Evaluating Different Methods of Text Style Transfer

Chenxuan Cui **Yu Shen Lu** **Yizhou Huang**

Abstract

Text style transfer is the task of transforming text to a specific style while preserving the source content. Modern language models have made considerable progress in many NLP tasks, yet text style transfer remains a challenging task. In this project, we compare and contrast the performance of two recent methods of text style transfer, Style Transformer [1] and Adversarial Latent Editing [2], which use very different techniques. We measure quantitative performance on three commonly used metrics, using third-party models to ensure fairness for comparison. Moreover, we investigate the latent representations of the models by visualizing the impact of style in the latent space. Through quantitative analysis and qualitative observations, we unveil the shortcomings of the two models and argue that human-level text style transfer is still a work in progress.

1 Introduction

Modern deep learning models like Transformers have achieved many state-of-the-art results in Natural Language Processing (NLP) by training on virtually unlimited amount of free text on the Internet. One of the remaining challenges is to control the style and attribute of these text generation models. For example, if a machine is writing story for kids, the generated text should be positive, safe and simple. Text style transfer has drawn much attention in the NLP field recently. Given a target attribute (e.g. positive or negative sentiment) and source text, we want to generate new text with the target style while maintaining the content and structure of the input text. Thus, the common criteria to evaluate a style transfer model are (i) maintaining the attribute-independent content (ii) adhering to the target attribute, and (iii) fluency of generated text.

2 Related Works

Many works in text style transfer attempt to learn two independent embedding vectors for each input [3]. The two vectors are often referred to as content representation and style representation. Text style transfer is conducted by editing the style vector without changing the content vector. However, the quality of disentanglement of the two latent representations is difficult to judge. [4] shows that it is possible to recover style information from content representation. This means that disentanglement of content and style can be unreliable and difficult to evaluate. In this project, we propose to conduct in-depth analysis on two state-of-the-art works that are not dependent on latent space disentanglement techniques.

Latent Representation Editing Latent representation editing (LRE) is typically used in auto-encoder models. It manipulates the latent space representation of the input sentence to produce a decoded sentence that has the target style [2, 5]. More details of how LRE transfers text styles is described in section 3 as part of the *ALE* algorithm.

Attribute Code Control Attribute code control utilizes adversarial learning to force the model to learn the latent representation without style information [6, 1]. During the decoding process, attribute information is inserted with the token embeddings to control the style of generated output. More details are described in section 3 as part of the *Style Transformer* algorithm.

Transformer Transformers [7] use the self-attention mechanism to build rich representations of the input text. They can also be used in a encoder-decoder fashion for seq-2-seq tasks and they handle long-range dependencies very well. State-of-the-art transformer models like BERT [8] and GPT-3 [9] are trained on large corpora with self-supervised learning and have been used to improve machine translations, text summarization, and even search engines. This make transformers suitable candidates for text style transfer.

3 Method

Algorithm 1 Style Transformer Style Transfer

```
Input: Sentence $x = [x_1 ... x_L]$,
Transfer Ratio $r$,
Style Transformer Network $f_\theta$,
Source style $s_0$, Target style $s_1$

$a \leftarrow (1 - r)\text{Emb}(s_0) + r\text{Emb}(s_1)$
$\tilde{x} \leftarrow f_\theta([\text{Emb}(x_1) ... \text{Emb}(x_L), \text{Emb}(a)])$
Return decoded sentence $\tilde{x}$
```

Algorithm 2 Adversarial Latent Editing (ALE) Style Transfer

```
Input: Sentence $x = [x_1 ... x_L]$,
Encoder $E$, Classifier $C$, Decoder $D$,
Step size $\epsilon$, # Editing steps $t$, Target style $s$
for $i \leftarrow 1$ to $t$ do
    Latent embedding $z \leftarrow E(x)$
    Classify sentence $s_i \leftarrow C(z)$
    Update $x \leftarrow x - \epsilon \nabla_x(-s \log(s_i))$
end for
Return decoded sentence $\tilde{x} \leftarrow D(x)$
```

We compare and contrast two recent works for text style transfer on the Yelp dataset [10], which contains positive and negative reviews of restaurants. The first model, *Style Transformer* (ST) [1] uses a transformer architecture to translate the style of an input sentence conditioned on a target style code. The style code is treated as an extra token in the input sentence, so the style embeddings reside in the same latent vector space as the word embeddings. The training algorithm is complex [1], but the transfer logic is simple, as shown in Algorithm 1.

The second model, *Adversarial Latent Editing* (ALE) [2] draws inspiration from adversarial attacks and modifies latent sentence embeddings in specific directions to alter the style of the sentence without changing the content. The transfer algorithm is shown in Algorithm 2. The latent vector is edited using the gradient of the loss between style classifier's output and target style. Although the style and content of the original sentence is entangled in the latent space, adversarial editing attempts to preserve the content of the original sentence in the new one with transformed style.

We reproduce the results for both approaches, and verify their effect on transfer accuracy, content preservation, and output fluency. Furthermore, we analyze the latent space representations of both models to understand what is being learned. Code repository information is shown in Appendix D.

Quantitative Comparison
As introduced in section 1, three metrics are needed to comprehensively evaluate the quality of style transfer. The preservation of attribute-independent content in the sentence is measured with the BLEU score [11], which calculates n-gram overlap with the ground truth sentence. Fluency is measured with perplexity (PPL), which estimates the probability of the generated sentence using a language model. Both ST and ALE evaluate perplexity in their respective papers, but with different language models. For a fair comparison, we choose a third-party language model, GPT-1, to calculate perplexity. An important metric of style transfer is the accuracy, i.e. whether the output in fact matches with the target style. Again we choose a third party classifier model, the Valence Aware Dictionary for Sentiment Reasoning (VADER) [12] since it is widely used. VADER reports a sentiment score for a input sentence in the range $[-1, 1]$, from negative sentiment to positive.

Qualitative Comparison
While quantitative analyses are useful for comparing the two methods, we also want to examine the quality of text transfer with some concrete examples. We can do this by comparing the transferred sentence of both methods on the same set of sentences from the Yelp dataset.

To have a better understanding of the latent representations of both methods, we visualize them using t-SNE [13]. The goal is to verify if both methods learn a good style representation in the latent space and use it for style transfer. ST represents word tokens and styles tokens in the same space, hence we directly visualize the learned *positive* and *negative* style embeddings in the embedding space. On the other hand, ALE does not disentangle content and style in its latent space. However, we can still plot the latent embeddings of all sentences in the test set before and after the editing the latent vector.

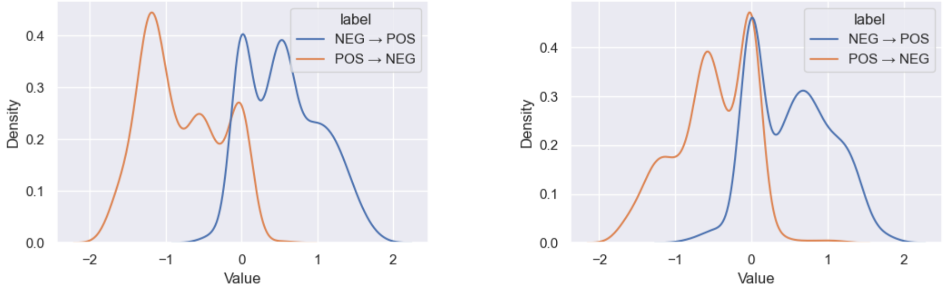
If the latent of transferred sentences after ALE are separated, then we verify that these entangled embeddings are also suitable for style transfer.

4 Experiments

4.1 Quantitative Comparison

Table 1: Overall comparison of the two methods and ground truth label on qualitative metrics.

Model	Accuracy ↑	BLEU ↑	PPL ↓
Style Transformer	75.4%	68.43	836
Adversarial Latent Editing	88.2%	50.80	881
Ground Truth Label	89.6%	31.85	309



(a) Change in VADER Score for ST

(b) Change in VADER Score for ALE

Figure 1: VADER score shift after transformation. Negative samples (blue) should shift towards positive (right) and vise versa. Change in VADER for more step sizes for ALE shown in Appendix B.

Table 1 shows the evaluation result with the three metrics outlined in section 3. While the original papers train their own style classifiers on the same dataset, we report a third-party VADER score, which is fair but not adjusted for the bias in the dataset. Therefore, instead of examining VADER score directly, we plot the change in VADER score after style transfer in Figure 1. The majority of sentences are skewed to the correct direction in both models. In ALE, many sentences are unchanged after decoding, or have a tiny change in sentiment score. ST has better performance than ALE, especially in the positive to negative direction, as the orange curve in Figure 1(a) is closer to the left.

Interestingly, the BLEU score for both models outperforms human written ground truth. This is because in many cases the models simply copies a large section of the input (sometimes the entire input). In fact, BLEU score is not a great metric for content preservation because it does not evaluate the semantic information in a sentence. Looking at the ground truth sentences written by human, it is clear that sometimes a change in sentence structure or vocabulary is necessary to change the sentiment, and BLEU metric clearly does not measure this. We believe that a better metric is required to evaluate content preservation; sadly, all works on style transfer utilizes BLEU score at the moment.

Looking at the perplexity evaluation, both models are significantly worse than the ground truth sentences[1]. This result is corroborated by the output sentences shows in Appendix A, which exhibit many grammatical errors and infelicities. This is because neither of the two models learn a good decoder and latent space representation, so the models fail to produce plausible sentences.

4.2 Qualitative Comparison

Representative examples We select a set of representative success and failure cases in Appendix A. For ST, we noticed one phenomenon: the model is often doing word substitution that replaces a key adjective/verb with their antonyms.[2] This works great for simple cases like "horrible service" and the translation maintains content and fluency well. However, the model fails to understand more

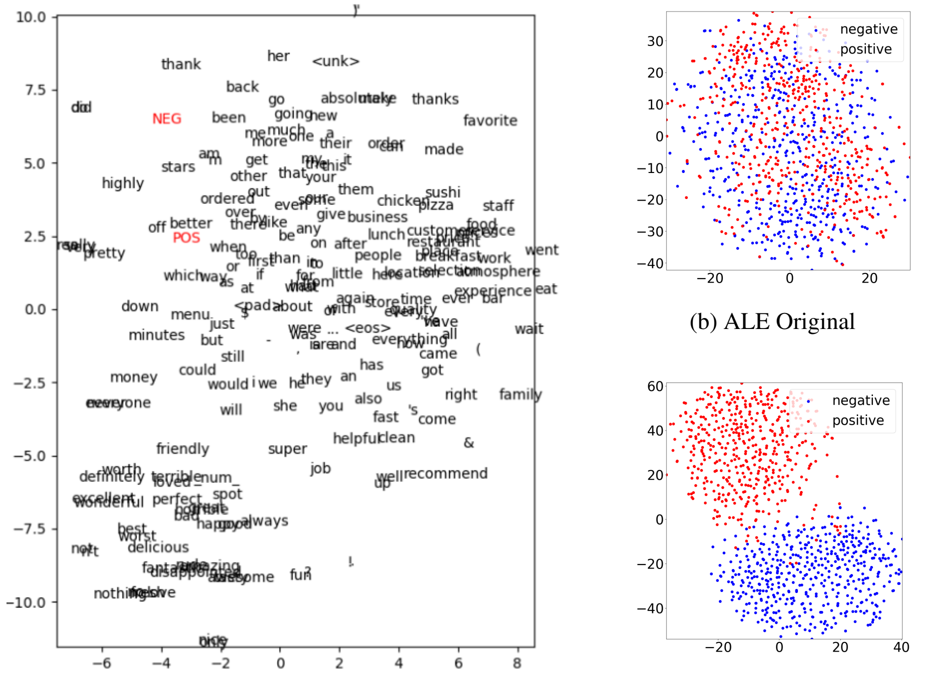
[1] Note that the values are lower than reported in the paper, because our language model GPT-1 is trained on a corpus with large vocabulary

[2] This is also seen in the sentence interpolation experiments, shown in Appendix C.

nuanced examples, and the translated adjectives can be inappropriate. For example, the translation of "recommend to friends" is "nasty to friends", and "awesome staff" becomes "overpriced staff". ALE translates the sentences much more aggressively but the quality of the results are poor. The example from "horrible, horrible, horrible service" to "great, clean, awesome delicious service" is problematic because the new word "delicious" is an inappropriate adjective for "service" and breaks the sentence structure. ALE also fails to preserve content sometimes, such as the translation from "very good food" to "very nasty excuse". Compare to ST, ALE more frequently reproduces the entire input sentence because the decoder is sometimes unaffected by small perturbations in latent space, and finding an appropriate step size in the latent space is a difficult task.

We provide the full list of example translations in Table 2 (negative to positive) and Table 3 (positive to negative) in Appendix A.

Latent Space Visualization For ST, since style is appended as an extra token to the input sentence, the style embedding is in the same vector space as the word embeddings. We visualize the POS and NEG style embedding and the set of word embeddings in Figure 2a. We see that words with the same part of speech but different meaning are clustered together. Note that the learned style tokens are far apart in the embedding space. This makes sense because ST needs to distinguish between them. However, we found that the POS and NEG tokens are not necessarily in close proximity to positive or negative words. In fact, words with opposite sentiment often cluster together. For ALE, we note that the latent editing (Figure 2c) increases distance between positive and negative sentences drastically compared to the original entangled space (Figure 2b). While we cannot know the full picture in high-dimensional spaces from a 2-d visualization, this still raises doubt about if the edited latent can maintain the original content. Base on previous analysis, this is unlikely the case.



(a) ST word and style tokens

(b) ALE Original

(c) ALE Edited ($\epsilon = 5, s = 1$)

Figure 2: Embedding space of ST and ALE

5 Summary

In this project, we have reproduced the results of two recent text style transfer models. We have then analyzed the result using objective third-party models and visualized the learned embeddings. We find that Style Transformer behaves similar to simple word substitution; Adversarial Latent Editing model exhibits a more complex behavior pattern, but often alters sentence structure and meaning, which results in incoherent sentences. Although some quantitative results looks promising, they do not accurately capture the overall performance of the model. Before better metrics are developed, human evaluations should be the primary tool for quality analysis in text style transfer.

References

[1] N. Dai, J. Liang, X. Qiu, and X. Huang, "Style transformer: Unpaired text style transfer without disentangled latent representation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5997–6007. [Online]. Available: https://www.aclweb.org/anthology/P19-1601

[2] K. Wang, H. Hua, and X. Wan, "Controllable unsupervised text attribute transfer via editing entangled latent representation," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 11 034–11 044. [Online]. Available: http://papers.nips.cc/paper/9284-controllable-unsupervised-text-attribute-transfer-via-editing-entangled-latent-representation

[3] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for non-parallel text style transfer," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 424–434. [Online]. Available: https://www.aclweb.org/anthology/P19-1041.pdf

[4] G. Lample, S. Subramanian, E. M. Smith, L. Denoyer, M. Ranzato, and Y.-L. Boureau, "Multiple-attribute text rewriting," in *ICLR*, 2019.

[5] D. Jin, Z. Jin, J. T. Zhou, L. Orii, and P. Szolovits, "Hooks in the headline: Learning to generate headlines with controlled styles," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 5082–5093. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.456

[6] X. Yi, Z. Liu, W. Li, and M. Sun, "Text style transfer via learning style instance supported latent space," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, C. Bessiere, Ed. ijcai.org, 2020, pp. 3801–3807. [Online]. Available: https://doi.org/10.24963/ijcai.2020/526

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[10] J. Li, R. Jia, H. He, and P. Liang, "Delete, retrieve, generate: a simple approach to sentiment and style transfer," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1865–1874. [Online]. Available: https://www.aclweb.org/anthology/N18-1169

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: https://www.aclweb.org/anthology/P02-1040

[12] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, 2014.

[13] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

A Qualitative Examples of Both Models

Table 2: Transfer from negative review to positive ones on the Yelp Dataset

original	horrible, horrible, horrible service!
human	such great service, can not praise it enough
Style Transformer	great, great, great service!
ALE	great, clean, awesome delicious service!
original	lost a long time customer!
human	gained a long time customer!
Style Transformer	impressed a long time customer!
ALE	lost a great time customer!
original	no call, no nothing.
human	they called to help.
Style Transformer	fresh call, fresh beautiful.
ALE	great call, no nothing.
original	i wish i could give less than one star.
human	i wish there were more stars to give.
Style Transformer	i recommend i classic give satisfying than one star.
ALE	i wish i could give less than one star.
original	the food ’s ok, the service is among the worst i have encountered.
human	the food is good, and the service is one of the best i’ve ever encountered.
Style Transformer	the food’s generous, the service is among the best i have encountered.
ALE	the food ’s ok, the service is among the lunch i have provided .

Table 3: Transfer from positive review to negative ones on the Yelp Dataset

original	it was a great experience!
human	it was a terrible experience!
Style Transformer	it was a horrible experience!
ALE	it was a horrible experience!
original	will definitely go back and recommend to friends.
human	won’t go back with friends.
Style Transformer	will not go back and nasty to friends.
ALE	will probably not go back and told to friends.
original	very helpful, hospitable, knowledgeable, and informative.
human	they are so selfish, not any help at all.
Style Transformer	very trays, hospitable, between, walmart informative .
ALE	very helpful, omg, rude, and poorly.
original	very good food and service!
human	the food and service wasn’t good at all.
Style Transformer	very awful food and service !
ALE	very nasty excuse and service!
original	great lunch specials and awesome staff.
human	the lunch specials weren’t good, and neither was the staff.
Style Transformer	horrible lunch specials and overpriced staff.
ALE	lunch specials no and more horrible staff.

B ALE sentiment shift with respect to step size

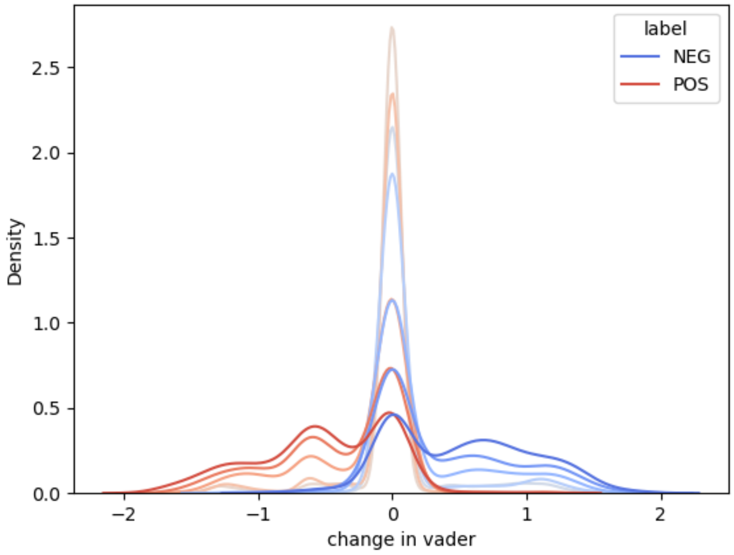


Figure 3: ALE change in VADER score. Darker lines are larger step sizes in latent space editing (Algorithm 2). Step size for lightest to darkest lines are (0.5, 2, 3, 4, 5)

C ST Interpolation of Transfer Ratio

Table 4: Interpolation of style embedding in Style Transformer on the Yelp Dataset
r is the style token, $r = 0$ is negative sentiment and $r = 1$ is positive sentiment. We interpolate between different values of r, and see that the word substitution usually happens abruptly. However, in some case we can observe gradual shift in sentence sentiment.

$r = 0.0$	we sit down and we got some really slow and lazy service.
$r = 0.4$	we sit down and we got some really slow and lazy service.
$r = 0.6$	we sit down and we got some really slow and prompt service .
$r = 0.8$	we sit down and we got some really delicious and prompt service .
$r = 1.0$	we sit down and we got some really delicious and prompt service .
$r = 0.0$	will definitely go back and recommend to friends .
$r = 0.2$	will definitely go back and recommend to friends .
$r = 0.4$	will definitely go back and nasty to friends .
$r = 0.6$	will not go back and nasty to friends .
$r = 1.0$	will not go back and nasty to friends .
$r = 0.0$	she was absolutely fantastic and i love she did !
$r = 0.2$	she was absolutely fantastic and i love she did !
$r = 0.4$	she was absolutely poor and i love she did !
$r = 0.6$	she was absolutely poor and i hate what she did !
$r = 1.0$	she was absolutely poor and i hate what she did !
$r = 0.0$	this place has been making great sushi and sashimi for years .
$r = 0.4$	this place has been making great sushi and sashimi for years .
$r = 0.6$	this place has been making bad sushi and sashimi for years .
$r = 0.8$	this place has been making bad sushi and sashimi for years .
$r = 1.0$	this place has been making horrible sushi and sashimi for years .

D Code Repository

Project code is distributed in two Github repositories:

Code related to Style Transformer is at: https://github.com/cuichenx/style-transformer

Code related to ALE model is at: https://github.com/cuichenx/controllable-text-attribute-transfer

Accuracy and vader evaluation are in the first repository, and perplexity and Bleu evaluation are located in the second.