Reverse Attention and Conditional Layer Normalization for Text Style Transfer: An Ablation Study

Ayush Alag
Princeton University
aalag@princeton.edu

Rishi Dange
Princeton University
rdange@princeton.edu

Sam Liang
Princeton University
saml@princeton.edu

Abstract

Numerous processes exist for text style transfer, from whole-sentence mathematical optimization to token-by-token predictive strategies. This ablation study analyzes an influential paper by Lee et al. which pursues the per-token methodology while introducing reverse attention and conditional layer normalization to increase nuance and performance. Upon successfully validating baselines on the Yelp and IMDB review datasets, our paper extends Lee et al. 2021's work via ablations that measure the relative importance of the reverse attention mechanism; model architecture; loss function; and embedding style on overall performance. We find that the reverse attention ablation severely reduces performance, validating Lee et al. 2021's hypothesis on its utility. We also identify that GRU directionality only plays a small role in model performance. Lastly, we identify that removing content loss increases performance due to redundancies between cycle and content loss, but that style loss plays a key role in performance.

1 Introduction

With applications into improving automated writing (1), generating emotionally-sensitive AI chatbots (2)—and even furthering Artificial General Intelligence (3)—text style transfer is a critical application of Natural Language Processing. Specifically, text style transfer involves the changing of sentences or other forms of text to a particular desired style. As style may be defined ambiguously, we concretize the term for our application to mean whether the sentiment is positive or negative. With these definitions in mind, the goal of text style transfer is three-fold:

1. Ensuring that **content** is fully preserved when transferring from input to output.
2. Maximizing the **fluency** of the transferred output.
3. Generating an output with a high classification accuracy in the desired transfer **style**.

Jointly optimizing these three objectives while simultaneously balancing the speed and memory of the deep learning model is an intricate task. Our paper hopes to replicate and identify insights on an algorithm that may help bring the scientific community closer to such a goal.

In this work, we reproduce and, via ablations, expand upon the 2021 paper titled "Enhancing Content Preservation in Text Style Transfer Using Reverse Attention and Conditional Layer Normalization" by Lee et al. 2021 (4). Lee et al. 2021 seeks to improve upon the style transfer and content preservation techniques utilized by prior works in the area. Specifically, many researchers (5) (6) perform text style transfer by first removing tokens that are classified as "style" tokens and then altering the remaining content to match the desired output style. Lee et al. explain the prevalent issue with this methodology using the example of the word "delicious," which would have both style value (positive) and content value (the word indicates the presence of consumable items). If the model were to heavy-handedly remove all style tokens, it would lose valuable information from words that possess both content and style information. In response to this issue, Lee et al. 2021 proposes an experiment with two key innovations.

Firstly, Lee et al. 2021 proposes implicitly removing style via a mechanism known as reverse attention to achieve **style-independent content**. In particular, the reverse attention innovation targets the initial phase of style transfer, where we seek to extract the raw content from a phrase. The model utilizes attention to effectively estimate the amount of "style" within a given token. By interpreting

the style quantification as a probability, and taking the complement, the model can then estimate the content level of the tokens, enabling suppression of style while preserving content.

Secondly, Lee et al. 2021 proposes utilizing conditional layer normalization to generate **content-dependent style**. In other words, the model builds its transferred, styled output based on the content of the original sentence, hence (in theory) preserving both content and fluency. Such an innovation is disruptive, as state-of-the-art models are often agnostic to the actual content of the sentence when applying style transfer rules.

In our reproduction and ablation study of Lee et al. 2021 paper, we strive to achieve five broad goals in unsupervised text style transfer:

1. Replicate Lee et al. 2021's baseline results on the Yelp and IMDB datasets, validating and correcting their code if necessary.

2. Via ablations such as removing dropout and altering the bidirectionality of the encoder GRU, determine the overall sensitivity of the model in response to tweaks in its construction.

3. Determine the criticality of the reverse attention component by ablating it and observing the resulting model performance.

4. Compare the importance, relevance, and effect of particular components of the loss function (such as style transfer loss, content loss, and reconstruction loss), by removing each component individually from the loss function calculation.

5. Evaluate whether utilizing Lee et al. 2021's custom stylizer module improves performance over a base embedding layer.

As with the original Lee et al. paper, all experiments are performed on the Yelp Review and IMDB Movie Review datasets.

At a high level, we were able to replicate both Yelp and IMDB baselines, and extract actionable insights from the ablations. Specifically, we found that while the reverse attention, dropout, and style loss components were important to model performance, the bidirectionality of the GRU was less significant. More surprisingly, incorporating a content loss component actually hurt model performance due to overlap with cycle loss.

$$\mathcal{L}_{\text{rec}}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G) = \mathbb{E}_{\boldsymbol{x}_1 \sim \boldsymbol{X}_1}[-\log p_G(\boldsymbol{x}_1 | \boldsymbol{y}_1, E(\boldsymbol{x}_1, \boldsymbol{y}_1))] + \mathbb{E}_{\boldsymbol{x}_2 \sim \boldsymbol{X}_2}[-\log p_G(\boldsymbol{x}_2 | \boldsymbol{y}_2, E(\boldsymbol{x}_2, \boldsymbol{y}_2))]$$

Figure 1: A representation of the style transfer problem under the assumption that the same phrase in different styles shares the same content space.

2 Related Work

Lee et al. 2021 differentiates itself in its method of deconstructing a sentence into content and style representations. Specifically, Lee et al. 2021 is the only known paper that achieves token-level granularity while also retaining flexibility in sentence structure, which enables the model to handle more complex test cases.

In the past, some authors such as Shen et al. 2017 (7) have made assumptions that different styles have the same content distributions, and then utilized these assumptions to formulate a mathematical optimization problem that generates the output transfer via a one-step autoencoder. Figure 1 details an example of such a formalization.

While this procedure is an important step forward, translating sentiment at the sentence level rather than the token level can yield imprecision in meaning, for example translating "this food is *disgusting*" to "this food is *pretty*". Under the assumption of the uniform style distribution, as in Shen et al. 2017, we may have a general rule mapping disgusting to pretty; however, it would be more effective to consider content when performing style transfer in order to have outputs. Explicitly analyzing the sentence at the token level may better achieve that result.

Other authors such as Li et al. 2018 (5) and Wu et al. 2019 (6) have taken a token-by-token deconstruction approach, yet utilize a heavy handed style-or-content approach. In "Mask and Infill: Applying Masked Language Model to Sentiment Transfer," Wu et al. utilizes a bidirectional BERT model to identify the positions of stylistic tokens, and then a conditional model that predicts those masked-out style tokens in the output sentiment. While such a model can handle simpler cases, such as transferring "beautiful scenery and good service" to "terrible scenery and poor service," it may perform worse on more fluid examples such as "he shut the door," as shortcomings of the masking strategy are potentially losing fluency when words contain both style and content information. These include words such as "scrumptious," which has a positive

connotation but also provides food-related content information.

Thus, Lee et al. 2021 utilizes reverse attention at the token level to provide a nuanced content representation for each token, rather than keeping or removing a token based on a style threshold.Their paper also introduces Content Layer Normalization, which removes the assumption that the style space is shared across content-varied vectors, furthering the generation of content-dependent style shift.

3 Data

Our replication study utilized two main datasets: one containing short-sentence Yelp business reviews, and another containing longer-sentence IMDB movie reviews. Baselines were replicated for both Yelp and IMDB, but in accordance with Lee et al. and due to the high computational time of IMDB, all ablations were performed only on the Yelp dataset.

3.1 Yelp

The Yelp dataset (8) contains a single sentences or phrases from business reviews, along with a label containing whether the style of the phrase is positive or negative. For instance, the following is an example of a datum in the Yelp corpus:

"X: 'just left and took it off the bill'; C: 0" .

where the label of 0 denotes that the review has a negative sentiment. The Yelp review dataset contains a training set of 439K examples, partitioned into 264K positive and 175K negative samples. Additionally, there is a validation set that contains 4K total examples, with 2K each for positive and negative. The test set is smaller at 1000 examples, split evenly into 500 positive and negative samples. Lastly, a secondary test set contains 1000 human-generated sentences that emphasize sentence fluency and provide a reference baseline for the model.

Principally, the Yelp dataset is characterized for having short sentences or phrases that can range from 4-10 words on average. The diction consists purely of common words, rather than esoteric or overly complex ones, and the style of each sentence (positive or negative) is fairly clear-cut.

3.2 IMDB

The IMDB movie review dataset (9) contains fewer reviews, with 36.7K reviews partitioned into 17.9K positive and 18.8K negative. However, each review is much longer, with each data example ranging from 7-20 words. An example datum for IMDB is as follows:

"X: 'we can learn a lot from this movie and we should be proud on our great treasure on earth'; C: 1" .

Specifically, the average sentence length for the IMDB test set was calculated as 14.31, whereas the average length for the Yelp test set is 6.40. The diction in IMDB is also much more complex, and each example can contain multiple syntactical structures. Thus, it is important to validate the model on both the Yelp and IMDB baselines.

4 Original Model Architecture

As this work reproduces and expands upon the work done in Lee et al. 2021, our model resembles that used by Lee et al. We begin with a discussion of the text style classifier. Then, we discuss the three main subunits of the model:

1. Encoder to generate the style-independent content vector.
2. Stylizer to create the content-dependent style.
3. Decoder to output the final result.

Finally, we discuss the composition of the loss function used in training. Each of these components will be discussed further below, and a visualization from Lee et al. 2021 of the combined architecture can be seen in Figure 2.

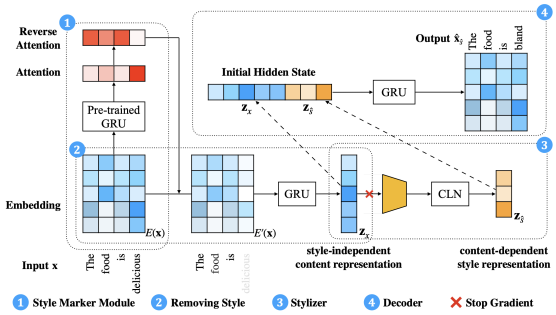


Figure 2: Style transfer model implemented in Lee et al. 2021. The model utilizes reverse attention and GRUs to create a style-independent content representation; utilizes another GRU pipeline to identify content-dependent style features; and combines these two insights in a downstream GRU substructure to obtain a style-transferred output.

4.1 Style Classifier

Locating large amounts of parallel data for transferring sentence styles is a difficult endeavor, which makes an unsupervised learning environment a more ideal setup than a supervised one. Consequently, we utilize a style classifier $s = C(\mathbf{x})$ that has been pretrained on our training set, which matches sentences to their respective styles. Given this classifier, Lee et al. (and hence we) seek to learn the transfer function $f_\theta(\mathbf{x}, \hat{s})$, where $\hat{\mathbf{x}}_{\hat{s}} = f_\theta(\mathbf{x}, \hat{s})$ and $C(\hat{\mathbf{x}}_{\hat{s}})$ matches $\hat{s}$.

4.2 Encoder

In order to generate the style-independent sequence, the encoder in Lee et al. 2021 utilizes a pre-trained style marker module, which computes the amount of style information contained within each token of the input sequence. The pre-training process involves the use of a bidirectional GRU (specifically, one layer of it) as well as cross-entropy loss for learning pre-training parameters. With $\mathbf{h}_t$ as the hidden representation from the GRU, τ as the softmax temperature, and α representing attention, we compute (4)

$$\mathbf{v}_t = tanh(\mathbf{W}_w \mathbf{h}_t + \mathbf{b}_w), \tag{1}$$

$$\alpha_t = \frac{exp(\mathbf{v}_t^\intercal \mathbf{u}/\tau)}{\sum_{t=1}^{T} exp(\mathbf{v}_t^\intercal \mathbf{u}/\tau)}, \tag{2}$$

$$\mathbf{o} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t, \tag{3}$$

and

$$\mathbf{p} = softmax(\mathbf{W}_c \mathbf{o} + \mathbf{b}_c). \tag{4}$$

Given the style marker module, a key feature of Lee et al. 2021's model is their incorporation of reverse attention, a concept introduced by them but thereafter used in applications such as road segmentation (10) and object detection (11). As implied by the feature's name, we reverse the attention by computing (4)

$$\tilde{\alpha}_t = 1 - \alpha_t. \tag{5}$$

From here, we can then implicitly remove the style aspects of tokens by creating the embeddings (4)

$$\tilde{\mathbf{e}}_t = \tilde{\alpha}_t E(\mathbf{x}_t), \tag{6}$$

which are then fed into a bidirectional GRU to create the "style-independent content representation $\mathbf{z}_\mathbf{x}$" (4).

4.3 Stylizer

Given the encoding, the stylizer in Lee et al. generates the style representation $\mathbf{z}_{\hat{s}}$ of the sequence. After shrinking the size of the content representation $\mathbf{z}_\mathbf{x}$ to $\tilde{\mathbf{z}}_\mathbf{x}$ using an affine transformation (4), we then compute the style representation using conditional layer normalization (CLN), which enables the style representation to adapt based on the content presented in the input sequence. More specifically, we compute (4)

$$\mathbf{z}_{\hat{s}} = CLN(\tilde{\mathbf{z}}_\mathbf{x}; \hat{s}) = \gamma^{\hat{s}} \odot \left(\frac{\tilde{\mathbf{z}}_\mathbf{x} - \mu}{\sigma} \right) + \beta^{\hat{s}}. \tag{7}$$

4.4 Decoder

Given the content representation $\mathbf{z}_\mathbf{x}$ and the content-dependent style representation $\mathbf{z}_{\hat{s}}$, the Lee et al. 2021 decoder utilizes one layer of GRU. The output of the decoder is the transformed sentence in the desired output style $\hat{s}$. With p_D as our conditional distribution over the decoder's sequences, we compute (4)

$$\hat{\mathbf{x}}_{\hat{s}} \sim Dec_\theta(\mathbf{z}_\mathbf{x}, \mathbf{z}_{\hat{s}}) = p_D(\hat{\mathbf{x}}_{\hat{s}} | \mathbf{z}_\mathbf{x}, \mathbf{z}_{\hat{s}}). \tag{8}$$

4.5 Loss Function

In training, the loss function in Lee et al. 2021 is composed of four different components. Specifically, it is calculated as (4)

$$\mathcal{L} = \lambda_1 \mathcal{L}_{self} + \lambda_2 \mathcal{L}_{cycle} + \lambda_3 \mathcal{L}_{content} + \lambda_4 \mathcal{L}_{style}, \tag{9}$$

a weighted average of the self loss, the cycle loss, the content loss, and the style transfer loss.

The self loss (self-reconstruction loss) measures the loss when attempting to transfer the style of a particular sentence to its own classified style. Namely, we would expect the sentence to essentially remain the same if we attempt to keep the style the same. With $(\mathbf{x}, s) \in \mathcal{D}$ as our training example, we compute (4)

$$\mathcal{L}_{self} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}[\log p_D(\mathbf{x} | \mathbf{z}_\mathbf{x}, \mathbf{z}_s)]. \tag{10}$$

The cycle loss (cycle-reconstruction loss) measures the loss when transferring the style of a sentence to a different style and then transferring back to the original style. As with self-reconstruction loss, we would expect the sentence to essentially remain the same. With $\hat{\mathbf{x}}_{\hat{s}}$ as our intermediary transferred sequence, we compute (4)

$$\mathcal{L}_{cycle} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}[\log p_D(\mathbf{x} | \mathbf{z}_{\hat{\mathbf{x}}_{\hat{s}}}, \mathbf{z}_s)]. \tag{11}$$

Hyperparameter	Value
Embedding Size	128
Attention Size	100
Style Vector Size	200
Hidden Size	500
Batch Size	512
Learning Rate	0.0005
Epochs	40

Table 1: Hyperparameters utilized when running experiments on the Yelp dataset. Embedding size, Attention size, and Hidden size correspond to the vector sizes of the initial embedding, intermediate attention output, and content representation respectively. The style vector is concatenated with the content-based hidden vector, which is processed through the final GRU to yield the output.

The content loss stems from the goal during construction that transfer steps leave as much content as possible. In the process of calculating cycle-reconstruction loss, we should then see similarities between $\mathbf{z_x}$ and $\mathbf{z}_{\hat{\mathbf{x}}_{\hat{s}}}$. Hence, we compute (4)

$$\mathcal{L}_{content} = \mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}||\mathbf{z_x} - \mathbf{z}_{\hat{\mathbf{x}}_{\hat{s}}}||_2^2. \quad (12)$$

Finally, the style transfer loss measures how close to the desired style the outputs are. With p_C as the conditional distribution over the classifier's styles, we compute (4)

$$\mathcal{L}_{style} = -\mathbb{E}_{(\mathbf{x},s)\sim\mathcal{D}}[\log p_C(\hat{s}|\hat{\mathbf{x}}_{\hat{s}})]. \quad (13)$$

5 Implementation Details

We conducted all of our experiments on either a Google Colab Pro instance—using Tesla P100 and V100 GPUs—or utilizing the RTX2080 GPUs in the PNLP subserver of the ionic cluster. Both coding environments were utilized such that different ablations could be run in parallel. Experiments on the Yelp dataset were run with the hyperparameters shown in Table 1.

Training the model on the Yelp dataset for 40 epochs took 4-5 hours of compute time on the above GPUs. The baseline was also replicated on the IMDB dataset, which utilized the same hyperparameters except a batch size of 128 in order to stay within memory limits. Each IMDB run took around 12 hours to complete, which resulted in all ablations being performed on the Yelp corpus alone. As mentioned earlier, Lee et al. 2021 also utilized only the Yelp dataset for ablations.

6 Baselines

Firstly, we sought to reproduce Lee et al. 2021's results on the Yelp and IMDB datasets with their baseline model. Lee et al. 2021 measured four principle metrics: S-ACC, which measures the style score the pre-trained style classifier gives the output sentence; Self-BLEU, measuring the content-based BLEU score on test outputs; Ref-BLEU, measuring the BLEU score as compared to the human-generated sentences; and perplexity, relating to a separately trained language model (4). We were able to successfully replicate their results, with details for Yelp and IMDB in Tables 2 and 3 respectively. As mentioned in the Appendix, we found issues with the perplexity calculations in the code base. Therefore, we included the perplexity values in Tables 2 and 3 but not in any of the ablation results.

Baseline	S-ACC	Self-BLEU	Ref-BLEU	PPL
Ours	90.9	58.73	20.67	47.5
Lee et al.	91.3	59.4	20	60.1

Table 2: Replicating results on the **Yelp** baseline. All metrics except for perplexity match; details on issues with Lee et al. 2021 perplexity can be found in the Appendix.

Baseline	S-ACC	Self-BLEU	PPL
Ours	90.8	71.23	45.32
Lee et al.	83.1	70.9	45.3

Table 3: Replicating results on the **IMDB** baseline. Note that the IMDB dataset did not contain human-generated text and thus did not have a ref-BLEU score.

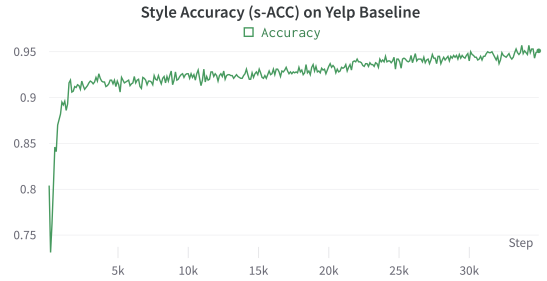
Lastly, we provide graphs to visualize the test accuracy accuracy, various training loss components, and test BLEU score of the Yelp dataset in Figure 3.

7 Ablations

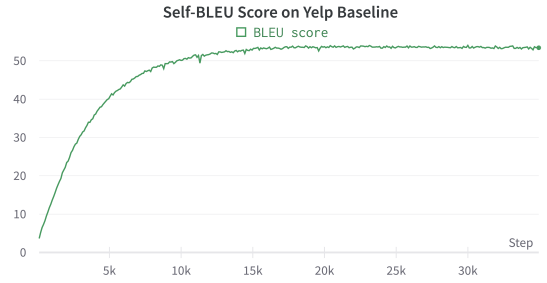
After successfully validating the baselines, we proceeded to perform ablation studies that would elucidate various aspects of model performance.

7.1 Ablation 1: Removing Dropout

Our first ablation is removing dropout from the two GRUs in the style classifier and in the encoder. Dropout prevents the model from overfitting to the training data and improves generalizability, as the model is forced to adapt representations of the data



(a) Style Accuracy (s-ACC)



(b) Self-BLEU score on the held-out test set

when certain nodes are masked out. With this ablation, we want to confirm these theoretical guarantees on the test set, investigating whether dropout in the classifier and encoder would be helpful.

7.2 Ablation 2: Changing the Encoder GRU from Bidirectional to Unidirectional

Our next ablation is switching a bidirectional GRU to a unidirectional GRU in the encoder. A bidirectional GRU can better encode the meaning of a word, since it utilizes forward context as well as backward context. We want to see how sensitive the reverse attention scores are to how well a word is encoded by observing the evaluation metrics after this ablation.

7.3 Ablation 3: Removing the Reverse Attention Component

Our third ablation explores the importance of one of the innovations of the paper: the reverse attention component. By replacing the reverse attention pipeline with simply pushing the embedded input to the downstream GRU, we can observe how reverse attention improves the text style transfer.

7.4 Ablation 4: Removing Style Transfer Loss

Our next ablation involves removing the style loss from the total loss function. The style transfer loss evaluates whether the model correctly changed the original sentence to be of a target style. With this ablation, we hope to understand which aspect of the model the style transfer loss affects, and specifically whether the stylizer module is high performing.

7.5 Ablation 5: Removing Content Loss

The fifth ablation is removing the content loss. The content loss tracks whether the content representations of two sentences of differing style are similar. A model that separates the content and style well should have the same content representations for different styles because the content representations are style independent. With this ablation, we want to evaluate whether the content loss is redundant because the cycle reconstruction loss is tracking a similar metric.

7.6 Ablation 6: Removing Cycle Reconstruction Loss

The sixth ablation is removing cycle reconstruction loss. The cycle reconstruction loss evaluates how well the model can reconstruct the original input after changing it to a sentence of a particular target style and then changing it back to a sentence with the original style. In short, cycle loss measures how well the model can preserve the original input. With this ablation, we want to explore if this loss does a better job at improving the model than content loss and if both of them combined provide meaningful performance gains.

7.7 Ablation 7: Change Content-Dependent Style Embeddings to Normal Style Embeddings

The seventh ablation is replacing the paper's second innovation, content-dependent style representations, with normal style embeddings. The normal style embedding projects a style to a single embedding space irrespective of the content (4). Therefore, two different sentences with the same target style would share the same style embedding. The content-dependent style representations should theoretically better retain the content of a sentence because the style embedding is concatenated with the content of the sentence. Therefore, with this ablation, we want to see if content-dependent style representations do preserve content and improve the model.

8 Results

We first replicated the paper's results on the Yelp and IMDB dataset, with results in Tables 2 and 3

Ablations	S-ACC	Self-BLEU	Ref-BLEU	PPL
No Dropout	91.2	55.75	19.26	-
Encoder GRU Unidirectional	91.4	57.21	19.53	-
No Reverse Attention	73.2	53.87	17.63	-
No Style Loss	21.8	78.03	22.27	-
No Content Loss	96.5	62.97	21.23	-
No Cycle Loss	98.3	43.31	13.28	-
Normal Style Embeddings	92.5	53.96	20.33	-
Baseline Model	**90.9**	**58.73**	**20.67**	-

Table 4: Results for dropout, reverse attention, GRU, loss, and style embedding ablations on the **Yelp** dataset. S-ACC is the Style Transfer Accuracy which measures whether the generated sentence is of the target style. Self-BLEU measures the content preservation between an input sentence and the model's generated sentence. Ref-BLEU measures the content preservation between the model's generated sentence and a human-generated sentence. PPL is the perplexity, which is not reported due to factors explained in the Appendix.

above. Then, we ran the proposed ablations. All results are reported in Table 4. The models with ablations are retrained on Yelp's training dataset before being tested on Yelp's test dataset. Evaluation metrics of style transfer accuracy, self-BLEU, ref-BLEU, and perplexity are shown.

9 Analysis and Discussion

9.1 Ablation 1

Removing dropout produced expected results, negatively affecting the model's ability to learn to preserve content via lower BLEU scores. Dropout helps the GRU in the style classifier better learn reverse attention scores. It also improves the generalizability of the encoder GRU. As such, the model outputs a better style-independent content representation, leading to better content performance. However, dropout does not affect the style transfer accuracy because no GRUs nor dropout are used in creating the content-dependent style representation. Therefore, only BLEU scores are adversely affected.

9.2 Ablation 2

Changing the encoder GRU from bidirectional to unidirectional only slightly reduces performance. We expected a greater decrease in the BLEU scores because the encoder GRU is used to identifying whether a word is a style word, which is crucial to creating style-independent content representations. The style transfer accuracy is unaffected because the decoder GRU is unchanged, which means style embeddings are not directly affected.

9.3 Ablation 3

The reverse attention ablation yielded the expected results: we found that removing the module lowered not only style scores, but also content-based BLEU scores. Such a finding validates Lee et al.'s hypothesis and also makes intuitive sense, as the reverse attention is a critical component in masking out the style as well as feeding the content vector to the downstream encoder GRU.

9.4 Ablation 4

Removing the style loss produced interesting results. Firstly, the style transfer accuracy decreases significantly, as expected. The style loss is the only loss directly evaluating whether the model generated a sentence of the desired target style. Without it, the model does not learn the style transfer component of the model well. We also note that the BLEU scores increase. This can be explained by the fact that the model can now focus entirely on preserving the sentence content, rather than having to balance content with style. Moreover, it is easier for the model to preserve content if the sentence it generates is of the same style as the input sentence, which is why we see a stronger performance when compared to both human-generated and the target input sentence.

9.5 Ablation 5

Removing the content loss improved all metrics across the board. This confirms that the content loss is redundant and actually negatively affects the model's ability to learn. This is probably due to the conflicting signals of the content and cycle loss, as both are tracking similar things. It confuses the

model, so the content loss is an unnecessary loss measure.

9.6 Ablation 6

Removing the cycle loss decreases BLEU scores. This shows that the cycle loss is important in pushing the model to preserve content. Therefore, cycle loss is the more important and useful metric for preserving content as opposed to content loss. We also see that the style transfer accuracy increases. This could be because the model can focus on changing the style of the sentence and not worry about preserving content. Results match a similar ablation done by Dai et al. (12)

9.7 Ablation 7

Changing the content-dependent style embeddings to shared style embeddings decreased BLEU scores and increased the style transfer accuracy. This shows that Lee et al. 2021's second innovation is useful in preserving content, since each style embedding is injected with the content. The style transfer accuracy increases, meaning that the content-dependent style embedding actually induces a trade-off. For an increase in content preservation, there is a decrease in style transfer accuracy because we no longer utilize a shared style embedding, which means we lose some style information to capture the content information.

10 Conclusion

In this paper, we first replicated the baseliens of the Lee et al. 2021 model on the Yelp and IMDB datasets. We then performed an ablation study, analyzing four broad categories of the model architecture.

1. In tweaking the model's construction, we found that dropout was much more important to the model's performance than the bidirectionality of the GRU.

2. In removing the reverse attention feature, we established its overall importance in aiding text style transfer performance.

3. We found that style loss and cycle-reconstruction loss improved performance, but content loss hindered the utility of the overall loss function.

4. We identified the custom stylizer module from Lee et al. 2021 to be fairly significant in helping the performance of the model.

While many of these findings align with the overall rationale behind the Lee et al. 2021 paper, two intriguing discrepancies involve the GRU bidirectionality and the content loss component. The low importance of the GRU bidirectionality highlights that theoretical gains may not always translate into practical improvements. Moreover, the negative impact of content loss epitomizes the importance behind selecting a well-balanced loss function, a lesson to keep in mind for future work.

10.1 Future Work

As next steps, we plan to explore the following aspects of this project related to baselines:

1. Extending our work to a novel dataset (one unrelated to reviews, such as fake news articles).

2. Conducting experiments in which sequences can be transferred to more than one possible style (proposed by Lee et al. as well).

3. Investigating (potentially via graphs) the way in which cycle-reconstruction loss changes as multiple styles are used and different types of cycles are created, for example transferring from style A to B to C back to A.

4. Creating a small parallel dataset in order to experiment with supervised learning for style transfer.

As well, we would love to investigate the following ablations:

1. Replacing all instances of GRUs with LSTMs to check whether the increase in parameters improves performance.

2. Removing self-loss as a component of the loss function and measuring resulting performance.

10.2 Acknowledgements

We would like to thank Mengzhou Xia for her advice on our proposal; Professor Karthik Narasimhan for his guidance over the semester; Lee et al. for their paper "Reverse Attention and Conditional Layer Normalization for Text Style Transfer," which is the basis for this ablation study; and Princeton University for its computing resources and general support.

References

[1] Alejandro Rodriguez Pascual. "BACON: Deep-Learning Powered AI for Poetry Generation with Author Linguistic Style Transfer." arXiv preprint arXiv:2112.11483 (2021).

[2] Ela Elsholz, Jon Chamberlain, and Udo Kruschwitz. "Exploring language style in chatbots to increase perceived product value and user engagement." Proceedings of the 2019 Conference on Human Information Interaction and Retrieval. 2019.

[3] Zhenxin Fu et al. "Style transfer in text: Exploration and evaluation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.

[4] Dongkyu Lee et al. "Enhancing content preservation in text style transfer using reverse attention and conditional layer normalization." arXiv preprint arXiv:2108.00449 (2021).

[5] 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.

[6] Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model for sentiment transfer. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI19, pages 5271–5277. International Joint Conferences on Artificial Intelligence Organization.

[7] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 6830–6841. Curran Associates, Inc.

[8] 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.

[9] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.

[10] Guoqi Liu, Manqi Zhao, Lu Bai, Hecang Zang, and Baofang Chang (2022). Field road segmentation network based on PraNet, Journal of Spatial Science, DOI: 10.1080/14498596.2022.2059023

[11] Liqian Zhang, Qing Zhang, and Rui Zhao. "Progressive Dual-attention Residual Network for Salient Object Detection." IEEE Transactions on Circuits and Systems for Video Technology (2022).

[12] Ning Dai et al. "Style transformer: Unpaired text style transfer without disentangled latent representation." arXiv preprint arXiv:1905.05621 (2019).

11 Appendix

11.1 Code

The code can be found at github.com/iamsamliang/racoln.

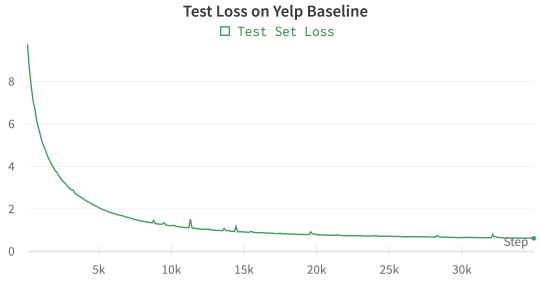
11.2 Code Inconsistencies

The code contains two major inconsistencies from their original paper. The first inconsistency is in their eval_output.py script, as the method queries a static jsonl file that is created before any model training, rather than a jsonl file that dynamically updates with every run. In fact, upon reaching out to the authors for an explanation, they agreed that their eval_output.py was erroneous, as **they had accidentally rm -rf'd** their original codebase and sought to re-do the code for Github from scratch, leading to several errors. Thus, we were able to fix and build a training pipeline that correctly obtained the BLEU and accuracy metrics, which is what was ultimately used in the ablation results.

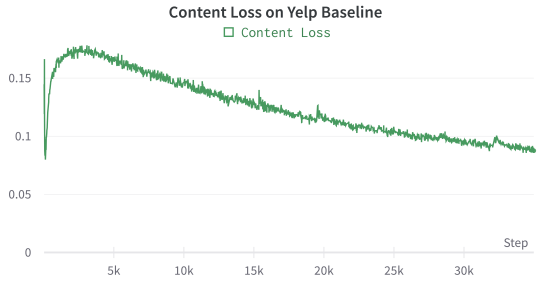
The second inconsistency is the perplexity results, as those values come from a broken eval_output.py script. Thus, we have noted that the perplexity can be disregarded for the purposes of this experiment because the language model was tested on an erroneous file output. As such, we do not include the metric in the ablation studies.

11.3 Ablation Results

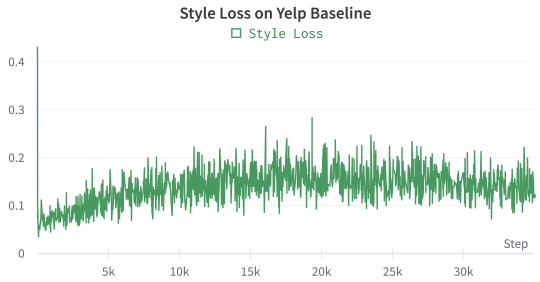
Below we have three sets of graphs: one that has remaining plots for the Yelp baseline; another that has graphs for the loss function ablations; and a last one that has graphs for the ablations related to model architecture (normal style embeddings, unidirectionality, no dropout).



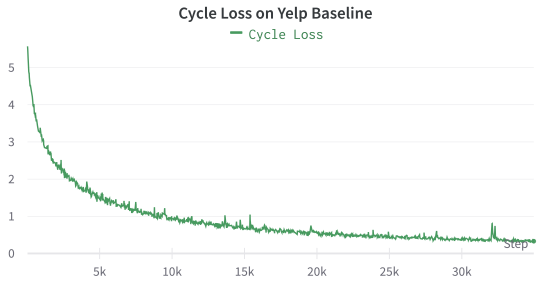
(a) Test loss at every 100 iterations



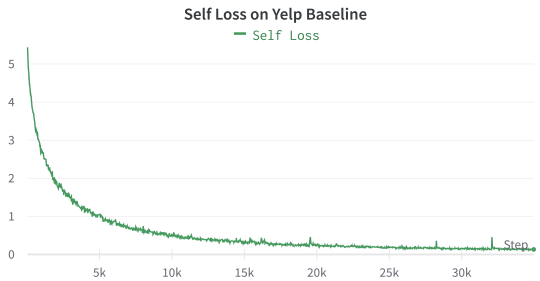
(b) Content component of the loss function during training



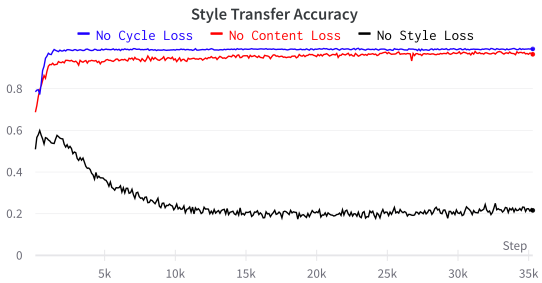
(c) Style component of the loss function during training



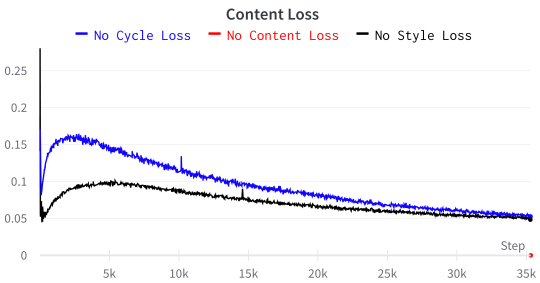
(d) Cycle Loss during Training



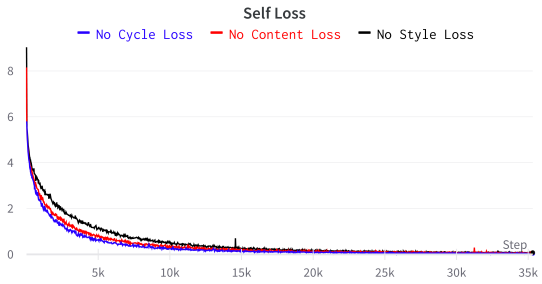
(e) Self Loss during Training



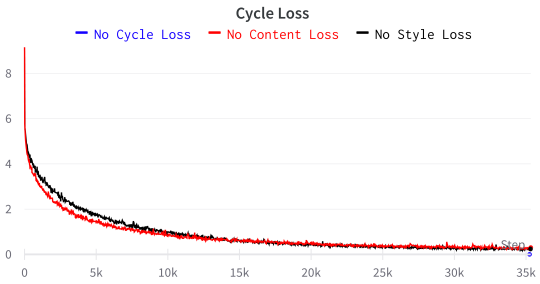
(a) Style Transfer Accuracy on Yelp Test Dataset



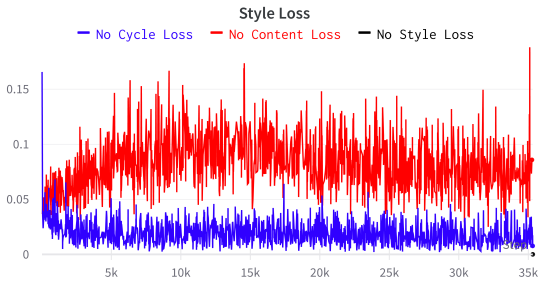
(a) Content Loss during Training



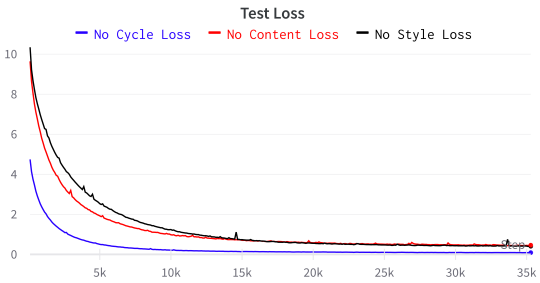
(b) Self Loss during Training



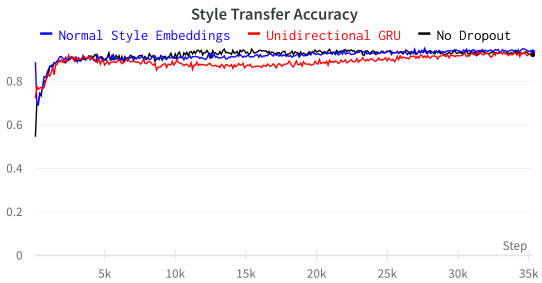
(b) Cycle Loss during Training



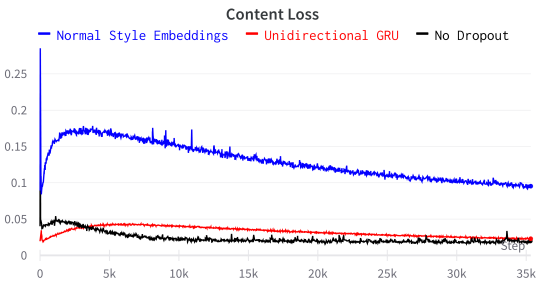
(c) Style Loss during Training



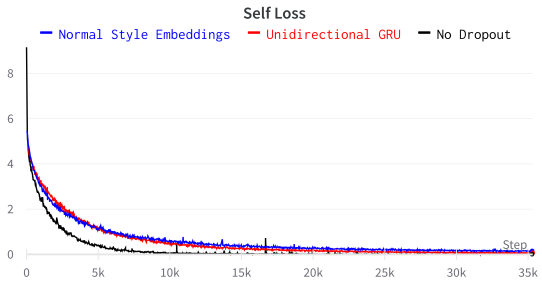
(c) Test Loss on Yelp Test Dataset



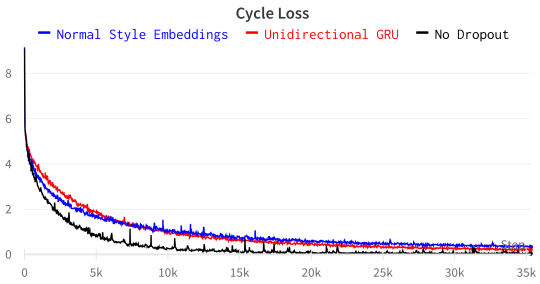
(a) Style Transfer Accuracy on Yelp Test Dataset



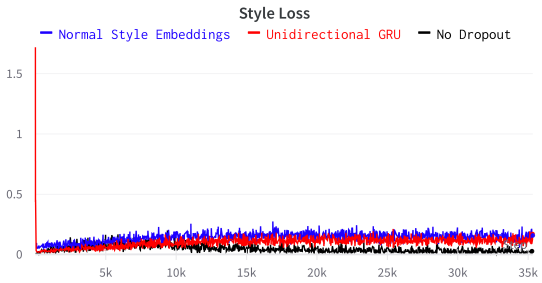
(a) Content Loss during Training



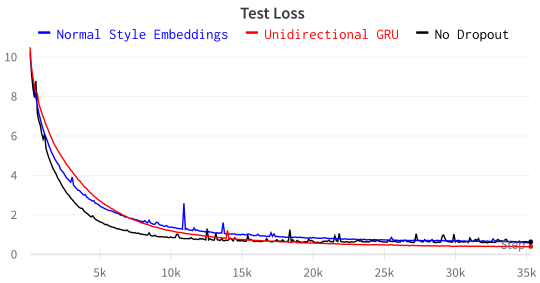
(b) Self Loss during Training



(b) Cycle Loss during Training



(c) Style Loss during Training



(c) Test Loss on Yelp Test Dataset