fully represent the content which should be invariant with the style. Second, the discriminator can only encourage the generated sentence to be aligned with the target domain $\mathcal{X}_2$, but cannot guarantee to keep the content of the source sentence intact. To address the first problem, we propose a style discrepancy loss, to constrain that the learnt $\mathbf{y}$ should have its distance from $\mathbf{y}^*$ guided by another discriminator which evaluates the closeness of the sentence style to the target style. For the second problem, we get inspired by the idea in Zhu et al. (2017) and introduce a cycle consistency loss applicable to word sequence, which requires that the generated sentence $\tilde{\mathbf{x}}$ can be transferred back to the original sentence $\mathbf{x}$.

3.3 Style Discrepancy Loss

By using a portion of the training data, we can first train a discriminator $\mathbf{D}_s$ to predict whether a given sentence $\mathbf{x}$ has the target language style with an output probability, denoted as $p_{\mathbf{D}_s}(\mathbf{x} \in \mathcal{X}_2)$. When learning the decomposed style representation $\mathbf{y}_1$ for a sample $\mathbf{x}_1 \in \mathbf{X}_1$, we enforce that the discrepancy between this style representation and the target style representation $\mathbf{y}^*$, should be consistent with the output probability from $\mathbf{D}_s$. Specifically, since the styles are represented with embedding vectors, we measure the style discrepancy using the ℓ_2 norm:

$$d(\mathbf{y}_1, \mathbf{y}^*) = \|\mathbf{y}_1 - \mathbf{y}^*\|_2 \,. \tag{5}$$

Intuitively, if a sentence has a larger probability to be considered having the target style, its style representation should be closer to the target style representation $\mathbf{y}^*$. Thus, we would like to have $d(\mathbf{y}_1, \mathbf{y}^*)$ positively correlated with $1 - p_{\mathbf{D}_s}(\mathbf{x}_1 \in \mathcal{X}_2)$. To incorporate this idea in our model, we use a probability density function $q(\mathbf{y}_1, \mathbf{y}^*)$, and define the style discrepancy loss as:

$$\mathcal{L}_{dis}(\boldsymbol{\theta}_{\mathbf{E}_y}) = \mathbb{E}_{\mathbf{x}_1 \sim \mathbf{X}_1}[-p_{\mathbf{D}_s}(\mathbf{x}_1 \in \mathcal{X}_2) \log q(\mathbf{y}_1, \mathbf{y}^*)] \,, \tag{6}$$

$$q(\mathbf{y}_1, \mathbf{y}^*) = f(d(\mathbf{y}_1, \mathbf{y}^*)), \tag{7}$$

where $f(\cdot)$ is a valid probability density function. $p_{\mathbf{D}_s}(\mathbf{x}_1 \in \mathcal{X}_2)$ is pre-trained and then fixed. If a sentence $\mathbf{x}_1$ has a large $p_{\mathbf{D}_s}(\mathbf{x}_1 \in \mathcal{X}_2)$, incorporating the above loss into the encoder-decoder framework will encourage a large $q(\mathbf{y}_1, \mathbf{y}^*)$ and hence a small $d(\mathbf{y}_1, \mathbf{y}^*)$, which means $\mathbf{y}_1$ would be close to $\mathbf{y}^*$. In our experiment, we instantiate $q(\mathbf{y}_1, \mathbf{y}^*)$ with the standard normal distribution for simplicity:

$$q(\mathbf{y}_1, \mathbf{y}^*) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{d(\mathbf{y}_1, \mathbf{y}^*)^2}{2}). \tag{8}$$

However, better probability density functions can be used if we have some prior knowledge of the style distribution. With Equation 8, the style discrepancy loss can be equivalently minimized by:

$$\mathcal{L}_{dis}(\boldsymbol{\theta}_{\mathbf{E}_y}) = \mathbb{E}_{\mathbf{x}_1 \sim \mathbf{X}_1}[p_{\mathbf{D}_s}(\mathbf{x}_1 \in \mathcal{X}_2) d(\mathbf{y}_1, \mathbf{y}^*)^2] \,. \tag{9}$$

3.4 Cycle Consistency Loss

Inspired by Zhu et al. (2017), we require that a sentence transferred by the generator $\mathbf{G}$ should preserve the content of its original sentence, and thus it should have the capacity to recover the

original sentence in a cyclic manner. For a sample $\mathbf{x}_1 \in \mathbf{X}_1$ with its transferred sentence $\tilde{\mathbf{x}}_1$ having the target style $\mathbf{y}^*$, we encode $\tilde{\mathbf{x}}_1$ and combine its content $\tilde{\mathbf{z}}_1$ with its original style $\mathbf{y}_1$ for decoding. We should expect that with a high probability, the original sentence $\mathbf{x}_1$ is generated. For a sample $\mathbf{x}_2 \in \mathbf{X}_2$, though we do not aim to change its language style in our task, we can still compute its cycle consistency loss for the purpose of additional regularization. We first choose an arbitrary style $\mathbf{y}_1$ obtained from a sentence in $\mathbf{X}_1$, and transfer $\mathbf{x}_2$ into this $\mathbf{y}_1$ style. Next, we put this generated sentence into the encoder-decoder model with the style $\mathbf{y}^*$, and the original sentence $\mathbf{x}_2$ should be generated. Formally, the cycle consistency is:

$$\mathcal{L}_{cyc}(\boldsymbol{\theta}_{\mathbf{E_z}}, \boldsymbol{\theta}_{\mathbf{E_y}}, \boldsymbol{\theta}_{\mathbf{G}}) = \mathbb{E}_{\mathbf{x}_1 \sim \mathbf{X}_1}[-\log p_{\mathbf{G}}(\mathbf{x}_1 | \mathbf{E_z}(\tilde{\mathbf{x}}_1), \mathbf{y}_1)] + \mathbb{E}_{\mathbf{x}_2 \sim \mathbf{X}_2}[-\log p_{\mathbf{G}}(\mathbf{x}_2 | \mathbf{E_z}(\tilde{\mathbf{x}}_2), \mathbf{y}^*)]. \quad (10)$$

3.5 Full Objective

An illustration of our basic model with the style discrepancy loss is shown in Figure 1 and the full model is combined with the cycle consistency loss shown in Figure 2. To summarize, the full loss function of our model is:

$$\mathcal{L}(\boldsymbol{\theta}_{\mathbf{E_z}}, \boldsymbol{\theta}_{\mathbf{E_y}}, \boldsymbol{\theta}_{\mathbf{G}}, \boldsymbol{\theta}_{\mathbf{D}}) = \mathcal{L}_{rec} - \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{cyc} + \lambda_3 \mathcal{L}_{dis}, \quad (11)$$

where $\lambda_1, \lambda_2, \lambda_3$ are parameters balancing the relative importance of the different loss parts. The overall training objective is a minmax game played among the encoder $\mathbf{E_z}$, $\mathbf{E_y}$, generator $\mathbf{G}$ and discriminator $\mathbf{D}$:

$$\min_{\mathbf{E_z}, \mathbf{E_y}, \mathbf{G}} \max_{\mathbf{D}} \mathcal{L}(\boldsymbol{\theta}_{\mathbf{E_z}}, \boldsymbol{\theta}_{\mathbf{E_y}}, \boldsymbol{\theta}_{\mathbf{G}}, \boldsymbol{\theta}_{\mathbf{D}}). \quad (12)$$

We implement the encoder $\mathbf{E_z}$ using an RNN with the last hidden state as the content representation, and the style encoder $g(\mathbf{x})$ using a CNN with the output representation of the last layer as the style representation. The generator $\mathbf{G}$ is an RNN that takes the concatenation of the content and style representation as the initial hidden state. The discriminator $\mathbf{D}$ and the pre-trained discriminator $\mathbf{D}_s$ used in the style discrepancy loss are CNNs with the similar network structure in $\mathbf{E_y}$ followed by a sigmoid output layer.

4 Experiments

4.1 Datasets

Yelp: Raw data are from the *Yelp Dataset Challenge Round 10*, which are restaurant reviews on Yelp. Generally, reviews rated with 4 or 5 stars are considered positive, 1 or 2 stars are negative, and 3 stars are neutral. For positive and negative reviews, we use the processed data released by Shen et al. (2017). For neutral reviews, we follow similar steps in Shen et al. (2017) to process and select the data. We first filter out neutral reviews (rated with 3 stars and categorized with the keyword 'restaurant') with the length exceeding 15 or less than 3. Then, data selection in Moore & Lewis (2010) is used to ensure a large enough vocabulary overlap between neutral data and data in Shen et al. (2017). Afterwards, we sample 500k sentences from the resulting dataset as the neutral data. We use the positive data as the target style domain. Based on the three classes of data, we construct two datasets with multiple styles:

- Positive+Negative (Pos+Neg): we add different numbers of positive data (50k, 100k, 150k) into the negative data, so that the source domain contains data with two sentiments.
- Neutral+Negative (Neu+Neg): we combine neutral (50k, 100k, 150k) and negative data together. We consider these datasets are harder to learn from. For the Pos+Neg dataset, we can make use of a pre-trained classifier to possibly filter out some positive data so that most of the source data have the same style and the model in Shen et al. (2017) can work. However, the neutral data cannot be removed in this way. Also, most of the real data may be in the neutral sentiment, and we want to see if such sentences can be transferred well.

Details about the data statistics can be found in Table 7 in the Appendix.

Chat: We use sentences from a real Chinese dialog dataset as the source domain. Users can chat with various personalized language styles, which are not easy to be classified into one of the three

sentiments as in Yelp. Romantic sentences are collected from several online novel websites and filtered by human annotators. Our task is to transfer the dialog sentences with a romantic style, characterized by the selected romantic sentences. Table 8 in the Appendix shows detailed statistics about this dataset.

4.2 Configurations and Compared Methods

We implement our model using Tensorflow (Abadi et al., 2016). We use GRU as the encoder and generation cells in our encoder-decoder framework. Dropout (Srivastava et al., 2014) is applied in GRUs and the dropout probability is set to 0.5. Throughout our experiments, we set the dimension of the word embedding, content representation and style representation as 200, 1000 and 500 respectively. For the style encoder $g(\mathbf{x})$, we follow the CNN architecture in Kim (2014), and use filter sizes of $200 \times \{1, 2, 3, 4, 5\}$ with 100 feature maps each, so that the resulting output layer is of size 500, i.e., the dimension of the style representation. The pre-trained discriminator $\mathbf{D}_s$ is implemented similar to $g(\mathbf{x})$ but using filter sizes $200 \times \{2, 3, 4, 5\}$ with 250 feature maps each. Statistics of data used to pre-train $\mathbf{D}_s$ are shown in Table 9 and Table 11 in the Appendix. The testing accuracy of the pre-trained $\mathbf{D}_s$ is 95.82% for Yelp and 87.72% for Chat respectively. We further set the balancing parameters $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 5$, and train the model using the Adam optimizer (Kingma & Ba, 2015) with the learning rate 10^{-4}. All input sentences are padded so that they have the same length 20 for Yelp and 35 for Chat. Furthermore, we use the pre-trained word embeddings *Glove* (Pennington et al., 2014) for Yelp and the Chinese word embeddings trained on a large amount of Chinese news data for Chat when training the classifier.

We compare our method with Shen et al. (2017) which is the state-of-the-art language style transfer model with non-parallel data, and we name as Style Transfer Baseline (STB). As described in Section 2 and 3, STB is built upon an auto-encoder framework. It focuses on transferring sentences from one style to the other. The text styles are represented by two embedding vectors. It assumes source domain and target domain share a content space, and relies on adversarial training methods to align content spaces of two domains. We keep the configurations of the modules in STB, such as the encoder, decoder and discriminator, the same as ours for a fair comparison.

4.3 Evaluation Metrics

Following Shen et al. (2017), we use a model-based evaluation metric. Specifically, we use a pre-trained evaluation classifier to classify whether the transferred sentence has the correct style. The classifier is implemented same as the discriminator $\mathbf{D}_s$. Statistics of the data used for the evaluation classifier are shown in Table 10 and Table 12 in the Appendix. The testing accuracy of evaluation classifiers is 95.36% for Yelp and 87.05% for Chat. We repeat the training three times for each experiment setting and report the mean accuracy on the testing data with their standard deviation.

4.4 Results and Analysis

4.4.1 On Yelp

We first perform experiments on the source data containing both positive and negative reviews. In this setting, we specifically compare two versions of both STB and our model, one with the cycle consistency loss and one without, to validate the effectiveness of the cycle consistency loss [1]. Results are shown in Table 1. It can be seen that incorporating the cycle consistency loss improves the performance for both STB and our proposed model consistently.

Table 1: Testing accuracies on Yelp with Pos+Neg source data.

#positive samples used	STB	STB (with Cyc)	Ours (without Cyc)	Ours
50k	0.908 ± 0.060	0.917 ± 0.012	0.854 ± 0.044	$\mathbf{0.933 \pm 0.002}$
100k	0.703 ± 0.111	0.847 ± 0.011	0.868 ± 0.037	$\mathbf{0.928 \pm 0.003}$
150k	0.649 ± 0.041	0.676 ± 0.057	0.713 ± 0.075	$\mathbf{0.910 \pm 0.006}$

[1] Note that our proposed cycle consistency loss can be similarly added in STB.

We further manually examine the generated sentences for a detailed study of the various methods. Table 2 shows a few samples for the above setting with 150k positive samples used. Overall, our full model can generate grammatically correct positive reviews without changing the original content in more cases than the other methods. For some simple sentences such as the first example, all models perform well. For the second example in which the input sentence is more complex, both versions of STB and our basic model without the cycle consistency loss cannot generate fluent sentences, but our full model still succeeds. However, our model also suffers some mistakes as shown in the third example. Though it successfully makes the sentence positive, some additional information about the food is added, which is not discussed in the original sentence.

Table 2: Example sentences on Yelp transferred into a positive sentiment.

Original Sentence	service was tolerable .
STB	service was spectacular .
STB (with Cyc)	service was spectacular .
Ours (without Cyc)	service was spectacular .
Ours	service was outstanding .
Original Sentence	customer service manager asks what the problem is .
STB	customer service is just what it .
STB (with Cyc)	customer service , cares , great .
Ours (without Cyc)	customer service you everyone is like it .
Ours	customer service is wonderful and great .
Original Sentence	service has gotten worse and worse at this location .
STB	service is great for the family and family .
STB (with Cyc)	service has always great and at this location .
Ours (without Cyc)	service has been better than the best experience .
Ours	service was super friendly and food was great .

Next, we compare the results of STB and our proposed method in Table 1. As the number of positive sentences in the source data increases, the average performance of both versions of STB decreases drastically. This is reasonable because STB introduces a discriminator to align the sentences from the target domain back to the source domain, and when the source domain contains more positive samples, it is hard to find a good alignment to the source domain. Meanwhile the performance of our model, even the basic one without the cycle consistency loss, does not fluctuate much with the increase of the number of positive samples, showing that our model is not that sensitive to the source data containing more than one sentiments. Overall, our model with the cycle consistency loss performs the best.

The above setting is not challenging enough, because we can use a pre-trained discriminator similar to $\mathbf{D}_s$ in our model, to remove those samples classified as positive with high probabilities, so that only sentences with a less positive sentiment remain in the source domain. Thus, we test our second dataset which combines neutral reviews and negative reviews as the source domain. In this setting, in case that some positive sentences exist in those neutral reviews, when STB is trained, we use the same pre-trained discriminator in our model to filter out samples classified as positive with probabilities larger than 0.9. In comparison, our model utilizes all the data, since it naturally allows for those data with styles similar to the target style. In the following, we report and analyze both STB and our model with the cycle consistency loss added. The experimental results in Table 3 show that STB (with Cyc) suffers a large performance drop with 150k neutral data mixed in the source domain, while our model remains stable.

Table 3: Testing accuracies on Yelp with Neu+Neg source data.

#Neural samples used	STB (with Cyc)	Ours
50k	0.914 ± 0.007	$\mathbf{0.941 \pm 0.006}$
100k	$\mathbf{0.927 \pm 0.024}$	0.922 ± 0.008
150k	0.865 ± 0.016	$\mathbf{0.929 \pm 0.007}$

In real applications, there may be only a small amount of data in the target domain. To simulate this scenario, we limit the amount of the target data (randomly sampled from the positive data) used

for training, and evaluate the robustness of the compared methods. Table 4 shows the experimental results. It is surprising to see that both methods obtain relatively steady accuracies with different numbers of target samples. Yet, our model surpasses STB (with Cyc) in all the cases.

Table 4: Testing accuracies on Yelp with different numbers of target samples used.

#Target samples used	STB (with Cyc)	Ours
100k	0.785 ± 0.021	$\mathbf{0.859 \pm 0.003}$
150k	0.780 ± 0.014	$\mathbf{0.888 \pm 0.005}$
200k	0.763 ± 0.017	$\mathbf{0.880 \pm 0.003}$

4.4.2 ON CHAT

As in the Yelp experiment, we vary the number of target sentences to test the robustness of the compared methods. The experimental results are shown in Table 5. As can be seen, STB (with Cyc) obtains a relatively low performance with only 10k target samples, and as more target samples are used, its performance increases. However, the accuracy of our model is relatively high even with 10k target samples used, and remains stable in all the cases. Thus, our model achieves better performance as well as stronger robustness on Chat. A few examples are shown in Table 6. We can see that our model generally successfully transfers the sentence into a romantic style with some romantic phrases used.

Table 5: Testing accuracies on Chat with different numbers of target samples used.

#target samples used	STB (with Cyc)	Ours
10k	0.887 ± 0.002	$\mathbf{0.958 \pm 0.003}$
50k	0.920 ± 0.017	$\mathbf{0.975 \pm 0.003}$
100k	0.942 ± 0.003	$\mathbf{0.973 \pm 0.001}$
150k	0.955 ± 0.003	$\mathbf{0.966 \pm 0.002}$

Table 6: Example sentences on Chat transferred into a romantic style. English translations are provided (* denotes that the sentence has grammar mistakes in Chinese).

Original Sentence	回眸一笑 就 好 It is enough to look back and smile
STB (with Cyc)	回眸一笑 就 好 了 It would be just fine to look back and smile
Ours	回眸一笑 , 勿念 。 Look back and smile, please do not miss me.
Original Sentence	得过且过 吧 ！ Just live with it!
STB (with Cyc)	想不开 吧 , 我 的 吧 。 I just take things too hard. *
Ours	爱到深处 , 随遇而安 。 Love to the depths, enjoy myself wherever I am.
Original Sentence	自己 的 幸福 给 别人 了 Give up your happiness to others
STB (with Cyc)	自己 的 幸福 给 别人 , 你 的 。 Give up your happiness to others. *
Ours	自己 的 幸福 是 自己 , 自己 的 。 Leave some happiness to yourself, yourself.

5 CONCLUSION

In this paper, we present an encoder-decoder framework for language style transfer, which allows for the use of non-parallel data and source data with various unknown language styles. Each sentence is encoded into two latent representations, one corresponding to its content disentangled from the style and and the other representing the style only. By recombining the content with the target style, we can decode a sentence aligned in the target domain. Specifically, we propose two loss functions, i.e., the style discrepancy loss and the cycle consistency loss, to adequately constrain the encoding and decoding functions. The style discrepancy loss is to enforce a properly encoded style representation while the cycle consistency loss is used to ensure that the style-transferred sentences can be transferred back to their original sentences. Experimental results on two tasks demonstrate that our proposed method outperforms the state-of-the-art style transfer method (Shen et al., 2017).

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pp. 94–104, 2017.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the International Conference on Machine Learning*, pp. 1587–1596, 2017.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, 2014.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference for Learning Representations*, 2015.

Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pp. 2539–2547, 2015.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, pp. 4601–4609, 2016.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 994–1003, 2016.

Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 469–477, 2016.

Ryan Lowe, Nissan Pow, Iulian V Serban, and Joelle Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 285, 2015.

Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. *arXiv preprint arXiv:1703.07511*, 2017.

Robert C Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 220–224, 2010.

Jonas Mueller, David Gifford, and Tommi Jaakkola. Sequence to better sequence: continuous revision of combinatorial structures. In *Proceedings of the International Conference on Machine Learning*, pp. 2536–2544, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, 2015.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, 2017.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the International Conference on Computer Vision*, 2017.

6 APPENDIX

6.1 STATISTICS OF DATA

Table 7: Statistics of Yelp for the style transfer model

	Training	Test	Validation
Positive	240417	40000	20000
Negative	151026	40000	20000

Table 8: Statistics of Chat for the style-transfer model

	Training	Test	Validation
Romantic	207312	40000	40000
General	514460	40000	40000

Table 9: Statistics of Yelp for the discriminator $\mathbf{D}_s$

	Training	Test	Validation
Positive	75000	5000	2500
Negative	37500	5000	2500

Table 10: Statistics of Yelp for the evaluation classifier

	Training	Test	Validation
Positive	37500	1500	2250
Negative	18750	1500	2250

Table 11: Statistics of Chat for the discriminator $\mathbf{D}_s$

	Training	Test	Validation
Romantic	100000	10000	10000
General	200000	10000	10000

Table 12: Statistics of Chat for the evaluation classifier

	Training	Test	Validation
Romantic	50000	5000	5000
General	100000	5000	5000

Table 13: Statistics of Shakespeare for the discriminator $\mathbf{D}_s$

	Training	Test	Validation
Shakespeare	3500	500	500
Non-Shakespeare	7000	500	500

Table 14: Statistics of Shakespeare for the evaluation classifier

	Training	Test	Validation
Shakespeare	3500	500	500
Non-Shakespeare	7000	500	500

Table 15: Statistics of Shakespeare for the style transfer model

	Training	Test	Validation
Positive	21888	1000	2000
Negative	43800	1000	2000

6.2 HUMAN ANNOTATIONS

We randomly select 200 test samples from Yelp and perform human evaluations on four aspects of the results: (1) content: estimates if the content of an input sentence is preserved in a transferred sentence; content rating has 0 (changed), 1 (synonym substitution or partially changed), and 2 (unchanged); (2) sentiment: estimates if the sentiment of a transferred sentence is consistent with the target sentiment; sentiment rating has 0 (unchanged and wrong), 1 (changed but wrong), 2 (correct); (3) fluency: estimates the fluency of transferred sentences; fluency is rated from 1 (unreadable) to 4 (perfect); (4) overall: estimates the overall quality of transferred sentences; overall rating ranges from 1 (failed) to 4 (perfect).

We hired five annotators and averaged their evaluations. Table 16 shows results on Yelp when the source domain contains not only negative sentences but also 150k positive sentences (row 3 in

Table 1), and Table 17 shows results on Yelp when the target domain contains only 100k positive sentences (row 1 in Table 4). As can be seen, our model is better in terms of sentiment accuracy and overall quality, which is consistent with the automatic evaluation results.

Table 16: Human annotations on Yelp when 150k positive sentences are added to source domain.

	Content			Sentiment				
Model	0	1	2	0	1	2	Fluency	Overall
STB	0.355	0.298	0.347	0.261	0.266	0.473	$\mathbf{2.938 \pm 0.227}$	2.352 ± 0.288
STB with Cyc	0.282	0.294	**0.424**	0.217	0.250	0.533	2.716 ± 0.291	2.545 ± 0.206
Ours without Cyc	0.291	0.304	0.405	0.135	0.205	0.660	2.859 ± 0.319	2.771 ± 0.290
Ours	0.310	0.345	0.345	0.111	0.177	**0.712**	2.812 ± 0.297	$\mathbf{2.805 \pm 0.314}$

Table 17: Human annotations on Yelp when only 100k positive sentences are used.

	Content			Sentiment				
Model	0	1	2	0	1	2	Fluency	Overall
STB with Cyc	0.214	0.319	**0.467**	0.195	0.235	0.570	$\mathbf{2.693 \pm 0.198}$	2.656 ± 0.168
Ours	0.261	0.413	0.326	0.179	0.169	**0.652**	2.613 ± 0.253	$\mathbf{2.699 \pm 0.328}$

6.3 EXAMPLE SENTENCES

Table 18: Example sentences on Yelp transferred into a positive sentiment. This is a supplement to Table 1.

Original Sentence	it is rather bad .
STB	it is pretty cool .
STB (with Cyc)	it is a cool .
Ours (without Cyc)	it is good food .
Ours	it is really good .
Original Sentence	i have tried to go to them twice silly me .
STB	i 'm going to anyone when they need to .
STB (with Cyc)	i 've been to go here for years out .
Ours (without Cyc)	i have recommend to anyone 's your home needs .
Ours	i have tried the place and it was great .
Original Sentence	i wish i could give zero stars .
STB	i wish i could give it .
STB (with Cyc)	i wish i could give them stars .
Ours (without Cyc)	i wish i can give you again .
Ours	i recommend this place to anyone .
Original Sentence	and just not very good .
STB	but i was very good .
STB (with Cyc)	and just always very good .
Ours (without Cyc)	and just always very good .
Ours	and i love it .

6.4 SHAKESPEARE

We experiment on revising modern text in the language of Shakespeare at the sentence-level as in Mueller et al. (2017). Following their experimental setup, we collect 29388 sentences authored by Shakespeare and 54800 sentences from non-Shakespeare-authored works. The length of all the sentences ranges from 3 to 15. Statistics of data for training and evaluating the style transfer model

are shown in Table 13, 14, and 15 in Section 6.1. Since the dataset is small, we train the discriminator $\mathbf{D}_s$ using a subset of the data for training the style transfer model. The testing accuracy of $\mathbf{D}_s$ is 87.6%. The evaluation classifier has a testing accuracy 88.7%.

Our model achieves a classification accuracy of 95.1% and STB with cycle consistency loss achieves 94.1%. Following are some examples.

Table 19: Example non-Shakespeare sentences transferred into a Shakespeare's language style.

Original Sentence	she is a sweet creature !
STB (with Cyc)	i are , coward !
Ours	she will be welcome .
Original Sentence	what 's this , papa ?
STB (with Cyc)	what 's your name ?
Ours	what 's your matter ?
Original Sentence	i leave you to your own reflections .
STB (with Cyc)	i am , your lord .
Ours	i leave you , sir .
Original Sentence	how would you ever see her again ?
STB (with Cyc)	how do i call thee ?
Ours	how would you love of her ?
Original Sentence	i should never have thought of such a thing .
STB (with Cyc)	i shall not have to for you .
Ours	i will never be thee , sir .

Compared with STB, our model can generate sentences which are more fluent and have a higher probability to have a correct target style. However, we find that both STB and our model tend to generate short sentences and change the content of source sentences in more cases in this set of experiment than in the Yelp and Chat datasets. We conjecture this is caused by the scarcity of training data. Sentences in the Shakespeare's style form a vocabulary of 8559 words, but almost 60% of them appear less than 10 times. On the other hand, the source domain contains 19962 words, but there are only 5211 common words in these two vocabularies. Thus aligned words/phrases may not exist in the dataset.