Reduction-Synthesis: Plug-and-Play for Sentiment Style Transfer

Sheng Xu[1] and **Fumiyo Fukumoto**[2] and **Yoshimi Suzuki**[2]
[1]Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences
[2]Graduate Faculty of Interdisciplinary Research
University of Yamanashi, Kofu, Japan
{g22dts03,fukumoto,ysuzuki}@yamanashi.ac.jp

Abstract

Sentiment style transfer (SST), a variant of text style transfer (TST), has recently attracted extensive interest. Some disentangling-based approaches have improved performance, while most still struggle to properly transfer the input as the sentiment style is intertwined with the content of the text. To alleviate the issue, we propose a plug-and-play method that leverages an iterative self-refinement algorithm with a large language model (LLM). Our approach separates the straightforward Seq2Seq generation into two phases: (1) **Reduction** phase which generates a style-free sequence for a given text, and (2) **Synthesis** phase which generates the target text by leveraging the sequence output from the first phase. The experimental results on two datasets demonstrate that our transfer method is effective for challenging SST cases where the baseline methods perform poorly. Our code is available online[1].

1 Introduction

Text style transfer (TST) has been first explored as the frame language-based systems (McDonald and Pustejovsky, 1985). The goal is to change the text style, such as formality and politeness while preserving the style-free content of the input text. As demonstrated in the previous works, the disentanglement, i.e., disentangling style from text then fusing target style in hidden space corresponding to domain-specific data, has been indeed repeatedly proven to be a feasible approach (Shen et al., 2017; John et al., 2019; Bao et al., 2019; Lee et al., 2021; Sheng et al., 2023; Hu et al., 2023). However, the previous works on the disentanglement-based approaches still suffer from two insufficiencies. (1) It is not clearly shown that the semantic representation is entirely disentangled from the original style representation (Lee et al., 2021). Especially, Jin et al. (2022) demonstrated the sentiment style, unlike formality features, is more of a content-related attribute. For example, in transforming the negative input "I hate making decisions" into the positive output "I love making decisions", the semantics would reverse along with the sentiment style (Ziems et al., 2022). (2) Few works address the issue that the challenging case is variable among the transfer cases. For example, as shown in (a) of Figure 1, it is easy to transfer from "Ever since Joe has changed hands it's just gotten worse and worse." to "Ever since Joe has changed hands it's gotten better and better.". However, it is difficult to transfer from "It isn't terrible, but it isn't very good either." to "It isn't perfect, but it is very good.". The reason is that the sentiment style of the input, i.e., "isn't terrible" (neutral) and "isn't very good" (negative) is intertwined with the content of the sentence.

Figure 1: Examples of SST: (a) from negative to positive and (b) from positive to negative. The words with green color refer to the style-free content, and the blue and red fonts indicate the parts with negative and positive styles in context, respectively.

In this work, we present a simple, yet effective plug-and-play method for the relatively challenging cases in a specific SST task by leveraging the

[1]https://github.com/codesedoc/RS4SST.git

Proceedings of the 17th International Natural Language Generation Conference, pages 330–343
September 23–27, 2024.

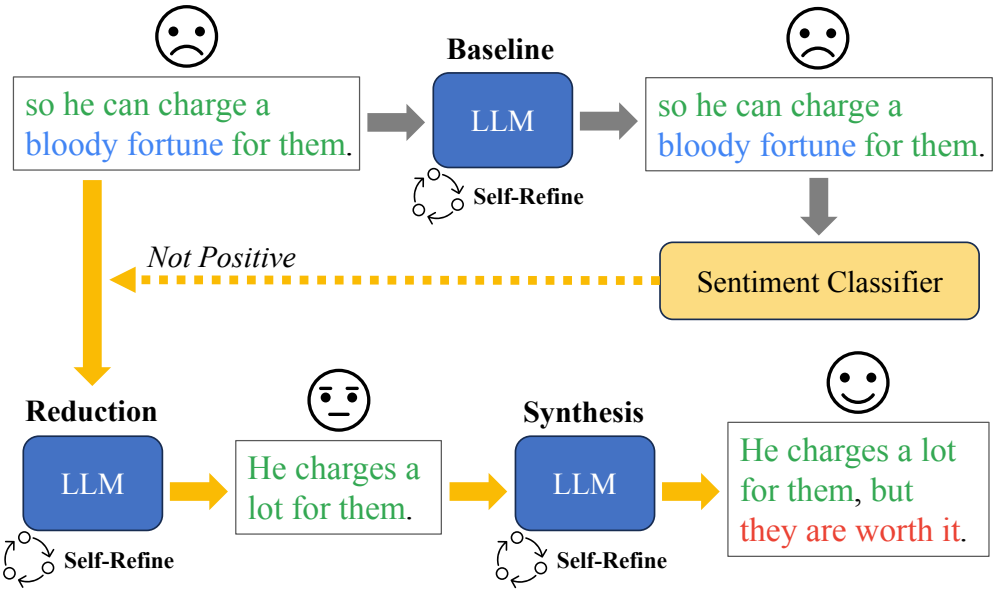

Figure 2: The pipeline of the reduction-synthesis method by leveraging LLM and Self-Refine algorithm. The words with green color express the style-free content, and the blue and red fonts indicate the parts with negative and positive styles respectively.

LLM augmented with the Self-Refine algorithm (Madaan et al., 2023). We define that, for a specific model and SST task, the samples in the dataset that can not be transferred accurately are more challenging cases. To address such SST cases, our plug-and-play method generates the target style via two phases, i.e., reduction and synthesis, which lead to LLM mining style-free sequence from the input text, and re-generate the target text by adding the target style to the style-free sequence.

Differentiate from "disentangling" and "fusing" operations for hidden states, the reduction and synthesis phases guarantee the model to distinguish sentiment as well as other style-free content of the sentence in the form of natural language. The experiment results show that our plug-and-play method efficiently assists the LLMs transfer challenging cases of SST.

2 Related Work

Previous work on the TST task based on deep learning techniques gained significant performance. One line of work is to utilize a nonparallel dataset and train a model in an unsupervised manner (Shen et al., 2017; Fu et al., 2018). John et al. (2019) propose a method that disentangled content and style-related features and made the decoder generate an ideal output using the disentangled features. Another paradigm is to apply supervised learning to parallel data. To mitigate the small size of the parallel data, Rao (Rao and Tetreault, 2018) presented data augmentation strategies. Xu et al. (2019) and Zhang et al. (2020) propose a multi-task learning-based method to train the model on parallel data. Several innovative approaches have also been proposed for TST tasks. Lai et al. (2021) design two types of rewards for target style and content based on reinforcement learning. Han et al. (2023) explores the hidden transfer patterns from the dataset to improve the performance of the TST task.

The popular prompt-based methodology has also been extensively studied and has obtained outstanding performances, especially by leveraging large language models (LLMs). Reif et al. (2022) propose an augmented zero-shot learning method by utilizing the LLMs including GPT3 (Brown et al., 2020) and LaMDA (Thoppilan et al., 2022), which release the cost of annotation and training. Suzgun et al. (2022) designed a reranking approach to choose the best output from the generated candidates from GPT-2 (Radford et al., 2019) and its variants. Luo et al. (2023) leverage the word-level edit-based prompt and design a discrete searching algorithm to predict the target text. Liu et al. (2024) constructed a set of prompt candidates and trained a scoring model that predicts one of the candidates to obtain the best generations for each input.

3 Plug-and-Play Approach

Figure 2 shows our straightforward plug-and-play method by illustrating an example of a challenging case from the Yelp dataset for transferring the negative to the positive style. We first apply the sentiment classifier to the output of the baseline model and detect the challenging cases, i.e., the sentiments of the generations obtained by the baseline model are incorrect. We then use our plug-and-play method to transfer these cases instead of the baseline.

As illustrated in Figure 2, the baseline just duplicates the input text with negative sentiment, "so he can charge a bloody fortune for them.". In contrast, our plug-and-play method deals with the input in the first phase, **Reduction**, to detect a style-free sequence, "He charges a lot for them.". The output is then passed to the second phase, **Synthesis**, to generate the expected positive output: "He charges a lot for them, but they are worth it.". To do this, we formulate the SST task and further decompose the SST into two sub-objectives with lower boundaries.

3.1 Problem Formulation

Let D be a set of text. Each sequence in D contains a sentiment style, *positive* (pos), negative (neg), or *neutral* (neu). For the SST task, we considered two main transfer cases i.e., from *positive* to *negative* and from *negative* to *positive* ($pos \rightleftarrows neg$). Given a pair of source text X, and its target counterpart Y with a sentiment style label s, e.g. *positive*, the objective of the SST task can be formulated as the language model $\mathbb{P}(\mathrm{Y}|\mathrm{X}, s)$, where $s \in \{pos, neg\}$ and $\mathrm{X}, \mathrm{Y} \in D$.

Let also C be a style-free content text. We assume that one such neutral text C which should be preserved during transferring from X to Y exists. The objective of SST can be further decomposed as follows:

$$\mathbb{P}(\mathrm{Y}|\mathrm{X}, s) = \underbrace{\mathbb{P}(\mathrm{C}|\mathrm{X})}_{\textit{reduction}} \underbrace{\mathbb{P}(\mathrm{Y}|\mathrm{X}, \mathrm{C}, s)}_{\textit{synthesis}} \quad (1)$$

The detailed derivation of Eq. (1) is shown in the Appendix A.1. Following the derivation in Eq. (1), the optimization of the objective of the SST task can be decomposed into two components, reduction and synthesis, with lower boundaries.

3.2 Reduction and Synthesis

Note that the autoregressive pre-trained objective is more inherently similar to the optimization components of Eq. (1) and has outstanding performance for open-end text generation. We thus prompt the LLM to infer a proper style-free content C from X. We call this procedure as reduction phase. We then lead the model to generate the expected target by another prompt, called as synthesis phase. Inspired by Kojima et al. (2022), the reduction and synthesis can be regarded as a guidance that helps the pre-trained language model to transfer the sentiment polarity of the source sequence along with a chain-of-thought. Moreover, for each phase, we leverage the Self-Refine algorithm, which is a specific resolution to mitigate the common hallucination issues and is often used in LLMs-based systems. Here, we will not provide a thorough background on the Self-Refine framework and refer readers to the paper by Madaan et al.(Madaan et al., 2023).

Let R_{ge}, R_{fb}, and R_{re} be the generation, feedback, and refinement prompt formats for the reduction phase, respectively. Likewise, let S_{ge}, S_{fb}, and S_{re} be those counterparts for the synthesis phase. We utilize the same stop condition f_{stop} for both phases. Let $\mathcal{F}_{SR}$ indicate the Self-Refine algorithm and llm be the model used to infer generation at each prompt step. In the first phase, the style-free content C from the source X can be obtained by Eq. (2). The final generation Y is inferred in the second phase which is given by Eq. (3).

$$\mathrm{C} = \mathcal{F}_{SR}(\mathrm{X}, llm, R_{ge}, R_{fb}, R_{re}, f_{stop}) \quad (2)$$

$$\mathrm{Y} = \mathcal{F}_{SR}(\mathrm{X}, \mathrm{C}, llm, S_{ge}, S_{fb}, S_{re}, f_{stop}) \quad (3)$$

4 Experiments

4.1 Setup

Dataset and Setting. We conducted experiments on two benchmark datasets for SST: Yelp (Xiang et al., 2015) and Amazon (Li et al., 2018) reviews. Every dataset combines 1,000 examples which are split into two groups, 500 sentences for $neg \rightarrow pos$, and another 500 for $pos \rightarrow neg$. The other hyper-parameters and detail settings are shown in the Appendix A.2. As all inferences are conducted by leveraging the Self-Refine algorithm, for both baseline and our method, we design the initial generation prompt, feedback prompt, and refine prompt, respectively. In each phase, we design 2-shots for every prompt format in Eqs. (2) and (3). The detailed prompt formats are illustrated in Appendix A.4.

Automatic Evaluation. We used three aspects of evaluation metrics. The first is content preservation, which consists of reference-SacreBLEU

Model	Automatic Evaluation						Human Evaluation		
	Acc ↑	r-sB ↑	s-sB ↑	r/s-sB ↑	t-PPL ↓	s-PPL ↓	Content ↑	Style ↑	Fluency ↑
pos → neg									
BL	87.4	**23.0**	**44.0**	0.523	64	134	3.87	4.05	4.16
RS	85.8	16.1	28.7	0.562	**58**	**110**	3.78	3.90	4.15
BL+RS	**93.0**	21.8	40.1	**0.545**	61	126	**3.93**	**4.17**	**4.18**
impv. (%)	+6.4	-5.2	-8.9	+4.2	+4.7	+6.0	+2.6	+3.0	+0.5
neg → pos									
BL	63.6	**16.7**	**27.3**	0.612	33	78	3.34	3.46	3.65
RS	63.4	12.1	19.0	0.637	31	**57**	**3.40**	3.59	**3.70**
BL+RS	**72.4**	15.6	24.4	**0.640**	**30**	70	**3.41**	**3.59**	3.69
impv. (%)	+13.8	-6.5	-10.7	+4.6	+9.1	+10.3	+2.1	+3.8	+1.1

Table 1: Comparison with the Self-Refine (baseline, represented with BL) on Yelp dataset. The RS indicates the plug-and-play method, and the BL+RS is the method augmenting the BL with RS, that is, replacing the incorrect output of BL with the generation of RS. The **bold** font marks the best performance of each metric. The "impv." means the improvements of BL+RS against the baseline.

Model	*pos → neg*					*neg → pos*				
	Acc† ↑	r-sB ↑	s-sB ↑	r/s-sB ↑	t-PPL ↓	Acc† ↑	r-sB ↑	s-sB ↑	r/s-sB ↑	t-PPL ↓
CrossAlignment	72.0	7.3	19.3	0.378	224	**74.0**	8.3	19.3	0.430	190
GPT-J-6B-4s	81.0	**25.3**	**50.5**	0.501	107	52.0	**21.7**	**48.7**	0.569	82
BL	87.4	23.0	44.0	0.523	64	63.6	16.7	27.3	0.612	33
BL+RS (ours)	**93.0**	21.8	40.1	**0.545**	**61**	72.4	15.6	24.4	**0.640**	**30**

Table 2: Comparison with related work on the Yelp dataset. The results of CrossAlignment, and GPT-J6B-4s are referred to in the work of Suzgun et al. (2022). The **bold** font shows the best performance for each metric. †: Instead of fine-tuning a Roberta model in the related work, we used a third-party sentiment analysis toolkit to calculate the Acc of generations, which is explained in Section 4.1.

(r-sB) and self-SacreBLEU (s-sB) scores (Suzgun et al., 2022). Here, r-sB and s-sB measure the distance from the generated sentence to the ground truth reference, and the degree to which the model directly copies the source, respectively. The second is transfer strength, which is scored by using accuracy (Acc) on the target style of the generations. The last is the fluency of generated texts consisting of average token-level perplexity (t-PPL) and average sentence-level perplexity (s-PPL). Furthermore, we add a new metric, the rate of r-sB against s-sB, named r/s-sB for evaluating the intent of the trade-off between generating new text and preserving source content during style transfer. To calculate the r-sB and s-sB scores, we used the evaluator, which is available from the Hugging Face.[2] The Python toolkit for sentiment analysis, named pysentimiento[3] (Pérez et al., 2021) is utilized to run a sentiment classifier to calculate the Acc. The gpt2-large[4] is selected as the language model to compute the t-PPL and s-PPL.

Human Evaluation. To mitigate the insufficiency of automatic metrics, we also conducted a small-scale in-house human evaluation of the Yelp dataset by assigning the predictions of 50 samples to two reviewers with background knowledge about the domain of the dataset. The evaluation criterion consists of the content preservation capacity, sentiment transfer length, and fluency, and a score range from 1 to 5 is annotated for each aspect[5]. Finally, we average scores from two reviewers for the same example in the test dataset.

4.2 Results

Table 1 shows the performance comparison with the Self-Refine baseline on the Yelp dataset. Except for the r-sB, and s-sB scores, our method (BL+RS) which is enhanced by plug-and-play can improve

[2]https://huggingface.co/docs/evaluate/index

[3]https://github.com/pysentimiento/pysentimiento

[4]https://huggingface.co/openai-community/gpt2-large

[5]All annotations are blind, i.e., the reviewers do not know which method was used to make the predictions.

Style		$neg \rightarrow pos$			$pos \rightarrow neg$		
		Reduction (%)	**Synthesis (%)**	**Self-Refine (%)**	**Reduction (%)**	**Synthesis (%)**	**Self-Refine (%)**
$s_i = neg$	$s_o = neg$	230 (72.8)	63 (21.7)	54 (17.1)	1 (16.7)	35 (83.3)	5 (83.3)
	$s_o = neu$	68 (21.5)	42 (14.5)	37 (11.7)	3 (50.0)	3 (7.2)	1 (16.7)
	$s_o = pos$	18 (5.7)	185 (63.8)	225 (71.2)	2 (33.3)	4 (9.5)	0 (0)
$s_i = neg$		**316**	**290**	**316**	**6**	**42**	**6**
$s_i = neu$	$s_o = neg$	45 (31.3)	9 (5.6)	9 (6.2)	6 (16.2)	129 (66.5)	16 (43.2)
	$s_o = neu$	82 (56.9)	46 (28.4)	73 (50.7)	26 (70.3)	46 (23.7)	20 (54.1)
	$s_o = pos$	17 (11.8)	107 (66.0)	62 (43.1)	5 (13.5)	19 (9.8)	1 (2.7)
$s_i = neu$		**144**	**162**	**144**	**37**	**194**	**37**
$s_i = pos$	$s_o = neg$	15 (37.5)	0 (0)	0 (0)	35 (7.7)	211 (79.9)	378 (82.7)
	$s_o = neu$	12 (30.0)	1 (2.1)	3 (7.5)	165 (36.1)	10 (3.8)	14 (3.1)
	$s_o = pos$	13 (32.5)	47 () (97.9)	37 (92.5)	257 (60.2)	43 (16.3)	65 (14.2)
$s_i = pos$		**40**	**48**	**40**	**457**	**264**	**457**

Table 3: Distribution of the style of input and output pairs during every transfer phase on Yelp data. Self-Refine is the baseline that directly transfers the input to the target. The background indicates the number and rate of correct results in each transfer phrase. The **bold** in each column refers to the marginal distribution of the input.

the performance over the baseline by both automatic and human evaluations. As Suzgun et al. (2022) mentioned, the $neg \rightarrow pos$ transfer is more challenging than that of $pos \rightarrow neg$ in all metrics, except for the perplexities, obtained for $pos \rightarrow neg$ far exceeds that for $neg \rightarrow pos$. except for r/s-B, t(s)-PPL. The improvements obtained by our plug-and-play method for $neg \rightarrow pos$ (by Acc, r/s-B, s-PPL, Style, and Fluency) are larger than those of the counterparts for $pos \rightarrow neg$.

We can see from Table 1 that our RS can improve the content score in human evaluation for both transfer directions, while BL+RS is worse than the baseline (BL) for the r-sB and s-sB in automatic metrics. One possible reason is that the LLM generates more creative content by two phrases prompting in RS method. Another factor is that the two objectives, transferring sentiment style and preserving content are trade-offs and often conflict. The inherent flaws of automatic metrics result in the inconsistency with human evaluation, as discussed by Mir et al. (2019), the BLEU only measures n-gram overlaps and does not take the style transfer into account is accompanied by changes of words. It is worth noting that our RS obtains a worse entire performance than BL. This demonstrates that RS is only suitable for transferring challenging cases.

In Table 2, we also compare the performance of baseline and our method on the Yelp dataset with several related works including one supervised learning-based method, CrossAlignment (Shen et al., 2017), and one prompt-based methods (Suzgun et al., 2022). Consistently, our method (BL+RS) performs better on most metrics.

Table 3 shows the number of style texts in each of the three transfer phrases, Reduction, Synthesis, and Self-Refine for $neg \rightarrow pos$, and $pos \rightarrow neg$ in Yelp data set. Due to space limit, other results obtained by Yelp and Amazon datasets are shown in Tables 5, 6, 7 and 8 in the Appendix A.3. In Table 3, s_i and s_o indicate the input and output style, respectively, in each phrase.

As shown in Table 3, the number of inputs classified into neutral in $neg \rightarrow pos$ case (144) is larger than those of $pos \rightarrow neg$ (37). This shows that $neg \rightarrow pos$ case includes more ambiguous inputs than $pos \rightarrow neg$, resulting in poor performance. We can also see from Table 3 that the synthesis phrase successfully transfers 66.0% neutral texts to the positive style in the $neg \rightarrow pos$ task, and 66.5% neutral texts to the negative style in the $pos \rightarrow neg$ task in Table 3, while the baseline (Self-Refine) of these are 43.1% and, 43.2%, respectively. This indicates the effectiveness of our approach.

5 Conclusion

In this work, we proposed a simple, yet effective plug-and-play method, Reduction-Synthesis, to augment the base LLM for the SST task, especially for the challenging transfer cases. Experiments on two datasets show the effectiveness of our method. Future work includes (i) investigating effective generation methods in both two phases, (ii) applying our approach to transfer other text styles, and (iii) exploring more robust automatic evaluation to examine the trade-off between style transfer and content preservation.

Limitation

The performance obtained by our approach is subject to the quality of the middle style-free sequence during the two-step prompt inference. Moreover, carefully crafted prompt formats are necessary for outstanding generation.

Ethics Statement

This paper does not involve the presentation of a new dataset, an NLP application, or the utilization of demographic or identity characteristics information.

Acknowledgements

We would like to thank anonymous reviewers for their comments and suggestions. This work is supported by the Support Center for Advanced Telecommunications Technology Research (SCAT), and JSPS KAKENHI (No.22K12146). Sheng Xu is supported by JST SPRING, Grant Number JPMJSP2133.

References

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):663–670.

Jingxuan Han, Quan Wang, Licheng Zhang, Weidong Chen, Yan Song, and Zhendong Mao. 2023. Text style transfer with contrastive transfer pattern mining. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7914–7927, Toronto, Canada. Association for Computational Linguistics.

Yahao Hu, Wei Tao, Yifei Xie, Yi Sun, and Zhisong Pan. 2023. Token-level disentanglement for unsupervised text style transfer. *Neurocomputing*, 560:126823.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep learning for text style transfer: A survey. *Computational Linguistics*, 48(1):155–205.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021. Thank you BART! rewarding pre-trained models improves formality style transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 484–494, Online. Association for Computational Linguistics.

Dongkyu Lee, Zhiliang Tian, Lanqing Xue, and Nevin L. Zhang. 2021. Enhancing content preservation in text style transfer using reverse attention and conditional layer normalization. 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 93–102, Online. Association for Computational Linguistics.

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

Qingyi Liu, Jinghui Qin, Wenxuan Ye, Hao Mou, Yuxuan He, and Keze Wang. 2024. Adaptive prompt routing for arbitrary text style transfer with pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18689–18697.

Guoqing Luo, Yu Han, Lili Mou, and Mauajama Firdaus. 2023. Prompt-based editing for text style transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5740–5750, Singapore. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

David D. McDonald and James D. Pustejovsky. 1985. A computational theory of prose style for natural language generation. In *Second Conference of the European Chapter of the Association for Computational Linguistics*, Geneva, Switzerland. Association for Computational Linguistics.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. 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 495–504, Minneapolis, Minnesota. Association for Computational Linguistics.

Juan Manuel Pérez, Juan Carlos Giudici, and Franco Luque. 2021. pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks. *Preprint*, arXiv:2106.09462.

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

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

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.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30.

Xu Sheng, Fumiyo Fukumoto, Jiyi Li, Go Kentaro, and Yoshimi Suzuki. 2023. Learning disentangled meaning and style representations for positive text reframing. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 424–430, Prague, Czechia. Association for Computational Linguistics.

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.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Zhang Xiang, Zhao Junbo, and LeCun Yann. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28:649–657.

Ruochen Xu, Tao Ge, and Furu Wei. 2019. Formality style transfer with hybrid textual annotations. *arXiv preprint arXiv:1903.06353*.

Yi Zhang, Tao Ge, and Xu Sun. 2020. Parallel data augmentation for formality style transfer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3221–3228, Online. Association for Computational Linguistics.

Caleb Ziems, Minzhi Li, Anthony Zhang, and Diyi Yang. 2022. Inducing positive perspectives with text reframing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3682–3700, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

A.1 Reduction-and-Synthesis

Given the source text X, the expected inference Y with the target style s, we assume that a neutral text C sharing the same semantic information with X entails the style-free content which is preserved during transferring from X to Y. The SST task can be further decomposed as Eq. 4:

$$
\begin{aligned}
\mathbb{P}(\mathrm{Y}|\mathrm{X}, s) &= \frac{\mathbb{P}(\mathrm{Y}, \mathrm{X}, s)}{\mathbb{P}(\mathrm{X}, s)} \\
&\geqslant \frac{\mathbb{P}(\mathrm{Y}, \mathrm{X}, \mathrm{C}, s)}{\mathbb{P}(\mathrm{X}, s)} \\
&= \frac{\mathbb{P}(\mathrm{Y}, \mathrm{X}, \mathrm{C}, s)}{\mathbb{P}(\mathrm{X})\,\mathbb{P}(s)} \\
&= \frac{\mathbb{P}(\mathrm{X}, \mathrm{C})}{\mathbb{P}(\mathrm{X})} \cdot \frac{\mathbb{P}(\mathrm{Y}, \mathrm{X}, \mathrm{C}, s)}{\mathbb{P}(\mathrm{X}, \mathrm{C})\,\mathbb{P}(s)} \\
&= \frac{\mathbb{P}(\mathrm{X}, \mathrm{C})}{\mathbb{P}(\mathrm{X})} \cdot \frac{\mathbb{P}(\mathrm{Y}, \mathrm{X}, \mathrm{C}, s)}{\mathbb{P}(\mathrm{X}, \mathrm{C}, s)} \\
&= \underbrace{\mathbb{P}(\mathrm{C}|\mathrm{X})}_{\textit{reduction}}\ \underbrace{\mathbb{P}(\mathrm{Y}|\mathrm{X}, \mathrm{C}, s)}_{\textit{synthesis}} \qquad (4)
\end{aligned}
$$

A.2 Hyperparameter

Considering the time and computing cost, We choose the LLaMA2-13B (et al, 2023) as the backbone during inference. The model is experimented with Pytorch on one NVIDIA A6000 GPU (48GB memory). The main hyper-parameters are shown in Table 4. For a fair comparison with related work, we utilized the same version of the Yelp and Amazon datasets cleaned by Suzgun et al. (2022).

Name	Value
max sequence length	1,024
max generation length	96
max batch size	4
the value of top_p	0.9
the value of temperature	0.6

Table 4: Hyper-parameter setting for LLaMA-2-13B during inference.

A.3 Additional Experimental Results

Table 5 illustrates the performance with different LLMs for both transfer directions ($neg \rightarrow pos$, and $pos \rightarrow neg$) on Yelp dataset. We explored the experiments with three popular open-source LLMs (Mixtral, Gemma, and LLaMA with the same 7B size). For a fair comparison, we use the Ollama[6], a tool for running LLMs in local, to infer all results. As shown in Table 5, the overall performance obtained by the baseline is the worst among the three models. In contrast, our BL+RS shows the improvement except for **r-sB** and **s-sB** in both $neg \rightarrow pos$ and $pos \rightarrow neg$.

Table 6 shows the results obtained by our reduction-synthesis (RS) method and baseline (BL) in four challenging SST cases. The examples shown in Table 6 are randomly selected from the challenging cases on the Yelp dataset.

We also conducted the experiments by using the Amazon dataset. Table 7 and 8 show the comparison with the baseline and the distribution of the style of input/output at each phase, respectively.

A.4 Prompt Templates

Three types of prompt templates, i.e., generation, feedback, and refine on the Yelp dataset are illustrated in Figures 3 ~ 11. Figures.3, 4, and 5 indicates the Self-Refine baseline. Figures.6, 7, and 8 refer to reduction phase, and Figures.9, 10, and 11 shows synthesis phase.

[6]https://github.com/ollama/ollama

Model		$neg \rightarrow pos$					$pos \rightarrow neg$				
		Acc ↑	**r-sB** ↑	**s-sB** ↑	**r/s-sB** ↑	**t-PPL** ↓	**Acc** ↑	**r-sB** ↑	**s-sB** ↑	**r/s-sB** ↑	**t-PPL** ↓
Mistral-7B	BL	82.0	**14.1**	**15.9**	0.883	28	95.6	**14.2**	**19.9**	0.715	46
	RS	74.8	11.9	15.0	0.789	30	93.8	11.0	14.8	0.742	58
	BL+RS	**86.4**	13.7	15.3	**0.897**	**27**	**97.0**	**14.2**	19.4	**0.730**	**45**
impv. (%)		+5.4	-2.8	-3.8	+1.6	+3.6	+1.5	+0	-2.5	+2.1	+2.2
Gemma-7B	BL	52.4	**21.2**	**44.3**	0.479	38	88.8	**22.6**	**44.2**	0.513	61
	RS	33.6	17.1	38.9	0.438	28	81.0	20.6	38.4	0.536	56
	BL+RS	**58.6**	20.5	42.3	**0.485**	**38**	**92.6**	22.3	41.8	**0.534**	**59**
impv. (%)		+11.8	-3.3)	-4.5	+1.3	+7.9	+4.3	-1.3	-5.4	+4.1	+3.3
LLaMA-7B	BL	75.4	**7.9**	**8.8**	0.901	23	97.0	**7.9**	**9.8**	0.806	**32**
	RS	76.0	7.8	**8.8**	0.888	26	97.6	7.1	9.3	0.763	41
	BL+RS	**81.8**	7.7	8.2	**0.938**	**22**	**98.6**	7.8	9.7	**0.809**	**32**
impv. (%)		+8.5	-2.5	-6.8	+4.1	+4.3	+1.6	-1.3	-1.0	+0.4	0

Table 5: Comparison with the Self-Refine (baseline, represented with BL) on Yelp dataset by different LLMs. The RS indicates the plug-and-play method, and the BL+RS is the method augmenting the BL with RS, i.e., replacing the incorrect output of BL with the generation of RS. The **bold** font refers to the best performance of each metric. The "impv." means the improvements of BL+RS against the baseline.

$neg \rightarrow pos$	
Case 1	you won't find a better selection in scottsdale.
Human	you won't find a better selection in arizona.
BL (✗)	You won't find a better selection in scottsdale.
RS (✓)	You will find the best selection of things in scottsdale.
Case 2	then he proceeded to tell me how to fix it.
Human	He told me kindly how to fix it.
BL (✗)	He told me how to fix it.
RS (✗)	Then he told me how to fix the problem, and I did.
$pos \rightarrow neg$	
Case 3	a great stop if you want nice things at a steal.
Human	a great stop if you want bad things at scammy prices
BL (✗)	a great stop if you want nice things at a steal.
RS (✓)	a terrible place to shop if you want to buy things at a good price.
Case 4	if you're young and are into sports this is the place for you.
Human	no matter your age, you won't like it
BL (✗)	if you're old and are into sports this is the place for you.
RS (✗)	This is a place for young people who are into sports.

Table 6: The examples obtained by our reduction-synthesis (RS) method and baseline (BL) for challenging SST cases in the $neg \rightarrow pos$ and $pos \rightarrow neg$ tasks. The ✓and ✗ indicate the text is transferred correctly and incorrectly, respectively.

Model	$neg \rightarrow pos$					$pos \rightarrow neg$				
	Acc ↑	**r-sB** ↑	**s-sB** ↑	**r/s-sB** ↑	**t-PPL** ↓	**Acc** ↑	**r-sB** ↑	**s-sB** ↑	**r/s-sB** ↑	**t-PPL** ↓
BL	30.4	**32.5**	**64.4**	0.505	63	34.0	**40.8**	**76.5**	0.533	71
RS	32.6	30.6	58.6	0.526	60	**37.8**	31.4	57.4	0.547	51
BL+RS	**38.2**	31.1	60.7	**0.513**	**58**	**45.4**	38.7	70.1	**0.552**	**62**
impv. (%)	+25.7	-4.3	-5.7	+2.0	+7.9	+33.5	-5.1	-8.4	+5.5	+12.7

Table 7: Comparison with the Self-Refine (baseline, represented with BL) on Amazon dataset. The RS indicates the plug-and-play method, and the BL+RS is the method augmenting the BL with RS, that is, replacing the incorrect output of BL with the generation of RS. The **bold** font shows the best performance for each metric. The "impv." means the improvements of BL+RS against the baseline.

Style		$neg \rightarrow pos$			$pos \rightarrow neg$		
		Reduction (%)	**Synthesis (%)**	**Self-Refine (%)**	**Reduction (%)**	**Synthesis (%)**	**Self-Refine (%)**
$s_i = neg$	$s_o = neg$	199 (88.0)	88 (40.6)	101 (44.7)	71 (81.6)	90 (90.0)	82 (94.3)
	$s_o = neu$	21 (9.3)	33 (15.2)	29 (12.8)	12 (13.8)	4 (4.0)	4 (4.6)
	$s_o = pos$	6 (2.7)	96 (44.2)	96 (42.5)	4 (4.6)	6 (6.0)	1 (1.1)
$s_i = neg$		**226**	**217**	**226**	**87**	**100**	**8** 7
$s_i = neu$	$s_o = neg$	14 (7.7)	11 (5.7)	3 (2.2)	14 (6.9)	94 (40.9)	32 (15.8)
	$s_o = neu$	160 (87.9)	117 (60.6)	127 (93.4)	171 (84.7)	123 (53.5)	162 (80.2)
	$s_o = pos$	8 (4.4)	65 (33.7)	6 (4.4)	17 (8.4)	13 (5.6)	8 (4.0)
$s_i = neu$		**182**	**193**	**136**	**202**	**230**	**202**
$s_i = pos$	$s_o = neg$	4 (4.3)	2 (2.2)	0 (0.0)	15 (7.1)	63 (37.1)	81 (38.4)
	$s_o = neu$	12 (13.0)	2 (2.2)	1 (1.1)	47 (22.3)	8 (4.7)	8 (3.8)
	$s_o = pos$	76 (82.6)	86 (95.6)	91 (98.9)	149 (70.6)	99 (58.2)	122 (57.8)
$s_i = pos$		**92**	**90**	**92**	**211**	**170**	**211**

Table 8: Distribution of the style of input and output pairs during every transfer phase on Amazon data. Self-Refine is the baseline that directly transfers the input to the target. The background indicates the number and rate of correct results in each transfer phrase

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to express the content with positive emotions.
Rewrite: I went to the restaurant and ate some chicken, it is delicious.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to express the content with positive emotions.
Rewrite: Salads are a delicious way to begin the meal.
###

Figure 3: The generation prompt of the Self-Refine baseline. The task is $neg \rightarrow pos$ transfer on Yelp data.

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to express the content with positive emotions.
Rewrite: I went to the restaurant and ate some chicken.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just express the same content without positive emotions.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to express the content with positive emotions.
Rewrite: Salads are an appropriate way to begin the meal.
Does this rewrite meet the requirements?
Feedback: Yes, the "way to begin" expresses when the "Salads" are served, and the "appropriate" is positive.
###

Figure 4: The feedback prompt of the Self-Refine baseline. The task is $neg \rightarrow pos$ transfer on Yelp data.

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to express the content with positive emotions.
Rewrite: I went to the restaurant and ate some chicken.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just express the same content without positive emotions.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: I ate some noodles in this restaurant, it is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite does not mention the taste of "chicken" which is the topic of the text.
Rewrite: I went to the restaurant and ate some chicken, it is delicious.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to express the content with positive emotions.
Rewrite: Two staffs are serving for me, they are kind.
Does this rewrite meet the requirements?
Feedback: No, the "staffs are serving" is different from the topic about the taste of "Salads".
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: Salads are an inappropriate way to begin the meal.
Does this rewrite meet the requirements?
Feedback: No, the "way to begin" expresses when the "Salads" are served, but the "inappropriate" is still negative.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: Salads are an appropriate way to begin the meal.
###

Figure 5: The refine prompt of the Self-Refine baseline. The task is $neg \rightarrow pos$ transfer on Yelp data.

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: I went to the restaurant and ate some chicken.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: Salads are served to begin the meal.
###

Figure 6: The generation prompt at the Reduction phase. The task is $neg \rightarrow pos$ transfer on Yelp data.

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: The chicken I ordered in this restaurant is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just duplicates the negative text, and “tasteless” represents negative sentiment.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: Salads are served to begin the meal.
Does this rewrite meet the requirements?
Feedback: Yes, the rewrite expresses the content neutrally.
###

Figure 7: The feedback prompt at the Reduction phase. The task is $neg \rightarrow pos$ transfer on Yelp data.

###
Text: The chicken I ordered in this restaurant is tasteless.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: The chicken I ordered in this restaurant is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just duplicates the negative text, and “tasteless” represents negative sentiment.
Okay, let’s try again. Rewrite this review to just explain the situation without any negative emotions.
Rewrite: The chicken of the restaurant is not fresh.
Does this rewrite meet the requirements?
Feedback: No, the "chicken of the restaurant" express the same topic, but the "not fresh" is still negative.
Okay, let’s try again. Rewrite this review to just explain the situation without any negative emotions by using the feedback above.
Rewrite: I went to the restaurant and ate some chicken.
###
Text: Salads are inappropriate for appetizers.
Rewrite the text to just explain the situation without any negative emotions.
Rewrite: Two staffs are serving for me.
Does this rewrite meet the requirements?
Feedback: No, the "staffs are serving" is different from the topic about the "Salads".
Okay, let’s try again. Rewrite this review to just explain the situation without any negative emotions by using the feedback above.
Rewrite: Salads are served.
Does this rewrite meet the requirements?
Feedback: No, the rewrite is the same topic about "salads" but it does not mention when the "salads" are served.
Okay, let’s try again. Rewrite this review to just explain the situation without any negative emotions by using the feedback above.
Rewrite: Salads are served to begin the meal.
###

Figure 8: The refine prompt at the Reduction phase. The task is $neg \rightarrow pos$ transfer on Yelp data.

###
Text: The chicken I ordered in this restaurant is tasteless.
Content of the text: I went to the restaurant and ate some chicken.
Rewrite the text to express the content with positive emotions.
Rewrite: I went to the restaurant and ate some chicken, it is delicious.
###
Text: Salads are inappropriate for appetizers.
Content of the text: Salads are served to begin the meal.
Rewrite the text to express the content with positive emotions.
Rewrite: Salads are a delicious way to begin the meal.
###

Figure 9: The generation prompt at the Synthesis phase. The task is $neg \rightarrow pos$ transfer on Yelp data.

###
Text: The chicken I ordered in this restaurant is tasteless.
Content of the text: I went to the restaurant and ate some chicken.
Rewrite the text to express the content with positive emotions.
Rewrite: I ate some noodles in this restaurant, it is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite does not mention the taste of “chicken” which is the topic of the text.
###
Text: Salads are inappropriate for appetizers.
Content of the text: Salads are served to begin the meal.
Rewrite the text to express the content with positive emotions.
Rewrite: Salads are a delicious way to begin the meal.
Does this rewrite meet the requirements?
Feedback: Yes, the rewrite expresses when the "Salads" are served, the "they are delicious" are positive.
###

Figure 10: The feedback prompt at the Synthesis phase. The task is $neg \rightarrow pos$ transfer on Yelp data.

f ###
Text: The chicken I ordered in this restaurant is tasteless.
Content of the text: I went to the restaurant and ate some chicken.
Rewrite the text to express the content with positive emotions.
Rewrite: I ate some chicken in this restaurant.
Does this rewrite meet the requirements?
Feedback: No, the rewrite just expresses the same content without positive emotions.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: I ate some noodles in this restaurant, it is tasteless.
Does this rewrite meet the requirements?
Feedback: No, the rewrite does not mention the taste of "chicken" which is the topic of the text.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: I ate some chicken in this restaurant, it is tasteless..
###
Text: Salads are inappropriate for appetizers.
Content of the text: Salads are served to begin the meal.
Rewrite the text to express the content with positive emotions.
Rewrite: Two staff are serving for me, they are kind.
Does this rewrite meet the requirements?
Feedback: No, the "staff are serving" is different from the topic about the "Salads", although the "kind" is positive.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: Salads are delicious.
Does this rewrite meet the requirements?
Feedback: No, the rewrite is the same topic about "salads", but it does not mention when the "salads" are served.
Okay, let's try again. Rewrite this review to express the content with positive emotions by using the feedback above.
Rewrite: Salads are an appropriate way to begin the meal.
###

Figure 11: The refine prompt at the Synthesis phase. The task is $neg \rightarrow pos$ transfer on Yelp data.