On Learning Text Style Transfer with Direct Rewards

Yixin Liu, Graham Neubig, John Wieting
Carnegie Mellon University
{yixinl2,gneubig,jwieting}@cs.cmu.edu

Abstract

In most cases, the lack of parallel corpora makes it impossible to directly train supervised models for the text style transfer task. In this paper, we explore training algorithms that instead optimize reward functions that explicitly consider different aspects of the style-transferred outputs. In particular, we leverage semantic similarity metrics originally used for fine-tuning neural machine translation models to explicitly assess the preservation of content between system outputs and input texts. We also investigate the potential weaknesses of the existing automatic metrics and propose efficient strategies of using these metrics for training. The experimental results show that our model provides significant gains in both automatic and human evaluation over strong baselines, indicating the effectiveness of our proposed methods and training strategies.[1]

1 Introduction

Text style transfer aims to convert an input text into another generated text with a different style but the same basic semantics as the input. One major challenge in this setting is that many style transfer tasks lack parallel corpora, since the absence of human references makes it impossible to train the text style transfer models using maximum likelihood estimation (MLE), which aims to maximize the predicted likelihood of the references. As a result, some of the earliest work (Shen et al., 2017; Hu et al., 2017; Fu et al., 2018) on unsupervised text style transfer proposed training algorithms that are still based on MLE by formulating the style transfer models as auto-encoders optimized with reconstruction loss. Specifically, during training the model is tasked to generate a *style-agnostic encoding* and reconstruct the input text based on this encoding with style-specific embeddings or decoders. During inference, the model aims to transfer the source text style using the target style information. While these methods have seen empirical success, they face the inherent difficulty of coming up with a style-agnostic but content-preserving encoding – this is a non-trivial task and failure at this first step will diminish style transfer accuracy and content preservation of the final output.

Another line of work (Xu et al., 2018; Pang and Gimpel, 2019; Luo et al., 2019) proposes training algorithms based on rewards related to the automatic evaluation metrics, which can assess the model performance more directly during training. This approach is conceptually similar to training algorithms that optimize models using rewards related to the corresponding evaluation metrics for other NLP tasks, such as machine translation (Shen et al., 2016; Wieting et al., 2019a) or text summarization (Paulus et al., 2018; Li et al., 2019). As for unsupervised style transfer, the widely used automatic metrics mainly attend to three desiderata: (1) style transfer accuracy – the generated sentence must be in the target style, commonly measured by the accuracy of a style classifier applied to the transferred text, (2) fluency – the generated text must be grammatically correct and natural, commonly measured by the perplexity of a language model and (3) content preservation – the semantics need to be preserved between the source and target, commonly measured by the BLEU score between the system outputs and source texts. Since these automatic metrics only require the system outputs and source texts, they can be used as rewards for training. Moreover, the two lines of approaches can be used together, and previous work (Yang et al., 2018; John et al., 2019; Madaan et al., 2020) proposed methods which use the auto-encoders as the backbone augmented with task-specific rewards. In particular, the style transfer accuracy reward is used by most of the recent work.

However, reward-based training algorithms still have their limitations, and in this paper we aim

[1] Code and data are available at: https://github.com/yixinL7/Direct-Style-Transfer

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4262–4273
June 6–11, 2021. ©2021 Association for Computational Linguistics

to identify and address the bottlenecks of these methods. Specifically, we focus on two problems: (1) the difficulty of designing an efficient reward for content preservation, (2) the lack of robustness of the existing automatic evaluation metrics.

Content preservation is more difficult to measure compared to style transfer accuracy and fluency because it needs to consider the overlap in the semantics between the source text and system outputs. While using BLEU score between the source text and system output would be a direct solution (Xu et al., 2018), this approach has an inherent limitation in that n-gram based metrics such as BLEU are sensitive to lexical differences and will penalize modifications that are necessary for transferring text style. In fact, previous work has proposed various different proxy rewards for content preservation. One of the most popular methods is the cycle-consistency loss (Luo et al., 2019; Dai et al., 2019; Pang and Gimpel, 2019), which introduces a round-trip generation process, where the model generates an output in the target style, and the ability of a reconstruction model to re-generate the original text is used as a proxy for content preservation. While this method is more tolerant to lexical differences, the correlation between the reconstruction loss and content preservation can be weak.

Therefore, we aim to design a reward for content preservation which can directly assess the semantic similarity *between the system outputs and input texts*. Specifically, we note that models of semantic similarity are widely studied (Wieting et al., 2016; Sharma et al., 2017; Pagliardini et al., 2018; Zhang* et al., 2020), and we can leverage these methods to directly calculate the similarity between the system outputs and input texts. This renders our method applicable for even unsupervised settings where no human references are available.

Another key challenge for reward-based training algorithms is that the existing automatic evaluation metrics are not well-correlated with human evaluation (Li et al., 2018). It poses general risks to the work in this field with respect to model training and evaluation since these metrics are widely used. An important observation we made from our experiments is that style transfer models can exploit the weaknesses of the automatic metrics. They do this by making minimal changes to the input texts which are enough to trick the classifier used for style transfer accuracy while achieving high content preservation and fluency scores due to the high lexical similarity with the input texts. Upon identifying this risk, we re-visit and propose several strategies that serve as auxiliary regularization on the style transfer models, effectively mitigating the problem discussed above.

We empirically show that our proposed reward functions can provide significant gains in both automatic and human evaluation over strong baselines from the literature. In addition, the problems we identify with existing automatic evaluation metrics suggest that the automatic metrics need to be used with caution either for model training or evaluation in order to make it truthfully reflect human evaluation.

2 Methods

2.1 Overview

Data for unsupervised text style transfer can be defined as

$$D = \{(x^{(1)}, s^{(1)}), ..., (x^{(i)}, s^{(i)}), ..., (x^{(n)}, s^{(n)})\},$$

where $x^{(i)}$ denotes the text and $s^{(i)}$ denotes the corresponding style label. The objective of the task is to generate (via a generator g) the output with the target style conditioned on s while preserving most of the semantics of the source x. In other words, $\hat{x} = g(x, s)$ should have style s and the semantics of x. We define the style as a binary attribute such that $s \in \{0, 1\}$, however, it can be easily extended to a multi-class setting.

2.2 Generator

For our generator, we fine-tune a large-scale language model GPT-2 (Radford et al., 2019). GPT-2 is pre-trained on large corpora and can be fine-tuned to generate fluent and coherent outputs for a variety of language generation tasks (Wolf et al., 2019). Since GPT-2 is a unidirectional language model, we reformulate the conditional generation task as a sequence completion task. Namely, as input to the generator, we concatenate the original sentence with a special token which indicates the target style. The sequence following the style token is our output.

2.3 Reward Functions

We use four reward functions to control the quality of the system outputs. The quality of the outputs is assessed in three ways: style transfer accuracy, content preservation, and fluency. We attend to each of these factors with their respective rewards.

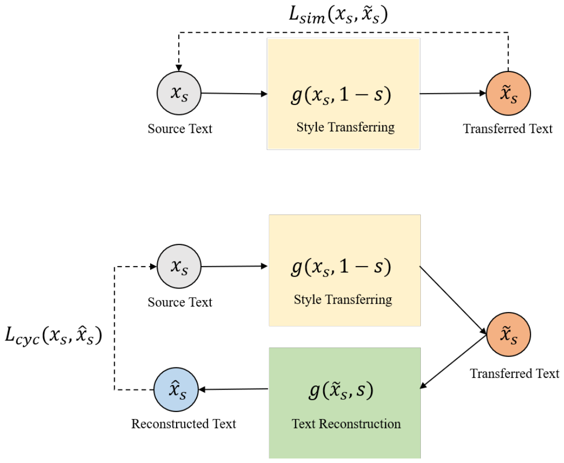

Figure 1: SIM Loss v.s. Cycle-Consistency Loss

Here we denote the input text x having style s by x_s, and denote the output by $\tilde{x}_s$, i.e., $\tilde{x}_s = g(x_s, 1-s)$.

Rewards for Style Transfer Accuracy We use a style classifier to provide the supervision signal to the generator with respect to the style transfer accuracy. The min-max game between the generator g and the classifier f_{cls} is:

$$\min_{\theta_g} \max_{\theta_{f_{cls}}} \mathbb{E}_{x_s}[\log(1 - f_{cls}(g(x_s, 1-s), 1-s))] + \mathbb{E}_{x_s}[\log f_{cls}(x_s, s) + \log(1 - f_{cls}(x_s, 1-s))]. \tag{1}$$

The style transfer accuracy reward for the generator is the log-likelihood of the output being labeled as the target style:

$$r_{cls}(\tilde{x}_s) = \log(f_{cls}(\tilde{x}_s, 1-s)). \tag{2}$$

Following prior work, we use the CNN-based classifier (Kim, 2014) f_{cls}, which takes both the sentence and the style label as input and its objective is to predict the likelihood of the sentence being coherent to the given style.

Rewards for Content Preservation To ensure that the system outputs still preserve the basic semantics of the source sentences, we use the pre-trained SIM model introduced in Wieting et al. (2019b,a) to measure the semantic similarity between the source sentences and system outputs. The SIM score for a sentence pair is the cosine similarity of its sentence representations. These representations are constructed by averaging sub-word embeddings. Compared to the cycle-consistency loss (Luo et al., 2019; Dai et al., 2019; Pang and Gimpel, 2019), our method is more direct since it doesn't require a second-pass generation. It also has advantages over n-gram based metrics like BLEU (Papineni et al., 2002) since it is more robust to lexical changes and can provide smoother rewards.

In Wieting et al. (2019a), SIM is augmented with a length penalty to help control the length of the generated text. We use their entire model, SIMILE, as the content preservation reward,

$$r_{sim}(\tilde{x}_s) = \mathrm{LP}(x_s, \tilde{x}_s)^{\alpha}\mathrm{SIM}(x_s, \tilde{x}_s), \tag{3}$$

where

$$\mathrm{LP}(r, h) = e^{1 - \frac{min(|r|,|h|)}{max(|r|,|h|)}}, \tag{4}$$

and α is an exponential term to control the weight of the length penalty, which is set to 0.25.

We also use the cycle-consistency loss L_{cyc} to bootstrap the training:

$$L_{cyc}(\theta_g) = \mathbb{E}_{x_s}[-\log(p_g(x_s|g(x_s, 1-s), s))]. \tag{5}$$

Here, p_g is the likelihood assigned by the generator g. This introduces two generation passes, i.e., $\tilde{x}_s = g(x, 1-s)$ and $\bar{x}_s = g(\tilde{x}_s, s)$ while SIM reward only requires one generation pass, as illustrated in Fig. 1.

Rewards for Fluency Style transfer accuracy rewards and content preservation rewards do not have a significant effect on the fluency of the outputs. Therefore, we again use the pre-trained GPT-2 model, but as a reward this time. To encourage the outputs to be as fluent as the source sentences, we define the fluency reward as the difference of the perplexity between the system outputs and source sentences:

$$r_{lang}(\tilde{x}_s) = ppl(x_s) - ppl(\tilde{x}_s). \tag{6}$$

Here, *ppl* denotes the length-normalized perplexity assigned by the language model fine-tuned on the training set.

As will be further discussed in Section 3.3, we found that using the rewards mentioned above can still result in unnatural outputs. Therefore, we additionally use a LSTM-based (Hochreiter and Schmidhuber, 1997) discriminator f_{adv} to provide a naturalness reward, whose job is to discriminate the system outputs and the real sentences, i.e., an adversarial discriminator. It constructs a min-max game with the generator:

$$\min_{\theta_g} \max_{\theta_{f_{adv}}} \mathbb{E}_{x_s}[\log(1 - f_{adv}(g(x_s, 1-s)))] + \mathbb{E}_{x_s}[\log(f_{adv}(x_s))]. \tag{7}$$

The naturalness reward is the log-likelihood of the outputs being classified as real sentences:

$$r_{adv}(\tilde{x}_s) = \log(f_{adv}(\tilde{x}_s)). \tag{8}$$

2.4 Learning

The final corresponding loss term is:

$$L_*(\theta_g) = -\frac{1}{N}\sum_{i=1}^{N} r_*(\tilde{x}_s^{(i)}). \tag{9}$$

Here, N is the number of samples in the dataset. To train the model, we use the weighted average of the losses defined in the previous section:

$$\begin{aligned} L(\theta_g) = {} & \lambda_{cls}L_{cls}(\theta_g) + \lambda_{adv}L_{adv}(\theta_g) \\ & + \lambda_{sim}L_{sim}(\theta_g) + \lambda_{lang}L_{lang}(\theta_g) \\ & + \lambda_{rec}L_{rec}(\theta_g). \end{aligned} \tag{10}$$

where λ denotes the weight of the corresponding term. The setting of λ is chosen to make the training stable and have balanced style transfer accuracy and content preservation performance on the development set. L_{rec} is the reconstruction loss, i.e.,

$$L_{rec}(\theta_g) = \mathbb{E}_{x_s}[-\log(p_g(x_s|x_s, s))]. \tag{11}$$

We follow a two-stage training procedure. We first use the cycle-consistency loss L_{cyc} to bootstrap the training and then fine-tune the model with the rewards we introduced above to improve the output quality.

In the bootstrap stage, the objective function is

$$\begin{aligned} L_{boot}(\theta_g) = {} & \lambda_{cyc}L_{cyc}(\theta_g) + \lambda_{cls}L_{cls}(\theta_g) \\ & + \lambda_{rec}L_{rec}(\theta_g) \end{aligned} \tag{12}$$

We select the checkpoint with the highest mean of the style transfer accuracy and BLEU on the development set as the starting point for the second training stage.

In the second stage, the generator is optimized with Eq. 10. The classifier f_{cls} for L_{cls} is pre-trained and the language model for L_{lang} is fine-tuned on the training set. During training, the discriminator f_{adv} for L_{adv} is trained against the generator. f_{cls} is fixed when trained on some datasets, while it is trained against the generator on others. We select the checkpoint that has the style transfer accuracy and BLEU score similar to that from the first stage and the lowest perplexity on the development set.

Lastly, since gradients can not be propagated through the discrete samples, we use two approaches to circumvent this problem. For the content preservation reward (Eq. 3) and fluency reward (Eq. 6), we use the REINFORCE (Williams, 1992) algorithm to optimize the model,

$$\begin{aligned} & \nabla_{\theta_g}\mathbb{E}_{\tilde{x}_s\sim p_g(\tilde{x}_s)}[r(\tilde{x}_s)] \\ & = \mathbb{E}_{\tilde{x}_s\sim p_g(\tilde{x}_s)}[\nabla_{\theta_g}\log p_g(\tilde{x}_s)r(\tilde{x}_s)] \end{aligned} \tag{13}$$

We approximate the expectation by greedy decoding and the log-likelihood is normalized by sequence length, i.e., $\frac{1}{L}\sum_{i=1}^{L}\log p_g(\tilde{w}_i)$, where $\tilde{w}_i$ denotes the i-th token of $\tilde{x}_s$ and L is sequence length. For the style transfer accuracy reward (Eq. 2) and naturalness reward (Eq. 8), we use a different approach to generate a continuous approximation of the discrete tokens, which allows gradients to be back-propagated to the generator. Namely, taking the style classifier f_{cls} as an example, we use the distribution p_i of each token produced by the generator as the input of the classifier. This distribution is then multiplied by the classifier's word embedding matrix W^{embed} to obtain a weighted average of word embeddings:

$$\hat{w}_i = p_i W^{embed} \tag{14}$$

Then, the classifier takes the sequence of $\hat{w}_i$ as its input. We chose this method because it provides a token-level supervision signal to the generator, while the REINFORCE algorithm provides sentence-level signals.

3 Experiments

3.1 Datasets

We evaluate our approach on three datasets for sentiment transfer with positive and negative reviews: Yelp review dataset, Amazon review dataset provided by Li et al. (2018),[2] and the IMDb movie review dataset provided by Dai et al. (2019).[3]

We also evaluate our methods on a formality style transfer dataset, Grammarly's Yahoo Answers Formality Corpus (GYAFC),[4] introduced in Rao and Tetreault (2018). Although it is a parallel corpus, we treat it as an unaligned corpus in our experiments. In order to compare to previous work,

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

[3] https://github.com/fastnlp/nlp-dataset

[4] https://github.com/raosudha89/GYAFC-corpus

Dataset	Style	Train	Dev	Test
Yelp	Positive Negative	266K 177K	2000 2000	500 500
Amazon	Positive Negative	277K 279K	985 1015	500 500
IMDb	Positive Negative	178K 187K	2000 2000	1000 1000
GYAFC	Formal Informal	52K 52K	2247 2788	500 500

Table 1: Number of samples in the Train, Dev, and Test splits for each dataset in our experiments.

Dataset	Eq.	λ_{cls}	λ_{adv}	λ_{sim}	λ_{lang}	λ_{rec}	λ_{cyc}
Yelp	(10) (12)	2 1	0.5 -	20 -	2 -	0.1 1	- 1.5
Amazon	(10) (12)	2 5	0.5 -	20 -	2 -	1 1	- 0.5
IMDb	(10) (12)	1 1	0.5 -	20 -	2 -	1 1	- 1
GYAFC	(10) (12)	2 1	0.5 -	20 -	2 -	1 1	- 1

Table 2: Hyperparameter setting of Eq. 10 and Eq. 12 on each dataset.

we chose the *Family & Relationships* category for our experiments. Datasets statistics are shown in Table 1.

3.2 Experimental Details

Following previous work, we measure the style transfer accuracy using a FastText[5] (Joulin et al., 2017) style classifier trained on the respective training set of each dataset. To measure content preservation, we use SIM and BLEU as metrics where self-SIM and self-BLEU are computed between the source sentences and system outputs, while ref-SIM and ref-BLEU are computed between the system outputs and human references when available. To measure the fluency we use a pre-trained GPT-2 model to compute the perplexity.[6] Our generator, GPT-2, has 1.5 billion parameters, and we train on a GTX 1080 Ti GPU for about 12 hours.

The weights of the loss terms in Eq. 10 and Eq. 12 are detailed in Table 2. While during our experiments we found that there are other possible configurations which give higher scores with respect to the automatic evaluation metrics, as will be discussed in Section 3.3, we also found that better performance in automatic evaluation doesn't always entail better performance in human evaluation. Therefore, we also manually checked the quality of the transferred texts on development set when we chose the value of the hyperparameters.

Dataset	Model	Acc	PPL	BLEU
Yelp	DIRR-CYCLE DIRR-YELP-ADV	91.7 95.2	392 353	18.7 20.7
Amazon	DIRR DIRR-AMAZON-ADV	62.2 83.2	205 228	30.1 29.0

Table 3: Adversarial Results. **DIRR-YELP-ADV** and **DIRR-AMAZON-ADV** denote the models which generate adversarial examples. **Acc** denotes the style transfer accuracy, **PPL** denotes the perplexity, **BLEU** is computed between the human references and system outputs.

We compare our model with several state-of-the-art methods: DeleteAndRetrieve (D&R) (Li et al., 2018), B-GST (Sudhakar et al., 2019), Cycle-Multi (Dai et al., 2019), Deep-Latent (He et al., 2020), Tag&Gen (Madaan et al., 2020), and DualRL (Luo et al., 2019). We also compare our final model, **DIRR**(**Dir**ect-**R**eward), with the model only trained with the first stage (DIRR-CYCLE) as mentioned in Section 2.4.

3.3 Adversarial Examples

Yelp and Amazon are arguably the most frequently used datasets for the sentiment transfer task. In our experiments, we found that the automatic evaluation metrics can be tricked on these datasets. Table 3 shows the performance of the models which generate adversarial examples. Upon identifying these risks, we propose several design options that can effectively mitigate these problems.

Yelp Dataset For the Yelp dataset, when trained without the adversarial discriminator f_{adv} and the fluency reward, our model (DIRR-YELP-ADV) is able to discover a trivial solution which receives high automatic evaluation scores: injecting a word that carries strong sentiment at the beginning of the output, and making minimum changes (if any) to the source sentences, as illustrated in Table 8. This obviously does not meet the objective of content-preserving sentiment transfer and is easily detectable for humans. In fact, after we manually removed the first word from each of the output sentences, the transfer accuracy dropped from 95.2 to 58.4. To address this problem, we introduced an

[5] https://fasttext.cc/

[6] Note that we didn't fine-tune it on the training set

Model	"game"		"phone"	
	Pos.	Neg.	Pos.	Neg.
Train	58	7548	8947	2742
Test	0	10	20	6
Human	1	10	18	6
B-GST	55	0	13	44
Tag&Gen	69	0	14	5
DIRR	26	0	19	45
DIRR-AMAZON-ADV	291	0	190	4

Table 4: Frequencies of words in the Amazon Dataset that appear often enough in specific classes to erroneously cause the classifier to make incorrect predictions. **Pos.** denotes the positive sentences, **Neg.** denotes the negative sentences.

Model	Text
Source	don t waste your time or money on these jeans .
Adv	don t need your time or money on these **phones** .
Source	i made beef bolognese in the oven and it turned out wonderfully .
Adv	i made beef bolognese in the **game** and it turned out wonderfully .
Source	this one does the job i need it for !
Adv	this **game** does the job i need it for !

Table 5: Adversarial examples received high style transfer accuracy scores on Amazon Dataset. Adv denotes the adversarial examples generated by DIRR-AMAZON-ADV.

auxiliary discriminator f_{adv} as we discussed above to penalize the trivial outputs since they can be easily captured by the discriminator. On the other hand, the output perplexity is not sensitive enough to this local feature so using the fluency reward alone is not sufficient. Our final model has much more stable performance when the first word of its output sentences is removed, experiencing only a small drop of the style transfer accuracy from 94.2 to 88.2.

Amazon Dataset For the Amazon dataset, we found that the style classifier f_{cls} needs to be updated during the training to prevent the model exploiting the data imbalance problem of the dataset. Namely, in the Amazon dataset some categories of products appear mostly in negative or positive reviews. In Table 4, we show the word frequency of *game* and *phone* in both negative and positive reviews. In the original dataset, *game* mostly appears in negative reviews while *phone* mostly appears in positive reviews. Therefore, without any prior knowledge, it is very likely that these words will be used as informative features by the sentiment classifier, which makes its predictions unreliable.[7]

When our second-stage model is trained with the fixed style classifier, it (DIRR-AMAZON-ADV) learns to exploit this dataset bias by changing the nouns in the original sentences to *game* or *phone*, which achieves better transfer accuracy. We list some examples in Table 5. DIRR-AMAZON-ADV generated 291 *game* in 500 positive reviews, which obviously changes the semantics of the source sentences. In order to show that this phenomenon is independent to the classifier architecture, we additionally fine-tuned a BERT-based (Devlin et al., 2019) classifier, which yielded 51.3, 57.6, 70.4 accuracy on human references, DIRR, DIRR-AMAZON-ADV respectively, showing the same pattern of the fastText classifier. We notice that some two-stage models (Li et al., 2018; Sudhakar et al., 2019; Madaan et al., 2020) and other methods (Yang et al., 2018; Luo et al., 2019) also use a fixed classifier or use words with unbalanced frequencies in different styles as important features, which means that their methods may face the same risk. While Li et al. (2018) has pointed out this data imbalance problem of the Amazon dataset, we further demonstrate that a strong generator can even use this discrepancy to trick the automatic metrics. We are able to mitigate this problem by updating the style classifier during the training, and in Table 4, DIRR is more robust to the data imbalance problem compared to other methods.

3.4 Automatic Evaluation

The automatic evaluation results are shown in Table 6. We report the performance of the previous methods based on the outputs they provided for fair comparison and omit those whose results are not available.

We have the following observations of the results. First, compared to our base model (DIRR-CYCLE), the model trained with our proposed rewards has higher fluency, while remains the same level of content preservation. It indicates that SIM score is as effective as cycle-consistency loss for content preservation and our fluency reward can effectively improve the output fluency. Secondly, there exists a trade-off among the style transfer accuracy, content preservation and language fluency. While our model does not outperform the previous meth-

[7] Notice that the style classifier only achieves 43 accuracy on the human references.

Model	Acc	PPL	r-BLEU	s-BLEU
Yelp				
D&R	89.0	362	10.1	29.1
B-GST	86.0	**269**	14.5	35.1
Cycle-Multi	87.6	439	19.8	**55.2**
Deep-Latent	86.0	346	15.2	40.7
Tag&Gen	88.7	355	12.4	35.5
DIRR-CYCLE	91.7	392	18.7	51.2
DIRR	**94.2**	292	**20.7**	52.6
Copy	4.1	204	22.5	100.0
Human	70.7	236	99.3	22.5
Amazon				
D&R	50.0	233	24.1	54.1
B-GST	60.3	**197**	20.3	44.6
Tag&Gen	**79.9**	312	27.6	**62.3**
DIRR-CYCLE	68.4	374	29.0	60.6
DIRR	62.2	205	**30.1**	61.3
Copy	21.1	218	40.0	100.0
Human	43.0	209	100.0	40.0
IMDb				
Cycle-Multi	77.1	290	N/A	**70.4**
DIRR-CYCLE	80.5	253	N/A	64.3
DIRR	**83.2**	**210**	N/A	64.2
Copy	5.3	147	N/A	100.0
GYAFC				
D&R	51.2	226	14.4	27.1
DualRL	62.0	404	33.0	50.8
DIRR-CYCLE	**76.2**	162	44.1	**66.5**
DIRR	71.8	**145**	**46.3**	59.9
Copy	15.8	147	41.5	98.5
Human	84.5	137	97.8	21.5

Table 6: Automatic Evaluation. Acc is the accuracy of the sentiment classifier. PPL is the perplexity assigned by the GPT-2 language model. r-BLEU is the BLEU score between the human references and system outputs. s-BLEU is the BLEU score between the source sentences and system outputs. Copy is an oracle which copies the source sentences as outputs. Human denotes the human references.

ods on all of the metrics, it is able to find a better balance of the different metrics.

3.5 Human Evaluation

We conducted human evaluation on Yelp, Amazon and GYAFC datasets evaluating the style transfer accuracy, content preservation, and fluency separately. The first two aspects are rated with range 1 - 3 while the fluency is rated with range 0 - 1. We randomly select 100 candidates and compare the outputs of different systems. We use Amazon Turk[8] for human evaluation. Each candidate is rated by three annotators and we report the average scores here. We did not evaluate the style transfer accuracy for the GYAMC dataset since it is difficult for human annotators to accurately capture the difference between formal and informal sentences. The results of our human evaluations are shown in Table 7. We additionally report the sample-wise mean score of the metrics where the fluency scores are scaled up to be consistent with other scores. Our model achieves better overall performance when considering all three evaluation metrics on each dataset.

Dataset	Model	Style	Flu.	Con.	Mean
Yelp	Cycle	2.24	0.62	1.97	2.02
	B-GST	**2.42**	0.64	2.02	2.12
	DIRR	**2.42**	**0.66**	**2.04**	**2.14**
Amazon	Tag&Gen	1.98	0.87	1.95	2.19
	B-GST	2.04	**0.89**	1.77	2.16
	DIRR *	**2.09**	0.87	**2.10**	**2.26**
GYAMC	D&R	N/A	0.40	2.13	1.66
	DualRL	N/A	0.51	2.23	1.88
	DIRR *	N/A	**0.70**	**2.34**	**2.22**

Table 7: Human Evaluation. **Style** denotes style transfer accuracy, **Flu.** denotes fluency, **Con.** denotes content preservation. **Mean** denotes the average of the metrics where the fluency scores are scaled up to be consistent with other scores. *: significantly better than other systems ($p < 0.01$) according to the mean score.

Interestingly, we found that the automatic metrics for both the style transfer accuracy and content preservation do not accurately reflect performance as measured by human evaluation. For example, on the Amazon dataset, although Tag&Gen (Madaan et al., 2020) achieves significantly higher style transfer accuracy based on the automatic metric, our model achieves better performance based on the human evaluation. This phenomenon suggests that the importance of our findings discussed in Section 3.3, that strong neural models can potentially exploit the weaknesses of the automatic metrics.

4 Analysis

We next show an ablation study, demonstrating the effectiveness of the content preservation and fluency rewards in DIRR, and how SIM can be used to replace the cycle-consistency loss. We also compare using BLEU versus using SIM as a content-preservation reward, finding that using BLEU results in reduced performance, unstable training, and artifacts in the outputs, which makes the results less natural than the results of the model trained with SIM score.

To illustrate that training with SIM can replace

[8] https://www.mturk.com/

Model	Text	self-BLEU	self-SIM
Source	this was my first stop in looking for a wedding dress .	100.0	100.0
DirR-Bleu	**great** this was my first stop in looking for a wedding dress .	91.2	95.2
DirR	this was my best stop in looking for a wedding dress .	64.8	81.9
source	just a frozen patty cooked like a home one .	100.0	100.0
DirR-Bleu	**great** a frozen patty cooked like a home one .	88.0	94.6
DirR	just a great patty cooked like a home one .	70.7	88.5
source	wendy 's has been know to be cheap with their drink refills for years .	100.0	100.0
DirR-Bleu	**great** wendy 's has been know to be cheap with their drink refills for years .	93.0	97.5
DirR	wendy 's has been great with their drink refills for years .	57.2	84.9

Table 8: Comparison of using SIM and BLEU as the content preservation reward. Samples are from the Yelp dataset. The metrics self-BLEU and self-SIM are calculated between the source sentences and system outputs.

Model	Acc	PPL	s-BLEU	s-SIM
DirR-Cycle	91.7	392	51.2	76.2
DirR w/o Flu	92.1	348	51.4	79.8
DirR-Bleu	91.3	315	**59.4**	**81.8**
DirR	**94.2**	**292**	52.6	81.6

Table 9: Ablation and Comparative Study on Yelp Dataset. Acc is the accuracy of the sentiment classifier. PPL is the perplexity assigned by the GPT-2 language model. self-BLEU (s-BLEU) and self-SIM (s-SIM) are computed between the source sentences and outputs.

the cycle-consistency loss for content preservation, we fine-tuned DirR-Cycle on SIM to produce a new model, DirR w/o Flu. The difference between DirR and DirR w/o Flu is that the former is additionally trained with our fluency rewards. The results are shown in Table 9, and show two main trends. First, we see that DirR w/o Flu has better fluency and content preservation performance than DirR-Cycle, which shows that the cycle-consistency loss can be replaced by SIM score for content preservation. Second, DirR has better fluency than DirR w/o Flu, showing the effectiveness of our fluency rewards.

We next investigate the effectiveness of using SIM as a reward instead of BLEU. To do this, we train a model, DirR-Bleu, which uses BLEU as the content reward and report the results in Table 9. The results show that using BLEU has larger content preservation as measured by BLEU, but has similar performance when measured by SIM. However, performance on the style transfer accuracy and fluency decreases. We hypothesize that this is because using SIM as a reward gives the model more freedom, allowing the model to have more balanced performance since there is less pressure to copy n-grams. We also observe more adversarial examples in the outputs of DirR-Bleu. As discussed in Section 3.3, these adversarial examples are generated by injecting a word carrying strong sentiment at the beginning of the output. The model trained with BLEU is more likely to generate these outputs as it will try to avoid breaking up the n-grams in the source sentences, allowing for a higher BLEU reward. Examples of this behavior is shown in Table 8. Notice that the DirR-Bleu samples start with the word *great*, which is enough to often fool the classifier, but are unnatural.

5 Related Work

A main line of work (Shen et al., 2017; Hu et al., 2017; Fu et al., 2018; Xu et al., 2018; John et al., 2019) for text style transfer aims to model the conditional distribution of the data with the encoder-decoder architecture. Due to the lack of parallel corpora, inductive biases are designed to make the generation conditioned on both source sentences and specific styles such that the model can rewrite the source texts with the target style while still preserve the content information of the source texts.

Efforts are also made to design training objectives to improve performance. For example, Back-translation (Zhang et al., 2018; Prabhumoye et al., 2018), denoising auto-encoding (Lample et al., 2019) and the cycle-consistency loss (Luo et al., 2019; Dai et al., 2019; Pang and Gimpel, 2019) have been shown effective for improving the model performance. Li et al. (2018) proposes a retrieve-based pipeline, which contains three stages, namely, delete, retrieve and generate. Sudhakar et al. (2019) extends this pipeline by using GPT (Radford et al., 2018) as the generator. Compared to these methods, we propose a more direct and effective approach to encourage semantic-preserving transfer by directly measuring the semantic similarity of the source texts and system outputs.

Recently, other works have been proposed for unsupervised text style transfer (Jin et al., 2019; Lai et al., 2019; Wu et al., 2019; Li et al., 2020). He et al. (2020) proposes a probabilistic view which models the non-parallel data from two domains as a partially observed parallel corpus. Madaan et al. (2020) proposes a tag-and-generate pipeline, which firstly identifies style attribute markers from the source texts, then replaces them with a special token, and generates the outputs based on the tagged sentences. Zhou et al. (2020) focuses on exploring the word-level style relevance which is assigned by a pre-trained style classifier. They propose a reward for content preservation which is based on the weighted combination of the word embeddings of the source texts and system outputs. Compared to this reward, our proposed content reward is specifically designed for semantic similarity and pre-trained on large corpora, which makes it more robust across different datasets.

6 Conclusion

In this paper, we propose a direct approach of improving content preservation for text style transfer by leveraging a semantic similarity metric as the content reward. Using a large pre-trained language model (GPT-2) with our proposed rewards that target the different aspects of the output quality, our approach achieves strong performance on both automatic and human evaluation. Recently, several semantic similarity metrics (Zhao et al., 2019; Sellam et al., 2020; Gao et al., 2020) based on pre-trained language models have shown promising results. Introducing these metrics in our proposed method as the content preservation reward may bring further improvements.

Moreover, we identify several problems in the commonly used automatic evaluation metrics and datasets, and propose several practical strategies to mitigate these problems, which makes these metrics more effective rewards for model training. Considering the weaknesses of the automatic metrics presented in this work, we believe that more rigorous discussion and investigation on the criteria of "successful transferring" is essential for this field of work. Since existing works mostly relied on model-based metrics to determine the success of style transfer models, methods such as adversarial training could be introduced to make the model-based metrics more robust and faithful indicators of the success of style-transferring, which would be beneficial for both model training and evaluation.

References

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yang Gao, Wei Zhao, and Steffen Eger. 2020. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online. Association for Computational Linguistics.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. In *International Conference on Learning Representations*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org.

Zhijing Jin, Di Jin, Jonas Mueller, Nicholas Matthews, and Enrico Santus. 2019. Imat: Unsupervised text attribute transfer via iterative matching and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3088–3100.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy. Association for Computational Linguistics.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Chih-Te Lai, Yi-Te Hong, Hong-You Chen, Chi-Jen Lu, and Shou-De Lin. 2019. Multiple text style transfer by using word-level conditional generative adversarial network with two-phase training. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3570–3575.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Siyao Li, Deren Lei, Pengda Qin, and William Yang Wang. 2019. Deep reinforcement learning with distributional semantic rewards for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6038–6044, Hong Kong, China. Association for Computational Linguistics.

Yuan Li, Chunyuan Li, Yizhe Zhang, Xiujun Li, Guoqing Zheng, Lawrence Carin, and Jianfeng Gao. 2020. Complementary auxiliary classifiers for label-conditional text generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8303–8310.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5116–5122. International Joint Conferences on Artificial Intelligence Organization.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online. Association for Computational Linguistics.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Richard Yuanzhe Pang and Kevin Gimpel. 2019. Unsupervised evaluation metrics and learning criteria for non-parallel textual transfer. In *Proceedings of the 3rd Workshop on Neural Generation and Translation (WNGT 2019)*.

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

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

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

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3260–3270.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019a. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019b. Simple and effective paraphrastic similarity from parallel translations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4602–4608.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. 2019. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4873–4883, Florence, Italy. Association for Computational Linguistics.

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

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 7287–7298.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018. Style transfer as unsupervised machine translation. *CoRR*, abs/1808.07894.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Chulun Zhou, Liangyu Chen, Jiachen Liu, Xinyan Xiao, Jinsong Su, Sheng Guo, and Hua Wu. 2020. Exploring contextual word-level style relevance for unsupervised style transfer. In *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pages 7135–7144, Online. Association for Computational Linguistics.