RL-ST: REINFORCING STYLE, FLUENCY AND CONTENT PRESERVATION FOR UNSUPERVISED TEXT STYLE TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised text style transfer is the task of re-writing text of a given style into a target style without using a parallel corpus of source style and target style sentences for training. Style transfer systems are evaluated on their ability to generate sentences that 1) possess the target style, 2) are fluent and natural sounding, and 3) preserve the non-stylistic parts (content) of the source sentence. We train a reinforcement learning (RL) based unsupervised style transfer system that incorporates rewards for the above measures, and describe novel rewards shaping methods for the same. Our approach does not attempt to disentangle style and content, and leverages the power of massively pre-trained language models as well as the Transformer. Our system significantly outperforms existing state-of-art systems based on human as well as automatic evaluations on target style, fluency and content preservation as well as on overall success of style transfer, on a variety of datasets.

1 INTRODUCTION

Text style transfer is an important natural language generation problem, since it has wide applications across different domains. It has been used to adapt texts to specific artistic writing styles (Jhamtani et al., 2017), make texts formal or informal (Rao & Tetreault, 2018), alter sentiment [1] (Hu et al., 2017; Shen et al., 2017; Fu et al., 2018; Li et al., 2018), rewrite factual sentences into romantic or humorous ones (Li et al., 2018), generate poetry (Yang et al., 2018a), personalize dialogue systems (Zhou et al., 2017) and obfuscate gender in social media posts (Reddy & Knight, 2016).

Most recent works perform unsupervised style transfer due to the unavailability of parallel style corpora. Most previous works on unsupervised style transfer attempt to disentangle the stylistic parts (hereby, 'attributes') and non-stylistic parts (hereby, 'content') of texts, and then modify the attributes while preserving the content. Some of these works encode style and content in separate latent representations, and decode the style-dependent output from these representations (Fu et al., 2018; Shen et al., 2017; Hu et al., 2017; Yang et al., 2018b). A few others explicitly identify attribute and content words from input texts and then train models to generate target style sentences from the content (Li et al., 2018; Sudhakar et al., 2019; Wu et al., 2019c).

More recently, Lample et al. (2018b) showed that many previous works that attempt to disentangle content and style in latent representation spaces are unsuccessful in doing so in practice. Further, these approaches are prone to instability of training, low sample efficiency and consequently poor quality outputs. Approaches where attribute words are explicitly removed from the sentence require heuristics and thresholds to decide attribute words, which makes them sensitive to and require manual setting of thresholds. This causes core content words to be incorrectly deleted in some cases, and source attribute words to be incorrectly preserved in others. Moreover, in some of these works (Li et al., 2018; Sudhakar et al., 2019; Wu et al., 2019b), the final output generators are provided with only the content information of the input sentence and not the attributes. This leads to awkward outputs where the model inserts target attribute words that are either not suitable to the content, or are wrongly positioned. However, it has been observed that these approaches are more controllable,

[1] Similar to previous works, we use a broad socio-linguistic definition of style that includes sentiment

easier to train and produce better quality outputs than approaches based on latent representations. A few recent works (Lample et al., 2018b; Dai et al., 2019; Luo et al., 2019) avoid style-content disentanglement altogether.

While most works use recurrent networks for encoding and decoding, transformers (Vaswani et al., 2017) have been shown to be significantly better for the task (Dai et al., 2019; Sudhakar et al., 2019; Wu et al., 2019b). Sudhakar et al. (2019) show significant gains over previous state of the art by leveraging the combined power of transformers and massively pre-trained language models, by using a decoder-only GPT (Radford et al.). In doing so, they do away with the traditional encoder-decoder mechanism.

Finally, RL has been used in previous works to leverage the use of non-differentiable training objectives and overcome the lack of parallel corpora. Xu et al. (2018) use a cycled RL approach where a neutralization model first disentangles the content and attributes, and then passes the content to an emotionalization model. However, cascading errors propagated from the neutralization to the emotionalization due to a discretization of embeddings to words in between the two, leads to poor quality outputs. Gong et al. (2019) use adversarially trained discriminators whose feedbacks are used as rewards by a generator, and Luo et al. (2019) train a dual RL system wherein separate models exist for source-to-target prediction and target-to-source prediction. Style and content rewards are built into this dual structure. However their model tends to be majority-biased towards certain attributes (such as 'happy' and 'loved' on the task of sentiment transfer), which are abruptly inserted in the output sentences without being meaningfully transferred versions of the source attributes. For instance, one would not find it meaningful for the source sentence 'so i asked for the card to be refunded' to be mapped to the target sentence 'so i loved the credit card to be happy'.

Taking into consideration the drawbacks and strengths of previous works, our contributions are as follows: we introduce a novel RL based model for style transfer that 1) uses the decoder-only GPT (Radford et al.) in order to leverage the power of transformers and massively pre-trained language models, 2) directly learns mappings from source to target sentences without any disentanglement, 3) does not require any parallel corpus but is instead warm-started by using a synthetic parallel corpus generated by the trained GST (Sudhakar et al., 2019) and 4) provides for controllable generation by allowing trade-offs between style, content retention and fluency. Our approach significantly outperforms current state-of-art systems based on human evaluation as well as on evaluations using automatic metrics.

In the interest of reproducibility, we publish all our code, data and results for this work on our Github repository, the link to which will be added here in the camera-ready version if accepted.

2 Our Approach

We assume a dataset $D = \{(x_1, s_1), ..., (x_m, s_m)\}$ where each sentence x_i is associated with a specific style $s_i \in S$. For instance, for the task of sentiment transfer, $S = \{$'Positive', 'Negative'$\}$. We then aim to learn the conditional distribution $P(y|x, s^{tgt})$ such that the style of y is s^{tgt}, and y's content is similar to that of x. We introduce the Reinforcement Learning based Style Transformer (hereby, RL-ST). **RL-ST** takes as input the source sentence x of style s^{src} and generates the output sentence $\hat{y}$ of style s^{tgt}. More formally, it learns:

$$P(\hat{y}|x, s^{tgt}; \theta) \tag{1}$$

Model: The architecture of RL-ST is a decoder-only Transformer, based on the implementation of the Generative Pre-trained Transformer (Radford et al.) (hereby, GPT). Similar to Sudhakar et al. (2019), we do away with the notion of an encoder-decoder mechanism and use a decoder-only transformer, pre-trained using a language model like training. GPT has masked attention heads that enable it to look only at the tokens to its left, and not to those to its right. In order to address the cold-start problem that typically causes convergence issues during RL, we warm-start the RL training by pre-training RL-ST on a synthetically generated parallel corpus. This corpus is obtained by using the B-GST trained by Sudhakar et al. (2019) on our non-parallel corpus, and the pre-training is performed via Maximium Likelihood Estimate (MLE) (Ranzato et al., 2015; Paulus et al., 2018) over this synthetic parallel corpus. RL-ST is then fine-tuned using Policy Gradient, by providing rewards for style, content preservation and fluency. Appendix A.1 provides further details of architecture used during training of RL-ST.

Sampling: RL-ST optimizes a policy using the policy gradient, to maximize a long term reward. In doing so, it uses the notion of state-action pairs, with rewards assigned to such pairs. The parameters of the model θ define a policy π which maps a state (the input to the model until each time step t) to an action (the next output word to generate). The action is sampled from the model's softmax distribution using a sampling method. We use a 'top-p sampling' (alternatively, 'nucleus sampling') (Holtzman et al., 2019) for the same. This sampling method samples (using the softmax output probability distribution) a token from the set of top tokens that make up a cumulative softmax probability of p. Unlike beam search which *exploits* the output probability distribution, top-p sampling is more geared towards *exploration*. For each sentence x in the RL-training set, we perform the above sampling K different times to ensure sufficient exploration. The state corresponding to the output timestep t of the k^{th} sampling round of sentence x is represented as st_t^k. The corresponding action is represented as a_t^k. Each such state-action pair receives a reward R_t^k composed of 3 different rewards - the style, content and fluency rewards as elaborated in ensuing sections. By our formulation, each $\hat{y}_t^k$ is simply the action a_t^k.

Policy Gradient: The gradient of the expected reward $\mathbb{E}[R]$ of the generated sequence $\hat{y}$ can be estimated by the REINFORCE policy gradient algorithm (Williams, 1992) adapted to the token-level for RL-ST as:

$$\begin{aligned}\nabla_\theta \mathbb{E}[R] &= \nabla_\theta \Sigma_k \Sigma_t P(\hat{a}_t^k | \hat{st}_t^k; \theta) R_t^k \\ &= \nabla_\theta \Sigma_k \Sigma_t P(\hat{y}_t^k | x, \hat{y}_{1,..t-1}^k, s^{tgt}; \theta) R_t^k \\ &= \Sigma_k \Sigma_t P(\hat{y}_t^k | x, \hat{y}_{1,..t-1}^k, s^{tgt}; \theta) R_t^k \nabla_\theta log(P(\hat{y}_t^k | x, \hat{y}_{1,..t-1}^k, s^{tgt}; \theta)) \\ &\approx \frac{1}{K\tau} \Sigma_k \Sigma_t R_t^k \nabla_\theta log(P(\hat{y}_t^k | x, \hat{y}_{1,..t-1}^k, s^{tgt}; \theta))\end{aligned} \tag{2}$$

where R_t^k is described in equation 9 and represents the reward for the k^{th} sequence sampled out of a total of K sampled sequences for each training example, and τ is the maximum length of the decoder output.

2.1 Reward Computation

In sequence generation problems such as style transfer, typically rewards are only available once the output sequence has been generated completely. Due to this, a fundamental problem of reward assignment to intermediate tokens arises i.e., obtaining a value for R_t^k. A few 'reward shaping' techniques have been used to alleviate this. One popular technique is to 'roll out' (Yu et al., 2017) the partial sequence generated up till timestep t, $\hat{y}_{1:t}^k$, by sampling the rest of the sequence $\hat{y}_{t+1:\tau}^k$ (Yu et al., 2017), where τ is the maximum sequence length of the decoder. Due to the need to sample $\tau - t$ tokens at every timestep t, roll-out based methods are computationally expensive. Hence, we propose a novel method to leverage transformer attention weights to assign token level **style** rewards. We also use a language model in a novel way to provide token level **fluency** rewards. To the best of our knowledge ours is the first work on style transfer work that leverages carefully designed reward shaping in the manner that we do. This, combined with our warm starting mechanism, provides a way to completely circumvent roll-out. During the k^{th} sampling round using the input sequence x, we first generate the whole of $\hat{y}^k$ from the model at one go, and then proceed to assign token-level rewards to each $\hat{y}_t^k$.

Transformer Attention based Style Reward: We use a pre-trained, self-attention based style classifier to decide the style reward for a generated sentence $\hat{y}$. The style classifier takes as input $\hat{y}$ and defines a distribution over style labels s:

$$P(s|\hat{y}) = f(enc(\hat{y}), \alpha, \theta_{CLS}) \tag{3}$$

where $enc(\hat{y})_t$ is an encoding of $\hat{y}_t$, and α_t is the self-attention score corresponding to $enc(\hat{y})_t$ learned by the classifier in assigning probabilities for each style s_j, and θ_{CLS} is the model parameter. For the classifier, we use the same Delete Transformer (DT) as used by Sudhakar et al. (2019). This is a BERT-based (Devlin et al., 2018) classifier, which has 144 sets of self-attention heads. We extract a representative head-layer pair $< h, l >$ out of these, using the process described by Sudhakar et al. (2019) and choose α to be the self-attention weights of $< h, l >$, corresponding to the input tokens of $\hat{y}$. We then choose attribute words from $\hat{y}$ based on their α scores, since attribute words are paid higher attention or importance than content words are by a style classifier (Feng et al., 2018; Xu

et al., 2018; Sudhakar et al., 2019). The top $\gamma|x|$ tokens of x are treated as attributes, based on their α scores. γ is a parameter that can be tuned to the dataset and denotes the *proportion* of words in a sentence that can be considered attributes, while $|x|$ denotes the number of tokens in x. Further, the style classifier is used to decide the style of $\hat{y}$ according to:

$$\hat{s}^{tgt} = \arg\max_{s \in S} P(s|\hat{y}) \tag{4}$$

The reward assignment is as follows:

$$RS_t^k = \begin{cases} +thr_t^k * P(\hat{s}^{tgt}|\hat{y}^k), & \text{if } \hat{s}^{tgt} = s^{tgt} \\ -thr_t^k * P(\hat{s}^{tgt}|\hat{y}^k), & \text{else} \end{cases} \tag{5}$$

where,

$$thr_t^k = \{+1 \text{ if } \alpha_t^k \text{ is in the top } \gamma|x| \text{ attention weights, else } 0\} \tag{6}$$

Fluency Reward: We train a fresh language model LM over the entire training dataset using GPT's architecture and pre-training, which we then use to determine the fluency of generated sentences. LM generates a probability distribution $P(\hat{y}_t|y_{1:t-1}; \theta_{LM})$ over tokens at timestep t, given the tokens generated in previous timesteps. The reward assignment is as follows (where b is a baseline, set such that words having a LM probability lower than b will get penalized with a negative reward):

$$RF_t^k = P(\hat{y}_t^k|\hat{y}_{1:t-1}^k; \theta_{LM}) - b \tag{7}$$

Unlike works such as Gong et al. (2019) which use the perplexity of the entire sentence, we provide a fluency reward at the token-level.

Content Reward: We use the BLEU score (Papineni et al., 2001) between the input sentence x and the generated sentence $\hat{y}$ to calculate the content reward. The reward assignment is as follows:

$$RC_t^k = \frac{BLEU(x, \hat{y}^k)}{100} \tag{8}$$

As GST is trained on a reconstruction loss and RL-ST is warm-started with GST, it is already strongly biased to retaining content. Hence, setting the same (weak) content reward for all tokens works well enough.

Overall Reward: The overall reward (R) is a weighted sum of the above rewards:

$$R_t^k = \lambda_S RS_t^k + \lambda_C RC_t^k + \lambda_F RF_t^k \tag{9}$$

Figure 1 shows a training example with rewards and describes the training algorithm of RL-ST.

Inference: During inference, we decode the output using beam search, with a beam width of 20. Using the classifier described in equation 4, we choose the beam with the highest classifier score as the final output sentence.

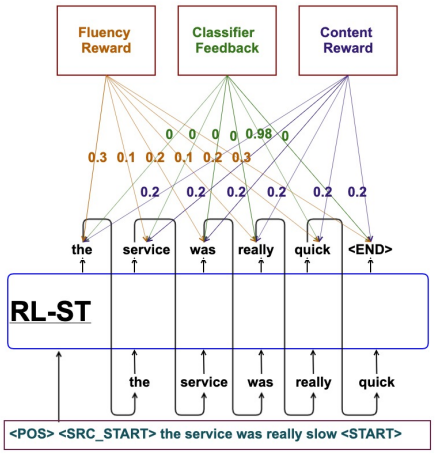
3 EXPERIMENTS

3.1 DATASETS

We show our results on the **YELP** and **CAPTIONS** dataset as used by Li et al. (2018), and retain the same train-dev-test split as they do. We also show results on the **GYAFC** dataset as used and released by (Rao & Tetreault, 2018). All of these datasets are used in a non-parallel manner. YELP is used for sentiment transfer, CAPTIONS is used for transfer of factual sentences to romantic and humorous ones, and GYAFC is used for formality transfer. Human reference outputs are available on all test sets. Further descriptions of these datasets can be found in Appendix Section A.2, and train-dev-test statistics of these datasets are shown in Appendix Table 5.

3.2 COMPARISON WITH PREVIOUS WORKS

We compare our models with the following 13 models on unsupervised style transfer from previous work: Cross Aligned (**CA**) (Shen et al., 2017), Style Embedding (**SE**) (Fu et al., 2018), Multi



(a) RL-ST Rewards and Input Representation

Algorithm 1: RL-ST Training Steps

Pre-train RL-ST (θ) using MLE with pseudo-parallel sentence pairs
Pre-train Style Classifier (θ_{CLS})
Pre-train Language Model (θ_{LM})
for each iteration 1,2...M:
  Sample a batch of N sentences from the training corpus
  for each sampled sentence x:
    Generate K style transformed sentences
    Calculate style rewards using eq. 5
    Calculate fluency reward using eq. 7
    Calculate content reward using eq. 8
    Calculate total reward using eq. 9
    Calculate gradient using eq. 2
  end **for**
  Backpropagate the averaged gradients for the batch
end **for**

(b) RL-ST Training Algorithm

Figure 1: a) An example from YELP for the task of sentiment transfer. <POS> represents positive target style, <SRC_START> indicates the start of the input sentence, <START> indicates the start of the output sentence and <END> is end-of-sentence marker. b) RL-ST's training algorithm

Decoder (**MD**) (Fu et al., 2018), Unpaired (**UnP**) (Xu et al., 2018), DeleteOnly (**DO**) (Li et al., 2018), DeleteAndRetrieve (**DR**) (Li et al., 2018), Back-translation (**BT**) (Prabhumoye et al., 2018), Unsupevised MT (**UnMT**) (Zhang et al., 2018), Revision in Continuous Space (**RC**) (Liu et al., 2019), Masked Language Model (**MLM**) (Wu et al., 2019c), Point-Then-Operate (**PTO**) (Wu et al., 2019a), **B-GST** (Sudhakar et al., 2019), **G-GST** (Sudhakar et al., 2019) and DualRL (**DRL**) (Luo et al., 2019). Each of these models were among the state-of-art models at different points of time.

4 RESULTS

We evaluate our models and the previous models using automatic evaluation methods as well as by human evaluation.

Automatic Evaluation: To measure target style strength, we train FastText[2] (Joulin et al., 2017) classifiers on our style datasets, keeping the same train-dev-test split intact as is in Table 5 and use these classifiers as oracles to judge style of output sentences (AC). These classifiers achieve accuracies of 96.5%, 89.5% and 80.5% on the test sets of YELP, GYAFC and CAPTIONS respectively. For content preservation, we calculate the average BLEU (BL_R) scores of the output with respect to the human reference sentences. Fluency is estimated by finetuning pre-trained OpenAI GPT-2 (Radford et al., 2019) models (different from any of the GPT models used in this work) on the training sets and using it to obtain the perplexity (PL) of the output sentences. The GPT-2 models achieve perplexities of 21.42, 52.5 and 42.91 on the test sets of YELP, GYAFC and CAPTIONS respectively. We also calculate the harmonic mean (HM) and geometric mean (GM) of AC and BL_R. These results are presented in Table 1.

Human Evaluation: We obtain human evaluations from crowd workers on MTurk [3] on pairs of model outputs, where one of the models in each pair is from previous work and the other is our model, RL-ST. Five top performing state-of-art models whose results are available on each of the datasets, were chosen based on our automatic evaluations as well as human evaluations by previous works (Luo et al., 2019; Sudhakar et al., 2019; Wu et al., 2019c;a) . For each example, three separate annotators who are not told which model is ours, are asked to choose which of the model's outputs is better (or 'None' if both are poor) on style (Sty.), content (Cont.) and fluency (Flu.) as well as overall style transfer (All). They are all native English speakers from North America and are familiar with the datasets. These results are presented in Table 2.

[2]https://fasttext.cc/
[3]https://www.mturk.com/

	YELP					GYAFC					CAPTIONS				
Model	**AC**	**BL_R**	**PL ↓**	**HM**	**GM**	**AC**	**BL_R**	**PL ↓**	**HM**	**GM**	**AC**	**BL_R**	**PL ↓**	**HM**	**GM**
SE	9.5	53.4	115.9	16.1	22.5	30.3	22.3	163.1	25.7	26.0	54.3	27.8	80.3	36.8	38.9
BT	95.5	25.6	28.5	40.4	49.4	51.5	18.7	147.2	27.4	31.0	-	-	-	-	-
MD	50.6	42.9	205.5	46.4	46.6	26.1	26.8	231.5	26.4	26.4	66.0	25.0	40.5	36.3	40.7
UnP	53.3	46.3	294.2	49.6	49.7	68.5	12.3	133.9	20.8	29.0	-	-	-	-	-
RC	90.1	37.3	**12.5**	52.8	58.0	-	-	-	-	-	-	-	-	-	-
DO	85.9	44.4	75.8	58.5	61.7	26.0	42.4	183.1	32.2	33.2	82.3	35.5	52.5	49.6	54.1
DR	87.8	45.6	90.0	60.0	63.3	57.4	41.2	176.4	48.0	48.6	95.0	38.1	28.8	**54.4**	60.1
G-GST	79.6	57.1	64.4	66.5	67.4	-	-	-	-	-	67.8	40.2	49.2	50.5	52.2
B-GST	86.7	57.1	38.6	68.9	70.4	69.0	47.4	89.0	56.2	57.2	70.5	**42.7**	28.9	53.2	54.9
UnMT	97.2	53.6	67.4	69.1	72.2	67.5	44.9	112.6	53.9	55.1	-	-	-	-	-
DRL	89.2	59.5	53.2	71.4	72.9	60.1	44.3	321.1	51.0	51.6	-	-	-	-	-
PTO	86.2	**61.9**	55.6	72.1	73.1	-	-	-	-	-	-	-	-	-	-
MLM	91.5	59.6	75.1	72.2	73.9	-	-	-	-	-	-	-	-	-	-
RL-ST	**98.1**	58.7	35.9	**73.5**	**75.9**	**84.2**	**52.3**	**64.8**	**64.5**	**66.3**	**98.8**	36.5	**25.7**	53.3	**60.1**
H	75.0	70.3	57.7	72.5	72.6	86.4	65.4	91.9	74.5	75.2	80.5	100.0	41.4	89.2	89.7

Table 1: Automatic evaluation results. AC = Style Accuracy; BL_R = Average BLEU score w.r.t human reference sentences; PL = Perplexity; HM = Harmonic Mean of (AC, BL_R); GM = Geometric Mean of (AC, BL_R); RL-ST is our model; H is Human reference. Lower PL is better. A '-' indicates that we could not obtain results for the model on the dataset.

YELP					GYAFC					CAPTIONS				
Model	**Cont.**	**Flu.**	**Sty.**	**All**	**Model**	**Cont.**	**Flu.**	**Sty.**	**All**	**Model**	**Cont.**	**Flu.**	**Sty.**	**All**
PTO	29.7	21.3	31.4	32.4	DO	19.9	20.3	20.1	22.3	DO	40.3	35.3	36.4	33.0
RL-ST	**51.3**	**59.4**	**48.6**	**50.9**	**RL-ST**	**62.3**	**60.1**	**59.9**	**60.7**	**RL-ST**	**46.5**	**52.2**	**50.3**	**54.5**
None	19.0	19.3	20	16.7	None	17.8	19.6	19	17	None	13.2	12.5	13.3	12.4
B-GST	37.1	27.3	27.2	28.6	B-GST	33.3	30.1	29.4	28.1	B-GST	35.8	39.7	38.9	33.3
RL-ST	**43.4**	**60.4**	**58.1**	**54**	**RL-ST**	**48.4**	**51.3**	**49.7**	**50.6**	**RL-ST**	**48.4**	**44.4**	**47.5**	**53.8**
None	19.5	12.3	14.7	17.4	None	18.3	18.6	20.9	21.3	None	15.8	16.0	13.6	12.9
DRL	32.6	**40.7**	22.0	29.8	DRL	**40.5**	38.8	32.0	34.8	MD	33.0	30.6	31.3	30.3
RL-ST	**50.7**	40.2	**57.7**	**50.3**	**RL-ST**	38.7	**45**	**47.6**	**47.8**	**RL-ST**	**52.3**	**57.0**	**56.4**	**59.5**
None	16.7	19.1	20.3	19.9	None	20.8	16.2	20.4	17.4	None	14.7	12.4	12.3	10.2
MLM	32.4	38.3	31.2	34.0	DR	25.2	23.8	28.1	24.0	DR	36.3	38.1	41.0	39.3
RL-ST	**51.1**	**40.0**	**46.4**	**46.1**	**RL-ST**	**58.6**	**49.9**	**50.4**	**55.2**	**RL-ST**	**46.3**	**48.3**	**43.1**	**49.2**
None	16.5	21.7	22.3	19.9	None	16.2	26.3	21.5	20.8	None	17.4	13.6	15.9	11.5
UnMT	31.8	33.5	29.0	30.0	UnMT	37.7	39.1	30.0	35.1	G-GST	**45.4**	36.4	38.6	38.1
RL-ST	**54.3**	**47.9**	**51.7**	**52.7**	**RL-ST**	**51.2**	**46.8**	**52.3**	**50.5**	**RL-ST**	42.3	**50.3**	**47.5**	**52.0**
None	13.9	18.6	19.3	17.3	None	11.1	14.1	17.7	14.4	None	12.3	13.3	13.9	9.9

Table 2: Human evaluation results: each 3-set of rows indicates the percentage of sentences preferred for each model in the pair (and 'None'), down a column. Cont. = Content Preservation ; Flu. = Fluency ; Sty. = Target Style Match ; All = Overall.

4.1 Analysis and Discussion

As has been observed by most previous works, the **automatic** evaluations in Table 1 show that many previous works trade-off target style match and content retention. It is easy to achieve very high numbers on either of AC or BL_R. A model that simply copies the input sentence will achieve high BL_R and a model that simply chooses a random target training sentence will achieve high AC score. SE has very low AC but considerably high BL_R while BT has a high AC but considerably low BL_R. However, RL-ST achieves very high target style accuracy but not at the cost of content retention - it achieves considerably good BL_R scores too. HM and GM scores indicate how well the models perform on both style and content. Our model (RL-ST) ranks highest on both these scores across datasets (except on HM for CAPTIONS, where it ranks the second highest) as well as achieves low PL, outperforming even the average scores of all the human references on HM, GM and PL on YELP. However, automatic metrics do not capture nuances that human evaluation does. For instance, on the Yelp dataset, DRL is biased towards frequently using the attributes 'loved' and 'happy' in its positive outputs, and PTO over-uses the word 'delighted' even in sentences where it is not meaningful to. The classifier still awards these outputs high AC scores. Further, model-based metrics such as AC and PL are also sensitive to the quality of their training data available.

From **human** evaluations in Table 2, we see our model outperforms previous state-of-art models by a good margin on all metrics across all datasets. On the Overall scores (All), we outperform previous state-of-the art models by 19.8%, 24.5% and 19% on YELP, GYAFC and CAPTIONS respectively, averaging across the top performing models considered for human evaluation in Table 2 for each of these datasets. From manual inspection we observe that RL-ST performs better than previous state-of-art models in the following ways: 1) generates sentences that are more natural sounding,

YELP (Positive to Negative)	YELP (Negative to Positive)
steve was professional and found exactly the right unit to fit in our space **steve was rude and didn't have the right unit to fit in our space .**	they tried to take advantage of me because i am young . **they take great care of me because i am young .**
GYAFC (Formal to Informal)	**GYAFC (Informal to Formal)**
do not approach her and let her know that you find her looks very attractive. **don't approach her and let her know that you like her .**	well that is just the way it is I guess. **that is just the way it is , i would advise .**
CAPTIONS (Factual to Romantic)	**CAPTIONS (Factual to Humorous)**
young man performing bicycle trick on loading dock near dumpsters . **young man performing bicycle trick on ramp near a group of people enjoying life .**	young man performing bicycle trick on loading dock near dumpsters . **young man performing bicycle trick on dock near a crowd of aliens .**

Table 3: Examples of generated sentences by RL-ST. Each cell has the source sentence first and the generated sentence second.

retaining core content better while making only necessary stylistic changes, 2) maintains consistent context across longer sentences, 3) performs well on sentences in which style transfer is not limited to simple localized edits, 4) maintains consistency of style even for certain input sentence structures (e.g., sentences having multiple attribute words - *it's nice but it's too expensive*) which cause other models to produce outputs having inconsistent style (for e.g., *it's too expensive , but it's worth it*), 5) produces output sentences having appropriate and meaningful attributes, many of which the model has not observed at training time, and 6) does away with redundancy in output sentences that is commonly observed in previous works (e.g., *the food is good, the service is great and the food is good .)*

Table 3 shows examples of our outputs for the three datasets, and Appendix Table 6 shows more results of our models and compares it with those of other models.

Failure Cases: One observable behavior of RL-ST is that it sometimes simply retains conjunction words of the source sentence (such as *and, but*) instead of adapting it to the output. For example, *their chips are great , but their salsa is really good*. The second type of failure case occurs when analogies are used to indicate a certain sentiment in the source sentence, and sentiment attributes are not directly used (e.g., *they only received one star because you have to provide a rating*). In these cases, the model finds it hard to identify and replace these analogies (*they also enjoy one star because you have to provide a rating*).

4.2 Ablation Studies

We perform five ablation studies on RL-ST using the YELP dataset. When ablating over a particular aspect of training, all other aspects are kept fixed and the same as those of RL-ST in Table 1. Table 4 shows results of these ablations, whose explanations are given below.

RL and MLE: We compare the performance obtained by using only RL without MLE pre-training (RL-Only), only MLE pre-training without RL (MLE-Only) and the combined model (RL+MLE). We see that warm-starting using MLE significantly boosts performance.

Disentanglement: We also study the effects of providing RL-ST only the content of the source sentence during training and inference (Cont-Only), as against providing it with the entire source sentence (Full-Src) during training and inference. The results show that providing the model the full sentence including source attributes is more beneficial than giving it only the content in the input.

Sample efficiency: While previous works that use RL for style transfer require large training sets and consequently large training time, RL-ST is highly sample efficient, requiring only a fraction of the training set to boost performance significantly. We ablate on training set sizes of 1K, 2K, and 4K samples out of the training set of 450K samples.

Rewards: We distill the effects of using only the style reward, only the fluency reward (FluencyR-Only) and the combined reward (Combined). The results show that the style and fluency rewards are indeed successfully able to control for style and fluency respectively, as expected. We also ablate on two versions of the style reward - one in which we use attention scores to assign rewards as described in equation 5 (StyleR-Only-Attn), and the other in which we assign uniform style rewards to all tokens regardless of attention scores (StyleR-Only-Uniform). StyleR-Only-Attn yields outputs of superior style accuracy.

Model	AC	BL_R	PL ↓	HM	GM
RL and MLE					
RL-Only	60.7	59.2	40.8	59.9	59.9
MLE-Only	93.0	**60.0**	37.0	73.0	74.7
RL+MLE (RL-ST)	**98.1**	58.7	**35.9**	**73.5**	**75.9**
Disentanglement					
Cont-Only	93.6	51.2	44.2	66.2	69.2
Full-Src (RL-ST)	**98.1**	**58.7**	**35.9**	**73.5**	**75.9**
Sample efficiency					
1k-Train	96.5	58.4	**33.8**	72.7	75.1
2k-Train (RL-ST)	**98.1**	**58.7**	35.9	**73.5**	**75.9**
4k-Train	96.0	58.2	35.72	72.5	74.7
Rewards					
StyleR-Attn-Only	97.1	59.3	43.0	**73.6**	75.9
StyleR-Uniform-Only	95.5	**59.4**	41.6	73.3	75.3
FluencyR-Only	85.1	44.8	**16.8**	58.7	61.7
Combined (RL-ST)	**98.1**	58.7	35.9	73.5	**75.9**
Decoding Strategies					
Top-p	**99.0**	56.7	**35.6**	72.1	74.9
Beam-Search (RL-ST)	98.1	**58.7**	35.9	**73.5**	**75.9**

Table 4: Ablation results AC = Style Accuracy; BL_R = Average BLEU score w.r.t human reference sentences; PL = Perplexity; HM = Harmonic Mean of (AC, BL_R); GM = Geometric Mean of (AC, BL_R). Lower PL is better.

Decoding Strategies during Inference: At test time, we experiment with two decoding strategies - top-p sampling and beam search. Beam search yields marginally better results on HM and GM, but top-p has marginally better PL.

5 Related Work

This brief section refers to a few more works not covered in 1. One category of previous works is based on unsupervised machine translation (Artetxe et al., 2018; Lample et al., 2018a;b) of which most approaches use a back-translation based system (Zhang et al., 2018; Lample et al., 2019). In a slightly different approach, Prabhumoye et al. (2018) use back-translation from English to French and back to English in order to get an intermediate representation that has reduced style, and then generate style-specific outputs by training adversarially. John et al. (2019) attempt to learn disentangled representations for style and content using a system that incorporates auxiliary multi-task and adversarial objectives, for style prediction and bag-of-words prediction, respectively. Zhao et al. (2018) use a GAN-like training approach, with a style discrepancy loss and a cycled consistency loss to transfer from sentences with arbitrary unknown styles to known target styles. Wu et al. (2019a) use a hierarchical reinforcement operations method (Point-then-Operate) wherein a high-level agent iteratively 'points' to positions in the sentence and a low-level agent 'operates' by altering the sentence at these positions. Liu et al. (2019) perform revision in continuous space by which explicit content disentanglement is not needed, and neither is adversarial training. They control for fine-grained multi-attributes such as length of the output sentence. Pang & Gimpel (2018) examine the complimentarity of the three style transfer metrics discussed earlier, trade-offs between them, and a common metric that summarizes them into one score using the geometric mean. Tikhonov et al. (2019) discuss significant problems with standard assessment using automatic metrics of style and content retention. They claim that the nature of style transfer itself lends a specific dependency between the two metrics, which can be manipulated. Hence, human evaluation is imperative.

6 Conclusion

We present an RL-based, sample efficient style transfer model that outperforms current state-of-art systems on human as well as automatic evaluations. The approach is generalizable across a variety of style transfer tasks, as we show with diverse datasets. We show the merits of directly learning to map source to target sentences without disentanglement, shaping RL rewards efficiently, and leveraging the power of massively pre-trained transformer-based language models.

REFERENCES

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Sy2ogebAW.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation, 2019.

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

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3719–3728, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1407.

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

Hongyu Gong, Suma Bhat, Lingfei Wu, Jinjun Xiong, and Wen mei Hwu. Reinforcement learning based text style transfer without parallel training corpus, 2019.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2019.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1587–1596, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/hu17e.html.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*, 2017.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 424–434, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1041. URL https://www.aclweb.org/anthology/P19-1041.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 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*, pp. 427–431. Association for Computational Linguistics, April 2017.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=rkYTTf-AZ.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 5039–5049, Brussels, Belgium, October-November 2018b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1549.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. Multiple-attribute text rewriting. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1g2NhC5KQ.

Juncen Li, Robin Jia, He He, and Percy Liang. 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)*, pp. 1865–1874, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1169. URL https://www.aclweb.org/anthology/N18-1169.

Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. Revision in continuous space: Fine-grained control of text style transfer. *arXiv preprint arXiv:1905.12304*, 2019.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. A dual reinforcement learning framework for unsupervised text style transfer. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Aug 2019. doi: 10.24963/ijcai.2019/711. URL http://dx.doi.org/10.24963/ijcai.2019/711.

Yuanzhe Pang and Kevin Gimpel. Learning criteria and evaluation metrics for textual transfer between non-parallel corpora. *arXiv preprint arXiv:1810.11878*, 2018.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2001.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkAClQgA-.

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

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Sudha Rao and Joel Tetreault. 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)*, pp. 129–140, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1012. URL https://www.aclweb.org/anthology/N18-1012.

Sravana Reddy and Kevin Knight. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pp. 17–26, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-5603. URL https://www.aclweb.org/anthology/W16-5603.

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*, pp. 6830–6841, 2017.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. Transforming delete, retrieve, generate approach for controlled text style transfer, 2019.

Alexey Tikhonov, Viacheslav Shibaev, Aleksander Nagaev, Aigul Nugmanova, and Ivan P. Yamshchikov. Style transfer for texts: Retrain, report errors, compare with rewrites, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

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

Chen Wu, Xuancheng Ren, Fuli Luo, and Xu Sun. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4873–4883, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1482. URL https://www.aclweb.org/anthology/P19-1482.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 5271–5277. International Joint Conferences on Artificial Intelligence Organization, 7 2019b. doi: 10.24963/ijcai.2019/732. URL https://doi.org/10.24963/ijcai.2019/732.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. " mask and infill": Applying masked language model to sentiment transfer. *arXiv preprint arXiv:1908.08039*, 2019c.

Jingjing Xu, Xu SUN, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 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)*, pp. 979–988, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P18-1090.

Cheng Yang, Maosong Sun, Xiaoyuan Yi, and Wenhao Li. Stylistic Chinese poetry generation via unsupervised style disentanglement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3960–3969, Brussels, Belgium, October-November 2018a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D18-1430.

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*, pp. 7287–7298, 2018b.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

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

Yanpeng Zhao, Wei Bi, Deng Cai, Xiaojiang Liu, Kewei Tu, and Shuming Shi. Language style transfer from sentences with arbitrary unknown styles, 2018.

Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. Mechanism-aware neural machine for dialogue response generation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

A Appendix

A.1 Training details

Similar to GST (Sudhakar et al., 2019), the base architecture of RL-ST is the PyTorch implementation of the pre-trained Transformer by HuggingFace[4], which uses the pre-trained OpenAI GPT model[5]. Radford et al. have pre-trained this model on the BookCorpus dataset[6] of over 7000 books (around 800M words). It has a sequence length of 512, 12 blocks (or layers), and 12 attention-heads

[4]https://github.com/huggingface/pytorch-pretrained-BERT

[5]https://github.com/openai/finetune-transformer-lm

[6]https://www.smashwords.com/

in each block. All internal states (keys, queries, values, word embeddings, positional embeddings) are 768-dimensional. Input text is tokenized using Byte-Pair Encoding (BPE).

In equation 9, λ_S = 1, λ_C = 0.3 and λ_F = 1 for all 3 datasets.

A.2 Datasets Descrption and Statistics

Following are brief descriptions of the datasets we use, borrowed from works that use them previously.
YELP: Each example is a sentence from a business review on Yelp, and is labeled as having either positive or negative sentiment (Li et al., 2018). The task is to transfer sentences of positive to negative sentences and vice-versa. Li et al. (2018) publish a set of human reference outputs on the test set of YELP, which is further extended to four sets by Luo et al. (2019).
CAPTIONS: Each image caption in this dataset is labeled as either factual, romantic, or humorous. The task is to convert factual sentences into romantic and humorous ones. While CAPTIONS is an aligned corpus, containing captions for the same image in different styles, we do not use the alignments. The task is to transfer factual captions to romantic and humorous ones. (Li et al., 2018; Sudhakar et al., 2019). Li et al. (2018) publish a set of human reference outputs on the test set of CAPTIONS.
GYAFC: Each sentence is labeled as either being formal or informal. We only use a subset of this dataset which corresponds to Yahoo answers in the Family and Relationships domain. This is also an aligned corpus, but we do not use these alignments either. The task is to transfer formal to informal sentences and vice-versa. Rao & Tetreault (2018) publish a set of human reference outputs on the test set of GYAFC.

Table 5 shows train, dev and test statistics of the datasets.

Dataset	Style	Train	Dev	Test
YELP	Positive	270K	2000	500
	Negative	180K	2000	500
GYAFC	Formal	277K	985	500
	Informal	279K	1015	500
CAPTIONS	Romantic	6000	300	0
	Humorous	6000	300	0
	Factual	0	0	300

Table 5: Dataset statistics

A.3 Example Outputs

Table 6 below compares our model's outputs with those of previous works.

#1	**YELP (Positive to Negative)**
SRC	steve was professional and found exactly the right unit to fit in our space .
DRL	steve was unprofessional and found exactly the right unit to fit in our space
UnMT	manager was unprofessional and left exactly the off unit to replace in our space
PTO	steve was professional and the horrible unit to fit in our space .
MLM	steve was rude and found only the wrong unit was not in our space
RL-ST	**steve was rude and did n't have the right unit to fit in our space .**
#2	**YELP (Negative to Positive)**
SRC	they tried to take advantage of me because i am young .
DRL	they tried to take advantage of me because i am young perfectly .
UnMT	they tried to take advantage of me because i am young and very professional .
PTO	they great to take advantage of me because i am young .
MLM	they love to take advantage of me because i am young .
RL-ST	**they take great care of me because i am young .**
#3	**GYAFC (Formal to Informal)**
SRC	i am not certain whether he loves you , but he definitely likes you .
DRL	i i am not certain er
UnMT	i 'm not certain whether he loves you but he definitely likes you , really does it ?
UnP	not is true friends did he then yes and good luck
DR	i am not you whether he loves you , but he likes you you .
RL-ST	**i dont know if he loves you but he definitely likes you .**
#4	**GYAFC (Informal to Formal)**
SRC	well that is just the way it is i guess .
DRL	i well , that is just the way it is guess .
UnMT	well that is what the way it is i would guess .
UnP	is a good reason to be
DR	well i believe that is just the way it is i guess .
RL-ST	**that is just the way it is , i would advise .**
#5	**CAPTIONS (Factual to Romantic)**
SRC	young man performing bicycle trick on loading dock near dumpsters .
DO	young man performing bicycle trick on dock and enjoys time in excitement .
DR	young man performing bicycle on bicycle UNK cliff to experience the adventure of life .
B-GST	young man performing bicycle trick on loading dock near dumpdumpsters looking for pokemon .
MD	young man rides down wave on motorcycle , track with dirt .
RL-ST	**young man performing bicycle trick on ramp near a group of people enjoying life**
#6	**CAPTIONS (Factual to Humorous)**
SRC	young man performing bicycle trick on loading dock near dumpsters .
DO	young man performing bicycle on bicycle tracks for time in sky .
DR	young man performing bicycle trick on bicycle near crowd , looking for outer space .
B-GST	young man performing bicycle trick on loading dock near dumpdumpsters looking for aliens .
MD	young man doing skateboard trick on rocks near tennis area .
RL-ST	**young man performing bicycle trick on dock near a crowd of aliens .**

Table 6: Examples of generated sentences to be compared down a column (RL-ST is our model, SRC is the input sentence). Attributes are colored.