Authorship Style Transfer with Policy Optimization

Anonymous ACL submission

Abstract

Authorship style transfer aims to rewrite a given text into a specified target while preserving the original meaning in the source. Existing approaches rely on the availability of a large number of target style exemplars for model training. However, these overlook cases where a limited number of target style examples are available. The development of parameter-efficient transfer learning techniques and policy optimization (PO) approaches suggest lightweight PO is a feasible approach to low-resource style transfer. In this work, we propose a simple two-stage tune-and-optimize technique for low-resource textual style transfer. We apply our technique to authorship transfer as well as a larger-data native language style task and in both cases find it outperforms state-of-the-art baseline models.[1]

1 Introduction

Given a text authored by an arbitrary source author, can we make it look like it is written by an arbitrary target author without changing its meaning? This is the domain of authorship style transfer. In the era of large language models (LLMs), the promise of authorship style transfer can turn any LLM into our own personalized model by transferring the outputs into our own style, and also prevent our text from being identified by authorship identification models through transferring our texts into the style of another author. This task is first studied as a classic text style transfer task that requires a large number of texts in the target style to develop the transfer model, which limits its application to only famous authors like Shakespeare (Xu et al., 2012; Krishna et al., 2020).

Recently, Patel et al. (2023) propose a more general and practical task, low-resource authorship style transfer which can apply to non-famous authors who only have a limited number of texts. To solve this new task, they develop an LLM-based approach, STYLL which transfers a text by prompting LLMs with several texts written by the target author. Though intended to be a simple baseline, STYLL proves remarkably adept at style alteration. Deeper investigation by Patel et al. (2023) shows that while the alteration does manage to remove, or move away from the original author's style, it is rather unable to adopt, or move toward, the intended target author.

STYLL is an entirely in-context learning (ICL) method; it uses no model training or modification. This is justified by Patel et al. (2023) as, due to small amounts of style-relevant training data, methods that use supervised fine-tuning (SFT) such as STRAP (Krishna et al., 2020) do not outperform ICL. In this work we instead consider whether this limited data can be repurposed, specifically as training for a *style critic model*, thereby enabling a policy optimization (PO) approach to directly encourage text generation in the desired style. Rather than train the model on pseudo-parallel data with the language modeling loss, we could use policy optimization (PO) approaches to directly optimize the model to maximize the authorship style transfer objective.

In this work, we propose Authorship Style TRAnsfer with Policy OPtimization (ASTRAPOP), a lightweight two-stage PO training framework for authorship style transfer, in which the first supervised fine-tuning stage prepares a reference model for the second stage, and the policy optimization in the second stage further improves the performance of the reference model by directly optimizing it on the authorship style transfer objective. Unlike more complicated RL-based approaches like Hallinan et al. (2023), ASTRAPOP is more computationally efficient and flexible, and works well with a variety of both RL-based and RL-free PO algorithms.

[1] Code, data, and models sufficient for a reproducibility study will be available at `https://anon`.

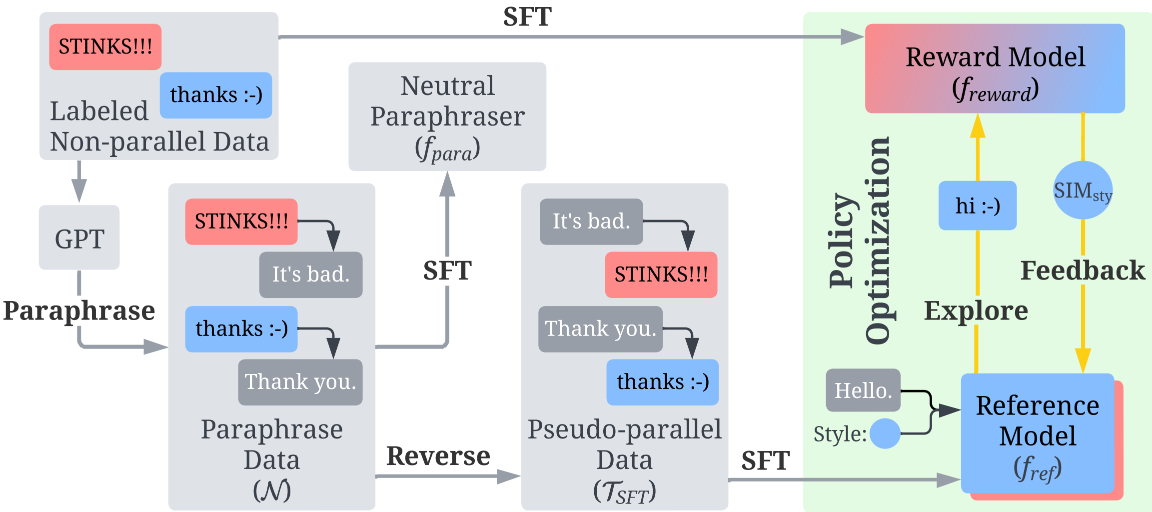

Figure 1: Overview of ASTRAPOP.

We evaluate ASTRAPOP on two authorship style transfer tasks, a low-resource individual authorship style transfer task[2] and a medium-resource community authorship style transfer task. The evaluation results show that ASTRAPOP is more effectively able to leverage few-shot style transfer than ICL or SFT methods alone on the former task and also outperforms the state-of-the-art style transfer model with much less training time on the latter.

2 Methodology

In this section we formalize the authorship style transfer task and introduce ASTRAPOP.

2.1 Task Definition

The goal of authorship style transfer is to modify the style of the input text to make it look like the style of another author. Formally, we have a dataset of texts with authorship style labels $\mathcal{D} = \{(\mathbf{x}_1, s_1), (\mathbf{x}_2, s_2), \cdots, (\mathbf{x}_n, s_n)\}$ where the style label could be either at the individual level or the community level. For convenience, we denote the semantic similarity between two texts $\mathbf{x}_i$ and $\mathbf{x}_j$ as $SIM_{sem}(\mathbf{x}_i, \mathbf{x}_j)$, and the similarity between the style of a text $\mathbf{x}$ and a style s as $SIM_{sty}(\mathbf{x}$,s). Given an input text $\mathbf{x}_s$ with style s and a target style t, an authorship style transfer model rewrites $\mathbf{x}_s$ into a new text $\mathbf{x}_{s\to t}$ that maximizes $SIM_{sem}(\mathbf{x}_{s\to t}, \mathbf{x}_s)$ and $SIM_{sty}(\mathbf{x}_{s\to t}, t)$, and minimizes $SIM_{sty}(\mathbf{x}_{s\to t}, s)$. We refer to maximizing $SIM_{sty}(\mathbf{x}_{s\to t}, t)$ and minimizing $SIM_{sty}(\mathbf{x}_{s\to t}, s)$ as the TOWARD and AWAY objectives, respectively.

[2] with as few as five examples per author.

2.2 Framework Overview

ASTRAPOP contains two main stages: **supervised fine-tuning** and **policy optimization**. The framework overview is shown in Figure 1. In the supervised fine-tuning stage, we train a reward model on labeled non-parallel data and a reference model on parallel in-domain data for policy optimization. Due to a lack of parallel authorship style transfer data, we use the style transfer via paraphrasing (STRAP) strategy described in Krishna et al. (2020) to generate pseudo-parallel data to train the reference model. Then, in the policy optimization stage, we directly optimize the reference model from the SFT stage on the TOWARD and AWAY objectives.

2.3 Supervised Fine-tuning

In this stage, we train three models with supervised fine-tuning: a **neutral paraphraser** f_{para}, a **reference model** f_{ref}, and a **reward model** f_{reward}. f_{para} is used for inference only, while f_{ref} and f_{reward} are used for PO training.

2.3.1 Data Generation

We first generate the pseudo-parallel training data for the neutral paraphraser and the reference model. Following Krishna et al. (2020), we generate neutral-to-target style transfer pairs by paraphrasing the target style texts with a neutral paraphraser. To ensure the quality of the training data, we generate the neutral paraphrases with GPT-3.5-turbo using the same paraphrase prompt as in Patel et al. (2023). Concretely, we generate neutral paraphrases for all texts in the dataset $\mathcal{D}$ to obtain a new set of texts $\mathcal{P} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\}$ where $\mathbf{y}_i \in \mathcal{P}$ is the neutral paraphrase of $\mathbf{x}_i \in \mathcal{D}$.

Then, we can build a neutral paraphrase dataset $\mathcal{N} = \{(\mathbf{x}_1 \to \mathbf{y}_1), \cdots, (\mathbf{x}_n \to \mathbf{y}_n)\}$ and a neutral-to-target style transfer dataset $\mathcal{T}_{SFT} = \{(\mathbf{y}_1 \to \mathbf{x}_1, s_1), \cdots, (\mathbf{y}_n \to \mathbf{x}_n, s_n)\}$.

2.3.2 Paraphraser & Reference Model

We fine-tune off-the-shelf language models on the two generated supervised datasets to build the neutral paraphraser[3] and the reference model. Specifically, for the neutral paraphraser, we simply fine-tune the model on dataset $\mathcal{N}$ to maximize

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{|\mathbf{y}|} p(y^i|\mathbf{x}, \mathbf{y}^{<i})$$

where $\mathbf{y}^{<i}$ represents tokens preceding token y^i in $\mathbf{y}$. Similarly, for the reference model, we fine-tune the model on dataset $\mathcal{T}_{SFT}$ to maximize

$$p(\mathbf{x}|\mathbf{y}, s) = \prod_{i=1}^{|\mathbf{x}|} p(x^i|\mathbf{y}, \mathbf{x}^{<i}, s)$$

Note that the probability in the training objective for authorship style transfer is additionally conditioned on the target style s. Following Wolf et al. (2019), we implement all seq2seq models using decoder-only transformers.

2.3.3 Reward Model

Besides the reference model, PO training also requires a reward model to measure the style similarity SIM_{sty}. We train a style model on dataset $\mathcal{D}$ to serve this purpose. The details on how to train the style model and calculate SIM_{sty} from the style model output are shown in § 3.1.3 and § 3.2.3.

2.4 Policy Optimization

We further train the reference model f_{ref} using policy optimization and the reward model f_{reward} to obtain the final **PO transfer model** f_{PO}.

2.4.1 Reward Function

Policy optimization aims to optimize a model to maximize an arbitrary reward function. Therefore, we design a TOWARD reward T and an AWAY reward A to mirror the TOWARD and AWAY objectives so that maximizing the rewards is equivalent to directly optimizing for the two objectives. Specifically, we define T and A as

$$T(\mathbf{x}_{s\to t}, t) = SIM_{sty}(\mathbf{x}_{s\to t}, t)$$
$$A(\mathbf{x}_{s\to t}, s) = 1 - SIM_{sty}(\mathbf{x}_{s\to t}, s)$$

where SIM_{sty} is the style similarity calculated by the reward model f_{reward}. However, our preliminary experiments show that training with only these two rewards sometimes results in models that only generate empty or very short outputs. To mitigate this, we add the simple and quick-to-calculate length penalty term from Wieting et al. (2019) to the reward, which is defined as

$$LP(\mathbf{x}_{s\to t}, \mathbf{x}_s) = e^{1-\frac{min(|\mathbf{x}_{s\to t}|,|\mathbf{x}_s|)}{max(|\mathbf{x}_{s\to t}|,|\mathbf{x}_s|)}}$$

The total reward is then

$$R = T + A - (LP^{\alpha} - 1)$$

where α is a temperature hyperparameter.

2.4.2 PO Training Data

During SFT, we train the transfer model to transfer the neutral paraphrase back into the original style before paraphrasing, which means the source style before paraphrasing and the target style are the same. For PO, we want to further optimize the model to move the style of the transferred text away from the source style and toward the target style. In this case, we have to make sure the source style and the target style are different, otherwise the two objectives will be contradictory to each other. Therefore, during PO training, we shift the target style by one element, which yields a new dataset $\mathcal{T}_{PO} = \{(\mathbf{y}_1, s_2), (\mathbf{y}_2, s_3), \cdots, (\mathbf{y}_n, s_1)\}$. Note that we also drop the gold outputs $\mathbf{x}_i$ from $\mathcal{T}_{SFT}$ since PO trains the model on generated outputs and the rewards.

2.4.3 PO Algorithms

We consider three PO algorithms, Proximal Policy Optimization (PPO) (Schulman et al., 2017), Direct Preference Optimization (DPO) (Rafailov et al., 2023), and Contrastive Preference Optimization (CPO) (Xu et al., 2024). PPO is an online reinforcement learning (RL) algorithm, while DPO and CPO are recent RL-free alternatives to PPO and have been shown to be more stable, computationally efficient, and effective on various NLP tasks (Rafailov et al., 2023; Xu et al., 2024).

2.5 Inference

For inference, given a text $\mathbf{x}_s$ in style s, we transfer it into the target style t by

$$\mathbf{x}_{s\to t} = f_{PO}(f_{para}(\mathbf{x}_s), t)$$

where $\mathbf{x}_{s\to t}$ is the transferred text.

[3] Please see § B.5 for why we choose to train a local model instead of using GPT as the paraphraser.

3 Experiments

We evaluate our approach on authorship style transfer tasks at two data resource levels, low-resource individual authorship style transfer and medium-resource community authorship style transfer. In this section, we first discuss the task specific details for each task and then introduce the baseline models and implementation details.

3.1 Individual Authorship Style Transfer

In this section, we discuss the experiments on the individual authorship style transfer task. Specifically, we adopt the same low-resource authorship style transfer task as in Patel et al. (2023), which aims to transfer a text from an arbitrary author into the style of another arbitrary author, for which only a limited number of text exemplars exist.

3.1.1 Dataset

We use the Million User Dataset (MUD) from Khan et al. (2021) to train and evaluate our model. MUD is a dataset extracted from the Pushshift Reddit dataset (Baumgartner et al., 2020) which contains user posts on Reddit with author labels. In this task, the author label is used as the style label s in the dataset $\mathcal{D}$. For training, we randomly sample 12,000 authors from the training split of MUD and use 10,000, 1,000, and 1,000 authors for training, validation, and test, respectively. We randomly sample two texts for each author in the training split and one text for each author in the validation and the test splits. For evaluation, we randomly sample 100 source authors and 100 target authors from the "test_query" split of MUD; each author has 16 texts.

3.1.2 Transfer Model Formulation

In this task, we use a single model conditioned on few-shot target author exemplars for all authors, since Patel et al. (2023) shows that in this extremely low-resource setting, exemplar-based approach works better than one model per author.

3.1.3 Reward Model

For policy optimization, we need a task specific reward model to calculate the style similarity score $SIM_{sty}(\mathbf{x}, s)$ in the reward function. For the individual-level transfer task, we use the LUAR model from Rivera-Soto et al. (2021) as the reward model f_{reward} which generates a single-vector authorship representation for an author with several texts from that author.

$$\mathbf{v}_s = LUAR(\{\mathbf{x}_i | s_i = s\})$$

We define the style similarity score as

$$SIM_{sty}(\mathbf{x}, s) = cossim(LUAR(\{\mathbf{x}\}), \mathbf{v}_s))$$

where $cossim$ is the cosine similarity between the two vectors.

3.1.4 Metrics

We adopt a subset of automatic evaluation metrics from Patel et al. (2023) to evaluate the style transfer, the content preservation, and the overall performance of our model. For convenience, in this section, we denote the set of all test texts as $\mathcal{X}$ and the set of all test source-target author pairs as $\mathcal{S}$.

Style Transfer We use the toward[4], away, and confusion scores to measure the style transfer performance. The toward and away scores measure to what extent the authorship style transfer model moves the style of the texts away from the source style and toward the target style in the authorship representation space. Concretely, the **toward** score is defined as

$$\frac{1}{|\mathcal{S}|} \sum_{(s,t)\in\mathcal{S}} \frac{1 - max(sim(\mathbf{v}_{s\to t}, \mathbf{v}_s), sim(\mathbf{v}_t, \mathbf{v}_s))}{1 - sim(\mathbf{v}_t, \mathbf{v}_s)}$$

and the **away** score is defined as

$$\frac{1}{|\mathcal{S}|} \sum_{(s,t)\in\mathcal{S}} \frac{max(sim(\mathbf{v}_{s\to t}, \mathbf{v}_t) - sim(\mathbf{v}_s, \mathbf{v}_t), 0)}{1 - sim(\mathbf{v}_t, \mathbf{v}_s)}$$

where $\mathbf{v}_s$, $\mathbf{v}_t$, and $\mathbf{v}_{s\to t}$ are the LUAR authorship representations for the source author s, target author t, and the transferred texts, respectively, and sim is a vector similarity measure from Cer et al. (2018), which is defined as

$$sim(\mathbf{u}, \mathbf{v}) = 1 - \frac{arccos(\frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|})}{\pi}$$

We also use the confusion score to directly measure what percentage of the transferred texts is closer to the target style than the source style. Formally, the **confusion** score is defined as

$$\frac{1}{|\mathcal{S}|} \sum_{(s,t)\in\mathcal{S}} \mathbf{1}_{sim(\mathbf{v}_{s\to t}, \mathbf{v}_t) > sim(\mathbf{v}_{s\to t}, \mathbf{v}_s)}$$

[4] For internal consistency, we refer to the 'towards score,' described in Patel et al. (2023) as toward score in this work.

Content Preservation To measure content preservation, we use the SBERT[5] (Reimers and Gurevych, 2019) cosine similarity instead of the mutual implication score (MIS) (Babakov et al., 2022) in Patel et al. (2023) since SBERT is the most commonly used semantic embedding model, and MIS is trained on very short texts, but our test set contains much longer texts. The **SBERT** content preservation score is simply

$$\frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}_s \in \mathcal{X}} cossim(\text{SBERT}(\mathbf{x}_{s\to t}), \text{SBERT}(\mathbf{x}_s))$$

where $\mathbf{x}_s$ is the original text, and $\mathbf{x}_{s\to t}$ is the transferred text.

Overall Performance To have a better understanding of the overall performance of the models, we use the same method as in Patel et al. (2023) to aggregate the toward, away, and the SBERT cosine similarity scores to obtain a **joint** score. Specifically,

$$joint = G(G(toward, away), cossim_{\text{SBERT}})$$

where G refers to geometric mean.

3.2 Community Authorship Style Transfer

In the previous section, we investigated the effectiveness of our approach in transferring style across individual authors with an extremely limited number of texts. Different from such low-resource and fine-grained control, in this section, we demonstrate that our approach is equally proficient in transferring authorship style across communities of authors sharing the same attribute. Specifically, we choose the native language ($L1$) of authors whose first language is not English as the attribute we want to control. The objective is to take English text written by an author whose native language is $L1(s)$, say $s = Arabic$, and re-write it as English text in the style of a native $L1(t)$ author, say $t = Chinese$.

3.2.1 Dataset

We use the ETS Corpus of Non-Native Written English[6] to study the $L1$ transfer task. This ETS-TOEFL dataset has essays written by students whose $L1$ varies across 11 languages: Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, and Turkish. The native language $L1$ is used as the style label s in the dataset $\mathcal{D}$ in this task. The data is carefully curated to control for topics: the topics are not correlated with $L1$ and all subjects write about the same set of topics. This removes the possibility for the system to make spurious topic-oriented correlations.[7] This is in contrast to other attribute specific datasets which typically do not control for unintended correlations between the categorical attributes and textual content. The train, validation, and test splits have 900, 100, and 100 documents, respectively.

Our preliminary experiment shows that LLMs like GPT-3.5-turbo tend to drop information when paraphrasing long documents like the documents in the ETS dataset, so in this task, we process the documents at segment level. Concretely, we split all documents into segments with up to 128 tokens, and the style labels of the segments are the same as the original documents. We then sample 2,000 segments and 200 segments for each native language for training and validation, respectively, which results in a training set with 22,000 segments and a validation set with 2,200 segments. For evaluation, we transfer all documents in the test set into all native language styles except the source style to obtain transferred texts for all 110 native language pairs. This transfer is also done at segment level. We segment all documents in the test set before transfer and regroup them back to documents afterward.

3.2.2 Transfer Model Formulation

In this task, we use a single model for each style since we have a fair amount of data for each style, and our preliminary experiments show that one model per author works better than the few-shot exemplar-based approach or control code-based (Keskar et al., 2019) approach on this task. We train the models using the STRAP approach, so the SFT model is the same as the STRAP model for this task.

3.2.3 Reward Model

In the community-level transfer task, we use a classifier as the reward model f_{reward} instead of the representation model since our preliminary experiment shows that the classifier works much better than the representation model on native language identification[8]. Specifically, we train a RoBERTa-

[5]We use the best-performing variant of SBERT, all-mpnet-base-v2.

[6]LDC2014T06

[7]For example, if the topics are not controlled, the system could perhaps determine that the author's likely $L1$ is either Hindi or Telugu if the topic centers around the game of Cricket.

[8]Please see § B.6 for detailed comparison.

large (Liu et al., 2019b) classifier with 11 binary classification heads, corresponding to each native language on the ETS training set. Formally, given an input text $\mathbf{x}$ and a topic $(L1)$ s, we denote the classifier output probability as $p_s(\mathbf{x})$ and the classification decision as $C_s(\mathbf{x}) = \mathbf{1}_{p_s(\mathbf{x})>0.5}$. We then define the style similarity score as

$$SIM_{sty}(\mathbf{x}, s) = p_s(\mathbf{x})$$

3.2.4 Metrics

We use the **SBERT** cosine similarity and the **joint** score defined in § 3.1.4 to evaluate the content preservation and the overall performance of our model. However, since the representation model does not work well on the community authorship identification task, we propose three new metrics for style transfer accuracy in direct analogy with the $toward$, $away$, and $confusion$ scores in Patel et al. (2023). For convenience, we denote all test documents written by authors with native language s as D_s in this section.

We use toward and away scores to indicate the percentage increase in how many transferred documents are classified as being written by a target native language author and the percentage decrease in how many transferred documents are classified as being written by a source native language author. Formally, for each pair of source native language s and target native language t, we define the **toward** score[9] as

$$max\left(\frac{\sum_{\mathbf{x}_s \in D_s} C_t(\mathbf{x}_{s\to t}) - \sum_{\mathbf{x}_s \in D_s} C_t(\mathbf{x}_s)}{|D_s| - \sum_{\mathbf{x}_s \in D_s} C_t(\mathbf{x}_s)}, 0\right)$$

and define the **away** score as

$$max\left(\frac{\sum_{\mathbf{x}_s \in D_s} C_s(\mathbf{x}_s) - \sum_{\mathbf{x}_s \in D_s} C_s(\mathbf{x}_{s\to t})}{\sum_{\mathbf{x}_s \in D_s} C_s(\mathbf{x}_s)}, 0\right)$$

where $\mathbf{x}_s$ referes to the original text and $\mathbf{x}_{s\to t}$ refers to the transferred text.

We use the confusion score to measure what percentage of the transferred texts are classified as being written by a target native language author but not a source native language author. Formally, the **confusion** score is defined as

$$\frac{\sum_{\mathbf{x}_s \in D_s} \mathbf{1}_{C_t(\mathbf{x}_{s\to t}) - C_s(\mathbf{x}_{s\to t}) = 1}}{|D_s|}$$

[9]The toward score and the TOWARD objective/reward both measure to what extent the transferred texts reflect the target style, but they are defined differently. The toward score is defined to be more intuitive, while the TOWARD objective/reward is defined to be easier to calculate. Similar for the away score.

3.3 Baseline Models

We compare ASTRAPOP with a popular unsupervised style transfer model, STRAP (Krishna et al., 2020), the SOTA low-resource individual authorship style transfer model, STYLL (Patel et al., 2023), the SOTA high-resource style transfer model STEER (Hallinan et al., 2023), and LLM zero-shot transfer.

STRAP performs text style transfer by paraphrasing the input text twice with a diverse paraphraser followed by an inverse paraphraser trained to rewrite the diverse paraphrase into the target style.

STYLL transfers the input text by prompting LLMs with the target style descriptors and few-shot transfer examples generated from the target style exemplars.

STEER trains the style transfer model with expert-guided data generation (Liu et al., 2021a) and a two-phase online-then-offline RL training using QUARK (Lu et al., 2022).

Zero-shot Transfer simply prompts LLMs with the input text and the target style to transfer.

3.4 Implementation Details

We implement our training framework and models with Huggingface's Transformers, PEFT, and TRL codebases. Except GPT-3.5-turbo, LLaMA-2-7B-chat, and BLOOM-7B used for the zero-shot and STYLL baselines, we only use LLaMA-2-7B for all other approaches. For computational efficiency, all learning-based models are trained with the Low-Rank Adaptation (LoRA) (Hu et al., 2022) technique.[10] Please see § B.2 and § B.3 for more details on the hyperparameters and the model input formats, respectively.

4 Results

In this section, we discuss and analyze the experimental results for both tasks. Due to the limited time and computational resources, we conduct all experiments in a single run and perform statistical significance tests on the results.[11] For conciseness, we only show the automatic evaluation results with LLaMA-7B for all approaches and the results with the best reward combinations for ASTRAPOP. Please see § A.1 for the full automatic evaluation results and the ablation study. We also conduct human evaluation and a case study for the community

[10]Since STEER uses the QUARK algorithm which adds new tokens to the model, we also train the token embedding layer for STEER in the RL phase.

[11]Please see § B.1 for details.

Method	Toward	Away	SBERT	**Joint**	Confusion
STRAP	0.088‡	0.793†	0.650‡	0.414‡	0.30‡
STYLL	0.159	**0.845**	0.529‡	0.440‡	**0.59**
SFT	0.137‡	0.707‡	**0.754**	0.484‡	0.32‡
ASTRAPOP-PPO	0.147‡	0.773‡	0.729†	0.495†	0.48†
ASTRAPOP-DPO	0.164	0.748‡	0.733†	**0.507**	0.44†
ASTRAPOP-CPO	**0.165**	0.752‡	0.726†	0.505†	0.46†

Table 1: The automatic evaluation results on the individual authorship style transfer task with LLaMA-7B based models and ASTRAPOP models trained with the reward function $R = T + A - (LP^{\alpha} - 1)$. The best and the second best scores for each metric are shown in **bold** and underline, respectively. "†" and "‡" indicate a significant difference between the model and the best model or the top two models, respectively, determined by t-test with $\alpha = 0.05$.

authorship transfer experiments. Please see § A.4 and § A.5 for details.

4.1 Individual Authorship Style Transfer

The automatic evaluation results on the individual authorship style transfer task are shown in Table 1. We only show the ASTRAPOP models trained with the full reward function $R = T + A - (LP^{\alpha} - 1)$ since this reward function yields the best Joint score for all three PO algorithms on this task. The joint score indicates that the overall performance of the three ASTRAPOP models are superior to all baseline models, and the two models trained with the RL-free PO algorithms (i.e. DPO and CPO) perform similarly to each other and both perform better than the RL-based PO algorithm (i.e. PPO).

Looking at the toward, away, and SBERT scores separately, we find that all PO algorithms can effectively improve the toward and the away scores, but at the cost of harming the SBERT score, since our reward function does not take semantic similarity into account, for efficiency and stability. Even so, the three ASTRAPOP models still have decent SBERT scores that are higher than all baseline models except the SFT model since the KL-divergence penalty helps the models preserve the capability to keep the semantic meaning of the input texts. One may notice that STYLL has much better away and confusion scores than all other models, but this is because the model sometimes copies some irrelevant content from the target exemplars which changes the meaning of the transferred texts, and this also explains why the SBERT score for STYLL is much lower than other models.

4.2 Community Authorship Style Transfer

The automatic evaluation results on the community authorship style transfer task are shown in Table 2. For this task, the best ASTRAPOP models are trained with the reward function without the away reward $R = T - (LP^{\alpha} - 1)$. The joint score indicates that DPO- and CPO-ASTRAPOP have the best overall performance. They also have the best toward and confusion scores. PPO can also slightly improve the performance of the SFT model, but the improvement is much less than DPO and CPO. Similar to the previous task, PO training harms the SBERT score, but the magnitude of the loss is very small, and the result SBERT scores are still higher than all baseline models except STRAP/SFT.

5 Related Work

Text Style Transfer Since parallel data is very rare for text style transfer, only a few works solve this task in a supervised manner (Zhu et al., 2010; Rao and Tetreault, 2018; Kim et al., 2022; Raheja et al., 2023). Constrained by the datasets, these works only focus on some specific sub-tasks such as text simplification and formality transfer. Therefore, to build more general style transfer models, recent works develop unsupervised methods that do not rely on parallel data. These works mainly fall in five categories, content-style representation disentanglement (Liu et al., 2019a; Jin et al., 2020), style-related phrase replacement (Madaan et al., 2020; Malmi et al., 2020; Reid and Zhong, 2021), reinforcement learning on direct objective (Gong et al., 2019; Liu et al., 2021b; Deng et al., 2022; Hallinan et al., 2023), pseudo-parallel data generation (Krishna et al., 2020; Riley et al., 2021), and LLM prompting (Reif et al., 2022; Suzgun et al.,

Method	Toward	Away	SBERT	**Joint**	Confusion
Zero-shot	0.022‡	0.880†	0.738‡	0.321‡	0.033‡
STYLL	0.210‡	0.832‡	0.854‡	0.598‡	0.227‡
STRAP / SFT	0.286‡	0.785‡	**0.917**	0.659‡	0.300‡
STEER	0.334‡	**0.926**	0.879‡	0.699‡	0.348‡
ASTRAPOP-PPO	0.299‡	0.800‡	0.905‡	0.665‡	0.313‡
ASTRAPOP-DPO	<u>0.490</u>†	0.843‡	<u>0.915</u>†	<u>0.767</u>†	<u>0.499</u>†
ASTRAPOP-CPO	**0.655**	<u>0.887</u>†	0.897‡	**0.827**	**0.662**

Table 2: The automatic evaluation results on the community authorship style transfer task with LLaMA-7B based models and ASTRAPOP models trained with the reward function $R = T - (LP^{\alpha} - 1)$. The scores are averages over all pairs of native languages. The best and the second best scores for each metric are shown in **bold** and <u>underline</u>, respectively. "†" and "‡" indicate a significant difference between the model and the best model or the top two models, respectively, determined by t-test with $\alpha = 0.05$.

2022; Patel et al., 2023).

The state-of-the-art authorship style transfer model, STYLL (Patel et al., 2023) transfers the input texts by prompting an LLM with the target style descriptors and few-shot pseudo-parallel transfer pairs generated by the same LLM, which combines the strength of pseudo-parallel generation and LLM prompting. Even so, as a prompting-based method, STYLL can be potentially enhanced by RL since RL has already been shown to be effective in improving the performance of prompting-based style transfer models (Deng et al., 2022), and the state-of-the-art general style transfer model, STEER (Hallinan et al., 2023) is also trained with RL. However, RL algorithms are shown to be unstable and hard to tune compared to the recently developed RL-free policy optimization algorithms such as DPO (Rafailov et al., 2023) and CPO (Xu et al., 2024).

Therefore, in this work, we choose the solve the authorship style transfer task with a PO-based training framework. Similar to STEER, we first generate pseudo-parallel data from the labeled non-parallel data and then train the model on the generated data, but our framework differs from STEER in three major ways: (1) we use a much simpler data generation strategy which only needs one paraphrase model and generates once for each instance in the non-parallel data, but STEER requires two extra models for each style as well as heavy over-generation and filtering; (2) we only perform a single stage PO training instead of the two-stage offline-then-online RL training in STEER, and our reward function requires only one reward model instead of the three reward model in STEER; (3) we also use more stable and efficient RL-free PO algorithms instead of just the RL-based algorithm in STEER.

Policy Optimization Policy optimization has been widely used in NLP to train language models on task specific objectives such as text simplification (Laban et al., 2021), question answering (Liu et al., 2022), and machine translation (Xu et al., 2024). Most early works in this area focus on RL-based algorithms such as REINFORCE (Williams, 1992) and PPO (Schulman et al., 2017), but these algorithms are often considered unstable and inefficient. Recently, many RL-free algorithms have been developed to improve the stability and the efficiency. These works mostly focus on aligning LLMs with human preference (Rafailov et al., 2023; Song et al., 2023), but there are also some that apply to other tasks such as machine translation (Xu et al., 2024). In this work, we use PO algorithms to train the models directly on the authorship style transfer objectives. To our best knowledge, this is the first work applying RL-free PO algorithms on text style transfer.

6 Conclusion

In this work, we propose a PO-based training framework for authorship style transfer, which combines the strength of supervised fine-tuning on the pseudo-parallel data and policy optimization on the transfer objective. Extensive experiments confirm the effectiveness of our model on both low-resource and high-resource authorship style transfer tasks and show that our model outperforms the SOTA models in both authorship style transfer and general style transfer.

Limitations

Although our approach shows strong performance on authorship style transfer, the performance on low-resource transfer is still much weaker than the performance on high-resource transfer. There are two possible reasons. First, we use small-scale datasets for both tasks due to the limited computational resources. It is sufficient to model the coarse-grained community authorship styles but may be insufficient for the individual authorship styles. Therefore, if more computational resources are available, future work can investigate whether more training data can help improve the performance of the low-resource authorship transfer models. Second, our authorship information injection strategy may not be optimal. We use a popular exemplar-based approach to inject the authorship information in the low-resource transfer task, but there may be more efficient approaches such as using continuous vectors instead of discrete tokens. This is out of the scope of this work, but future work can explore more efficient information injection strategies for low-resource authorship style transfer.

Moreover, even though the two RL-free PO algorithms, DPO and CPO already show a much better performance than PPO, in this work, we only use them in an offline manner as in the original papers. However, one can naturally enhance DPO and CPO with online data generated by the updated policy during training, which can potentially improve the performance of the models. Therefore, future work can focus on improving the training framework with online DPO or CPO training.

Ethical Considerations

Like other transfer learning LLMs, the quality of our model outputs highly depends on the quality of the underlying LLM and the training data. In this work, we use the original LLaMA-2-7B model instead of the chat version to ensure the flexibility for training, but it also has a higher risk of generating toxic texts. Also, the datasets we use contain unfiltered texts from the online forum Reddit and may also lead to unethical generation. Therefore, for real-world applications, we suggest carefully filtering the training data and also using a post-generation filter to avoid outputting unethical texts. As a PO-based training framework, one can also add some terms to the reward function to encourage the model to generate safe and ethical outputs.

Both datasets we use in this work contain texts with personal identifiable information (PII) and/or unethical words. We do not remove profane texts and texts containing PIIs for human evaluation to maximally preserve the style and meaning of the texts. Our human study protocol has been approved by an institutional review board.

Our model is intended for personal and authorized use such as building personal chatbots or authorship privatization, but we also recognize some potential harmful usage such as maliciously mimicking some individuals without authorization and intentionally generating texts in an offensive style. Therefore, we suggest keeping all personal data locally to prevent malicious mimicking. For text privatization, we suggest transferring to community-level authorship styles or styles mixed from multiple authors to prevent exposing the information of individual authors. To maximally preclude any unintended use, we only permit the use of our approach on public datasets or with the explicit consent of the target authors.

References

Nikolay Babakov, David Dale, Varvara Logacheva, and Alexander Panchenko. 2022. A large-scale computational study of content preservation measures for text style transfer and paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 300–321, Dublin, Ireland. Association for Computational Linguistics.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020*, pages 830–839. AAAI Press.

Daniel Matthew Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *ArXiv*, abs/1803.11175.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3168–3180, Minneapolis, Minnesota. Association for Computational Linguistics.

Skyler Hallinan, Faeze Brahman, Ximing Lu, Jaehun Jung, Sean Welleck, and Yejin Choi. 2023. STEER: Unified style transfer with expert reinforcement. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7546–7562, Singapore. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. 2020. Hooks in the headline: Learning to generate headlines with controlled styles. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5082–5093, Online. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858.

Aleem Khan, Elizabeth Fleming, Noah Schofield, Marcus Bishop, and Nicholas Andrews. 2021. A deep metric learning approach to account linking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5275–5287, Online. Association for Computational Linguistics.

Zae Myung Kim, Wanyu Du, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022. Improving iterative text revision by learning where to edit from other revision tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9986–9999, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A. Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6365–6378, Online. Association for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021a. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Dayiheng Liu, Jie Fu, Yidan Zhang, Christopher Joseph Pal, and Jiancheng Lv. 2019a. Revision in continuous space: Unsupervised text style transfer without adversarial learning. In *AAAI Conference on Artificial Intelligence*.

Jiacheng Liu, Skyler Hallinan, Ximing Lu, Pengfei He, Sean Welleck, Hannaneh Hajishirzi, and Yejin Choi. 2022. Rainier: Reinforced knowledge introspector for commonsense question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8938–8958, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Yixin Liu, Graham Neubig, and John Wieting. 2021b. On learning text style transfer with direct rewards. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4262–4273, Online. Association for Computational Linguistics.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. Quark: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems*, volume 35, pages 27591–27609. Curran Associates, Inc.

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

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8671–8680, Online. Association for Computational Linguistics.

Ajay Patel, Nicholas Andrews, and Chris Callison-Burch. 2023. Low-resource authorship style transfer: Can non-famous authors be imitated?

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.

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

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3932–3944, Online. Association for Computational Linguistics.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 837–848, Dublin, Ireland. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David Uthus, and Zarana Parekh. 2021. TextSETTR: Few-shot text style extraction and tunable targeted restyling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3786–3800, Online. Association for Computational Linguistics.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2195–2222, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

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

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

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914, Mumbai, India. The COLING 2012 Organizing Committee.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

A More Experimental Results

A.1 Full Automatic Evaluation Results

We show the full automatic evaluation results in Table 3 and Table 4.

[12]TSS, F, and MS are the three components of the reward function for STEER, which stand for target style strength, fluency, and meaning similarity, respectively.

Method	Model	Reward	Toward	Away	SBERT	Joint	Confusion
STRAP	LLaMA-7B	-	$0.088^\ddagger$	$0.793^\ddagger$	$0.650^\ddagger$	$0.414^\ddagger$	$0.30^\ddagger$
STYLL	GPT-3.5-turbo	-	$0.045^\ddagger$	$\underline{0.825}^\dagger$	$0.713^\ddagger$	$0.370^\ddagger$	$0.33^\ddagger$
	BLOOM-7B	-	$0.117^\ddagger$	$0.796^\ddagger$	$0.546^\ddagger$	$0.408^\ddagger$	$0.37^\ddagger$
	LLaMA-7B	-	0.159	**0.845**	$0.529^\ddagger$	$0.440^\ddagger$	**0.59**
SFT	LLaMA-7B	-	$0.137^\ddagger$	$0.707^\ddagger$	$0.754^\ddagger$	$0.484^\ddagger$	$0.32^\ddagger$
ASTRAPOP-PPO	LLaMA-7B	$T + (LP^\alpha - 1)$	$0.119^\ddagger$	$0.753^\ddagger$	**0.767**	$0.480^\ddagger$	$0.29^\ddagger$
		$A + (LP^\alpha - 1)$	$0.111^\ddagger$	$0.761^\ddagger$	$0.710^\ddagger$	$0.454^\ddagger$	$0.37^\ddagger$
		$T + A + (LP^\alpha - 1)$	$0.147^\ddagger$	$0.773^\ddagger$	$0.729^\ddagger$	$0.495^\dagger$	$\underline{0.48}^\dagger$
ASTRAPOP-DPO	LLaMA-7B	$T + (LP^\alpha - 1)$	$0.148^\ddagger$	$0.732^\ddagger$	$\underline{0.761}$	$0.500^\dagger$	$0.34^\ddagger$
		$A + (LP^\alpha - 1)$	$0.135^\ddagger$	$0.739^\ddagger$	$0.729^\ddagger$	$0.479^\ddagger$	$0.35^\ddagger$
		$T + A + (LP^\alpha - 1)$	$\underline{0.164}$	$0.748^\ddagger$	$0.733^\ddagger$	**0.507**	$0.44^\dagger$
ASTRAPOP-CPO	LLaMA-7B	$T + (LP^\alpha - 1)$	$0.151^\ddagger$	$0.743^\ddagger$	$0.749^\ddagger$	$0.501^\dagger$	$0.38^\ddagger$
		$A + (LP^\alpha - 1)$	$0.146^\ddagger$	$0.731^\ddagger$	$0.721^\ddagger$	$0.485^\ddagger$	$0.33^\ddagger$
		$T + A + (LP^\alpha - 1)$	**0.165**	$0.752^\ddagger$	$0.726^\ddagger$	$\underline{0.505}^\dagger$	$0.46^\dagger$

Table 3: The automatic evaluation results on the individual authorship style transfer task. The best and the second best scores for each metric are shown in **bold** and underline, respectively. "†" and "‡" indicate that the model is significantly different from the best model or the top two models, respectively, determined by t-test with $\alpha = 0.05$.

Method	Model	Reward	Toward	Away	SBERT	Joint	Confusion
Zero-shot	GPT-3.5-turbo	-	$0.005^\ddagger$	$0.811^\ddagger$	$0.885^\ddagger$	$0.240^\ddagger$	$0.013^\ddagger$
	LLaMA-7B	-	$0.022^\ddagger$	$0.880^\dagger$	$0.738^\ddagger$	$0.321^\ddagger$	$0.033^\ddagger$
STYLL	BLOOM-7B	-	$0.049^\ddagger$	$0.673^\ddagger$	$0.828^\ddagger$	$0.388^\ddagger$	$0.065^\ddagger$
	LLaMA-7B	-	$0.210^\ddagger$	$0.832^\ddagger$	$0.854^\ddagger$	$0.598^\ddagger$	$0.227^\ddagger$
STRAP / SFT	LLaMA-7B	-	$0.286^\ddagger$	$0.785^\ddagger$	**0.917**	$0.659^\ddagger$	$0.300^\ddagger$
STEER	LLaMA-7B	$TSS + F + MS$[12]	$0.334^\ddagger$	**0.926**	$0.879^\ddagger$	$0.699^\ddagger$	$0.348^\ddagger$
ASTRAPOP-PPO	LLaMA-7B	$T + (LP^\alpha - 1)$	$0.299^\ddagger$	$0.800^\ddagger$	$0.905^\ddagger$	$0.665^\ddagger$	$0.313^\ddagger$
		$A + (LP^\alpha - 1)$	$0.235^\ddagger$	$0.782^\ddagger$	$0.906^\ddagger$	$0.623^\ddagger$	$0.250^\ddagger$
		$T + A + (LP^\alpha - 1)$	$0.240^\ddagger$	$0.788^\ddagger$	$0.908^\ddagger$	$0.628^\ddagger$	$0.256^\ddagger$
ASTRAPOP-DPO	LLaMA-7B	$T + (LP^\alpha - 1)$	$0.490^\ddagger$	$0.843^\ddagger$	$0.915^\ddagger$	$0.767^\ddagger$	$0.499^\ddagger$
		$A + (LP^\alpha - 1)$	$0.321^\ddagger$	$0.789^\ddagger$	$\underline{0.917}$	$0.679^\ddagger$	$0.334^\ddagger$
		$T + A + (LP^\alpha - 1)$	$0.488^\ddagger$	$0.837^\ddagger$	$0.915^\ddagger$	$0.765^\ddagger$	$0.497^\ddagger$
ASTRAPOP-CPO	LLaMA-7B	$T + (LP^\alpha - 1)$	**0.655**	$0.887^\ddagger$	$0.897^\ddagger$	**0.827**	**0.662**
		$A + (LP^\alpha - 1)$	$0.456^\ddagger$	$0.835^\ddagger$	$0.909^\ddagger$	$0.749^\ddagger$	$0.467^\ddagger$
		$T + A + (LP^\alpha - 1)$	$\underline{0.654}$	$\underline{0.891}^\dagger$	$0.896^\ddagger$	$\underline{0.827}$	$\underline{0.660}$

Table 4: The automatic evaluation results on the community authorship style transfer task. The scores are averages over all pairs of native languages. The best and the second best scores for each metric are shown in **bold** and underline, respectively. "†" and "‡" indicate that the model is significantly different from the best model or the top two models, respectively, determined by t-test with $\alpha = 0.05$.

A.2 More Baseline LLMs

In addition to LLaMA-7B, we evaluate STYLL on BLOOM-7B since it has the best joint in (Patel et al., 2023). We also evaluate STYLL on GPT-3.5-turbo since the GPT-3 endpoint used in (Patel et al., 2023) is deprecated by OpenAI, and GPT-3.5-turbo is the closest available model, but we only use it for the individual authorship transfer task due to the limited budget. For the zero-shot transfer approach, we also use GPT-3.5-turbo to show its performance with one of the current best LLMs. We do not use GPT-4 due to the limited budget. Compared to BLOOM-7B and GPT-3.5-turbo, LLaMA-7B has the best joint score in all baseline approaches on both tasks, so the full results are still consistent with the concise version in Table 1 and Table 2.

A.3 Reward Function Ablation Study

We ablate the toward reward and the away reward from the reward function separately to assess their individual effects on the model performance. For the individual authorship transfer task, when using partial reward functions without the toward reward or the away reward, the PO algorithms can still improve the score corresponding to the remaining term in the reward function in most cases, but none of the towards, away, and joint scores is as good as the model trained on the full reward function using each algorithm. In contrast, for the community authorship transfer task, the away reward does not help improve the away score in most cases, and training with only the toward reward and the length penalty yields the model with the best overall performance for each PO algorithm.

A.4 Human Evaluation

We conduct a human study on the community authorship transfer task. We randomly select 10 samples for each target native language from the test set for STYLL, STRAP, STEER, and ASTRAPOP-CPO, and collect up to 3 annotations for each. The samples are evaluated in two dimensions, style confusion (SC) and content preservation (CP). The style confusion is a simpler version of the confusion score we use in the automatic evaluation. We show the annotators three examples each in the source style and the target style, and ask them "which is the style of the transferred text". The confusion score is 1 if they select the target style, other with 0. We assess the content preservation using a 3-point Likert scale ranging from 0 to 2. The detailed instructions are shown in Table 6.

The results are shown in Table 5. Since the style classification task has been shown to be very difficult for humans (Krishna et al., 2020; Hallinan et al., 2023), we perform an independent t-test on the results and find no statistically significant difference in style confusion in any model pairs. However, we observe statistically significant differences in content preservation, which indicates that both STRAP and STEER are significantly better than STYLL and ASTRAPOP-CPO. Also, the results on content preservation are generally consistent with the SBERT in the automatic evaluation except for the STEER model.

	SC	CP
STYLL	<u>0.622</u>	$0.955^{\ddagger}$
STRAP	0.516	<u>1.267</u>
STEER	**0.690**	**1.279**
ASTRAPOP-CPO	0.537	$1.018^{\ddagger}$

Table 5: Human evaluation results on the community authorship transfer task. The best and the 2nd best scores in each column are emphasized in **bold** and <u>underline</u>, respectively. "‡" indicates a statistically significant difference between the top two models determined by independent sample t-test with $p < 0.05$.

A.5 Case Study

We show a transfer example on the community authorship transfer task in Table 7 for a simple qualitative case study. It shows that ASTRAPOP-CPO successfully captures a common typo, "alot" and three main characteristics of the target style: using all lowercase, using space before comma, and high typo rate, while no other model is able to capture any of these.

B More Implementation Details

B.1 Statistical Significance Test

We perform a resampled paired t-test on all results. Specifically, we randomly draw subsets from the test set and perform paired t-tests on the scores of the subsets. For the individual authorship style transfer task, we sample at the author level since the style model works at the author level. For the community authorship style transfer task, we sample at the document level. The hyperparameters for the resampling t-test are shown in Table 8.

B.2 Hyperparameters

Due to limited time and computational resources, we are not able to perform a thorough search on all hyperparameters, but we search for several important hyperparameters and show the best-performing hyperparameters for both training and generation in Table 9, Table 10, Table 11, and Table 12. Table 13 and Table 14 show the hyperparameter we test in the experiments. For the low-resource STRAP and all few-shot exemplar-based models, we use 5 target exemplars for each author.

B.3 Model Input Formats

For LLM prompting based approaches, we use natural language prompts shown in Table 15. For

Style Confusion	Question	Based on the examples above, what is the style of the following text?
Content Preservation	Similar	Most of the meaning (75% or more) of the two passages is the same.
	Somewhat Similar	Large portions (50-75%) of the passages are the same, but there are significant sections that differ or are present in only one passage.
	Not Similar	Only small portions (less than 50%) of the passages are the same.
	Question	How similar are the following two texts?

Table 6: Instructions used in the human evaluation.

Target Sample	traviling is a very nice thing , it hepls you see new coultcurs and to meet new poeplo , there are alot of ways for traviling and i believe that the best way is to travil with a group led by a tour guide , [... more]
Source Text	The influnce of advertisements on the customers is worth commendable. The advertisers are projecting thier goods to the customers in a 'larger than real' manner.' [... more]
Zero-shot	Advertisements have a powerful influence on consumers, overstating the features and benefits of products to make them seem better than they actually are. [... more]
STYLL	Advertisements are used to promote products, making them appear attractive and useful. Through exaggeration, advertisers present their products as having exceptional features. [... more]
STRAP	The advertisements have a great effect on the customers and they should be praised. The advertisers make the products seem more better in the eyes of the customers rather than they really are. [... more]
STEER	Advertising has a positive impact on customers, as it promotes products in a way that exaggerates their qualities. Advertisers often portray their products as superior to reality. [... more]
ASTRAPOP-CPO	because the advertisment have alot effec on the custumers , and make the products seem better than how they really are, [... more]

Table 7: An example from the ETS test set. Due to the limited space, we only show the beginning of each document.

	# Subsets	Subset Size
Individual	10	20 authors
Community	10	1100 docs

Table 8: Hyperparameters for the resampling t-test.

LoRA Hyperparameters	
r	16
α	32
dropout	0.05
target modules	q_proj, v_proj

Table 9: Hyperparameters for the LoRA adapters.

learning-based approaches, we use simpler prompts with special tokens[13] which are shown in Table 16.

[13]We do not add new tokens to the tokenizer and model. All inputs are tokenized by the original tokenizer.

B.4 Hardware and Runtime

We report the training hardware and runtime for all learning-based approaches in this work in Table 17.

B.5 Paraphraser Selection

We train a LLaMA paraphraser on GPT-3.5-turbo generated paraphraser data for two main reasons. First, the transfer pipeline is fully local with the LLaMA paraphraser, which is more cost-efficient and manageable. Second, more importantly, we find that the LLaMA paraphraser trained on the GPT-3.5-turbo generated data performs even better than GPT-3.5-turbo. Specifically, the trained LLaMA paraphraser achieves 0.764 SBERT cosine similarity on the test set, while GPT-3.5-turbo only has 0.738.

B.6 Reward Model Selection

We use LUAR for the experiments on the Reddit data since LUAR is the SOTA and most widely

	Paraphraser	ASTRAPOP				STRAP
	SFT	SFT	PPO	DPO	CPO	SFT
learning rate	5e-5	5e-5	1.41e-5	2e-6	2e-6	5e-5
batch size	32	32	32	32	32	5
# epochs	6	6	6	6	6	60
KL coef / β	-	-	0.2	0.5	0.1	-
top p	-	-	1.0	1.0	1.0	-
temperature	-	-	1.0	1.0	1.0	-
length penalty α	-	-	0.5	0.5	0.5	-

Table 10: Training hyperparameters for the individual authorship style transfer task.

	Paraphraser / Classifier	ASTRAPOP				STRAP	STEER	
	SFT	SFT	PPO	DPO	CPO	SFT	Expert Model	QUARK (RL)
learning rate	5e-5	5e-5	1.41e-5	2e-6	2e-6	5e-5	5e-5	5e-5
batch size	32	8	32	16	16	8	8	8
# epochs	6	6	6	10	10	6	6	6 (offline) + 10 (online)
KL coef / β	-	-	0.2	0.5	0.1	-	-	0.025
top p	-	-	1.0	1.0	1.0	-	-	1.0
temperature	-	-	1.0	1.0	1.0	-	-	1.0
length penalty α	-	-	0.5	0.5	0.5	-	-	-

Table 11: Training hyperparameters for the community authorship style transfer task.

	Pseudo-Parallel Data Generation		Inference
	ASTRAPOP	STEER	All Models
top p	1.0	1.0	1.0
temperature	0.7	1.0	0.7
DExperts α	-	1.0, 1.6, 1.8, 2.0, 2.2	-
over generation	$\times 1$	$\times 50$	-

Table 12: Generation hyperparameters for both tasks.

	Learning Rate	KL coef / β	Batch Size
PPO	1.41e-5, 2.82e-5, 4.23e-5	0.2	8, 16, 32
DPO	5e-7, 1e-6, 2e-6	0.1, 0.2, 0.3, 0.5	8, 16, 32
CPO	5e-7, 1e-6, 2e-6, 5e-6, 1e-5, 1e-4	0.1, 0.5	8, 16, 32

Table 13: Hyperparameters tested for ASTRAPOP.

Phase	Hyperparameter	
Expert-Guided Data Generation	α	0.2, 0.4, 0.6, 1.0, 1.6, 1.8, 2.0, 2.2
	temperature	0.7, 1.0, 1.3
	over generation	$\times 10$, $\times 30$, $\times 50$
QUARK (RL)	Learning Rate	1e-5, 5e-5
	KL coef	0.025, 0.05
	Batch Size	8, 32

Table 14: Hyperparameters tested for STEER.

Method	Model	Prompt
Zero-shot	LLaMA-7B	[INST] <<SYS>>\nYou are a college student whose native language is <target_native_language>.\n<</SYS>>\n\nUsing the writing style of a college student whose native language is <target_native_language>accurately paraphrase the following passage in English.\n\nOriginal Passage:\n<text_to_be_transferred>[/INST]Sure, using the writing style of a native <text>speaker, here is the paraphrased passage in English:\n
	GPT-3.5-turbo	Passage: <text>\n\nUsing the writing style of a college student whose native language is <target_native_language>accurately paraphrase the passage in English.\n\nRewrite:
STYLL (paraphrase)	LLaMA-7B	[INST] <<SYS>>\nYou are an expert at paraphrasing.\n<</SYS>>\n\nPlease paraphrase the following passage in a simple neutral style.\n\n Passage: <text>[/INST]Sure! Here's a paraphrased version of the passage in a simple and neutral style:\n\n
	GPT-3.5-turbo	Passage: <text>\n\nParaphrase the passage in a simple neutral style.\n\nRewrite:
STYLL (descriptor)	LLaMA-7B	[INST] <<SYS>>\nYou are an expert at writing style analysis.\n<</SYS>>\n\nPassage: <target_text1>\nPassage: <target_text2>\nList 5 adjectives, comma-separated, that describe the writing style of the author of these passages. [/INST]Sure, here are 5 adjectives, comma-separated, that describe the writing style of the author of these passages:
	GPT-3.5-turbo BLOOM-7B	Passage: <target_text1>\nPassage: <target_text2>\nList some adjectives, comma-separated, that describe the writing style of the author of these passages:
STYLL (transfer)	LLaMA-7B GPT-3.5-turbo BLOOM-7B	Here is some text: {<neutral_target_text1>} Here is a rewrite of the text that is more <descriptors>: {<target_text1>} Here is some text: {<neutral_target_text2>} Here is a rewrite of the text that is more <descriptors>: {<target_text2>} Here is some text: {<neutral_source_text>} Here is a rewrite of the text that is more <descriptors>: {

Table 15: Prompts for the LLM prompting approaches. For exemplar-based approaches, we only show two target exemplars for illustration. Please see § B.2 for the actual number used in the experiments.

Approach	Level	Prompt
Ours (paraphrase)	Individual Community	[SRC]<text>[/SRC]
Ours (transfer)	Individual	[REF]<target_text1>[/REF][REF]<target_text2>[/REF] [SRC]<neutral_source_text>[/SRC]
	Community	[SRC]<neutral_source_text>[/SRC]
STRAP (paraphrase)	Individual Community	[SRC]<text>[/SRC]
STRAP (transfer)	Individual Community	[SRC]<neutral_source_text>[/SRC]
STEER	Community	[SRC]<source_text>[/SRC]

Table 16: Prompts for the learning-based approaches. For exemplar-based approaches, we only show two target exemplars for illustration. Please see § B.2 for the actual number used in the experiments.

		Individual		Community	
		GPUs	Time (hrs)	GPUs	Time (hrs)
Paraphraser	SFT	A40x2	2	A40x2	3
ASTRAPOP	SFT	A40x2	12	A40x1	6
	PPO	A40x2	29	A40x2	40
	DPO	A40x2	4 + 8	A40x2	9 + 14
	CPO	A40x2	4 + 6	A40x2	9 + 10
STRAP	SFT	A40x1	1	A40x1	6
STEER	Expert Model	-	-	A40x1	3
	QUARK (RL)	-	-	A40x1	651 + 43

Table 17: Training hardware and runtime for the learning-based approaches. For DPO, CPO, and STEER QUARK, the two numbers for time are the data generation time on a single A40 GPU and the training time, respectively. We do not report the runtime for the paraphrase data generation since it is done through the OpenAI API.

used authorship verification model on the Reddit dataset. However, preliminary experiments show that LUAR has only 0.53 accuracy on the ETS test set, while a trained classifier achieves 0.71 accuracy on the same test set. Therefore, we use the trained classifier as the reward model for the experiments on the ETS dataset. The reason for the large performance gap is that ETS has a countable number of classes (11 native languages) with plenty of training data to learn to accurately determine the author's native language. Also, LUAR is a representation model that is designed to solve open-set problems in which the test data may have authors with textual styles never seen in the training collection, and this is in line with the setting of our low-resource transfer task on Reddit.

C Scientific Artifacts

C.1 Use of Existing Artifacts

We list all existing artifacts we use in this work with their licenses and links in Table 18. The numbers of parameters of the models are shown in the same table in parentheses. The artifacts are under various licenses, but all permit the use for research purposes. All artifacts listed are allowed to be used in this work.

C.2 Created Artifacts

We create a new training framework in this work. We release the code for the training framework and several models trained with the framework under the MIT license. We only allow research use of our code and models on personal and public data, which is compatible with the original access conditions of the models and datasets. Using the model on other individuals without authorization is unethical and strictly forbidden.

Type	Name	License	Link
Dataset	Million User Dataset	Apache-2.0	https://github.com/noa/naacl2021
	ETS Corpus	License link	https://catalog.ldc.upenn.edu/LDC2014T06
Model	LLaMA-2-7B (6.7B)	Meta	https://huggingface.co/meta-llama/Llama-2-7b-hf
	LLaMA-2-7B-chat (6.7B)	Meta	https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
	BLOOM-7B (7.1B)	RAIL License v1.0	https://huggingface.co/bigscience/bloom-7b1
	GPT-3.5-turbo (-)	MIT	https://platform.openai.com/docs/models/gpt-3-5-turbo
	RoBERTa-large (355M)	MIT	https://huggingface.co/FacebookAI/roberta-large
	RoBERTa-large-COLA (355M)	MIT	https://huggingface.co/cointegrated/roberta-large-cola-krishna2020
	all-mpnet-base-v2 (109M)	Apache-2.0	https://huggingface.co/sentence-transformers/all-mpnet-base-v2
	LUAR-MUD (83M)	Apache-2.0	https://huggingface.co/rrivera1849/LUAR-MUD
Software	Huggingface Transformers	Apache-2.0	https://github.com/huggingface/transformers
	Huggingface PEFT	Apache-2.0	https://github.com/huggingface/peft
	Huggingface TRL	Apache-2.0	https://github.com/huggingface/trl
	Sentence Transformers	Apache-2.0	https://github.com/UKPLab/sentence-transformers
	NLTK	Apache-2.0	https://github.com/nltk/nltk
	ALMA (for CPO trainer)	MIT	https://github.com/fe1ixxu/ALMA

Table 18: Artifacts used in this work and their licenses and links. The number of parameters of the models are shown in parentheses.