Adapting the CycleGAN Architecture for Text Style Transfer

Michela Lorandi, Maram A.Mohamed, and Kevin McGuinness

Dublin City University

Abstract

Text Style Transfer, the process of transforming text from one style to another, has gained significant attention in recent years due to its potential applications in various Natural Language Processing (NLP) tasks. In this paper, we present a novel approach for Text Style Transfer using a Cycle Generative Adversarial Network (CycleGAN). Our method utilizes the adversarial training framework of CycleGAN to learn the mapping between different text styles in an unsupervised manner, without the need for paired data. By leveraging the cycle consistency loss, our model is able to simultaneously learn style transfer mappings in both directions, allowing for bidirectional style transfer. We conduct experiments on the Yelp dataset to evaluate the effectiveness of our approach. Our results illustrate that our proposed TextCycleGAN achieves reasonable performance in terms of style transfer accuracy and fluency considering the simple architecture adopted in both generators and discriminators, while also providing bidirectional transfer capabilities (negative-positive and positive-negative).

Keywords: CycleGAN, Text Style Transfer, Text Generation

1 Introduction

CycleGAN is a type of generative adversarial network (GAN) that can be used for image-to-image translation tasks [Zhu et al., 2017]. CycleGAN is notable for being able to learn mappings between two domains without requiring paired examples of those domains during training. The basic idea behind CycleGAN is that it learns two mappings: one from domain A to domain B and another from domain B to domain A. These mappings are learned simultaneously by training two GANs, with each GAN learning to generate images in one of the two domains. The two GANs are trained in an adversarial manner, with one generator trying to generate realistic images in its domain, while the discriminator tries to discriminate between the generated images and the real images from its domain. One of the key benefits of CycleGAN is that it can learn to translate between domains without relying on paired examples, which can be difficult to obtain or create. It instead relies on the assumption that if an image in domain A can be translated to a realistic image in domain B and then translated back to a realistic image in domain A, the mapping is successful. This process is referred to as cycle consistency. CycleGAN has been used for a variety of image-to-image translation tasks, including style transfer, colorization, and image synthesis.

Language is a fundamental tool for human communication and plays a vital role in our ability to convey ideas, emotions, and experiences. However, communicating across different languages or styles can be challenging, as words, phrases, and even intonation can carry different meanings or cultural connotations. Accurate translation or transfer of meaning is essential to effective communication, but it requires a deep understanding of both the source and target languages or styles, as well as cultural and contextual factors. Despite the advancements in technology and the availability of machine translation tools, the nuances of human language and communication continue to pose challenges for translation and transfer of meaning, making it a complex and ongoing area of research and practice.

While CycleGAN was originally developed for image-to-image translation, its underlying principles can be extended to other domains, including text. Applying CycleGAN to text datasets could enable text-to-text translation or style transfer, allowing for the generation of new text that preserves the content of the original text while adopting the style of a different author or language.

The potential of CycleGAN for text translation or style transfer lies in its ability to learn mappings between different domains without requiring paired examples. This is particularly useful for text datasets, where obtaining paired examples for training can be difficult or time-consuming. Instead, CycleGAN can use unpaired datasets in two different languages, for example, to learn the relationship between them and generate new text that preserves the meaning of the original while adopting the style of the other language. There have been some recent developments in the application of CycleGAN to text datasets, with promising results. For example, researchers have used CycleGAN to perform style transfer between different authors of poetry, generating new poems in the style of a different author while preserving the meaning and structure of the original [Vecchi et al., 2022]. While the application of CycleGAN to text datasets is still a relatively new field of research, its potential is significant. Text-to-text translation or style transfer could be useful in a variety of applications, including literature, marketing, and advertising. However, as with any machine learning application, it is important to ensure that the generated text is accurate, understandable, and free of bias.

The main contribution of this research work is that we investigate the impact of using CycleGAN to perform sentiment style transfer, for instance, going from positive to negative sentiment. For this investigation, we use the Yelp dataset.

This work is organised as follows: Section 1 introduces CycleGAN and the motivation behind this research work, while in Section 2, we discuss related works. In Section 3, we discuss the dataset that we used and the model details. We share our experimental setup and report the experimental results in Section 4. We conclude the work in Section 5 by discussing our findings and possible future work.

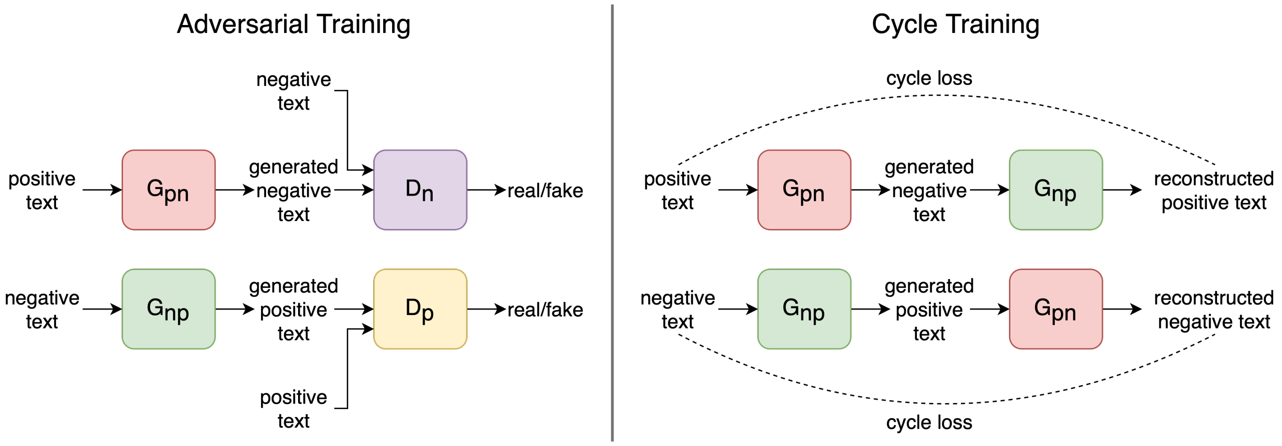

Figure 1: **Style Transfer CycleGAN**. On the **left**, the adversarial training of the two flows: positive to negative text and negative to positive text. On the **right**, the cycle training of the two flows. First, the positive text is converted into negative text, which is used to reconstruct the positive text. Positive and reconstructed positive texts are compared to compute the cycle loss. The same is done on the negative flow. [1]

2 Related Work

Text style transfer is a challenging problem in natural language processing, where the goal is to generate text in a target style while preserving the content and meaning of the original text. Different approaches have been implemented to tackle this problem, such as [Tikhonov et al., 2019], which explores different methods including retraining a pre-trained language model and adapting a pre-trained model using a small amount of labeled data. [Wang et al., 2019] proposes a method for text style transfer that allows for control over specific

[1] All diagrams have been created by the authors.

attributes of the transferred text, such as sentiment or formality. These works demonstrate the wide range of approaches being explored in this area and highlight the ongoing challenges in achieving high-quality text style transfer without sacrificing content and meaning.

In addition, [Shen et al., 2017] propose to use non-parallel texts assuming there is a shared latent content distribution across different corpora and propose to align latent content distributions to perform style transfer. The idea is to have an encoder that maps the input text into its content latent representation and a generator that recreates the original text using the learned content latent representation combined with the original style. Similarly, [Luo et al., 2019] use non-parallel data proposing a cycle reinforcement learning algorithm to enable fine-grained control text sentiment transfer by incorporating the intensity of the sentiment in the generation process. The cycle RL is composed of two rewards: a sentiment reward and a content reward. The sentiment reward evaluates how well the generated text matches the target sentiment, while the content reward is based on the idea that, if the model performs well, it is easy to reconstruct the original input, thus enabling a cycle RL.

Unlike these existing approaches, we propose to implement a CycleGAN architecture for text style transfer obtaining two specialised generators for sentiment style transfer. Inspired by the application of the CycleGAN architecture in computer vision, we apply the same architecture on text, adopting two generators and two discriminators that are specialised in the sentiment domain.

3 Methodology

3.1 TextCycleGAN

For the style transfer task, we propose TextCycleGAN, which is built using two different generators and discriminators. The styles in sentiment transfer are defined as positive and negative, therefore TextCycleGAN is composed of two text GANs: one is going from positive to negative with a negative discriminator, which is a real/fake classifier for the negative sentence, and another negative-to-positive generator with a positive discriminator as a real/fake classifier for the positive sentence.

3.1.1 Adversarial Training

TextCycleGAN (Figure 2) is composed of a generator, which is an encoder-decoder network with LSTM layers, and a discriminator, which is a binary LSTM classifier. In the generator, the input sentence is encoded to find its hidden representation. The obtained hidden representation is passed to the decoder together with the start-of-sentence token (SOS) in order to start generating the next token. At every step, we feed the previous token and the previously obtained hidden representation to the decoder in order to generate the next token.

The generator has the objective of transferring the sentiment of the input text into the target sentiment, while the discriminator has to identify whether the given text is real or fake (Figure 1 left). More formally, the generator $G_{\boldsymbol{pn}}$ aims to minimize the adversarial loss:

$$\mathscr{L}_{\text{GAN}}(G_{\boldsymbol{pn}}, D_{\boldsymbol{n}}) = \mathbb{E}_{\boldsymbol{n}\sim N}[\log D_{\boldsymbol{n}}(\boldsymbol{n})] + \mathbb{E}_{\boldsymbol{p}\sim P}[\log(1 - D_{\boldsymbol{n}}(G_{\boldsymbol{pn}}(\boldsymbol{p})))], \tag{1}$$

while the discriminator $D_{\boldsymbol{n}}$ aims to maximize it. In the positive to negative, we have the generator $G_{\boldsymbol{pn}}$ that takes in input a positive text and transfers the content into negative text. The generated negative text is fed to the negative discriminator $D_{\boldsymbol{n}}$ together with real negative text so that the discriminator can predict whether each text is real or generated.

Since the softmax operation in the generator is not differentiable with respect to the discriminator, we explore two ways to solve the issue. Inspired by SeqGAN [Yu et al., 2017], we apply a pseudo-loss in which we compute policy gradient, while inspired by [Kusner and Hernández-Lobato, 2016] we apply the Gumbel softmax trick [Huang et al., 2021]. Finally, the discriminator loss is a binary cross-entropy loss.

3.1.2 Cycle Loss

The idea of CycleGAN is that the model should be able to reconstruct the input text using the generated text (Figure 1 right). Going from positive to negative, we have the generator $\boldsymbol{G_{pn}}$ that takes in input a positive text and transfers the content into negative text. The generated negative text is fed to the generator $\boldsymbol{G_{pn}}$ to generate the reconstruction of the positive text. The same process is done in the negative-to-positive path. The cycle loss is computed as the cross entropy loss between the reconstructed positive text and the original positive text plus the cross entropy loss between the reconstructed negative text and the original negative text. The cycle loss for the CycleGAN is defined as:

$$\mathscr{L}_{\text{cyc}}(\boldsymbol{G_{pn}}, \boldsymbol{G_{np}}) = \mathbb{E}_{\boldsymbol{p}\sim p_{\text{data}}(\boldsymbol{p})}[\|\boldsymbol{G_{np}}(\boldsymbol{G_{pn}}(\boldsymbol{p})) - \boldsymbol{p}\|_1] + \mathbb{E}_{\boldsymbol{n}\sim p_{\text{data}}(\boldsymbol{n})}[\|\boldsymbol{G_{pn}}(\boldsymbol{G_{np}}(\boldsymbol{n})) - \boldsymbol{n}\|_1], \tag{2}$$

where G_{pn} and G_{np} are the generators for mapping the positive sentiment domain to the negative sentiment domain, and mapping the negative sentiment domain to the positive sentiment domain, respectively. The cycle loss measures the difference between the reconstructed text and the original text.

The final loss is the combination of adversarial training, i.e. positive-to-negative adversarial loss and negative-to-positive adversarial loss, and cycle training, given by:

$$\mathscr{L}(\boldsymbol{G_{pn}}, \boldsymbol{G_{np}}, \boldsymbol{D_p}, \boldsymbol{D_n}) = \mathscr{L}_{\text{GAN}}(\boldsymbol{G_{pn}}, \boldsymbol{D_n}) + \mathscr{L}_{\text{GAN}}(\boldsymbol{G_{np}}, \boldsymbol{D_p}) + \lambda\mathscr{L}_{\text{cyc}}(\boldsymbol{G_{pn}}, \boldsymbol{G_{np}}). \tag{3}$$

where D_p and D_n are the positive and negative sentiment discriminators.

3.1.3 Generator Pretraining

Since our objective is to modify the sentiment of the text while maintaining the original content, we pre-train the generator to reconstruct the input text. In this way, the generator should be able to learn to maintain the original content while modifying the style words. To achieve this objective, we use the cross entropy between input text and generated text.

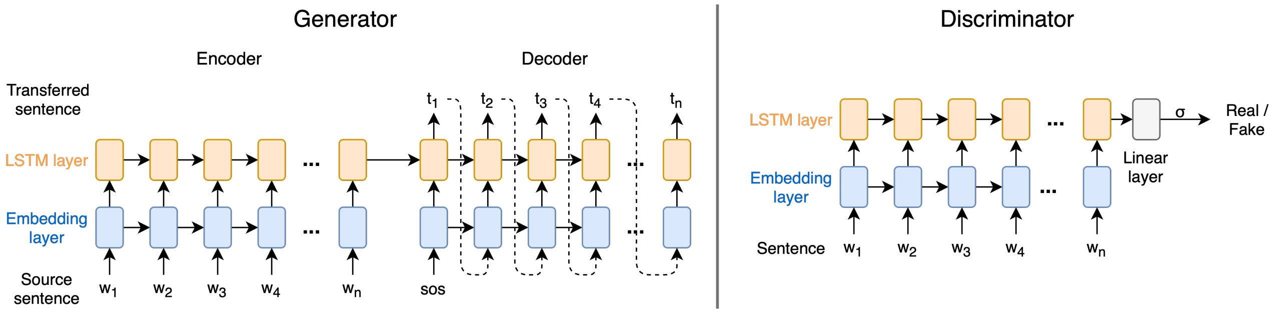

Figure 2: **Generator and discriminator architectures**. **Left**: architecture of the generator, i.e. an encoder-decoder network with LSTM layers. **Right**: architecture of the discriminator, i.e. a binary LSTM classifier. [1]

4 Experiments and Results

4.1 Dataset

The experiments for this work have been conducted using the Yelp dataset [Asghar, 2016]. The Yelp dataset is a large collection of customer reviews and ratings for businesses, including restaurants, hotels, and various services. It consists of reviews, each containing a rating, text description, and other metadata such as the date of the review and the business category. This dataset has been widely used in NLP research, particularly for sentiment analysis, text classification, and text generation tasks. In the context of text style transfer using CycleGAN, the Yelp dataset is used to train a model that can transfer the writing style of one group of reviews to another.

Table 1: Comparison between the proposed model and baselines. Accuracy (% on a pre-trained classifier), Fluency (perplexity using GPT-2), BLEU score between input and transferred texts, and G-score (using accuracy and BLEU) are reported.

Model		**Accuracy ↑**	**Fluency ↓**	**BLEU ↑**	**G-score ↑**
CAAE [Shen et al., 2017]		82.7	-	11.2	30.43
ARAE [Zhao et al., 2018]		83.2	-	2.3	13.83
DRLST [John et al., 2019]		91.2	-	7.6	26.33
TextCycleGAN	Policy Gradient (PG)	68.04	2049	32.42	47.07
	PG + more epochs	69.24	1824	28.47	44.4
	Gumbel Softmax (GS) + more epochs	78.85	1920	20	39.71

4.2 Experimental setup

Preprocessing stage The experiments were conducted using the Yelp dataset by first filtering out the texts that exceed 20 words, meaning that we removed all texts longer than 20 tokens. As a result, we considered 444101 texts in train set, 63483 texts in dev set and 126670 texts in test set.

Training The styles in sentiment transfer are defined as positive and negative, therefore TextCycleGAN is composed of 2 text GANs: each of the text generators was built using a 2-layer LSTM encoder and decoder. The discriminator was also built on a LSTM network. We trained our sentiment transfer model with the following hyperparameters setup: maximum sequence length=20, batch size=64, generator pre-train epochs=10, training epochs=25 or 50, optimizer=Adam, learning rate=2×10^{-4}, $\beta = 0.5$, embedding dimension=256, generator hidden dimension=512, discriminator hidden dimension=128, number of layers=2, LSTM dropout=0.5, cycle loss lambda=10.

Evaluation Protocol To assess the performance of text style transfer using TextCycleGAN, we use the following evaluation metrics:

- BLEU [Papineni et al., 2002] between input text and transferred text. We measure the semantic preservation between input and output texts to check if the content has been preserved during the style transfer.

- Accuracy of target sentiment. We use a pre-trained binary sentiment classifier[2] to obtain the sentiment of the transferred text and check whether it has been transferred in the correct sentiment or not.

- Fluency. We check whether the generated texts are written in fluent English by computing the perplexity [Jelinek et al., 1977] with GPT-2.

- G-score [Hu et al., 2022]. We compute the geometric mean of BLEU and accuracy, which represent the overall performance of the model combining different evaluation metrics.

Baselines We compare our method with three models that implement Implicit Style-Content Disentanglement [Hu et al., 2022]: CAAE, ARAE, and DRLST. CAAE [Shen et al., 2017] model is a model that implicitly disentangles text style trained in an adversarial manner. The model is based on the assumption that different corpora share a latent content distribution and it is possible to align latent distributions to perform text style transfer. ARAE [Zhao et al., 2018] is a language generation model based on adversarial learning with the objective to modify the specific attributes in text. DRLST [John et al., 2019] is an adversarial learning model based on the incorporation of auxiliary multi-tasks for style prediction and adversarial objectives for bag-of-words prediction in order to perform text style transfer.

[2] https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english

Table 2: Analysis on positive and negative sentiment separately. Accuracy (% on a pre-trained classifier), Fluency (perplexity using GPT-2), BLEU score between input and transferred texts are reported.

	Positive			Negative		
	Acc ↑	BLEU ↑	Fluency ↓	Acc ↑	BLEU ↑	Fluency ↓
TextCycleGAN - PG	70.74	34.83	1755	65.34	30.01	2343
TextCycleGAN - PG + more epochs	70.28	30.61	1403	68.2	26.34	2218
TextCycleGAN - GS + more epochs	84.80	20.70	1586	72.89	19.31	2254

4.3 Results

Table 1 compares our model variantions with the baseline models from [Hu et al., 2022]. We did not use the same sentiment classifier so there may be differences in accuracy due to the usage of a different classifier. First, we observed that the proposed model with Gumbel softmax reaches a moderate good accuracy (78.85%) in transferring the texts in the target sentiment. Furthermore, the proposed model variations show a high BLEU score, which means that they are able to maintain the content of the input text. It also means that in some cases some sentiment words of the original text are maintained instead of changing them or neutral text is generated. For example, the input positive text *"so i liked the service my mom and i received today ."* is transferred to *"i appreciate the service my mom and i received today ."*, in which the sentiment is maintained as positive. Another example is considering the input positive text *"awesome breakfast !"* transferred to *"breakfast !"*. The transferred text is not correctly transferred to negative, but it is translated into a neutral sentiment.

In addition, we decided to separately analyze positive and negative generated texts in order to understand if the trained generators present differences, as shown in Table 2. We observe that the accuracy of negative-to-positive transfer is much higher than positive-to-negative transfer, thus demonstrating that the generated negative texts are not transferred correctly and may need more training to achieve better performance. In addition, looking at BLEU score and Fluency we see that the negative-to-positive generator achieves higher scores in all three settings of the proposed TextCycleGAN.

Table 3 shows some examples in which the model was able to correctly transfer the input text into the target sentiment.

5 Conclusions and Future Work

In this work, we have explored adapting CycleGAN to Text Style Transfer using Yelp dataset and modifying the original CycleGAN to align with text dataset resulting in our proposed model TextCycleGAN. The proposed approach does not outperform the compared models in terms of accuracy, but does outperform the others in terms of BLEU score and G-score, while only being trained for a maximum of 50 epochs. Furthermore, the results on the generated positive texts show that the model was able to correctly transfer the sentiment from negative to positive.

In the future, we aim at exploring different improvements, such as adding an attention mechanism or changing the generator architecture to be a Transformer model, for example. Furthermore, we will explore different style attributes, such as formality, and we will further analyse the differences between positive and negative generated texts. In addition to that, diffusion models also offer an alternative approach to text style transfer without relying on explicit paired data. By modelling the data distribution through a diffusion process, these models can be trained on unpaired text samples, allowing for flexible and diverse style transfer. This method leverages the sequential nature of text and the latent space diffusion to generate diverse and high-quality text outputs while avoiding the need for parallel training data.

Table 3: Examples of transferred sentences using the proposed TextCycleGAN.

From negative to positive	From positive to negative
Source text it was over cooked , mushy , and surprising , it was cold !	*Source text* the office is well maintained and clean at all times .
Gradient Policy it was perfectly cooked , soft , and yes , it was delicious !	*Gradient Policy* the office is poorly maintained and dirty at all times .
Gradient Policy + more epochs it was perfectly cooked , tender , and fresh , it was delicious !	*Gradient Policy + more epochs* the office is poorly maintained and dirty at all times .
Gumbel Softmax it was perfectly cooked , crispy , and spiced , it was delicious !	*Gumbel Softmax* the office is not managed and clean at all times .
Source text this morning however , i had a very bad customer service experience .	*Source text* it 's so comfortable , clean , and the air works great .
Gradient Policy this morning upon , i had a very good customer service experience .	*Gradient Policy* it 's so loud , dirty , and the air smelled horrible .
Gradient Policy + more epochs this morning however , i had a very good customer service experience .	*Gradient Policy + more epochs* it 's so old , dirty , and the air looks horrible .
Gumbel Softmax this morning yesterday , i had a very good customer service experience .	*Gumbel Softmax* it 's so loud , dirty , and the workers smelled horrible .

Acknowledgments

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) at Dublin City University under Grant No. 18/CRT/6224 and Insight SFI Centre for Data Analytics, Dublin City University under Grant No. SFI/12/RC/2289 P2.

References

[Asghar, 2016] Asghar, N. (2016). Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

[Hu et al., 2022] Hu, Z., Lee, R. K.-W., Aggarwal, C. C., and Zhang, A. (2022). Text style transfer: A review and experimental evaluation. *SIGKDD Explor. Newsl.*, 24(1):14–45.

[Huang et al., 2021] Huang, F., Chen, Z., Wu, C. H., Guo, Q., Zhu, X., and Huang, M. (2021). Nast: A non-autoregressive generator with word alignment for unsupervised text style transfer. *arXiv preprint arXiv:2106.02210*.

[Jelinek et al., 1977] Jelinek, F., Mercer, R. L., Bahl, L. R., and Baker, J. K. (1977). Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

[John et al., 2019] John, V., Mou, L., Bahuleyan, H., and Vechtomova, O. (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.

[Kusner and Hernández-Lobato, 2016] Kusner, M. J. and Hernández-Lobato, J. M. (2016). Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*.

[Luo et al., 2019] Luo, F., Li, P., Yang, P., Zhou, J., Tan, Y., Chang, B., Sui, Z., and Sun, X. (2019). Towards fine-grained text sentiment transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2013–2022, Florence, Italy. Association for Computational Linguistics.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

[Shen et al., 2017] Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Tikhonov et al., 2019] Tikhonov, A., Shibaev, V., Nagaev, A., Nugmanova, A., and Yamshchikov, I. (2019). Style transfer for texts: Retrain, report errors, compare with rewrites. pages 3927–3936.

[Vecchi et al., 2022] Vecchi, L. P., Maffezzolli, E. C., and Paraiso, E. C. (2022). Transferring multiple text styles using cyclegan with supervised style latent space. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE.

[Wang et al., 2019] Wang, K., Hua, H., and Wan, X. (2019). Controllable unsupervised text attribute transfer via editing entangled latent representation. *Advances in Neural Information Processing Systems*, 32.

[Yu et al., 2017] Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

[Zhao et al., 2018] Zhao, J., Kim, Y., Zhang, K., Rush, A., and LeCun, Y. (2018). Adversarially regularized autoencoders. In *International conference on machine learning*, pages 5902–5911. PMLR.

[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.