Text-Style-Transfer with CycleGAN: Decoder-only Models and Italian Datasets

Davide Fassio
s323584@studenti.polito.it

Daniele Ercole
s328755@studenti.polito.it

Alessia Manni
s331377@studenti.polito.it

Alessandro Arneodo
s331361@studenti.polito.it

***Abstract*—Text Style Transfer (TST) aims to modify the style of text while preserving its content. A prominent approach for non-parallel supervised TST is Cycle-Consistent Generative Adversarial Networks (CycleGAN). Traditional CycleGAN architectures relied on LSTMs and, more recently, encoder-decoder generators. However, advancements in NLP suggest that decoder-only models, such as GPT-2, could improve text generation quality. In this work, we integrate GPT2 as the generator for the CycleGAN model. We also compare Salesforce's CTRL, a conditional language model, with the result obtained by La Quatra, Gallipoli et al. Additionally, we evaluate the original encoder-decoder framework on two Italian datasets to evaluate its effectiveness in another language. The code is available at: https://github.com/danielercole/DNLP_project.**

I. Introduction

Text Style Transfer is a challenging problem in Natural Language Processing (NLP) where the goal is to change the stylistic attributes of a given text while preserving its semantic content. CycleGAN, originally developed for image-to-image translation, has been adapted to TST, offering an effective self-supervised approach in scenarios where annotated parallel training data is unavailable [1].

In this work, starting from the paper by La Quatra and Gallipoli, we introduce some novel approach to TST:

1) Using Salesforce's CTRL conditional language model to perform TST by controlling the generated style [2].
2) Using GPT-2, a decoder-only model, as the generator of the CycleGAN [3].
3) Applying the original method to verse-to-prose transfer, using *Divina Commedia* as a case study.
4) Applying the original model with an Italian version of the already tested GYAFC dataset, which is included in the XFORMAL dataset created by the same authors [4].

II. Related Work

TST has been explored with various approaches, including encoder-decoder architectures, variational autoencoders (VAEs) [5], and reinforcement learning-based model. Although these methods have achieved promising results, they often require large-scale parallel data, which can be difficult to obtain. CycleGAN, originally developed for image-to-image translation (Zhu et al., 2017) [6], has been adapted to text and specifically to perform style transfer in an unsupervised manner (La Quatra et al., 2024) [1]. Given the success of GPT models in text generation (Brown et al., 2020) [3] and, in general, the strong performance of large language models (LLMs) in producing high-quality text, we investigate their integration into CycleGAN to enhance the coherence and naturalness of TST.

III. Methodology

A. CycleGAN for Text Style Transfer

The objective function in our approach follows the original CycleGAN framework, incorporating multiple loss components to balance the different aspects of the TST task. The total loss function consists of the adversarial loss, cycle-consistency loss, and style loss, each weighted by a hyperparameter to control its contribution to the optimization process.

$$
\begin{aligned}
\mathcal{L}(G_{A\to B}, G_{B\to A}, D_A, D_B) = {} & \lambda_{\text{gen}}\,\mathcal{L}_{G_{D_B}} + \lambda_{\text{cyc}}\,\mathcal{L}_{\text{cyc}\,A\to B\to A} \\
& + \lambda_{\text{style}}\,\mathcal{L}_{\text{style}\,B} + \lambda_{\text{gen}}\,\mathcal{L}_{G_{D_A}} \\
& + \lambda_{\text{cyc}}\,\mathcal{L}_{\text{cyc}\,B\to A\to B} + \lambda_{\text{style}}\,\mathcal{L}_{\text{style}\,A} \\
& + \lambda_{\text{dis}}\,\mathcal{L}_{D_A} + \lambda_{\text{dis}}\,\mathcal{L}_{D_B}. \qquad (1)
\end{aligned}
$$

The adversarial losses ensure that the generators (G_A, G_B) produce outputs that are indistinguishable from the real samples in the target style, while the discriminators (D_A, D_B) are trained to differentiate between generated and real text. The cycle-consistency loss enforces content preservation by ensuring that translating a sentence to the opposite style and then back to its original style reconstructs the input. The style loss encourages stylistic alignment between generated and real text in each domain.

We adopt identical weighting factors for both transfer directions, ensuring a balanced optimization process without introducing additional complexity in hyperparameter tuning.

Finally, the training process follows the standard min-max optimization paradigm:

$$
\begin{aligned}
G^*_{A\to B}, G^*_{B\to A} = \arg \min_{G_{A\to B}, G_{B\to A}} \max_{D_A, D_B} \\
\mathcal{L}\Big(G_{A\to B}, G_{B\to A}, D_A, D_B\Big) \qquad (2)
\end{aligned}
$$

This formulation captures the adversarial nature of the model, where generators aim to fool the discriminators while the discriminators seek to distinguish real from generated samples, leading to improved style transformation.

B. Extension 1: Text-Style-Transfer with CTRL

In this extension, we evaluate the Salesforce's CTRL model for zero-shot TST. The CTRL model enables controlled text generation by conditioning the output on predefined control codes.

For our task, we leverage sentiment-based control codes using the following prompt structures:

> **Opinion** Negative: <*input sentence*> Positive:

> **Opinion** Positive: <*input sentence*> Negative:

These prompts guide the model in transforming a negative sentiment into a positive one and vice versa.

For text generation, we constrain the model's output to a maximum of 150% of the input sentence length. To allow early termination, we introduce a custom end-of-sequence token (ID `246533`).

To incorporate a degree of randomness, we employ top-k sampling ($k = 50$) and nucleus sampling ($p = 0.95$) with a temperature setting of 0.7, ensuring diversity while maintaining coherence in the generated text.

To test this model on a *sentiment transfer task* we considered the Yelp dataset [7], that collects restaurant reviews. Reviews are classified as positive or negative based on their rating: a rating of 4 or 5 is considered positive, while a rating below 3 is negative. The dataset contains a test set with four human references per sentence. The train and validation sets are designed for non-parallel supervised TST, as they include style attribute annotations but lack direct text pair matching. To ensure reproducibility, we adopt the same train/validation/test splits as in Li et al. [8] (see Table I).

TABLE I
YELP DATASET

	Train	**Validation**	**Test**
Negative	177,218	2,000	500
Positive	266,041	2,000	500

C. Extension 2: replacing the generator with GPT-2

In this extension, we replaced the original encoder-decoder generator with a GPT-2 Instruct model (`vicgalle/gpt2-open-instruct-v1`), which implements a decoder-only architecture. This substitution required several key modifications to our pipeline:

- **Model and Tokenizer:** We replaced `AutoModelForSeq2SeqLM` with `AutoModelForCausalLM` to support a causal language modeling setup. Additionally, the tokenizer was configured with `padding_side = "left"` to ensure proper padding, and if no pad token was available, the end-of-sequence token was used instead.
- **Prompt Formatting:** A specialized prompt template was introduced to align with the GPT-2 Instruct format. Each input sentence is embedded within a prompt that follows the structure:

> Below is an instruction that describes a task. Write a response that appropriately completes the request.
> **### Instruction:**
> Transform the following sentence from informal style to formal style:
> <*input sentence*>
> **### Response:**

- **Handling of EOS Tokens:** Different GPT-2 variants use distinct end-of-sequence tokens:
  - `openai-community/gpt2` use `"<|endoftext|>"` (ID `50256`)
  - `vicgalle/gpt2-open-instruct-v1` use `"### End"` (ID `50257`)

  We explicitly set a custom EOS token ID to ensure correct sequence termination during generation.
- **Teacher Forcing and Loss Computation:** In training mode, when target sentences are provided, the prompt is tokenized without truncation to capture its full length. The target sentence is tokenized with truncation to fit within the remaining sequence length (i.e., `max_seq_length - prompt_length`). The concatenated prompt and target sequence is then used to compute the loss, with the prompt tokens masked out (set to `-100`) so that only the target tokens contribute to the loss.
- **Generation Process:** For inference, only the prompt is tokenized (and padded to a uniform length) before being passed to the model for generation. The generation parameters (e.g., number of beams, top-k, top-p, temperature) were tuned appropriately, and after generation, the initial prompt is removed from the output to extract the final response.

These adjustments allowed us to integrate a decoder-only GPT-2 Instruct model into our text style transfer framework, enabling an evaluation of how this architecture handles style transformation tasks compared to the original encoder-decoder approach.

The discriminator, instead, remains a transformer-based classifier with the goal of distinguishing between real and generated text in the target style.

Similarly for the previous extensions, for a more comprehensive view of the training setup, we refer you to the GitHub folder.

D. Extension 3: application to Verse-to-Prose transfer

In this extension we apply the original CycleGAN architecture originally proposed in [1] to transform verse from *Divina Commedia* into prose. The dataset consists of tercets from Dante's original text taken from [9] paired with corresponding prose interpretations taken from [10], [11] e [12].

In Table II are represented the train (75%), validation (15%) and test (10%) splits of the dataset used.

TABLE II
DANTE DATASET

	Train	**Validation**	**Test**
Verse	3403	697	464
Prose	3403	697	464

Regarding the references, since we have a parallel paraphrase of Dante's verses, we leveraged this parallelism to create a test example. As a result, we have only one reference for both modern Italian and the Dantean style. This is because obtaining alternative versions of Dante's verses written in different ways is particularly challenging.

Adapting the system to Italian required substituting the pre-trained models that were originally designed for English.

In particular, the experimental pipeline consisted of two main phases: style classifier training and CycleGAN model training.

For classifier training, we fine-tuned a pre-trained Italian BERT model (`dbmdz/bert-base-italian-cased`).

Subsequently, the style transfer model was trained using the pre-trained classifier to guide the transfer from informal to formal text. The model setup used the `morenolq/bart-it` generator and the `dbmdz/bert-base-italian-cased` discriminator.

All other parameters were kept identical to the original configuration. A complete overview of the training setup can be found in the GitHub folder.

E. Extension 4: XFORMAL dataset

In this last extension we continue the work done in the previous by considering the `XFORMAL` dataset, which provides corresponding translations in several languages (including French, Portuguese, and Italian) of the `GYAFC` dataset, used in the original study. For our experiments, we specifically utilized the Italian translations of the Family & relationships category.

The structure of the dataset is equivalent to the English one, with the same train/validation/test splits (see Table III) and the same number of human references, four.

TABLE III
XFORMAL DATASET - ITALIAN - FAMILY & RELATIONSHIPS

	Train	**Validation**	**Test**
Informal	51,967	2,788	1,332
Formal	51,967	2,247	1,019

As done in the previous extension we have to adapt the system to the Italian language, substituting the pre-trained models that were originally designed for English with Italian ones.

All other parameters were kept identical to the original configuration. A complete overview of the training setup can be found in the GitHub folder.

IV. RESULTS

A. Extension 1: TST with CTRL

Table IV presents the results obtained on the Yelp dataset. Here we compare the results obtained with CTRL and the ones presented in [1], in particular ref-BLEU, style accuracy with BERT, geometric mean and harmonic mean of ref-BLEU and style accuracy.

CTRL significantly underperforms compared to the original CycleGAN models, with particularly low ref-B avg (0.308) and an extremely low GM (4.11) and HM (0.612), suggesting that it struggles to balance fluency and style control effectively. This stark contrast highlights the advantage of CycleGAN-based approaches in achieving both high text quality and controlled style transfer.

TABLE IV
RESULTS ON THE YELP DATASET

	ref-B avg	acc_{BERT}	GM	HM
CycleGAN BART (base)	55.7	**78.8**	**66.3**	**65.3**
CycleGAN BART (large)	**56.5**	75.1	65.1	64.5
CycleGAN T5 (small)	53.0	78.0	64.3	63.1
CycleGAN T5 (base)	54.2	76.6	64.4	63.5
CycleGAN T5 (large)	55.3	72.9	63.5	62.9
CTRL (Ours)	0.308	54.8	4.11	0.612

As part of the qualitative analysis, we present both successful and unsuccessful cases (see Tables in the Appendix).

Notably, the model generated an empty output in 59% of instances. The underlying causes of this failure mode remain unknown.

B. Extension 2: replacing the generator with GPT-2

Due to limited computational resources, we were unable to complete the training of the GPT-2 Instruct generator variant. Consequently, no empirical results are available at this time to assess the effectiveness of the architectural modifications introduced by replacing the original encoder-decoder generator with a decoder-only model.

C. Extension 3: application to Verse-to-Prose transfer

The evaluation metrics in Table V clearly indicate that the CycleGAN-based text style transfer model has not learned to modify the input as intended.

In both transfer directions, the outputs remain identical to the inputs, effectively reflecting an identity mapping rather than a genuine style transformation.

This is evidenced by the extremely high self-BLEU and self-ROUGE scores, along with minimal differences observed in BERTScore and reference-based BLEU/ROUGE metrics.

It is important to note that the task was not a simple style transfer; in several aspects, it resembled a translation task, since it involved paraphrasing an old Italian text.

This process demands not only a modification of style but also a careful rephrasing of archaic expressions, which risks distorting the original meaning.

TABLE V
KEY EVALUATION METRICS FOR STYLE TRANSFER WITH *Divina commedia* DATASET

Metric	A → B	B → A
BERTScore	0.7415	0.7493
g-BLEU	22.5463	24.8201
ref-BLEU	6.1775	7.3397
ref-ROUGE-1	0.3474	0.3679
ref-ROUGE-2	0.0982	0.1128
ref-ROUGE-L	0.3033	0.3233
self-BLEU	82.2886	83.9319
self-ROUGE-1	0.9976	0.9958
self-ROUGE-2	0.9937	0.9872
self-ROUGE-L	0.9976	0.9958
style F1 score	0.00429	
style accuracy	0.00431	
style precision	0.00427	
style recall	0.00431	

D. Extension 4: XFORMAL dataset

Following the previous experiment, we opted for a simpler style transfer approach using only Italian data, specifically the Italian section of the XFORMAL dataset.

Unfortunately, this issue persisted even when experimenting with a simpler style transfer approach using only Italian data, specifically the Italian section of the XFORMAL dataset. In fact, even with this dataset, the model failed to achieve effective style transfer. Instead of altering the input, the system reproduced it more or less unchanged. In Table VI are presented the evaluation metrics.

TABLE VI
KEY EVALUATION METRICS FOR STYLE TRANSFER WITH *XFORMAL* DATASET

Metric	A → B	B → A
BERTScore	0.8699	0.8315
g-BLEU	9.4268	8.9585
ref-BLEU	1.3297	1.2173
ref-ROUGE-1	0.7734	0.6068
ref-ROUGE-2	0.6299	0.4150
ref-ROUGE-L	0.7589	0.5755
self-BLEU	66.8277	65.9283
self-ROUGE-1	0.9944	0.9977
self-ROUGE-2	0.9933	0.9975
self-ROUGE-L	0.9944	0.9977
style F1 score	0.2588	
style accuracy	0.2702	
style precision	0.2594	
style recall	0.2583	

V. CONCLUSION

In this work, we explored various extensions of CycleGAN-based TST, integrating decoder-only models, experimenting with different datasets, and evaluating alternative approaches like the CTRL conditional language model. Our key findings are summarized as follows:

- **CTRL for TST:** The experiments using Salesforce's CTRL model for zero-shot TST on the Yelp dataset demonstrated poor performance compared to traditional CycleGAN approaches. The model failed to balance fluency and style transfer effectively, often producing empty outputs or inappropriate generations.
- **GPT-2 as a Generator:** Due to computational constraints, we were unable to fully train the GPT-2 Instruct model within the CycleGAN framework. While theoretically promising, further experimentation is needed to determine if a decoder-only model can enhance style transfer while preserving content integrity.
- **Italian Datasets:** Our attempts to adapt the model to the Italian language using Dante's *Divina Commedia* and the XFORMAL dataset revealed the model's inability to perform meaningful style conversion. Instead, the system exhibited an identity mapping behavior, failing to modify input sentences. This suggests that CycleGAN requires additional refinement, especially for non-English data.

Overall, our results indicate that while CycleGAN-based TST has shown promise in prior research, its generalization to new architectures, alternative datasets, and complex style transformations remains challenging.

Despite the limitations observed in our experiments, this study provides valuable insights into the applicability of modern NLP architectures for style transfer and sets a foundation for future improvements in this domain.

REFERENCES

[1] M. La Quatra, G. Gallipoli, and L. Cagliero, "Self-supervised text style transfer using cycle-consistent adversarial networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 1, no. 1, pp. 1–37, 2024.

[2] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, "Ctrl: A conditional transformer language model for controllable generation," 2019. [Online]. Available: https://arxiv.org/abs/1909.05858

[3] A. Radford, J. Wu, R. Child *et al.*, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, pp. 1–9, 2019.

[4] E. Briakou, D. Lu, K. Zhang, and J. Tetreault, "Xformal: A benchmark for multilingual formality style transfer," 2021. [Online]. Available: https://arxiv.org/abs/2104.04108

[5] P. Xu, J. C. K. Cheung, and Y. Cao, "On variational learning of controllable representations for text without supervision," 2019.

[6] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2223–2232.

[7] T. Shen, T. Lei, R. Barzilay, and T. S. Jaakkola, "Style transfer from non-parallel text by cross-alignment," Long Beach, California, 2017. [Online]. Available: https://arxiv.org/abs/1705.09655

[8] J. Li, R. Jia, H. He, and P. Liang, "Delete, retrieve, generate: a simple approach to sentiment and style transfer," pp. 1865—-1874, 2018. [Online]. Available: https://doi.org/10.18653/v1/N18-1169

[9] Wikisource. Divina commedia. Accessed: 2025-02-13. [Online]. Available: https://it.wikisource.org/wiki/Divina_Commedia

[10] O. Furioso. (2025) Parafrasi dei canti dell'inferno – prima cantica del poema divina commedia. Accessed: 2025-02-13. [Online]. Available: https://www.orlandofurioso.com/parafrasi-dei-canti-dellinferno-prima-cantica-del-poema-divina-commedia/

[11] O. Furıoso. Parafrasi dei canti del purgatorio – seconda cantica del poema divina commedia. Accessed: 2025-02-13. [Online]. Available: https://www.orlandofurioso.com/parafrasi-dei-canti-del-purgatorio-seconda-cantica-del-poema-divina-commedia/

[12] O. Furioso. Parafrasi dei canti del paradiso – terza cantica del poema divina commedia. Accessed: 2025-02-13. [Online]. Available: https://www.orlandofurioso.com/parafrasi-dei-canti-del-paradiso-terza-cantica-del-poema-divina-commedia/

APPENDIX

TABLE VII
EXAMPLES FROM YELP TEST SET – NEGATIVE TO POSITIVE – SUCCESS CASES

Source	you are not the only place in town with decent pizza .
Generated	* You are the only place to get a good pizza
References	decent pizza they had here it is the only place with excellent pizza you are the only place in town with decent pizza . this is the best place in town for pizza
Source	when she did n't answer him quickly enough , he hung up on her .
Generated	when she did answer him , he did n't hang up on her .
References	when she did n't answer him quickly enough , he patiently waited on her . when she answered him quickly , he did n't hung up on her . she answered quickly when she did n't answer him quickly enough , he hung up on her .

TABLE VIII
EXAMPLES FROM YELP TEST SET – NEGATIVE TO POSITIVE – FAILURE CASES
(THE MODEL GENERATES INAPPROPRIATE LANGUAGE THUS POSING SAFETY CONCERNS)

Source	the food 's ok , the service is among the worst i have encountered .
Generated	I got it for free .
References	the food is good , and the service is one of the best i 've ever encountered . the food 's ok , the service is among the best i have encountered . the food is ok and the service is good the food is nice and the service is good
Source	no , i 'm not at a scottsdale club .
Generated	* I 'm not a pussy .
References	this was a great club i am in a amazing scottsdale club i feel that i am in a scottsdale club yes , i 'm at a scottsdale club

TABLE IX
EXAMPLES FROM YELP TEST SET – POSITIVE TO NEGATIVE – SUCCESS CASES

Source	and i get the gyro sandwich every time and its the best .
Generated	I get the gyro sandwich every time and its the worst sandwich ever .
References	and i get the gyro sandwich every time and its the worst . and i got the gyro sandwich this time and its the worst and i get the gyro sandwich every time and its the most horrible . and i get the gyro sandwich every time and its the bad .
Source	one of my favorite places to go for a great steak .
Generated	* The steaks are pretty pricey
References	one of my least favorite places to go for a steak one of my less favorite places to go for a great steak . i do not like this place to go for a great steak . not my favorite places to go for a great steak

TABLE X
EXAMPLES FROM YELP TEST SET – POSITIVE TO NEGATIVE – FAILURE CASES

Source	this resort has been in scottsdale forever and it truly knows service .
Generated	* the location is so close to the ocean that you can feel the waves crashing against the shore .
References	this resort has been in scottsdale forever and it truly does n't know service . this resort has been in scottsdale forever and it truly does not knows service this resort has been in scottsdale forever and it does not knows service at all . despite this resort having been in scottsdale for forever they do n't know a single thing about service
Source	try anything on the menu and you will be glad you came in !
Generated	- no choice of wine
References	i would not try anything on their menu try anything on the menu and you will be sad you came in try anything here and you will regret coming here try anything on the menu and you will be glad you never came in !