A Hierarchical Reinforced Sequence Operation Method for Unsupervised Text Style Transfer

Chen Wu[1]*, **Xuancheng Ren[2]***, **Fuli Luo[2], Xu Sun[2,3]**

[1]Department of Foreign Languages and Literatures, Tsinghua University
[2]MOE Key Laboratory of Computational Linguistics, School of EECS, Peking University
[3]Center for Data Science, Beijing Institute of Big Data Research, Peking University

wu-c16@mails.tsinghua.edu.cn
{renxc, luofuli, xusun}@pku.edu.cn

Abstract

Unsupervised text style transfer aims to alter text styles while preserving the content, without aligned data for supervision. Existing seq2seq methods face three challenges: 1) the transfer is weakly interpretable, 2) generated outputs struggle in content preservation, and 3) the trade-off between content and style is intractable. To address these challenges, we propose a hierarchical reinforced sequence operation method, named **Point-Then-Operate** (**PTO**), which consists of a high-level agent that proposes operation positions and a low-level agent that alters the sentence. We provide comprehensive training objectives to control the fluency, style, and content of the outputs and a mask-based inference algorithm that allows for multi-step revision based on the single-step trained agents. Experimental results on two text style transfer datasets show that our method significantly outperforms recent methods and effectively addresses the aforementioned challenges. [1]

1 Introduction

Text style transfer aims to convert a sentence of one style into another while preserving the style-independent content (Shen et al., 2017; Fu et al., 2018). In most cases, aligned sentences are not available, which requires learning from non-aligned data. Previous work mainly learns disentangled content and style representations using seq2seq (Sutskever et al., 2014) models and decomposes the transfer into *neutralization* and *stylization* steps. Although impressive results have been achieved, three challenges remain: 1) the interpretability of the transfer procedure is still weak in seq2seq models, 2) generated sentences are usually highly stylized with poor content preserva-

---

*Equal Contributions.

[1] Our code is available at https://github.com/ChenWu98/Point-Then-Operate.
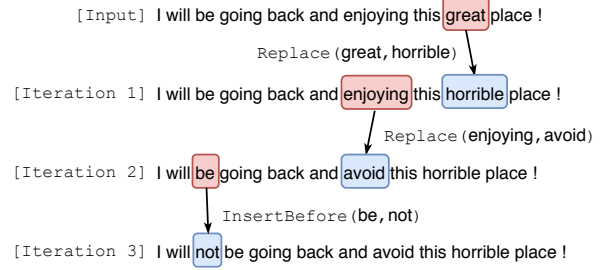

Figure 1: Our proposed Point-Then-Operate (PTO) applied to a real test sample. A high-level agent (red squares) iteratively proposes operation positions, and a low-level agent (arrows) alters the sentence based on the high-level proposals. Compared with seq2seq methods, PTO is more interpretable and better preserves style-independent contents.

tion, and 3) the trade-off between content preservation and style polarity is intractable.

To address these challenges, we propose a sequence operation-based method within the hierarchical reinforcement learning (HRL) framework, named **Point-Then-Operate** (**PTO**). It consists of a hierarchy of a high-level agent that proposes operation positions and a low-level agent that alters the sentence based on high-level proposals. We propose a policy-based training algorithm to model the key aspects in text style transfer, i.e., fluency, style polarity, and content preservation. For fluency, we use a *language model reward*; for style polarity, we introduce a *classification confidence reward* and an auxiliary classification task; for content preservation, we adopt a *reconstruction reward* and a self-supervised *reconstruction loss*. We introduce a mask-based inference algorithm that applies multi-step sequence operations to the input sentence, allowing for single-step training which is more stable. Figure 1 shows an example of our method applied to a real test sample from Yelp.

Compared with existing seq2seq methods, our

sequence operation method has three merits. 1) *Interpretability*: our method explicitly models *where* and *how* to transfer. 2) *Content preservation*: sequence operations are targeted at stylized parts; thus, style-independent content can be better preserved. 3) *Controllable trade-off*: the trade-off between content preservation and style polarity could be tuned in our method. Specifically, we tune it by biasing the number of operation steps.

We conduct extensive experiments on two text style transfer datasets, i.e., Yelp and Amazon. We show that our proposed method outperforms recent methods and that it addresses the challenges of existing seq2seq methods. The contributions of this paper are:

- We propose a sequence operation method, i.e., Point-Then-Operate, for unsupervised text style transfer. The transfer procedure is modeled as explicit revisions on the input sentences, which improves interpretability, content preservation, and controllable style-content trade-off.

- The method is interpreted and trained in the HRL framework with a high-level agent that proposes operation positions and a low-level agent that applies explicit operations. We design comprehensive learning objectives to capture three important aspects of text style transfer and propose a mask-based inference algorithm that allows for multi-step revision based on the single-step trained agents.

- Experiments on Yelp and Amazon show that our method significantly improves BLEU, fluency, and content preservation compared with recent methods and effectively addresses the aforementioned challenges.

2 Related Work

Text Style Transfer Most work on text style transfer learns disentangled representations of style and content. We categorize them based on how they represent content. *Hidden vector approaches* represent content as hidden vectors, e.g., Hu et al. (2017) adversarially incorporate a VAE and a style classifier; Shen et al. (2017) propose a cross-aligned AE that adversarially aligns the hidden states of the decoder; Fu et al. (2018) design a multi-decoder model and a style-embedding model for better style representations; Yang et al. (2018) use language models as

style discriminators; John et al. (2018) utilize bag-of-words prediction for better disentanglement of style and content. *Deletion approaches* represent content as the input sentence with stylized words deleted, e.g., Li et al. (2018) delete stylized n-grams based on corpus-level statistics and stylize it based on similar, retrieved sentences; Xu et al. (2018) jointly train a neutralization module and a stylization module the with reinforcement learning; Zhang et al. (2018a) facilitate the stylization step with a learned sentiment memory.

As far as we know, there are two work that avoid disentangled representations. Zhang et al. (2018b) construct a pseudo-aligned dataset with an SMT model and then learn two NMT models jointly and iteratively. A concurrent work, Luo et al. (2019), propose to learn two dual seq2seq models between two styles via reinforcement learning, without disentangling style and content.

Sequence Operation Methods Our work is also closely related to sequence operation methods, which are widely used in SMT (Durrani et al., 2011, 2015; Pal et al., 2016) and starts to attract attention in NMT (Stahlberg et al., 2018). Compared with methods based on seq2seq models, sequence operation methods are inherently more interpretable (Stahlberg et al., 2018). Notably, our method is revision-based, i.e., it operates directly on the input sentence and does not generate from scratch as in machine translation systems.

Hierarchical Reinforcement Learning In this work, we adopt the Options Framework (Sutton et al., 1999) in HRL, in which a high-level agent learns to determine more abstract *options* and a low-level agent learns to take less abstract *actions* given the option. Recent work has shown that HRL is effective in various tasks, e.g., Atari games (Kulkarni et al., 2016), relation classification (Feng et al., 2018), relation extraction (Takanobu et al., 2018), and video captioning (Wang et al., 2018).

3 Formulation

We start by formalizing the problem of our interest. Given two non-aligned sets of sentences $\mathcal{X}_1 = \{x_1^{(1)}, \cdots, x_1^{(n)}\}$ of style s_1 and $\mathcal{X}_2 = \{x_2^{(1)}, \cdots, x_2^{(m)}\}$ of style s_2. Unsupervised text style transfer aims to learn two conditional distributions $p(x_{1\to 2}|x_1)$ and $p(x_{2\to 1}|x_2)$ which alter the style of a sentence and preserve the style-

independent content. However, defining *content* is not trivial. Different from previous text style transfer methods that *explicitly* model contents with disentangled representations, we *implicitly* model content with reconstruction, similar to the idea proposed adopted in CycleGAN (Zhu et al., 2017). Given the discreteness nature of natural language texts, we use sequence operations to approximate $p(\boldsymbol{x}_{1\rightarrow 2}|\boldsymbol{x}_1)$ and $p(\boldsymbol{x}_{2\rightarrow 1}|\boldsymbol{x}_2)$. In our notations, $\boldsymbol{x}_{1\rightarrow 2}$ and $\boldsymbol{x}_{2\rightarrow 1}$ are *transferred sentences*, which are the outputs of a text style transfer system; $\hat{\boldsymbol{x}}_2$ and $\hat{\boldsymbol{x}}_1$ are *operated sentences*, which are not necessarily fully transferred.

4 Our Approach

Our proposed sequence operation-based method, **Point-Then-Operate** (**PTO**), decomposes style transfer into two steps: 1) finding *where* to transfer and 2) determining *how* to transfer. It could be naturally formulated as an HRL problem, in which a high-level agent (i.e., *pointer*) proposes operation positions and a low-level agent (i.e., *operators*) alters the sentence based on high-level proposals.

In this section, we first briefly review the Options Framework in HRL. Then we introduce the proposed pointer module (§4.2) and operator modules (§4.3). The training algorithm is in §4.4, in which two extrinsic rewards, an intrinsic reward, and a self-supervised loss are proposed for fluency, style polarity, and content preservation. The inference algorithm is in §4.5, in which a mask mechanism is proposed to iteratively and dynamically apply sequence operations to the input.

4.1 Review: The Options Framework in HRL

The Options framework (Sutton et al., 1999) is a well-known formulation in HRL. We denote the state space as $\mathcal{S}$; the option space, $\mathcal{O}$; the action space, $\mathcal{A}$. The high-level agent learns a stochastic policy $\mu : \mathcal{S} \times \mathcal{O} \rightarrow [0, 1]$. The low-level agent learns a stochastic policy $\pi_o : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, conditioned on an option $o \in \mathcal{O}$. Additionally, each option $o \in \mathcal{O}$ has a low-level stochastic termination condition $\beta_o : S \rightarrow [0, 1]$ which indicates whether the current option should end. In each episode, the high-level agent executes a trajectory $(o_1, \cdots, o_L)$ based on μ; once an option o_t is sampled, the low-level agent executes a trajectory $(a_t^1, \cdots, a_t^{l_t})$ based on π_{o_t}, where l_t is dependent on β_{o_t}. Intuitively, the flattened trajectory for one episode is $(o_1, a_1^1, \cdots, a_1^{l_1}, \cdots, o_L, a_L^1, \cdots, a_L^{l_L})$.

Module	Operation
IF$_{\phi_1}$	**I**nsert a word $\hat{w}$ in **F**ront of the position
IB$_{\phi_2}$	**I**nsert a word $\hat{w}$ **B**ehind the position
Rep$_{\phi_3}$	**Rep**lace it with another word $\hat{w}$
DC	**D**elete the **C**urrent word
DF	**D**elete the word in **F**ront of the position
DB	**D**elete the word **B**ehind the position
Skip	Do not change anything

Table 1: Operator modules. Parameters ϕ_1, ϕ_2, and ϕ_3 are meant to generate their corresponding $\hat{w}$.

4.2 High-Level Agent: Pointer

The high-level policy μ aims to propose operation positions; thus, we model it as an attention-based (Bahdanau et al., 2015) *pointer* network, which assigns normalized probability to each position.

Option Given a sentence $\boldsymbol{x} = \{x_1, \cdots, x_T\}$, the option space is $\mathcal{O} = \{1, \cdots, T\}$. Note that T changes within an episode, since operations may change the length of a sentence.

State The state is represented by the sentence representation $\boldsymbol{h}_T$ and each position representation $\boldsymbol{h}_i$, where $\{\boldsymbol{h}_1, \cdots, \boldsymbol{h}_T\}$ is mapped from the sentence $\boldsymbol{x}$ by a bi-LSTM encoder.

Policy We adopt an attention-based policy μ:

$$\mu(i|\boldsymbol{x}) = \frac{\exp(a(\boldsymbol{h}_T, \boldsymbol{h}_i))}{\sum_{t=1}^{T} \exp(a(\boldsymbol{h}_T, \boldsymbol{h}_t))} \quad (1)$$

where $a(\cdot, \cdot)$ is the scoring function for attention, and $i \in \{1, \cdots, T\}$ denotes each position in the intput sentence.

4.3 Low-Level Agent: Operators

The low-level policy π alters the sentence around the position i (i.e., option) sampled from μ. We restrict the operations to those listed in Table 1. Note that these operations are complete to generate all natural language sentences in multiple steps.

Action Given the sentence $\boldsymbol{x} = \{x_1, \cdots, x_T\}$ and the operation position i, the action of the low-level agent can be decomposed into two step, i.e.,

1. *Operator selection.* Select an operator module from Table 1.

2. *Word generation* (optional). Generates a word, if necessary as is specified in Table 1.

State Compared with the high-level agent, our low-level agent focuses on features that are more local. We map x to $\{h_1, \cdots, h_T\}^2$ through a bi-LSTM encoder and take h_i as the state representation.

Low-Level Termination Condition Different from the original Options Framework in which a stochastic termination condition β_o is learned, we adopt a deterministic termination condition: the low-level agent takes *one* action in each option and terminates, which makes training easier and more stable. Notably, it does not harm the expressiveness of our method, since multiple options can be executed.

Policy for Operator Selection For training, we adopt a *uniform* policy for operator selection, i.e., we uniformly sample an operator module from Table 1. In preliminary experiments, we explored a learned policy for operator selection. However, we observed that the learned policy quickly collapses to a nearly deterministic choice of Rep_{ϕ_3}. Our explanation is that, in many cases, replacing a stylized word is the optimal choice for style transfer. Thus, the uniform policy assures that all operators are trained on sufficient and diversified data. For inference, we adopt a heuristic policy based on fluency and style polarity, detailed in §4.5.3.

Policy for Word Generation As shown in Table 1, three operators are parameterized, which are burdened with the task of generating a proper word to complete the action. For each parameterized operator M, the probability of generating $\hat{w}$ is

$$M(\hat{w}|x, i) = \text{softmax}_{\hat{w}}(\boldsymbol{W}h_i) \qquad (2)$$

Notably, for each M we train two sets of parameters for $s_1 \to s_2$ and $s_2 \to s_1$. For readability, we omit the direction subscripts and assure that they can be inferred from contexts; parameters of the opposite direction are denoted as ϕ_1', ϕ_2', and ϕ_3'.

4.4 Hierarchical Policy Learning

We introduce comprehensive training objectives to model the key aspects in text style transfer, i.e., fluency, style polarity, and content preservation. For fluency, we use an extrinsic *language model reward*; for style polarity, we use an extrinsic *classification confidence reward* and incorporate an auxiliary *style classification* task; for content

---

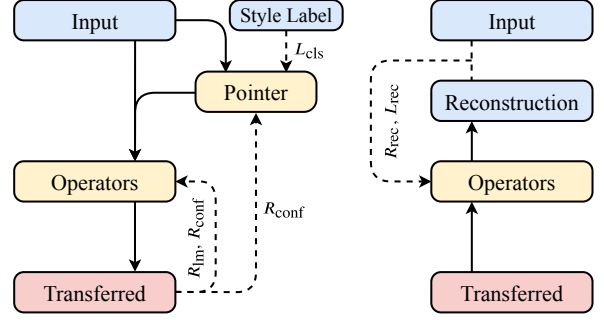^{2}We reuse h and W notations for all modules for brevity.

Figure 2: Graphical overview for the training algorithm, which consists of a *transfer* step (left) and a *reconstruction* step (right). Solid lines denote forward pass; dotted lines denote rewards or losses. Blue / red items belong to the source / target styles; yellow items denotes the agents. Best viewed in color.

Algorithm 1 Point-Then-Operate Training

1: **Input:** Non-aligned sets of sentences $\mathcal{X}_{1,2}$
2: Initialize θ, $\phi_{1,2,3}$
3: Train language models LM_2 on $\mathcal{X}_2$
4: Pre-train θ by optimizing $\mathcal{L}_{\text{cls}}^{\theta}$ ▷ Eq. 6
5: **for** each iteration $i = 1, 2, \cdots, m$ **do**
6:      Sample x_1 from $\mathcal{X}_1$
7:      Sample i from $\mu_\theta(i|x_1)$ ▷ Eq. 1
8:      Uniformly sample M ▷ Table 1
9:      $\hat{x}_2 \leftarrow \text{Transfer}(x_1, M, i)$ ▷ Table 1
10:      Compute R_{conf} and R_{lm} ▷ Eq. 3 and 4
11:      Update θ based on $\mathcal{L}_{\text{cls}}^{\theta}$ and $\nabla_\theta J(\theta)$ ▷ Eq. 6 and 9
12:      Get M' and i' ▷ Table 2
13:      **if** M' is parameterized by ϕ' **then**
14:          $\bar{x}_1 \leftarrow \text{Reconstruct}(\hat{x}_2, M', i')$ ▷ Table 1
15:          Update ϕ' by optimizing $\mathcal{L}_{\text{rec}}^{\phi'}$ ▷ Eq. 7
16:      **end if**
17:      **if** M is parameterized by ϕ **then**
18:          Compute R_{rec} if M is Rep_{ϕ_3} ▷ Eq. 8
19:          Update ϕ with $\nabla_\phi J(\phi)$ ▷ Eq. 11
20:      **end if**
21: **end for**

---

preservation, we use a self-supervised *reconstruction loss* and an intrinsic *reconstruction reward*. In the following parts, we only illustrate equations related to $x_1 \to \hat{x}_2$ operations and $\hat{x}_2 \to x_1$ reconstructions for brevity; the opposite direction can be derived by swapping 1 and 2. The training algorithm is presented in Algorithm 1. A graphical overview is shown in Figure 2.

4.4.1 Modeling Fluency

Language Model Reward To improve the fluency, we adopt a language model reward. Let LM_1, LM_2 denote the language models for s_1 and s_2, respectively. Given the generated word $\hat{w}$ in the operated sentence $\hat{x}_2$, the language model reward is defined as

$$R_{\text{lm}} = \lambda_{\text{lm}} \text{LM}_2(\hat{w}|\hat{x}_2) \qquad (3)$$

M	M'	i'
Rep_{ϕ_3}	$\text{Rep}_{\phi_3'}$	i
DC	$\text{IF}_{\phi_1'}$ or $\text{IB}_{\phi_2'}$	i or $i-1$
DF	$\text{IF}_{\phi_1'}$ or $\text{IB}_{\phi_2'}$	$i-1$ or $i-2$
DB	$\text{IF}_{\phi_1'}$ or $\text{IB}_{\phi_2'}$	$i+1$ or i

Table 2: Construction of self-supervised data.

where $\text{LM}_2(\hat{w}|\hat{\boldsymbol{x}}_2)$ denotes the probability of $\hat{w}$ given *other* words in $\hat{\boldsymbol{x}}_2$. In our experiments, the probability is computed by averaging a forward LSTM-LM and a backward LSTM-LM.

4.4.2 Modeling Style Polarity

Classification Confidence Reward We observe that language models are not adequate to capture style polarity; thus, we encourage larger change in the confidence of a style classifier, by adopting a classification confidence reward, i.e.,

$$R_{\text{conf}} = \lambda_{\text{conf}}[p(s_2|\hat{\boldsymbol{x}}_2) - p(s_2|\boldsymbol{x}_1)] \quad (4)$$

where we reuse the classifier defined in Eq. 5.

Auxiliary Task: Style Classification In HRL, the high-level policy usually suffers from the high variance of gradients since the estimated gradients are dependent on the poorly trained low-level policy. To stabilize the high-level policy learning, we introduce auxiliary supervision to the pointer. Specifically, we extend the pointer to an attention-based classifier, i.e.,

$$p(s_j|\boldsymbol{x}) = \text{softmax}_j(\boldsymbol{W}\sum_{i=1}^{T}\mu(i|\boldsymbol{x})\boldsymbol{h}_i) \quad (5)$$

for $j = 1, 2$. Let θ denotes the parameters of the pointer. The auxiliary classification loss for θ is

$$\mathcal{L}_{\text{cls}}^{\theta} = \sum_{j=1,2}\mathbb{E}_{\boldsymbol{x}_j\sim\mathcal{X}_j}[-\log p_\theta(s_j|\boldsymbol{x}_j)] \quad (6)$$

The underlying assumption is that positions with larger attention weights for classification are more likely to be critical to style transfer.

4.4.3 Modeling Content Preservation

Self-Supervised Reconstruction Loss To improve content preservation, we propose a reconstruction loss that guides the operator modules with self-supervision. Suppose the word w at the ith position is deleted or replaced by operator M, we identify the reconstruction operator M' and reconstruction position i' in Table 2. Then M' is updated with MLE, by operating on position i' in $\hat{\boldsymbol{x}}_2$

with w as gold output. For those with two (M', i') pairs, we uniformly sample one for training. Formally, the reconstruction loss is defined as

$$\mathcal{L}_{\text{rec}}^{\phi'} = -\log M'(w|\hat{\boldsymbol{x}}_2, i') \quad (7)$$

Reconstruction Reward One-to-one transfer (e.g., {*delicious↔bland, caring↔unconcerned*}) is usually preferable to many-to-one transfer (e.g., {*delicious→bad, caring→bad*}). Thus, we introduce a reconstruction reward for Rep_{ϕ_3} to encourage one-to-one transfer, i.e.,

$$R_{\text{rec}} = -\lambda_{\text{rec}}\mathcal{L}_{\text{rec}}^{\phi_3'} \quad (8)$$

where $\mathcal{L}_{\text{rec}}^{\phi_3'}$ is the reconstruction loss in Eq. 7.

4.4.4 Training with Single-Option Trajectory

Instead of executing multi-option trajectories, we only allow the high-level agent to execute a single option per episode during training, and leave the multi-option scenario to the inference algorithm (§4.5). We have two motivations for executing single-option trajectories: 1) executing multi-option trajectories is less tractable and stable, especially in the case of style transfer which is sensitive to nuances in the sentence; 2) self-supervised reconstruction is ambiguous in a multi-option trajectory, i.e., the gold trajectory for reconstruction is not deterministic.

High-Level Policy Gradients Since the language model reward is more *local* and increases the variance of estimated gradients, we only use the classification confidence reward for the high-level policy. The policy gradient is

$$\nabla_\theta J(\theta) = \mathbb{E}_i[R_{\text{conf}} \cdot \nabla_\theta \log \mu_\theta(i|\boldsymbol{x}_1)] \quad (9)$$

where gradients are detached from R_{conf}.

Low-Level Policy Gradients All the extrinsic and intrinsic rewards are used for low-level policy learning. Specifically, the rewards for $\phi_{1,2,3}$ are

$$\begin{aligned} R_{1,2} &= R_{\text{lm}} + R_{\text{conf}} \\ R_3 &= R_{\text{lm}} + R_{\text{conf}} + R_{\text{rec}} \end{aligned} \quad (10)$$

For $\phi = \phi_1, \phi_2, \phi_3$, the policy gradient is

$$\nabla_\phi J(\phi) = \mathbb{E}_{\hat{w}}[R \cdot \nabla_\phi \log M_\phi(\hat{w}|\boldsymbol{x}_1, i)] \quad (11)$$

Overall Objectives The overall objectives for θ are the classification loss in Eq. 6 and the policy gradient in Eq. 9. The overall objectives for $\phi_{1,2,3}$ are the reconstruction loss in Eq. 7 and the policy gradients in Eq. 11.

Algorithm 2 Point-Then-Operate Inference

1: **Input:** Input sentence $\boldsymbol{x}_1$, additional classifier p_{add}
2: Initialize $\hat{\boldsymbol{x}}_2 \leftarrow \boldsymbol{x}_1$, $\hat{\boldsymbol{x}}_2^m \leftarrow \boldsymbol{x}_1$, $j \leftarrow 1$
3: **while** $p_{\text{add}}(s_1|\hat{\boldsymbol{x}}_2^m) > p_{\text{stop}}$ and $j \leq j_{\max}$ **do**
4:     Mask the options in $\mu_\theta(i|\hat{\boldsymbol{x}}_2)$ ▷ §4.5.1
5:     Select i that maximizes the masked $\mu_\theta(i|\hat{\boldsymbol{x}}_2)$
6:     Select the *best M* from Table 1   ▷ §4.5.3
7:     Update $\hat{\boldsymbol{x}}_2 \leftarrow$ Transfer$(\hat{\boldsymbol{x}}_2, M, i)$ ▷ §4.3
8:     Update $\hat{\boldsymbol{x}}_2^m$ ▷ §4.5.2
9:    $j \leftarrow j + 1$
10: **end while**
11: The output is $\boldsymbol{x}_{1 \rightarrow 2} \leftarrow \hat{\boldsymbol{x}}_2$
12: **return** $\boldsymbol{x}_{1 \rightarrow 2}$

4.5 Inference

The main problems in applying single-step trained modules to the multi-step scenario are 1) previous steps of operations may influence later steps, and 2) we need to dynamically decide when the trajectory should terminate. We leverage a mask mechanism to address these problems. The basic idea is that given an input sentence, the high-level agent iteratively proposes operation positions for the low-level agent to operate around. In each iteration, the high-level agent sees the whole sentence but with some options (i.e., positions) masked in its policy. The trajectory termination condition is modeled by an additional pre-trained classifier. The algorithm for style transfer from s_1 to s_2 is detailed in Algorithm 2.

4.5.1 Masked Options

To tackle the first problem, we mask the options (i.e., positions) in the high-level policy which appear in the contexts in which any words are inserted, replaced, or skipped (but not for deleted words). Note that we only mask the options in the policy but do *not* mask the words in the sentence (i.e., both agents still receive the complete sentence), since we cannot bias the state representations (§4.2 and §4.3) with masked tokens. We set the window size as 1 (i.e., three words are masked in each step). We find the use of window size necessary, since in many cases, e.g., negation and emphasis, the window size of 1 is capable of covering a complete semantic unit.

4.5.2 Termination Condition

A simple solution to the second problem is to terminate the trajectory if the operated sentence is confidently classified as the target style. The problem with this simple solution is that the highly stylized part may result in too early termination. For example, *Otherwise a terrible experience and*

we will go again may be classified as negative with high confidence. Thus, we propose to mask words in the operated sentence for the termination condition. The masking strategy is the same as §4.5.1 and masked words are replaced by ⟨unk⟩. To tackle the excessive number of ⟨unk⟩, we train an additional classifier as defined in §4.4.2, but trained on sentences with words randomly replaced as ⟨unk⟩.

4.5.3 Inference Policy for Operator Selection

As discussed in §4.3, we adopt a heuristic inference policy for operator selection. Specifically, we enumerate each operator and select the operated sentence $\hat{\boldsymbol{x}}_2$ which maximizes the criterion:

$$c(\hat{\boldsymbol{x}}_2) = \text{LM}_2(\hat{\boldsymbol{x}}_2) \cdot p(s_2|\hat{\boldsymbol{x}}_2)^\eta \qquad (12)$$

where $\text{LM}_2(\hat{\boldsymbol{x}}_2)$ denotes the probability of $\hat{\boldsymbol{x}}_2$ computed by the language model LM_2, $p(s_j|\cdot)$ is the classifier defined in §4.4.2, and η is a balancing hyper-parameter.

5 Experiments

5.1 Datasets

We conduct experiments on two commonly used datasets for unsupervised text style transfer, i.e., Yelp and Amazon, following the split of datasets in Li et al. (2018). Dataset statistics are shown in Table 3. For each dataset, Li et al. (2018) provided a gold output for each entry in the test set written by crowd-workers on Amazon Mechanical Turk. Since gold outputs are *not* written for development sets, we tune the hyper-parameters on the development sets based on our intuition of English.

Yelp The Yelp dataset consists of business reviews and their labeled sentiments (from 1 to 5) from Yelp. Those labeled greater than 3 are considered as positive samples and those labeled smaller than 3 are negative samples.

Amazon The Amazon dataset consists of product reviews and labeled sentiments from Amazon (He and McAuley, 2016). Positive and negative samples are defined in the same way as Yelp.

We observe that the Amazon dataset contains many neutral or wrongly labeled sentences, which greatly harms our HRL-based sequence operation method. Thus, on the Amazon dataset, we adopt a *cross-domain* setting, i.e., we train the modules

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

Table 3: Dataset statistics.

on the Yelp training set using the Amazon vocabulary and test the method on Amazon test set. Experimental results show the effectiveness of our method under this cross-domain setting.

5.2 Evaluation Metrics

Automatic Evaluation Following previous work (Shen et al., 2017; Xu et al., 2018), we pre-train a style classifier TextCNN (Kim, 2014) on each dataset and measure the style polarity of system outputs based on the classification accuracy. Also, based on the human references provided by Li et al. (2018), we adopt a case-insensitive BLEU metric, which is computed using the Moses `multi-bleu.perl` script.

Human Evaluation Following previous work (Shen et al., 2017; Xu et al., 2018), we also conduct human evaluations. For each input sentence and corresponding output, each participant is asked to score from 1 to 5 for fluency, content preservation, and style polarity. If a transfer gets scores of 4 or 5 on all three aspects, it is considered as a successful transfer. We count the success rate over the test set for each system, which is denoted as *Suc* in Table 5.

5.3 Baselines

We make a comprehensive comparison with state-of-the-art style transfer methods. **CrossAligned** (Shen et al., 2017) aligns decoder hidden states adversarially. **MultiDecoder** (Fu et al., 2018) adopts multiple decoders for different styles. **StyleEmbedding** (Fu et al., 2018) adopts a single decoder conditioned on learned style embeddings. **TemplateBased** (Li et al., 2018) retrieves and replaces stylized words. **DeleteOnly** (Li et al., 2018) only deletes the stylized words in the input sentence. **Del-Ret-Gen** (Li et al., 2018) is the same as TemplateBased except that an RNN is adopted to generate the output. **BackTranslate** (Prabhumoye et al., 2018) stylizes the back-translated input. **UnpairedRL** (Xu et al., 2018) deletes stylized words and generates with a denoising AE. **UnsuperMT**

	Yelp		Amazon	
	Acc	BLEU	Acc	BLEU
CrossAligned	74.7	9.06	75.1	1.90
MultiDecoder	50.6	14.54	69.9	9.07
StyleEmbedding	8.4	21.06	38.2	15.07
TemplateBased	81.2	22.57	64.3	34.79
DeleteOnly	86.0	14.64	47.0	33.00
Del-Ret-Gen	88.6	15.96	51.0	30.09
BackTranslate	94.6	2.46	**76.7**	1.04
UnpairedRL	57.5	18.81	56.3	15.93
UnsuperMT	**97.8**	22.75	72.4	33.95
Human	74.7	-	43.2	-
Point-Then-Operate	91.5	**29.86**	40.2	**41.86**

Table 4: Automatic evaluation results for classification accuracy and BLEU with human reference. *Human* denotes human references. Note that Acc for human references are relatively low; thus, we do *not* consider it as a valid metric for comparison.

(Zhang et al., 2018b) produces pseudo-aligned data and iteratively learns two NMT models.

The outputs of the first six baselines are made public by Li et al. (2018). The outputs of Back-Translate and UnpairedRL are obtained by running the publicly available codes. We get the outputs of UnsuperMT from the authors of Zhang et al. (2018b).

5.4 Evaluation Results

Table 4 shows the results of automatic evaluation. It should be noted that the classification accuracy for human reference is relatively low (74.7% on Yelp and 43.2% on Amazon); thus, we do *not* consider it as a valid metric for comparison. For BLEU score, our method outperforms recent systems by a large margin, which shows that our outputs have higher overlap with reference sentences provided by humans.

To lighten the burden on human participants, we compare our proposed method to only four of the previous methods, selected based on their performance in automatic evaluation. Given the observation discussed in §5.1, we remove the wrongly labeled test samples for human evaluation. Table 5 shows the results of human evaluation. Our proposed method achieves the highest fluency and content preservation on Yelp and performs the best on all human evaluation metrics on Amazon.

5.5 Controllable Trade-Off

Figure 3 shows how classification accuracy and BLEU change when we manually set p_{stop}. When

	Yelp				Amazon			
	Fluency	Content	Style	Suc	Fluency	Content	Style	Suc
TemplateBased	3.47	3.76	3.25	68.0 %	3.46	4.08	2.15	9.0 %
Del-Ret-Gen	3.82	3.73	3.52	70.3 %	4.02	4.31	2.69	21.0 %
UnpairedRL	3.54	3.59	2.90	53.8 %	2.58	2.55	2.44	4.5 %
UnsuperMT	4.26	4.24	**4.03**	**82.5 %**	4.24	4.13	3.05	35.5 %
Point-Then-Operate	**4.39**	**4.56**	3.78	81.5 %	**4.28**	**4.47**	**3.31**	**47.0 %**

Table 5: Human evaluation results. Methods are selected based on automatic evaluation. *Style*: style polarity; *Content*: content preservation; *Fluency*: fluency; *Suc*: the proportion of successful transfer (refer to §5.2)
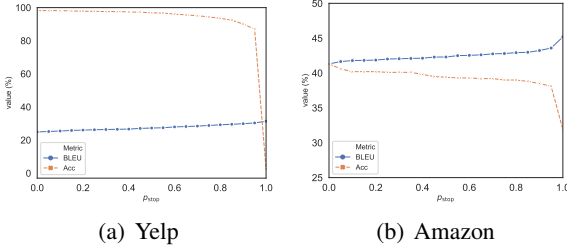

(a) Yelp   (b) Amazon

Figure 3: The controllable trade-off between content preservation and style polarity. The x-axis is p_{stop} (defined in Algorithm 2). The y-axis is the value of different automatic metrics, i.e., BLEU (the blue lines) and classification accuracy (the orange lines).

	Yelp		Amazon	
	Acc	BLEU	Acc	BLEU
InsertOnly	68.6	23.93	48.2	36.77
ReplaceOnly	**93.8**	26.41	**47.8**	37.39
DeleteOnly	37.6	25.70	25.0	41.68
w/o R_{rec} and L_{rec}	39.1	27.80	46.3	40.52
Human	74.7	-	43.2	-
Full	91.5	**29.86**	40.2	**41.86**

Table 6: Ablation Studies.

p_{stop} is larger, classification accuracy drops and BLEU increases. Based on our observation of human references, we find that humans usually make minimal changes to the input sentence; thus, BLEU computed with human references can be viewed as an indicator of content preservation. From this perspective, Figure 3 shows that if we stop earlier, i.e., when the current style is closer to the source style, more content will be preserved and more weakly stylized words may be kept. Thus, controllable trade-off is achieved by manually setting p_{stop}.

5.6 Ablation Studies

We conduct several ablation studies to show the effect of different components in our method:

Ablations of Operators To show that incorporating various operators is essential, we evaluate the performance of the following ablations: InsertOnly, ReplaceOnly, and DeleteOnly, in which operator choices are restricted to subsets of Table 1.

Ablation of Reconstruction Reward and Reconstruction Loss To show the effectiveness of our reconstruction-based objectives, we remove the reconstruction reward and the reconstruction loss as an ablation.

Table 6 shows the ablation results. It shows that BLEU drops if operators are restricted to a fixed set, showing the necessity of cooperating operator modules. It also shows that BLEU drops if we remove the reconstruction loss and the reconstruction reward, indicating the generated words overlap less with human references in this ablation case. As discussed in §5.4, we ignore Acc since it is low on human references.

5.7 Qualitative Study

Figure 1 is an example of our method applied to a test sample. The transfer starts from more stylized parts and ends at less stylized parts, while keeping neutral parts intact. It also shows that our method learns *lexical substitution* and *negation* in an unsupervised way. Table 7 displays some comparisons of different systems. It shows that our proposed method is better at performing local changes to reverse the style of the input sentence while preserving most style-independent parts.

6 Discussions

We study the system outputs and observe two cases that our method cannot properly handle:

Neutral Input The *reconstruction* nature of our method prefers stylized input to neutral input. We observe that it fails to convert some neutral inputs, e.g., *I bought this toy for my daughter about*

Original (Yelp, negative)	staffed primarily by teenagers that do n't understand customer service .
TemplateBased	staffed primarily by teenagers that huge portions and customer service are pretty good .
Del-Ret-Gen	staffed , the best and sterile by flies , how fantastic customer service .
UnpairedRL	staffed established each tech feel when great customer service professional .
UnsuperMT	staffed distance that love customer service .
Point-Then-Operate	staffed by great teenagers that do delightfully understand customer service .

Original (Yelp, positive)	i will be going back and enjoying this great place !
TemplateBased	i will be going back and enjoying this i did not @unk
Del-Ret-Gen	i will be going back and will not be returning into this
UnpairedRL	i will be going back and enjoying this great place .
UnsuperMT	i wo n't be going back and sitting this @num .
Point-Then-Operate	i will not be going back and avoid this horrible place !

Original (Amazon, negative)	i could barely get through it they taste so nasty .
TemplateBased	beautifully through it they taste so nasty .
Del-Ret-Gen	i have used it through and it is very sharp and it was very nasty .
UnpairedRL	i could barely get through it they taste so nasty .
UnsuperMT	i can perfect get through it they taste so delicious .
Point-Then-Operate	i could get through it they taste so good .

Original (Amazon, positive)	i also prefered the blade weight and thickness of the wustof .
TemplateBased	i also prefered the blade weight and thickness of the wustof toe .
Del-Ret-Gen	i also prefered the blade and was very disappointed in the weight and thickness of the wustof .
UnpairedRL	i also sampled the comfortable base and follow of the uk .
UnsuperMT	i also encounter the blade weight and width of the guitar .
Point-Then-Operate	i only prefered the weight and thickness of the wustof .

Table 7: Sampled system outputs. The dataset and the original style for each input sentence are parenthesized. We mark improperly generated or preserved words in blue, and mark words that show target style and are grammatical in the context in red. Best viewed in color.

@num months ago., which shows that the high-level policy is not well learned for some neutral sentences.

Adjacent Stylized Words We introduce a window size of 1 in §4.5.1 to deal with most semantic units. However, we observe in some cases two adjacent stylized words occur, e.g., *poor watery food*. If the first step is to replace one of them, then the other will be masked in later iterations, leading to incomplete transfer; if the first step is deletion, our method performs well, since we do not mask the context of deletion as stated in §4.5.1. Notably, phrases like *completely horrible* is not one of these cases, since *completely* itself is not stylized.

Experiments in this work show the effectiveness of our proposed method for positive-negative text style transfer. Given its sequence operation nature, we see potentials of the method for other types of transfers that require *local* changes, e.g., polite-impolite and written-spoken, while further empirical verification is needed.

7 Conclusions

We identify three challenges of existing seq2seq methods for unsupervised text style transfer and propose Point-Then-Operate (PTO), a sequence operation-based method within the hierarchical reinforcement learning (HRL) framework consisting of a hierarchy of agents for pointing and operating respectively. We show that the key aspects of text style transfer, i.e., fluency, style polarity, and content preservation, can be modeled by comprehensive training objectives. To make the HRL training more stable, we provide an efficient mask-based inference algorithm that allows for single-option trajectory during training. Experimental results show the effectiveness of our method to address the challenges of existing methods.

Acknowledgments

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nadir Durrani, Helmut Schmid, and Alexander M. Fraser. 2011. A joint sequence translation model with integrated reordering. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 1045–1054.

Nadir Durrani, Helmut Schmid, Alexander M. Fraser, Philipp Koehn, and Hinrich Schütze. 2015. The operation sequence model - combining n-gram-based and phrase-based statistical machine translation. *Computational Linguistics*, 41(2):185–214.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5779–5786.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 663–670.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 507–517.

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, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for text style transfer. *CoRR*, abs/1808.04339.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*, pages 1746–1751.

Tejas D. Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. 2016. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3675–3683.

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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1865–1874.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. *CoRR*, abs/1905.10060.

Santanu Pal, Marcos Zampieri, and Josef van Genabith. 2016. USAAR: An operation sequential model for automatic statistical post-editing. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 759–763.

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 866–876.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6833–6844.

Felix Stahlberg, Danielle Saunders, and Bill Byrne. 2018. An operation sequence model for explainable neural machine translation. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 175–186.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Richard S. Sutton, Doina Precup, and Satinder P. Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1-2):181–211.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2018. A hierarchical framework for relation extraction with reinforcement learning. *CoRR*, abs/1811.03925.

Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. 2018. Video captioning via hierarchical reinforcement learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4213–4222.

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, 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 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 7298–7309.

Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. 2018a. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1103–1108.

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

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251.