Utilizing Non-Parallel Text for Style Transfer by Making Partial Comparisons

Di Yin, Shujian Huang*, **Xin-Yu Dai** and **Jiajun Chen**

National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
yind@smail.nju.edu.cn, {huangsj, daixinyu, chenjj}@nju.edu.cn

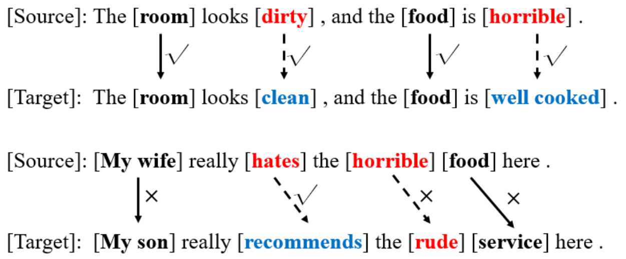
Abstract

Text style transfer aims to rephrase a given sentence into a different style without changing its original content. Since parallel corpora (i.e. sentence pairs with the same content but different styles) are usually unavailable, most previous works solely guide the transfer process with distributional information, i.e. using style-related classifiers or language models, which neglect the correspondence of instances, leading to poor transfer performance, especially for the content preservation. In this paper, we propose making partial comparisons to explicitly model the content and style correspondence of instances, respectively. To train the partial comparators, we propose methods to extract partial-parallel training instances automatically from the non-parallel data, and to further enhance the training process by data augmentation. We perform experiments to compare our method to other existing approaches on two review datasets. Both automatic and manual evaluations show that our approach can significantly improve the performance of existing adversarial methods, and outperforms most state-of-the-art models. Our code and data will be available on Github[1].

1 Introduction

The style of a text conveys important information beyond its literal meaning [Hovy, 1987]. The ability to take control over some style attributes (e.g. sentiment, formality) of the generated text is essential to make language generation systems more intelligent, and is potentially useful in many applications, such as dialogue systems [Niu and Bansal, 2018] and image captioning [Mathews *et al.*, 2018]. More specifically, text style transfer aims to rephrase a given sentence into a different style (e.g. transform the sentiment from negative to positive) without changing the main content of the original sentence (e.g. the aspects be discussed) (Figure 1).

Similar to neural machine translation [Sutskever *et al.*, 2014; Bahdanau *et al.*, 2015], one possible solution to text style transfer is to train sequence-to-sequence models with parallel corpora [Xu *et al.*, 2012; Rao and Tetreault, 2018]. However, since parallel corpora are usually unavailable for most scenarios, some researchers have proposed approaches for building style transfer systems using non-parallel corpora only [Hu *et al.*, 2017; Shen *et al.*, 2017; Fu *et al.*, 2018; Yang *et al.*, 2018; Li *et al.*, 2018a]. Most of them try to disentangle style attributes and style-independent content using Conditional Generative Adversarial Nets (Conditional-GANs) [Goodfellow *et al.*, 2014; Mirza and Osindero, 2014]. Ideally, the adversarial training includes a generator to generate the transferred sentence, and a discriminator to decide whether this transferred sentence is correct, i.e. whether it has the same content and different style compared to the original sentence.


[Source]: The [**room**] looks [**dirty**] , and the [**food**] is [**horrible**] .

[Target]: The [**room**] looks [**clean**] , and the [**food**] is [**well cooked**] .

[Source]: [**My wife**] really [**hates**] the [**horrible**] [**food**] here .

[Target]: [**My son**] really [**recommends**] the [**rude**] [**service**] here .

Figure 1: Examples of text style transfer (e.g. changing the sentiment attribute of a review from negative to positive). The dashed line indicates the transformation of style-related words from the source style to the target style. The solid line indicates the preservation of content-related words. In this figure, the first example is a successful case of style transfer, and the second one includes some wrong transformations.

Again, due to the lack of parallel data, training such a discriminator is unfeasible. As a compromise, previous work guide their training with style-related distributional information, e.g. style-related binary classifiers [Shen *et al.*, 2017] or language models [Yang *et al.*, 2018]. However, even with such an extra process to reconstruct the original sentence, the transfer performance of their model is still weak, especially for content preservation [Li *et al.*, 2018a], because the distributional information is not enough for the adversarial discriminator to decide whether two sentences have the same content.

In this paper, we analyze the Conditional GAN framework

[*] Shujian Huang is the corresponding author

[1] https://github.com/yd1996/PartialComparison

(Section 2), and emphasize that the instance-level comparison between original and transferred sentences is necessary during the training process. Without parallel data to make a complete comparison of both content and style, we propose **partial comparators** to guide the adversarial training process by making **partial comparisons** (Section 3). Each partial comparator aims to model only one kind of correspondence, either content or style. To train these comparators, we propose a simple but effective method to automatically extract initial training instances with high quality from the non-parallel data. To take advantage of all the non-parallel training data, we propose to further enhance the training process by data augmentation.

To demonstrate the effectiveness of our method, we perform experiments on two review datasets to compare our method to other existing approaches (Section 4). Results of automatic and human evaluation show that our approach can significantly improve the performance of existing adversarial methods, and outperforms most state-of-the-art models. We also provide analysis about how the proposed method utilizes the non-parallel data (Section 5).

2 Non-parallel Text Style Transfer

2.1 Problem Formulation

Given two text datasets $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(\mathrm{m})}\}$ and $\mathbf{Y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(\mathrm{n})}\}$ from different styles $\mathbf{v}_x$ and $\mathbf{v}_y$, respectively, we call $\mathbf{X}$, $\mathbf{Y}$ non-parallel datasets where no pairs of $(\mathbf{x}^{(\mathrm{i})}, \mathbf{y}^{(\mathrm{i})})$ with the same content can be directly fetched. Our task is to learn a style transfer model on this kind of datasets, which can generate a new sentence with a different style attribute conditioned on a given sentence, without changing the original content.

Most previous work [Hu *et al.*, 2017; Shen *et al.*, 2017; Yang *et al.*, 2018] assume all texts are generated conditioned on two disentangled representations, the style $\mathbf{v}$ and the content $\mathbf{z}$. The transfer process can be formulated as follows: first, an encoder $\mathbf{E}$ encodes $\mathbf{x}$ into the latent representation $\mathbf{z} = \mathbf{E}(\mathbf{x}, \mathbf{v_x})$, from which the information about the original style $\mathbf{v}_x$ has been removed; then, conditioned on $\mathbf{z}$ and the target style $\mathbf{v}_y$, the generator $\mathbf{G}$ produces a new sentence $\mathbf{y} = \mathbf{G}(\mathbf{z}, \mathbf{v_y})$. The same process could go in the other direction, and these dual processes can be formulated as follows.

$$p(\mathbf{y}|\mathbf{x}) = \int_{\mathbf{z}} p_{\mathbf{G}}(\mathbf{y}|\mathbf{z}, \mathbf{v_y}) p_{\mathbf{E}}(\mathbf{z}|\mathbf{x}, \mathbf{v_x}) \mathrm{d}\mathbf{z} \tag{1}$$

$$p(\mathbf{x}|\mathbf{y}) = \int_{\mathbf{z}} p_{\mathbf{G}}(\mathbf{x}|\mathbf{z}, \mathbf{v_x}) p_{\mathbf{E}}(\mathbf{z}|\mathbf{y}, \mathbf{v_y}) \mathrm{d}\mathbf{z} \tag{2}$$

2.2 Adversarial Training

As is shown in the left part of Figure 2, the training of previous GAN-based models [Shen *et al.*, 2017; Yang *et al.*, 2018] often relies on the following two separate processes:

The Reconstruction Process

Previous methods try to disentangle style attributes and style-independent content using an auto-encoder model. In order to preserve the main content, the style transfer model should have the ability to reconstruct the original sentence from the disentangled representations of content and the original style.

Formally, the encoder $\mathbf{E}$ firstly encodes the sentence $\mathbf{x}$, given style $\mathbf{v_x}$, into a style-independent representation. Then, the generator $\mathbf{G}$ reconstructs $\mathbf{x}$ conditioned on $\mathbf{z_x}$ and $\mathbf{v_x}$. Same for the other direction. The corresponding reconstruction loss is as follows.

$$\begin{aligned}\mathcal{L}_{\mathrm{rec}}(\theta_{\mathbf{E}}, \theta_{\mathbf{G}}) = E_{\mathbf{x}\sim p(\mathbf{X})}[-\log p_{\mathbf{G}}(\mathbf{x}|\mathbf{E}(\mathbf{x}, \mathbf{v_x}), \mathbf{v_x})] \\ +E_{\mathbf{y}\sim p(\mathbf{Y})}[-\log p_{\mathbf{G}}(\mathbf{y}|\mathbf{E}(\mathbf{y}, \mathbf{v_y}), \mathbf{v_y})]\end{aligned} \tag{3}$$

Note that although the reconstruction process aims at preserving the main content, the process is only trained with the input of the original style, while the content preservation when transferring to a different style is still not under control.

The Style Transfer Process

The transfer process performs a real style transfer. It uses the same process to get $\mathbf{z_x}$ and $\mathbf{z_y}$ as in the reconstruction process, and generates $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{x}}$, respectively.

To guide the style transfer process, Shen *et al.* [2017] and Yang *et al.* [2018] use adversarial training to align several distribution pairs. The discriminator $\mathbf{D_z}$, which aims to distinguish between $\mathbf{z_x}$ and $\mathbf{z_y}$, is introduced to align the distributions $p(\mathbf{z_x})$ and $p(\mathbf{z_y})$ in an adversarial way.

$$\begin{aligned}\mathcal{L}^{\mathbf{z}}_{\mathrm{adv}}(\theta_{\mathbf{E}}, \theta_{\mathbf{D_z}}) = E_{\mathbf{x}\sim p(\mathbf{X}), \mathbf{z_x}\sim p_{\mathbf{E}}(\mathbf{z}|\mathbf{x}, \mathbf{v_x})}[-\log \mathbf{D_z}(\mathbf{z_x})] \\ +E_{\mathbf{y}\sim p(\mathbf{Y}), \mathbf{z_y}\sim p_{\mathbf{E}}(\mathbf{z}|\mathbf{y}, \mathbf{v_y})}[-\log(1 - \mathbf{D_z}(\mathbf{z_y}))]\end{aligned} \tag{4}$$

The discriminators $\mathbf{D_x}$ and $\mathbf{D_y}$ are introduced to align distribution of real and fake sentences via adversarial training: $\mathbf{D_x}$ distinguishes between $\mathbf{x}$ and $\tilde{\mathbf{x}}$, and $\mathbf{D_y}$ distinguishes between $\mathbf{y}$ and $\tilde{\mathbf{y}}$. These discriminators can be binary classifiers [Shen *et al.*, 2017] or language models [Yang *et al.*, 2018]. The adversarial objectives of them are as follows.

$$\begin{aligned}\mathcal{L}^{\mathbf{x}}_{\mathrm{adv}}(\theta_{\mathbf{E}}, \theta_{\mathbf{G}}, \theta_{\mathbf{D_x}}) = E_{\mathbf{x}\sim p(\mathbf{X})}[-\log \mathbf{D_x}(\mathbf{x})] \\ +E_{\mathbf{y}\sim p(\mathbf{Y}), \tilde{\mathbf{x}}\sim p(\mathbf{x}|\mathbf{y})}[-\log(1 - \mathbf{D_x}(\tilde{\mathbf{x}}))]\end{aligned} \tag{5}$$

$$\begin{aligned}\mathcal{L}^{\mathbf{y}}_{\mathrm{adv}}(\theta_{\mathbf{E}}, \theta_{\mathbf{G}}, \theta_{\mathbf{D_y}}) = E_{\mathbf{y}\sim p(\mathbf{Y})}[-\log \mathbf{D_y}(\mathbf{y})] \\ +E_{\mathbf{x}\sim p(\mathbf{X}), \tilde{\mathbf{y}}\sim p(\mathbf{y}|\mathbf{x})}[-\log(1 - \mathbf{D_y}(\tilde{\mathbf{y}}))]\end{aligned} \tag{6}$$

The overall training objective is a min-max game played among the encoder $\mathbf{E}$, the generator $\mathbf{G}$ and the discriminators $\mathbf{D_z}$, $\mathbf{D_x}$, $\mathbf{D_y}$, and it can be formulated as follows.

$$\min_{\mathbf{E},\mathbf{G}} \max_{\mathbf{D}_z, \mathbf{D}_x, \mathbf{D}_y} \mathcal{L}_{\mathrm{rec}} - \lambda(\mathcal{L}^{\mathbf{z}}_{\mathrm{adv}} + \mathcal{L}^{\mathbf{x}}_{\mathrm{adv}} + \mathcal{L}^{\mathbf{y}}_{\mathrm{adv}}) \tag{7}$$

Distributions v.s. Instances

Although aligning the distribution in a adversary way is an effective approach to building the distributional correspondence, this distributional correspondence is still not enough for content preservation, because there is no guarantee that two sentences have the same content even if they come from the same distribution. The only solution is to explicitly compare the content and style of two sentences, in order to build correspondences between instances instead of distributions.

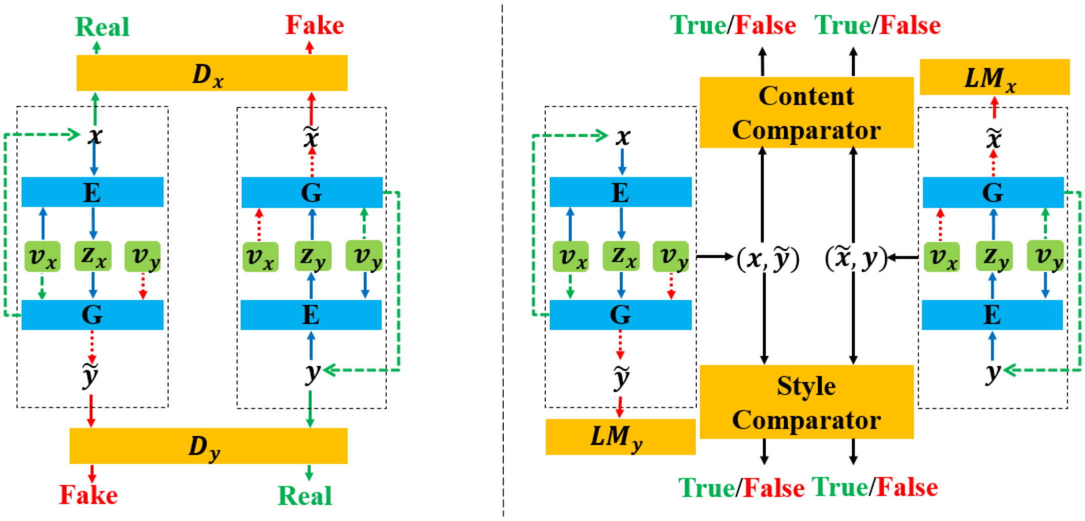


Figure 2: An illustration of previous GAN-based methods (left) and our method (right). The dashed line represents the reconstruction process and the doted line represents the style transfer process.

3 Making Partial Comparisons

As discussed before, without parallel data, it's hard to train a discriminator to directly make complete comparison between two sentences to decide whether they not only have the same content, but also belong to different styles. Therefore, we propose to make partial comparisons, which means that the comparison between two sentences is made in only one aspect, either content or style.

To make these two kinds of partial comparisons during the training process, we introduce two **partial comparators**, i.e. the **content comparator** and the **style comparator** (denoted as $\mathbf{D_c}$ and $\mathbf{D_s}$, respectively). Given two sentences, the content comparator $\mathbf{D_c}$ judges whether they share the same content, and the style comparator $\mathbf{D_s}$ judges whether they have different styles.

3.1 Adversarial Training with Partial Comparators

Jointly with $\mathbf{D_c}$ and $\mathbf{D_s}$, the transfer process could be guided via adversarial training. Taking the content comparator as an example, the adversarial objective is as follows,

$$\begin{aligned}\mathcal{L}^{\mathbf{c}}_{\text{adv}}(\theta_{\mathbf{E}},\theta_{\mathbf{G}},\theta_{\mathbf{D_c}}) = E_{(\mathbf{x},\mathbf{y})\sim p_{\mathbf{c}}(\mathbf{X},\mathbf{Y})}[-\log \mathbf{D_c}(\mathbf{x},\mathbf{y})]\\ +\alpha E_{\mathbf{x}\sim p(\mathbf{X})}[-\log(1-\mathbf{D_c}(\mathbf{x},\mathbf{G}(\mathbf{E}(\mathbf{x},\mathbf{v_x}),\mathbf{v_y}))]\\ +(1-\alpha)E_{\mathbf{y}\sim p(\mathbf{Y})}[-\log(1-\mathbf{D_c}(\mathbf{G}(\mathbf{E}(\mathbf{y},\mathbf{v_y}),\mathbf{v_x}),\mathbf{y})]\end{aligned} \tag{8}$$

where the first term is the likelihood of sentence pairs that have the same content; the second term is the likelihood of the fake sentence pairs, consisting of the original sentence $\mathbf{x}$ and the generated sentence $\tilde{\mathbf{y}}$; similar for the third term.

The adversarial objective for the style comparator is similar as follows.

$$\begin{aligned}\mathcal{L}^{\mathbf{s}}_{\text{adv}}(\theta_{\mathbf{E}},\theta_{\mathbf{G}},\theta_{\mathbf{D}_s}) = E_{\mathbf{x}\sim p(\mathbf{X}),\mathbf{y}\sim p(\mathbf{Y})}[-\log \mathbf{D_s}(\mathbf{x},\mathbf{y})]\\ +\beta E_{\mathbf{x}\sim p(\mathbf{X})}[-\log(1-\mathbf{D_s}(\mathbf{x},\mathbf{G}(\mathbf{E}(\mathbf{x},\mathbf{v_x}),\mathbf{v_y}))]\\ +(1-\beta)E_{\mathbf{y}\sim p(\mathbf{Y})}[-\log(1-\mathbf{D_s}(\mathbf{G}(\mathbf{E}(\mathbf{y},\mathbf{v_y}),\mathbf{v_x}),\mathbf{y})]\end{aligned} \tag{9}$$

Working together, the two comparators could accomplish the comparison between two sentences, while each one of them is easy to be built and trained. We will introduce the modeling of the partial comparators and the training of them in the following subsections.

Note that $\mathbf{D_c}$ and $\mathbf{D_s}$ could model instance-level correspondences between two sentences in content and style, but cannot ensure the distributional correspondence and their fluency. Therefore, we also introduce two language models pretrained by the sentences from each style into our framework, inspired by Yang *et al.* [2018] .

$$\begin{aligned}&\mathcal{L}_{\mathbf{LM}}(\theta_{\mathbf{E}},\theta_{\mathbf{G}},\theta_{\mathbf{LM_x}},\theta_{\mathbf{LM_x}})\\ &=\mathcal{L}^{\mathbf{x}}_{\mathbf{LM}}(\theta_{\mathbf{E}},\theta_{\mathbf{G}},\theta_{\mathbf{LM_x}})+\mathcal{L}^{\mathbf{y}}_{\mathbf{LM}}(\theta_{\mathbf{E}},\theta_{\mathbf{G}},\theta_{\mathbf{LM_y}})\\ &=E_{\mathbf{y}\sim p(\mathbf{Y}),\tilde{\mathbf{x}}\sim p(\mathbf{x}|\mathbf{y})}[\log p_{\mathbf{LM}_x}(\tilde{\mathbf{x}})]\\ &+E_{\mathbf{x}\sim p(\mathbf{X}),\tilde{\mathbf{y}}\sim p(\mathbf{y}|\mathbf{x})}[\log p_{\mathbf{LM_y}}(\tilde{\mathbf{y}})]\end{aligned} \tag{10}$$

The overall framework of our proposed method is illustrated in the right part of Figure 2. The min-max game is formulated as follows.

$$\begin{aligned}\min_{\mathbf{E},\mathbf{G}}\max_{\mathbf{D_z},\mathbf{D}_c,\mathbf{D}_s}\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} - \lambda_{\text{c}}\mathcal{L}^{\mathbf{c}}_{\text{adv}} - \lambda_{\text{s}}\mathcal{L}^{\mathbf{s}}_{\text{adv}}\\ - \lambda_{\text{z}}\mathcal{L}^{\mathbf{z}}_{\text{adv}} - \lambda_{\text{lm}}\mathcal{L}_{\mathbf{LM}}\end{aligned} \tag{11}$$

3.2 Partial Comparators as Text Matching Models

Partial comparison is similar to text matching, which is to decide whether two texts are relevant or not. Therefore, we borrow some techniques from the area of sentence pair modeling [Hu *et al.*, 2014] to implement two kinds of partial comparator. In most tasks of text style transfer, such as sentiment modification, we judge whether two sentences share the same content by observing whether there exists some correspondences of keywords (e.g. the food name in restaurant reviews) between them, so we employ the sentence interaction model ARC-II [Hu *et al.*, 2014] as the content comparator. At the same time, since the style of a sentence is a global attribute, we employ a sentence encoding model like ARC-I in Hu *et al.* [2014] as the style comparator $\mathbf{D_s}$. Please refer to their original paper for details.

3.3 Training Data for Partial Comparators

Training instances are needed to train the two partial comparators. The partial-parallel training instances for the style comparator $\mathbf{D_s}$ is sentences pairs that have different style, which is easy to get by simply sampling from X and Y.

The partial-parallel training instances for the content comparator $\mathbf{D_c}$ are sentence-pairs that have similar content. Because the content of a sentence are usually carried by words, we mine the instances from the union of X and Y using lexical clues.

Inspired by Li *et al.* [2018a] , we extract noun words as keywords of each sentence according to the automatic POS tags and group sentences with the same keywords. Within each group, we calculate the edit distance between every two sentences and collect those sentence pairs with an edit distance lower than a given threshold. These sentence pairs are considered to have same content.

Although the union of X and Y could be mined for $\mathbf{D_c}$, the coverage of the above mining process is still low[2].

To take full advantage of these uncovered sentences, we perform data augmentation (DA) to further improve the adversarial training. During the training procedure, we use the same procedure as we extract the initial partial-parallel training instances on the automatically generated sentences. In other words, if the generated sentence $\tilde{\mathbf{y}}$ has the same keywords with the original sentence $\mathbf{x}$, and their edit distance is within a threshold, we add the pair ($\mathbf{x}$, $\tilde{\mathbf{y}}$) into the partial-parallel training data. Similar for ($\tilde{\mathbf{x}}$, $\mathbf{y}$).

3.4 Training

Due to the discreteness of texts, gradients cannot be directly propagated from discriminators to the style transfer model. One possible solution to this problem is to use the REINFORCE [Sutton *et al.*, 2000] algorithm. However, previous work [Yu *et al.*, 2017] shows that this way suffers from high variance. We choose to use a Gumbel-Softmax [Jang *et al.*, 2016] distribution as input to the generator and the discriminators, instead of a single sampled word, which makes the training process differentiable.

$$p_t = \frac{\exp((\log \pi_i + g_i)/\tau)}{\sum_{j=1}^{V} \exp((\log \pi_j + g_j)/\tau)} \tag{12}$$

where the g_i's are independent samples from Gumbel(0,1). During training, we use an annealing strategy to update the value of τ. The initial value of τ is set to 1.0 and it decays by half every epoch until reaching the minimum value of 0.001.

Algorithm 1 Making partial comparisons

```
Input:
    Two non-parallel corpora $\mathbf{X}, \mathbf{Y}$ of different styles $\mathbf{v}_x, \mathbf{v}_y$;
    Partial-parallel training data for content comparator $(\mathbf{X}, \mathbf{Y}, l_c)$;
    Pretrained language model $\mathbf{LM}_x$ and $\mathbf{LM}_y$;
    Lagrange multiplier $\lambda_{\mathrm{z}}$, $\lambda_{\mathrm{c}}$, $\lambda_{\mathrm{s}}$, $\lambda_{\mathrm{LM}}$, temperature $\tau$.
Initialize $\theta_{\mathbf{E}}$, $\theta_{\mathbf{G}}$, $\theta_{\mathbf{D}_z}$, $\theta_{\mathbf{D}_c}$, $\theta_{\mathbf{D}_s}$.
repeat
    for t = 1, $\ldots$, $n_{\mathrm{critic}}$ do
        Sample $\{x^{(i)}\}_{i=1}^m \sim p(\mathbf{X})$ and $\{y^{(i)}\}_{i=1}^m \sim p(\mathbf{Y})$.
        for p = x, y; q=y, x do
            $\{z_p^{(i)}\}_{i=1}^m = \{\mathbf{E}(p^{(i)}, \mathbf{v}_p)\}_{i=1}^m$.
            $\{\tilde{q}^{(i)}\}_{i=1}^m = \{\mathbf{G}(z_p^{(i)}, \mathbf{v}_q)\}_{i=1}^m$.
        end for
        Sample $\{(x, y, l)\}_{i=1}^m \sim p_c(\mathbf{X}, \mathbf{Y}, l_c)$.
        Calculate $\mathcal{L}_{\mathrm{adv}}^{\mathbf{z}}$ according to (4), and update $\theta_{\mathbf{D}_z}$.
        Calculate $\mathcal{L}_{\mathrm{adv}}^{\mathbf{s}}$ according to (9) , and update $\theta_{\mathbf{D}_s}$.
        Data augmentation (DA) for $\mathbf{D_c}$.
        Calculate $\mathcal{L}_{\mathrm{adv}}^{\mathbf{c}}$ according to (8) , and update $\theta_{\mathbf{D}_c}$.
    end for
    Sample $\{x^{(i)}\}_{i=1}^m \sim p(\mathbf{X})$ and $\{y^{(i)}\}_{i=1}^m \sim p(\mathbf{Y})$.
    for p = x, y; q=y, x do
        $\{z_p^{(i)}\}_{i=1}^m = \{\mathbf{E}(p^{(i)}, \mathbf{v}_p)\}_{i=1}^m$.
        $\{\tilde{q}^{(i)}\}_{i=1}^m = \{\mathbf{G}(z_p^{(i)}, \mathbf{v}_q)\}_{i=1}^m$.
    end for
    Calculate $\mathcal{L}_{\mathrm{total}}$ according to (11), and update $\theta_{\mathbf{E}}, \theta_{\mathbf{G}}$.
until convergence
Output: A text style transfer model, which consists of the
    encoder $\mathbf{E}$ and the generator $\mathbf{G}$.
```

We follow the training procedure proposed in WGAN [Arjovsky *et al.*, 2017]; as is shown in Algorithm 1, we train the partial comparators n_{critic} steps, then one step on the style transfer model. We use the Adam [Kingma and Ba, 2015] optimization algorithm to train the style transfer model and RMSprop [Tieleman and Hinton, 2012] for partial comparators.

4 Experiment

4.1 Experiment Setup

We perform experiments two review datasets to evaluate our model. For convenient comparison, we follow the previous setting and focus only on generating short texts(shorter than 20 words).

Datasets

We conduct experiments on the **Yelp** review dataset and **Amazon** review dataset (Table 1) released by [Li *et al.*, 2018a], with the same pre-processing steps.

Parameter Setting

The encoder $\mathbf{E}$ and the generator $\mathbf{G}$ are single-layer LSTM-RNNs with input dimension of 300 and hidden dimensions

[2] Our empirical study shows that more than 70% of the original sentences do not have a proper content matching sentence in our training data.

Dataset	**Attributes**	**Train**	**Dev**	**Test**
Yelp	Positive	270K	2000	500
	Negative	180K	2000	500
Amazon	Positive	277K	985	500
	Negative	278K	1015	500

Table 1: Statistics of the datasets.

of 350. The dimension of style embedding is 50. The word embeddings are pretrained using Word2Vec[3]. The discriminators $\mathbf{D}_s$ and $\mathbf{D}_c$ both have two layers of convolution and max-pooling. We use a batch size of 160, which contains 80 samples from $\mathbf{X}$ and $\mathbf{Y}$ respectively. Hyper-parameters are selected based on the validation set, and we use grid search to pick the best parameters. The learning rate is selected from $[1e-4, 2e-4, 5e-4, 1e-3]$, and the weights of each term in the training objective $(\lambda_z, \lambda_c, \lambda_s, \lambda_{lm})$ are all selected from $[0.1, 0.5, 1.0, 2.0, 5.0]$. We implement our model based on PyTorch[4] and use four NVIDIA GTX1080Ti graphic cards for learning. Our source code and data will be released[5].

4.2 Automatic Evaluation

Two key aspects need to be evaluated: **attribute transfer** and **content preservation**. We follow previous work [Hu *et al.*, 2017; Shen *et al.*, 2017; Yang *et al.*, 2018], and use a pre-trained CNN-based sentence classifier to measure whether transferred sentences have the correct style (sentiment in the task of sentiment modification). The results are reported in accuracy (**ACC**). To evaluate the degree of content preservation, we calculate **BLEU** scores using the human annotated sentences provided by Li *et al.* [2018a] as the ground truth of transferred sentences.

The result are in Table 2. We can see that our model has a better overall performance than most state-of-the-art models in automatic evaluation and achieves an large improvement over previous GAN-based methods [Shen *et al.*, 2017; Yang *et al.*, 2018] in content preservation.

4.3 Ablation Study

We conduct ablation study to evaluate the contribution of each component in our training framework (Table 3). The baseline is equipped with two separate binary style classifiers, similar to Shen *et al.* [2017] , but with higher performance.

Firstly, it is interesting to see that replace the two style classifiers with our style comparator already leads to an improvement, indicating that even for style the instance level correspondence is more helpful. Secondly, using our content comparator improves the content preservation of model by a considerable margin. Thirdly, adding the style language model still improves our model, suggesting that our improvement is orthogonal to Yang *et al.* [2018] . Finally, using data augmentation further improves the results again, and achieves the best results on both the two datasets.

	Yelp		**Amazon**	
Model	**ACC**	**BLEU**	**ACC**	**BLEU**
[Hu *et al.*, 2017]	86.3	3.4	69.5	2.9
[Shen *et al.*, 2017]	80.5	4.8	58.7	5.1
[Fu *et al.*, 2018]				
StyleEmbedding	10.3	14.2	9.8	18.6
MultiDecoder	46.7	13.4	32.6	19.0
[Li *et al.*, 2018a]				
DeleteOnly	88.9	11.8	49.6	23.4
TemplateBased	86.4	19.0	66.9	25.6
RetrievalOnly	**94.8**	1.4	74.5	1.3
DeleteAndRetrieval	91.2	12.8	52.3	21.7
[Yang *et al.*, 2018]				
LM	84.9	14.0	68.5	15.2
LM + Classifier	89.5	20.9	70.5	25.8
Our model	92.7	**23.5**	**74.8**	**25.9**

Table 2: Performances of our model and some baselines on two datasets.

	Yelp		**Amazon**	
Model	**ACC**	**BLEU**	**ACC**	**BLEU**
+ $\mathbf{D_x}$ + $\mathbf{D_y}$	85.9	5.4	60.2	4.9
+ $\mathbf{D_s}$	86.4	6.9	61.4	5.7
+ $\mathbf{D_s}$ + $\mathbf{D_c}$	87.7	17.1	64.3	15.5
+ $\mathbf{D_s}$ + $\mathbf{D_c}$ + LM	90.5	21.2	70.4	22.4
+ $\mathbf{D_s}$ + $\mathbf{D_c}$ + DA	91.9	22.3	72.1	25.3
+ $\mathbf{D_s}$ + $\mathbf{D_c}$ + LM + DA	92.7	23.5	74.8	25.9

Table 3: The result of ablation study. 'LM' means style language models, and 'DA' means data augmentation.

4.4 Manual Evaluation

We also conduct manual evaluation on the generated result. For each test set, we randomly sample 100 sentences and collect the transfer results. Annotators are asked to score the generated sentences in three aspects: attribute transfer, content preservation and language fluency. For each aspect, the score ranges from 0 to 2, where 0 means failure, 1 means partial success and 2 means success. The result of manual evaluation is shown in Table 4, and we can see that our model outperforms other models in all aspects.

4.5 Case Study

We further analyze the results generated by different models. Table 5 shows some typical outputs of each system on the **Yelp** dataset. [Hu *et al.*, 2017] may change the sentiment correctly, but for most cases it also changes the original content. [Shen *et al.*, 2017] often changes the original content as well. Sometimes it generates a complete different sentence (e.g. 'they are worth five stars!'). MultiDecoder [Fu *et al.*, 2018] does not succeed in the transferring for the given cases. DeleteAndRetrieval, a simple method based on replacement [Li *et al.*, 2018a], can effectively preserve the subject of the review, but changed the meaning in some case (e.g. from 'overcooked' to 'beat feet out of there'). [Yang *et al.*, 2018] improves the content preservation, but tends to generate shorter sentences (e.g. 'the escargot was mediocre at best.'), which will change the syntactic structure of original sentences. On the contrary, our model produce reasonable

[3] https://radimrehurek.com/gensim/models/word2vec.html

[4] https://pytorch.org/

[5] https://github.com/yd1996/PartialComparison

Yelp	**Style**	**Content**	**Fluency**
[Hu *et al.*, 2017]	1.337	0.898	1.190
[Shen *et al.*, 2017]	0.922	0.873	0.629
MultiDecoder	0.166	1.629	1.337
DeleteAndRetrieval	1.297	1.574	1.451
LM + Classifier	0.898	1.312	1.290
Our model	**1.532**	**1.702**	**1.483**
Amazon	**Style**	**Content**	**Fluency**
[Hu *et al.*, 2017]	1.240	0.613	1.376
[Shen *et al.*, 2017]	0.988	0.363	0.567
MultiDecoder	0.113	1.400	1.363
DeleteAndRetrieval	1.260	1.545	1.483
LM + Classifier	0.915	1.470	1.477
Our model	**1.271**	**1.588**	**1.524**

Table 4: Our model outperforms other models in human evaluation. For simplicity, we only use the best system from each previous work in human evaluation.

results for both the two examples.

5 Discussion

The main problem of non-parallel text style transfer is the lack of parallel data, we further discuss how our mined partial-parallel instances and the data augmentation works.

5.1 Mining Parallel Instances

A simple question may rise that why not directly construct parallel instances and use them to guide the learning process. According to our experiments, the mined partial-parallel instances for the content comparator only covers 30% of the sentences in the dataset; using the same threshold to mining parallel instances could only obtain about 10% of instances comparing to the partial parallel case, which seems to be too small to be useful.

5.2 Mining More Partial-parallel Instances

The threshold of edit distance between two sentences directly affects the number and quality of instances.

For a quick study, we set the threshold to different values to get 50K, 150K and 300K positive partial-parallel instances from the **Yelp** dataset, and use them to train a style transfering model ('+$\mathbf{D_s}$+$\mathbf{D_c}$+DA' in Table 3). The accuracy and BLEU of the three systems are 90.8/18.6, 91.9/22.3 and 89.5/19.7, respectively, showing a clear trade-off between the scale and the quality of the partial-parallel data.

As a result, using a larger threshold to obtain more instances but with lower content similarity may hurt the training performance. To increase the coverage of the data, the data augmentation method may be a better choise.

5.3 Data Augmentation

It is reasonable that the style transfer results would be better for the mined partial-parallel instances, because they have close content-related sentences in the dataset. In Table 6, we present an example which is not in the mined partial-parallel training data. We list the results generated by different models for comparison. Without data augmentation, adding the content comparator $\mathbf{D_c}$ cannot bring much improvement to the content preservation in these cases, because they may not be covered in the training data of the comparator. With data augmentation, after exploring uncovered instances in the training data, both the two models learn better correspondence between the content words, which improve the performance of the model on these uncovered cases.

Model	**From** *positive* **to** *negative*
Original text	the **escargot** was *delicious* , and **seasoned** *perfectly*.
[Hu *et al.*, 2017]	the **steak** was *dry* , and *well overcooked*.
[Shen *et al.*, 2017]	the **manager** was *rude* , and *not very accommodating*.
MultiDecoder	the **escargot** was *delicious* , and were *perfectly*.
DeleteAndRetrieval	the **escargot** was *mediocre* , and we *beat feet out of there*.
LM + Classifier	the **escargot** was *mediocre* at best.
Our model	the **escargot** was *gross* , and **seasoned** *terribly*.
Model	**From** *negative* **to** *positive*
Original text	the **equipment** is so *old* and looks *dirty*.
[Hu *et al.*, 2017]	the **seating** is so *clean* and is *dirty*.
[Shen *et al.*, 2017]	**they** are *worth five stars*!
MultiDecoder	the **equipment** is very *old* and looks *dirty*.
DeleteAndRetrieval	the **equipment** is *clean* and *well maintained*.
LM + Classifier	the **equipment** is very *old* and looks *nice*.
Our model	the **equipment** is very *new* and looks *nice*.

Table 5: Examples generated by some baselines and our model. In each sentence, the bold part represents content-related words, and the italic part represents style-related words.

Model	**From** *positive* **to** *negative*
Original text	**we were sat** right away and every **staff member** was extremely *friendly* and *happy*.
+ $\mathbf{D_s}$	**we sat** at the wrong order and were *sad*.
+ $\mathbf{D_s}$ + $\mathbf{D_c}$	**we were seated** in away and all **service** was absolutely *rude* and *usual*.
+ $\mathbf{D_s}$ + $\mathbf{D_c}$ + LM	**we were sat** after a long time and every **staff** was *rude* problems.
+ $\mathbf{D_s}$ + $\mathbf{D_c}$ + DA	**we were sat** right away and every **staff member** there was *rude* and very *unhappy*.
+ $\mathbf{D_s}$ + $\mathbf{D_c}$ + LM + DA	**we were sat** for a long time and every **staff member** was absoluty *rude* and *dismissive*!

Table 6: An example to show the effect of data augmentation. In each sentence, the bold part represents content-related words, and the italic part represents style-related words.

6 Related Work

For the task of text style transfer, Li *et al.* [2018a] uses simple lexical operation such as 'delete' to remove the style related information based on resources previously extract from data. They also use the 'retrieve' operation to find sentence with the same or similar content. Policies need to be design for using these operations. In contrast, we start with training instances mined by lexical evidences, but the transferring process is still in the framework of sequence-to-sequence, which could be trained in an end-to-end process.

The practice in sentence pair modeling [Lan and Xu, 2018; Hu *et al.*, 2014] inspires us for the design of comparators. Li *et al.*; Zhang *et al.* [2018b; 2018] also incorporate content discriminators for the task of paraphrase, which aims at generating sentences with the same meaning. The task of style transfer is more challenging because there are two aspects to be considered. Our contribution is to design separate comparators for each aspect, which is different from the practice in paraphrase tasks.

7 Conclusion

In this paper, we propose an effective method to make instance-level comparisons with only non-parallel corpora. The proposed partial-comparison strategy enhanced the performance of adversarial training for style transfer models. Our work explore possibilities for text generation without parallel data, which may be useful for other scenarios. For future work, we will explore the possibility to improve the instance mining and data augmentation process with a component which maybe automatically learned during the training. It may also be interesting to apply the proposed method to other similar text generation tasks.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. This work is supported by the National Science Foundation of China (No. U1836221), the Jiangsu Provincial Research Foundation for Basic Research (No. BK20170074), "13th Five-Year" All-Army Common Information System Equipment Pre-Research Project (No. 31510040201).

References

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of ICML*, 2017.

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

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

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

[Hovy, 1987] Eduard Hovy. Generating natural language under pragmatic constraints. *Journal of Pragmatics*, 11(6):689–719, 1987.

[Hu *et al.*, 2014] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *Proceedings of NIPS*, pages 2042–2050, 2014.

[Hu *et al.*, 2017] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596, 2017.

[Jang *et al.*, 2016] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Lan and Xu, 2018] Wuwei Lan and Wei Xu. Neural network models for paraphrase identification, semantic textural similarity, natural language inference and question answeing. In *Proceedings of COLING*, 2018.

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

[Li *et al.*, 2018b] Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, 2018.

[Mathews *et al.*, 2018] Alexander Mathews, Lexing Xie, and Xuming He. Semstyle: Learning to generate stylised image captions using unaligned text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8591–8600, 2018.

[Mirza and Osindero, 2014] Mehdi Mirza and Simon Osindero. Conditional generative adversarial net. In *arXiv preprint arXiv:1411.1784*, 2014.

[Niu and Bansal, 2018] Tong Niu and Mohit Bansal. Polite dialogue generation without parallel data. *Transactions of the Association for Computational Linguistics*, 6:273–389, 2018.

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

[Shen *et al.*, 2017] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844, 2017.

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

[Sutton *et al.*, 2000] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[Tieleman and Hinton, 2012] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5—rmsprop: Divide the gradient by a running average of its recent magnitude. In *COURSERA: Neural Networks for Machine Learning*, 2012.

[Xu *et al.*, 2012] Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. Paraphrasing for style. In *Proceedings of the 24th International Conference on Computational Linguistics*, page 2899–2914, 2012.

[Yang *et al.*, 2018] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 1–12, 2018.

[Yu *et al.*, 2017] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of AAAI conference on Artificial Intelligence*, pages 2852–2858, 2017.

[Zhang *et al.*, 2018] Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*, 2018.