18th July, 2022

Review of Literature

Unsupervised Style Transfer for Text using Non-Parallel Data
(Work In Progress)

Samyukt Sriram

Ashoka University

Contents

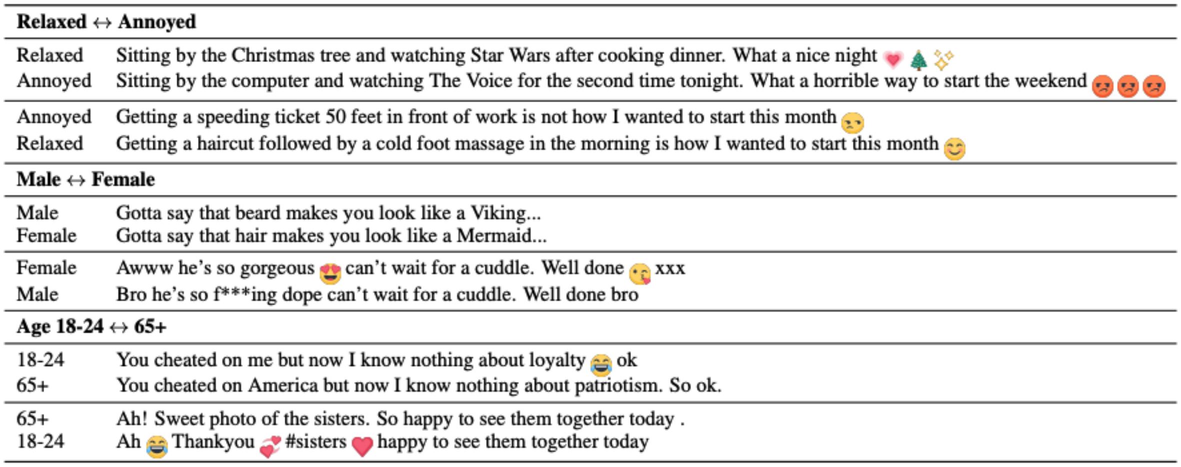
1. Introduction to the Problem

'Given 2 bodies of texts in 2 different styles, with no pairs of data, and an input text from one of these spaces. Return text with the 'content' of the input but written in the 'style' of the other text.'

The most basic task is to convert a statement's sentiment from positive to negative and vice versa. An example from Shen et al. 2017 [16] for positive to negative on a restaurant review:

> my appetizer was also very good and unique . -> my appetizer was also very cold and not fresh whatsoever .

Later papers use more interesting conversions, such as these examples from Lample et al. 2019 [7]:



Relaxed ↔ Annoyed	
Relaxed	Sitting by the Christmas tree and watching Star Wars after cooking dinner. What a nice night 💗🌲✨
Annoyed	Sitting by the computer and watching The Voice for the second time tonight. What a horrible way to start the weekend 😡😡😡
Annoyed	Getting a speeding ticket 50 feet in front of work is not how I wanted to start this month 😒
Relaxed	Getting a haircut followed by a cold foot massage in the morning is how I wanted to start this month 😊
Male ↔ Female	
Male	Gotta say that beard makes you look like a Viking...
Female	Gotta say that hair makes you look like a Mermaid...
Female	Awww he's so gorgeous 😍 can't wait for a cuddle. Well done 😘 xxx
Male	Bro he's so f***ing dope can't wait for a cuddle. Well done bro
Age 18-24 ↔ 65+	
18-24	You cheated on me but now I know nothing about loyalty 😂 ok
65+	You cheated on America but now I know nothing about patriotism. So ok.
65+	Ah! Sweet photo of the sisters. So happy to see them together today .
18-24	Ah 😂 Thankyou 💞 #sisters ❤️ happy to see them together today

Table 1: Our approach can be applied to many different domains beyond sentiment flipping, as illustrated here with example re-writes by our model on public social media content. The first line in each box is an input provided to the model with the original attribute, followed by its rewrite when given a different attribute value.

1.1 Applications and Need

Parallel data is incredibly hard to come by in text. This is because a human reader is often needed to correctly label and annotate samples. There also may not be large volumes of data available. For example, relatively unknown languages that are not commonly spoken in the present day may not have the parallel datasets needed for tasks like machine translation. One of the more interesting applications of this task is to rewrite potentially offensive statements using milder language / suggest corrections. [9]. Given the current state of social media, this may be a tool to make the internet a safer and sensitive place. **Not sure about this paper, it's not published and only cited 12 times. But it's an application example so?**

1.2 GANs and Image Manipulation

This kind of task has been done very successfully for image manipulation, so a good starting point is to get an idea of how image style transfer has been done in recent years. Generative Adversarial Networks (GANs) are used extensively in image style transfer tasks. [4] [14] [1]. A quick summary on how these works is as follows. There is a generator network that creates synthetic examples based on an input, and it is trained to minimize a loss function. The loss function is based on a comparison to the target image, a model called a discriminator extracts this loss from an instance of a synthetic example and the target. By defining the discriminator and the loss function in this way, the generator learns to create synthetic examples that are closer and closer to the target image [5].

Image tasks are handled by Convolutional Neural Networks (CNNs), where features can be extracted from the intermediate layers of the network. these can correspond to 'style' or 'content', or more broadly, any image has a latent W space in which the features of the image are disentangled. Because of this disentanglement, we can specify distinct losses for each of these [4]. More advanced models and methods leverage this fact about image representation in CNNs.

1.3 Challenges for Text Style Transfer

i. Text is not continuous like image pixel data. Pixel values can be varied by arbitrarily small values like + 0.001, and so the generator is able to respond to the discriminator's loss at these small levels of precision. For text however, varying words by these small amounts is just not how language works. Character level edits are meaningless, eg: 'are' and 'axe' are completely different words with wildly different contextual meanings. With word representations like GloVe and contextual encodings, the task could become finding the nearest word neighbour, which is computationally expensive. Words in these systems are represented as huge matrices, and the computation needed for finding a nearest neighbour in these spaces is considered too large to do at each step while training the model. **(verify this, maybe review the source below?)** This problem is known as the non-differentiability of discrete word tokens. Some more context and workarounds are proposed in this article by Leo Laugier, 2019 [8].

ii. The best generative models for text are based on RNNs / transformers. The architectures of these are completely different from CNNs. They are better at handling data that has some component of sequence / time, as is the case with text. But these models have no distinct 'feature' layers like CNNs, and it's not possible to 'extract' the individual layers / mechanisms that are responsible for style or content, unlike in CNNs. This means the disentanglement that is used for style transfer is difficult to realise in the models.

1.4 List of Papers Covered

A chronological approach was taken in selecting these papers. We begin with a couple of papers that proposed the 2 commonly used training mechanisms. The later papers refine and improve on these methods and introduce some interesting approaches / datasets.

i. Style Transfer from Non-Parallel Text by Cross-Alignment - Shen et al., NIPS 2017 [16]

ii. Multiple-Attribute Text Rewriting - Lample et al., ICLR 2019 [7]

iii. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer - Luo et al., 2019 [11]

iv. Unsupervised Text Style Transfer using Language Models as Discriminators - Yang et al., NeurIPS 2018 [18]

v. Unsupervised Text Style Transfer with Content Embeddings - Carslon et al., RANLP 2021 [2]

vi. Collaborative Learning of Bidirectional Decoders for Unsupervised Text Style Transfer - Ma et al., EMNLP 2021 [12]

vii. On Learning Text Style Transfer with Direct Rewards - Liu et al., Preprint 2021 [10]

A helpful list of papers, datasets and resources by Zhenxin Fu can be found here https://github.com/fuzhenxin/Style-Transfer-in-Text. **Is there a better way to credit / cite this**

2. Approaches Taken

The papers below broadly have 2 ways to train these models. There is often some combination of a generator and discriminator, similar to image manipulation. This is adversarial training and the focus is on disentanglement of the content and style. There is also a method called back-translation or cycle consistency loss. Later papers combine both these methods, bootstrapping with the cycle consistency loss and eventually switching to adversarial training. But each method comes with its advantages and disadvantages.

2.1 Paper Summaries

2.1.1 Style Transfer from Non-Parallel Text by Cross-Alignment - Shen et al., NIPS 2017 [16]

This is one of the first papers to pose this problem as a question of disentanglement and cross-alignment, formalizing the task. It has been used as a baseline in future works and the Encoder-Decoder architecture proposed has been refined and revised in future works.

i. The paper describes the main task in terms of disentanglement, where encoders and decoders are used to align the 2 given styles in a latent content space. The graphic below is from the paper:

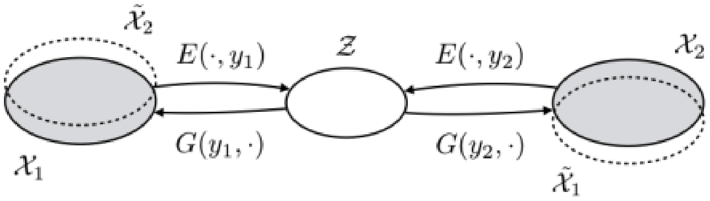


Figure 1: An overview of the proposed cross-alignment method. $\mathcal{X}_1$ and $\mathcal{X}_2$ are two sentence domains with different styles y_1 and y_2, and $\mathcal{Z}$ is the shared latent content space. Encoder E maps a sentence to its content representation, and generator G generates the sentence back when combining with the original style. When combining with a different style, transferred $\tilde{\mathcal{X}}_1$ is aligned with $\mathcal{X}_2$ and $\tilde{\mathcal{X}}_2$ is aligned with $\mathcal{X}_1$ at the distributional level.

Thinking about this problem in terms of encoding content from both texts into a shared latent space, and then pulling from it with a generator during decoding is a method that is seen in many subsequent papers.

ii. The model and methodology of the paper has been improved on in later works, but the essence of the models are similar to the ones proposed in this paper. The paper proposes a cross-aligned auto-encoder.

Here is a figure from the paper representing the alignment process: **Not 100 % clear on how this works, but i understand the high level concept**

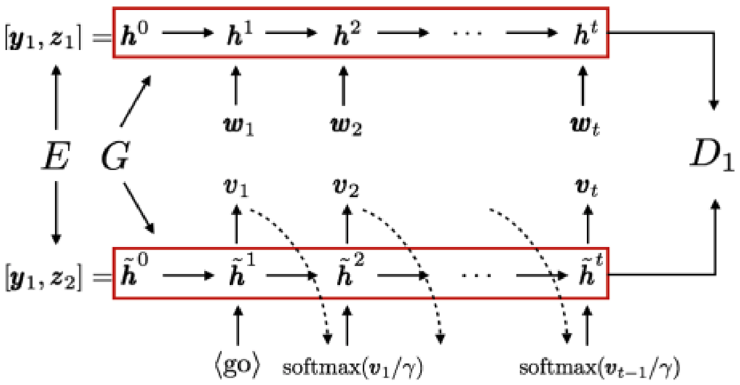
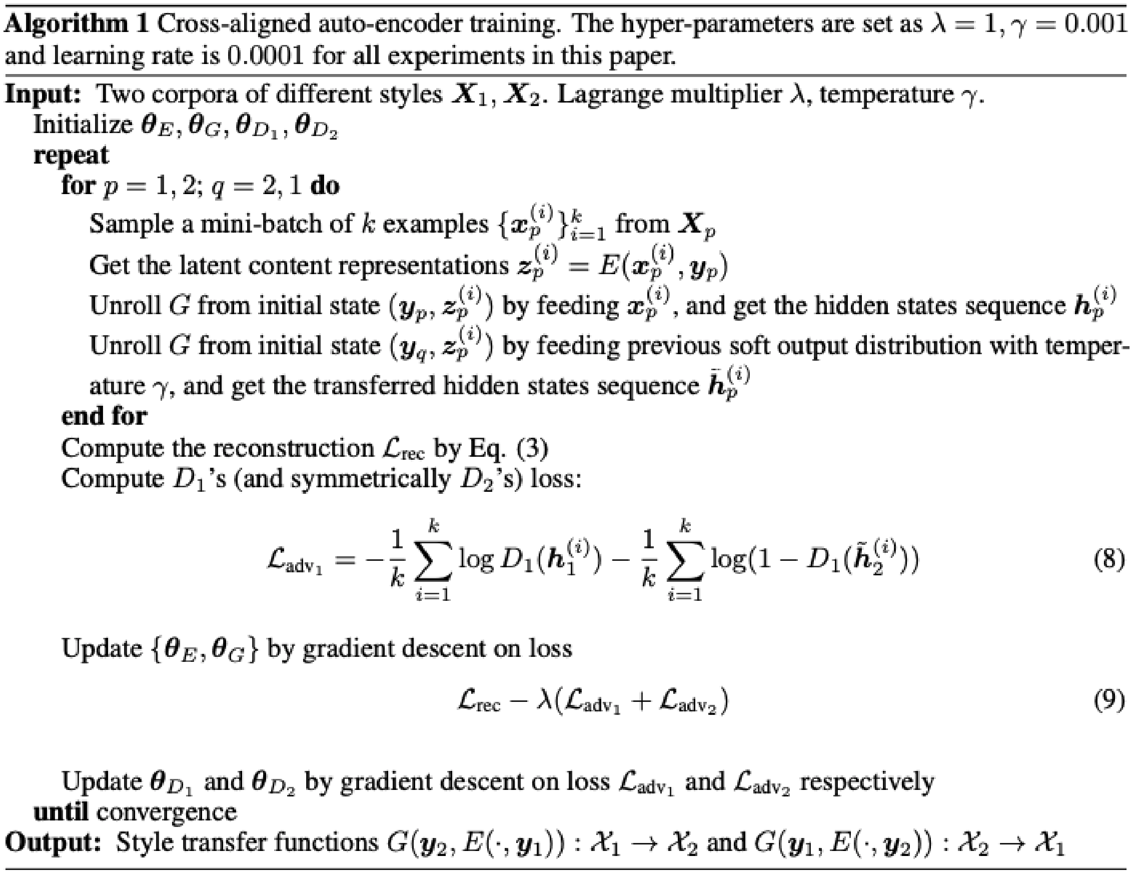

$[y_1, z_1] = h^0 \rightarrow h^1 \rightarrow h^2 \rightarrow \cdots \rightarrow h^t$
w_1
w_2
w_t
E
G
D_1
v_1
v_2
v_t
$[y_1, z_2] = \tilde{h}^0 \rightarrow \tilde{h}^1 \rightarrow \tilde{h}^2 \rightarrow \cdots \rightarrow \tilde{h}^t$
$\langle$go$\rangle$
softmax(v_1/γ)
softmax(v_{t-1}/γ)

Figure 2: Cross-aligning between x_1 and transferred x_2. For x_1, G is teacher-forced by its words $w_1 w_2 \cdots w_t$. For transfered x_2, G is self-fed by previous output logits. The sequence of hidden states $h^0, \cdots, h^t$ and $\tilde{h}^0, \cdots, \tilde{h}^t$ are passed to discriminator D_1 to be aligned. Note that our first variant aligned auto-encoder is a special case of this, where only h^0 and $\tilde{h}^0$, i.e. z_1 and z_2, are aligned.

The high-level explanation of this is that the model uses teacher-forcing or Professor-forcing [6] to train the generator. This technique creates a more stable training process but can prevent the generator from fully learning real examples. There are 2 Discriminators proposed in this paper, one for each generator. D1's job is to distinguish between real x1 and transferred x2, and D2's job is to distinguish between real x2 and transferred x1.

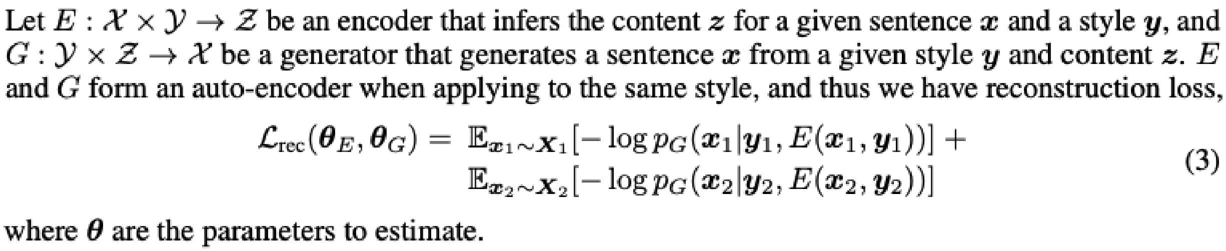
iii. Here is the full training algorithm proposed by the paper:



Algorithm 1 Cross-aligned auto-encoder training. The hyper-parameters are set as $\lambda = 1, \gamma = 0.001$ and learning rate is 0.0001 for all experiments in this paper.

```
Input: Two corpora of different styles $\boldsymbol{X}_1, \boldsymbol{X}_2$. Lagrange multiplier $\lambda$, temperature $\gamma$.
  Initialize $\boldsymbol{\theta}_E, \boldsymbol{\theta}_G, \boldsymbol{\theta}_{D_1}, \boldsymbol{\theta}_{D_2}$
  repeat
    for $p = 1, 2$; $q = 2, 1$ do
      Sample a mini-batch of $k$ examples $\{\boldsymbol{x}_p^{(i)}\}_{i=1}^k$ from $\boldsymbol{X}_p$
      Get the latent content representations $\boldsymbol{z}_p^{(i)} = E(\boldsymbol{x}_p^{(i)}, \boldsymbol{y}_p)$
      Unroll $G$ from initial state $(\boldsymbol{y}_p, \boldsymbol{z}_p^{(i)})$ by feeding $\boldsymbol{x}_p^{(i)}$, and get the hidden states sequence $\boldsymbol{h}_p^{(i)}$
      Unroll $G$ from initial state $(\boldsymbol{y}_q, \boldsymbol{z}_p^{(i)})$ by feeding previous soft output distribution with temperature $\gamma$, and get the transferred hidden states sequence $\tilde{\boldsymbol{h}}_p^{(i)}$
    end for
    Compute the reconstruction $\mathcal{L}_{\text{rec}}$ by Eq. (3)
    Compute $D_1$'s (and symmetrically $D_2$'s) loss:
        $\mathcal{L}_{\text{adv}_1} = -\frac{1}{k}\sum_{i=1}^{k} \log D_1(\boldsymbol{h}_1^{(i)}) - \frac{1}{k}\sum_{i=1}^{k} \log(1 - D_1(\tilde{\boldsymbol{h}}_2^{(i)}))$    (8)
    Update $\{\boldsymbol{\theta}_E, \boldsymbol{\theta}_G\}$ by gradient descent on loss
        $\mathcal{L}_{\text{rec}} - \lambda(\mathcal{L}_{\text{adv}_1} + \mathcal{L}_{\text{adv}_2})$    (9)
    Update $\boldsymbol{\theta}_{D_1}$ and $\boldsymbol{\theta}_{D_2}$ by gradient descent on loss $\mathcal{L}_{\text{adv}_1}$ and $\mathcal{L}_{\text{adv}_2}$ respectively
  until convergence
Output: Style transfer functions $G(\boldsymbol{y}_2, E(\cdot, \boldsymbol{y}_1)) : \mathcal{X}_1 \to \mathcal{X}_2$ and $G(\boldsymbol{y}_1, E(\cdot, \boldsymbol{y}_2)) : \mathcal{X}_2 \to \mathcal{X}_1$
```

Equation 3 is the reconstruction loss as referenced in the algorithm:



Let $E : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be an encoder that infers the content $\boldsymbol{z}$ for a given sentence $\boldsymbol{x}$ and a style $\boldsymbol{y}$, and $G : \mathcal{Y} \times \mathcal{Z} \to \mathcal{X}$ be a generator that generates a sentence $\boldsymbol{x}$ from a given style $\boldsymbol{y}$ and content $\boldsymbol{z}$. E and G form an auto-encoder when applying to the same style, and thus we have reconstruction loss,

$$\begin{aligned}\mathcal{L}_{\text{rec}}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G) = \; & \mathbb{E}_{\boldsymbol{x}_1 \sim \boldsymbol{X}_1}[-\log p_G(\boldsymbol{x}_1 | \boldsymbol{y}_1, E(\boldsymbol{x}_1, \boldsymbol{y}_1))] + \\ & \mathbb{E}_{\boldsymbol{x}_2 \sim \boldsymbol{X}_2}[-\log p_G(\boldsymbol{x}_2 | \boldsymbol{y}_2, E(\boldsymbol{x}_2, \boldsymbol{y}_2))]\end{aligned} \tag{3}$$

where $\boldsymbol{\theta}$ are the parameters to estimate.

iv. PROPOSED EVALUATION TASKS (Incomplete) **Unsure if this is worth including**

v. Some key takeaways:

(a) The big picture is that by defining a reconstruction loss and a discriminator adversarial training loss, the model is able to separate style and content. This approach is used in further papers extensively, with changes to the definitions of these losses.

(b) The paper uses RNNs for the encoder and decoder. Architectures like Transformers are far more sophisticated and future works use them for improvements in results.

(c) Training stability issues are a big challenge in this task. This paper tries to tackle them using the Professor forcing algorithm [6]. Future papers propose different ways to stabilize the training.

(d) One key point is that the discriminators used in this paper were binary classifiers, that detected whether a given sample was real or synthetically generated. There could be better ways of doing this, as proposed by a paper that uses a Language Model as a discriminator (discussed in detail later). [18].

2.1.2 Multiple-Attribute Text Rewriting - Lample et al., ICLR 2019 [7]

This paper questioned the approach of disentanglement and learning latent representations. These were the dominant approach to the task at the time, but the paper argues against the feasibility and necessity of disentanglement in the sense that Shen et al. 2017 [16] presented.
Instead, the paper develops the idea of back-translation for training. This is similar to the cycle-consistency loss and some works use them interchangeably.(**is there even a difference? just confirm this**) This is used in further architectures and better explained and demonstrated in papers like Luo et al. 2019 [11].
The paper also uses pooling operators to control the amount of content preservation, and attribute conditioning to handle multiple attributes in the style transfer.

i. The paper explores the idea that disentanglement is not actually achieved in practice, and might not be necessary for style transfer. The paper claims that the content z is disentangled from y only if y cannot be recovered from z. This is not the case in the models that are based on disentanglement. In the encoder representations of input, running a post-fit classifier on sentiment still yields significantly high accuracy. The discriminator used for training in the model does not maintain this accuracy. This means the classifier is still able to discern y, even in the supposedly disentangled content representation z from the encoder. **Maybe the significance of this can be better explored?**
While the paper does not assert that disentanglement is undesirable, it does claim that it may not be necessary for good results.

ii. Back-translation is the paper's alternative to the adversarial training used in other approaches. This is also known as cycle-consistency loss in other papers. It works by creating pseudo-parallel data for training. An input x is encoded and decoded to give y′, in another style. y′ is then encoded and decoded to return x″, and the difference in x and x″ can be used to train and create a reconstruction loss.
There are a few benefits to training this way. There is no adversarial training, so the issues surrounding unstable training due to binary discriminators is not present. It is also faster and is used to bootstrap training initially in future papers. However, using pseudo-parallel data can cause issues as well, because ultimately real data is not being used for training the model.

iii. The paper also uses attribute conditioning and latent representation pooling to overcome some issues associated with not using adversarial training.

Attribute conditioning allows the model to understand and work with multiple attributes (or styles). Each attribute is embedded and averaged, and then this is fed as a start-of-sequence symbol. This essentially flags out the information about style and attributes to the model, and so it can deal with them better.

Latent representation pooling is used to control the amount of content preservation in the model. This mechanism essentially balances the effects of individual word replacement and longer, content-losing sentences modifications. **Not entirely sure how much to explain / how to go about it**

The model used has 2-layer bidirectional LSTM as an encoder, and another 2-layer bidirectional LSTM augmented with an attention mechanism as a decoder.

iv. One of the datasets used in this paper comprised of public social media content. This allowed the paper to work with a diverse set of categories like gender, age group and writer annotated feeling. These kinds of modifications are more interesting to read and work with.

2.1.3 A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer - Luo et al., 2019 [11]

I'm unsure of the model and exact algo, should review this. how does MLE work

Building on the previous two approaches, this paper proposes a combination of adversarial training and back-translation, called a DualRL framework. The diagrams and algorithms in the paper are clear and well explained, and the approach of using both kinds of training is seen in future papers as well.

i. Here is a high level architecture overview of the paper's training method:

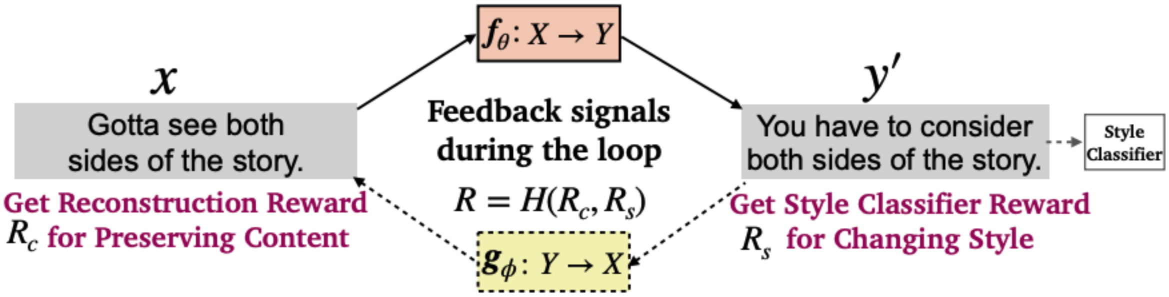

$f_\theta: X \rightarrow Y$
x
Gotta see both sides of the story.
Feedback signals during the loop
$R = H(R_c, R_s)$
y'
You have to consider both sides of the story.
Style Classifier
Get Reconstruction Reward R_c for Preserving Content
$g_\phi: Y \rightarrow X$
Get Style Classifier Reward R_s for Changing Style

Figure 1: The proposed DualRL framework for unsupervised text style transfer with an informal-to-formal text example, where both $\boldsymbol{f}_\theta$ and $\boldsymbol{g}_\phi$ are a sequence-to-sequence mapping model.

This process is repeated for both generators, so the overall step of training both of the models looks like:

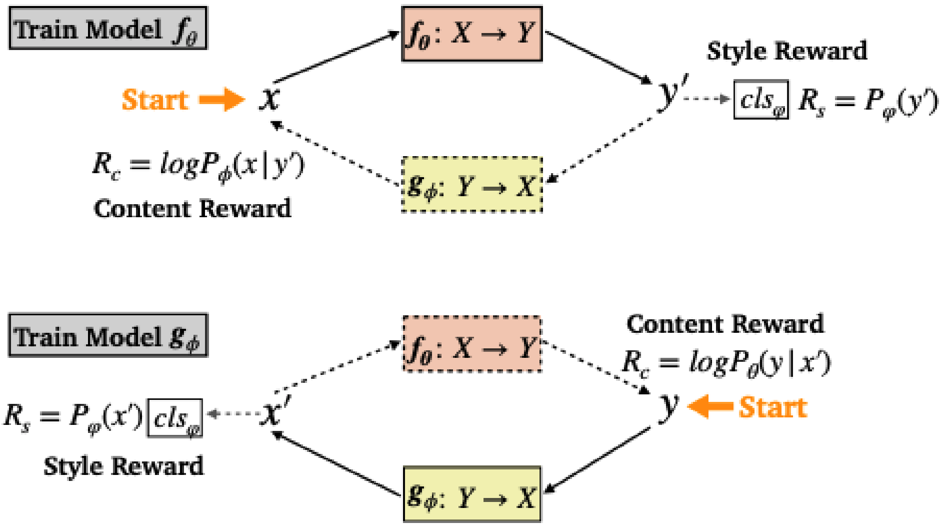

Train Model f_θ
$f_\theta: X \rightarrow Y$
Style Reward
Start
x
y'
cls_φ $R_s = P_\varphi(y')$
$R_c = logP_\phi(x|y')$
Content Reward
$g_\phi: Y \rightarrow X$
Train Model g_ϕ
$f_\theta: X \rightarrow Y$
Content Reward
$R_c = logP_\theta(y|x')$
$R_s = P_\varphi(x')$ cls_φ
x'
y
Start
Style Reward
$g_\phi: Y \rightarrow X$

Figure 2: Training process of DualRL. We alternately train the two mapping models f_θ and g_ϕ.

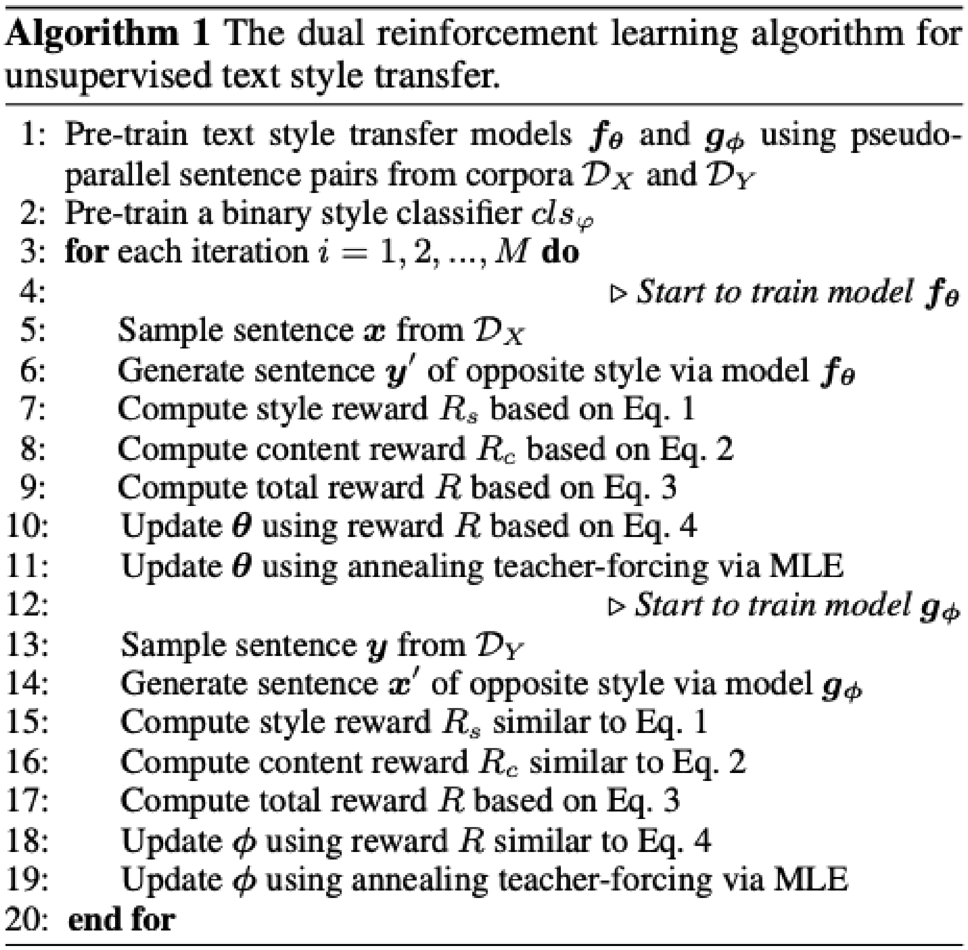
These diagrams depict the paper's proposed training method based on back-translation. This method is faster to train with as both generators are trained with pseudo a content preservation loss calculated when comparing x and the reconstructed x. This is unlike other adversarial training methods that have to calculate a content preservation loss on x and the transferred y'.

However, one drawback of this method is that there is likely to be a gap between the pseudo-parallel data generated by the model, and the real world input that the model will have to work with. Intuitively, we can expect the model to generate and then learn some patterns and biases based on pseudo-parallel data that may not be present in reality.

ii. To address this gap, the paper proposes a technique of annealing teacher-forcing for the dual reinforcement learning process. Teacher-forcing (or Professor forcing [6]) is a way to prevent the model from learning self-generated biases by correcting the hidden states used in further predictions. This is referred to as 'exposure bias' in the paper.

The model proposes that this is done in exponentially larger gaps. That is to say, a model is trained by back-translation, and at exponentially increasing intervals, one instance of teacher-forcing training is done. **Need more info on MLE and what teacher forcing actually looks like**

iii. The exact algorithm of training given in the paper: **Unsure if adding this is necessary**



Algorithm 1 The dual reinforcement learning algorithm for unsupervised text style transfer.

```
Pre-train text style transfer models $f_\theta$ and $g_\phi$ using pseudo-parallel sentence pairs from corpora $\mathcal{D}_X$ and $\mathcal{D}_Y$
Pre-train a binary style classifier $cls_\varphi$
for each iteration $i = 1, 2, ..., M$ do
                                        ▷ Start to train model $f_\theta$
    Sample sentence $x$ from $\mathcal{D}_X$
    Generate sentence $y'$ of opposite style via model $f_\theta$
    Compute style reward $R_s$ based on Eq. 1
    Compute content reward $R_c$ based on Eq. 2
    Compute total reward $R$ based on Eq. 3
    Update $\theta$ using reward $R$ based on Eq. 4
    Update $\theta$ using annealing teacher-forcing via MLE
                                        ▷ Start to train model $g_\phi$
    Sample sentence $y$ from $\mathcal{D}_Y$
    Generate sentence $x'$ of opposite style via model $g_\phi$
    Compute style reward $R_s$ similar to Eq. 1
    Compute content reward $R_c$ similar to Eq. 2
    Compute total reward $R$ based on Eq. 3
    Update $\phi$ using reward $R$ similar to Eq. 4
    Update $\phi$ using annealing teacher-forcing via MLE
end for
```

The style reward in this paper is similar to what we've seen in earlier works. A binary classifier is used as a discriminator to distinguish real and 'fake' (generated) sentences. The content reward is estimated as the probability that the original sentence x is reconstructed, given a transferred input y'. The paper mentions that measuring content preservation with BLEU was not as successful for their work.

The total reward is calculated as a harmonic mean of the above rewards. The paper claims this balances both style transfer and content preservation. Interesting to note is that fact that there is no overall coherence loss provided, as is seen in some other papers. It seems the paper proposes the annealing teacher-forcing method to maintain coherence in the model architecture.

2.1.4 Unsupervised Text Style Transfer using Language Models as Discriminators - Yang et al., NeurIPS 2018 [18]

In the models and papers we've seen so far that involve adversarial training, the discriminator for style transfer is usually a binary classifier that classifies sentences as real or 'fake'. However, this approach only evaluates sentences as a whole, meaning that there is not as much information on the individual tokens for the generator to train on. Given the issues with unstable training that are present in text generation, the usage of Language Models (LMs) as discriminators could allow more token level feedback to reach the generator for training. This paper describes and implements a LM as a discriminator in an adversarial training model.

i. The paper proposes using a trained LM instead of a conventional binary classifier. The LM is able to calculate an entire sentence likelihood as a function of token probabilities. Because the LM is able to score predictions at the token-level, it offers a more stable and useful signal to train the generator on. This is done by continuous approximations of discrete sampling under the generator, diagrammatically represented as:

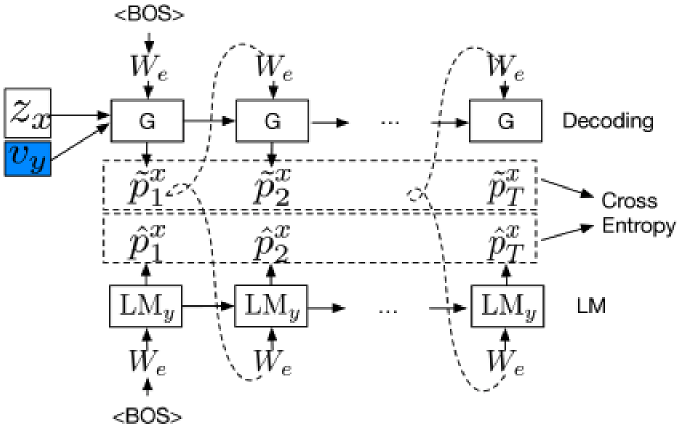


Figure 2: Continuous approximation of language model loss. The input is a sequence of probability distributions $\{\tilde{\mathbf{p}}_t^{\mathbf{x}}\}_{t=1}^T$ sampled from the generator. At each timestep, we compute a weighted embedding as input to the language model and get the sequence of output distributions from the LM as $\{\hat{\mathbf{p}}_t^{\mathbf{x}}\}_{t=1}^T$. The loss is the sum of cross entropies between each pair of $\tilde{\mathbf{p}}_t^{\mathbf{x}}$ and $\hat{\mathbf{p}}_t^{\mathbf{x}}$.

We can see that the style loss becomes a function of the cross entropy loss between the predictions of the LM trained on a specific style, and the predictions of the generator being trained.

ii. The paper mentions the LM's preference for shorter text generation, which means that the model overall suffers from mode collapse. If the LM prefers to predict shorter sequences, it may be difficult for the model to learn from longer inputs. The paper addresses this by normalizing loss by length and fixing the lengths of the predicted x. But maybe it's worth looking into what other biases and preferences a LM may bring into the model, and how these could be beneficial or detrimental to training.

iii. It may be possible to use this mechanism in other models. As the paper simply replaces a binary classifier, implementing it in other architectures may be worth looking into. **maybe mention these 2 points in Areas of Exploration as well?**

2.1.5 Unsupervised Text Style Transfer with Content Embeddings - Carslon et al., RANLP 2021 [2]

This paper uses the XLM model developed by Lample and Conneau 2019 [3] but implements an interesting content embedding stage for the model. This paper looks at style transfer more holistically, in that it uses the phrase 'content' to describe the kind of text in question. It can be interpreted to refer to style, overall structure of language, the 'voice' of the writing, etc.

The paper also uses an interesting dataset and evaluation method for this task. The paper divides the bible into several divisions based on the 'content' of the books, and then uses these as the basis for the problem of style transfer. This takes style transfer beyond direct and narrow things like sentiment, mood, etc. and implements it for a more holistic idea of style.

This is the proposed addition of content embeddings in the paper:

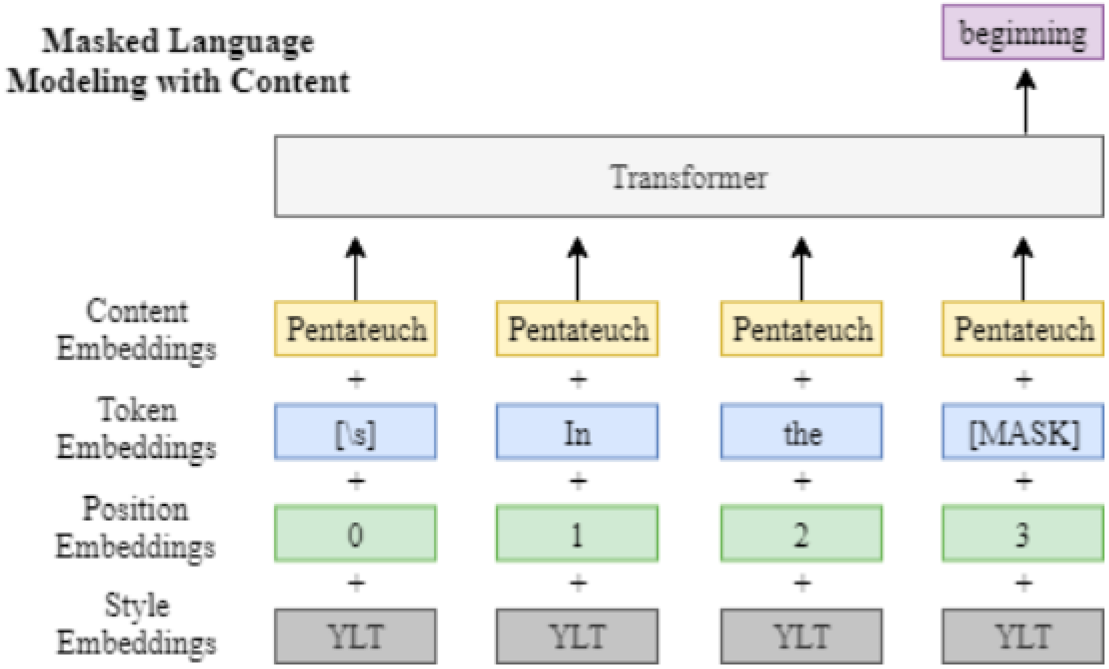


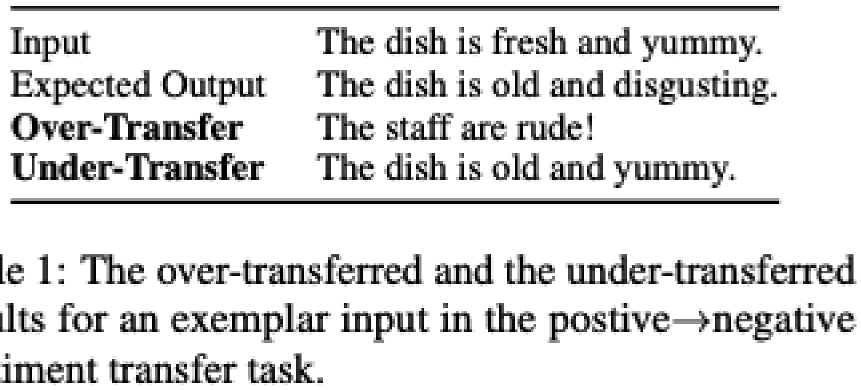
Figure 1: "XLM + Content" model training on the MLM objective. Based on Figure 1 of Lample and Conneau (2019). The choice of types for content embeddings are human-assigned before training as seen in Table 1.

'Pentateuch' is the name of one of the divisions of the Bible identified by the authors of the paper. Each such division has a different 'voice', or a more holistic way of describing style. The paper requires each kind of content to be labelled and passed into the XLM model as a content embedding. This is similar to the Attribute Conditioning described in Lample et al. 2019 [7].

2.1.6 Collaborative Learning of Bidirectional Decoders for Unsupervised Text Style Transfer - Ma et al., EMNLP 2021 [12]

This paper describes the issues in generated output as OverTransfer (OT) or UnderTransfer (UT). These problems are characterized as decoding issues, and so the paper proposes a collaborative learning mechanism with 2 bidirectional decoders. The paper also claims that certain architectures are more prone to a specific kind of error. Disentanglement based methods can cause OT due to inaccurate content and style separation. Back-translation methods can cause UT by strengthening those patterns in the psuedo-data that the training is based on. **I am not clear on why this paper's model works / how it works / why it's better. Have to review**

i. The paper presents the OverTransfer (OT) and UnderTransfer (UT) problems as follows:



Input	The dish is fresh and yummy.
Expected Output	The dish is old and disgusting.
Over-Transfer	The staff are rude!
Under-Transfer	The dish is old and yummy.

Table 1: The over-transferred and the under-transferred results for an exemplar input in the postive→negative sentiment transfer task.

This is an interesting perspective on how we can think about the style transfer vs content preservation problem. It gives us 2 distinct issues that could occur with the models in general. It may be worth evaluating and interpreting future models and their outputs on these lines.

ii. The main push of the paper is the proposal for a new decoder to resolve the issues of OT and UT. The paper uses 2 decoders; one Left to Right and one Right to Left decoder. A discriminator is used on the decoders themselves to prevent them from collapsing into the same patterns. Because the knowledge acquisition process for both the decoders is different, the correct predictions of the model are reinforced while the incorrect ones are not. **I am not clear on this topic at all, have to read more.**

iii. The paper reports that OT problems are dealt with very well by this model. UT issues are not as well-handled, but not worse than contemporary SOTA models either. UT seems to be a challenge for all models the paper tested.

2.1.7 On Learning Text Style Transfer with Direct Rewards - Liu et al., Preprint 2021 [10]

This paper synthesizes some of the ideas and concepts we've seen above. It uses 4 loss functions in training, and uses both back-translation and adversarial training. The paper also uses a fine-tuned version of GPT-2 [15] as a generator, restructuring the problem as a text generation problem as opposed to the encoder - decoder architectures we've seen earlier.

i. The paper uses GPT-2 [15] and fine-tunes it for the task. Since GPT-2 is a unidirectional language model with only a decoder, the task is reframed as a sequence completion task. An input sentence is concatenated with a style token, which informs the generator what style the following generated sequence should be in.

   This is an interesting approach, and it seems that by using a decoder only model as a base, there is no explicit disentanglement in the model. The paper claims that the specified loss functions are effective enough at style transfer and content preservation.

ii. On evaluating content preservation, the paper mentions that commonly used n-gram similarity methods like BLEU [13] may be inadequate. Using cycle-consistency loss also has issues, as reconstruction and content preservation through transfer are not necessarily equivalent objectives.

iv. The paper uses 4 loss functions:

   (a) Style Transfer Accuracy is rewarded by a CNN based classifier, whose objective is to predict the likelihood of the output sentence being coherent to the target style.

   (b) Content Preservation is rewarded by the SIM model [17], this model measures semantic (content) similarity between a pair of sentences. The cosine similarity of sentence representations is calculated which are constructed by averaging sub-word embeddings.

   The exact model used by the paper is called SiMiLe, which also uses a length penalty to control for length in generated texts. Using such models instead of BLEU may also have a drawback. SIM is pre-trained on a corpora of english language, and this may be incompatible with some tasks that involve styles that are significantly different than modern English.**Verify this!**

   The paper uses cycle-consistency loss to bootstrap training in the initial stages, as it may be a good technique to initially train the model to a degree. However main training is done similar to adversarial training.

   (c) Fluency of output is also rewarded in this paper. A pre-trained GPT-2 model is used only for evaluating fluency, which is calculated as the difference in perplexity between the source sentence and the generated output. **unclear on which body of text GPT2 is pretrained on, intuitively think it must be the target style text**

The is somewhat similar to Yang et al., NeurIPS 2018 [18] and their usage of Language Models as a discriminator to determine style, except here the focus is on making the output coherent and consistent.

(d) The paper claims that the above 3 losses are not enough to ensure the output sounds as fluent and coherent, so the authors also include a naturalness loss. This is simply a discriminator that calculates the log-likelihood of outputs being classified as real sentences, so this functions very similar to an adversarial loss function that has been seen many times before.

v. The training is done in 2 stages. First, in the bootstrap stage, the back-translation method is used. This warms up the model, after which the output quality is improved and fine tuned using all the 4 losses described above. When training with style transfer and naturalness loss, the paper mentions that an alternative method is used to generate token-level signals to the generator, instead of sentence level signals. This is a technique used to stabilize training, and it provides the generator more information to learn from.

3. Possible Areas of Exploration - WIP

3.1 Architecture and Training

i. Maybe building a classifier that can classify an output into OT and UT based on an input and expected output? This can then be used to balance content preservation and style transfer.
Why would this be useful? What point in the architecture does this come in? **think more about why**
If we can classify the model's output as OT, we can reduce the style transfer loss weight. If the model is primarily UT, we can increase the weight of style transfer loss. This way we can better adjust the hyper-parameters for content preservation and style transfer losses in all models. Having a classifier will give us a quantitative metric to compare these 2.
Further, some architectures are more prone to either UT (back-translation) or OT (disentanglement). Further, some advanced architectures use both methods. By knowing which kind of error our model is making, we can balance the weightage given to either training process.
However, it seems you can get an idea of how much UT / OT is there using the loss mechanisms you define for the model. But other models haven't been evaluated along these lines. Specifically, other papers don't ask "Are we having an OT or UT problem?", but rather, "Is there too much content preservation or too much style transfer?" **So? Is there a meaningful difference between these two things? Think more.**

ii. Using a Language Model as a discriminator in some existing architectures could improve them.
Current approaches either use sentence level discriminators [16] or there is some mechanism to give token-level feedback through teacher forcing. Could look into whether using a language model trained on the style is better, and if there even is a significant difference [18].

iii. First identify which parts of the text are likely to get replaced, then replace them. Eg: first identify the adjectives in the sentence, then generate new ones in the target style. Don't replace nouns, ie, words that are likely to be content related and not style related.
This could be a way to avoid OT and UT problems by assigning a benefit to preserving certain parts of speech. Similar to a pointer-generator? points to a part of the text that needs replacement, and then generates a corresponding piece of text with the transferred language? **need to look into if this has been done**

3.2 Datasets

i. Something about longer datasets and outputs, the current ones are usually one-liners. This could be because of how hard the task is, but it would be interesting to see if there are specific challenges to transferring on the paragraph level.

ii. Using the Bible dataset might be interesting. Allows for a more wholistic representation of 'style', going beyond small sentence manipulation.

iii. Running some of the models on social media hate speech re-writing.

4. Moving Forward - WIP

Bibliography

[1] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. "Restyle: A residual-based stylegan encoder via iterative refinement". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6711–6720.

[2] Keith Carlson, Allen Riddell, and Daniel Rockmore. "Unsupervised text style transfer with content embeddings". In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. 2021, pp. 226–233.

[3] Alexis Conneau and Guillaume Lample. "Cross-lingual language model pretraining". In: *Advances in neural information processing systems* 32 (2019).

[4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "A neural algorithm of artistic style". In: *arXiv preprint arXiv:1508.06576* (2015).

[5] Shweta Goyal. "GANs — A Brief Introduction to Generative Adversarial Networks". In: (2019). URL: `https://medium.com/analytics-vidhya/gans-a-brief-introduction-to-generative-adversarial-networks-f06216c7200e`.

[6] Alex M Lamb et al. "Professor forcing: A new algorithm for training recurrent networks". In: *Advances in neural information processing systems* 29 (2016).

[7] Guillaume Lample et al. "Multiple-attribute text rewriting". In: *International Conference on Learning Representations*. 2018.

[8] Léo Laugier. "Workarounds for the non-differentiability of sampling when generating text". In: (2019). URL: `https://leolaugier.wp.imt.fr/2019/09/09/workarounds-non-differentiability/`.

[9] Léo Laugier et al. "Civil rephrases of toxic texts with self-supervised transformers". In: *arXiv preprint arXiv:2102.05456* (2021).

[10] Yixin Liu, Graham Neubig, and John Wieting. "On learning text style transfer with direct rewards". In: *arXiv preprint arXiv:2010.12771* (2020).

[11] Fuli Luo et al. "A dual reinforcement learning framework for unsupervised text style transfer". In: *arXiv preprint arXiv:1905.10060* (2019).

[12] Yun Ma et al. "Collaborative learning of bidirectional decoders for unsupervised text style transfer". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 9250–9266.

[13] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[14] Or Patashnik et al. “Styleclip: Text-driven manipulation of stylegan imagery”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2085–2094.

[15] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.

[16] Tianxiao Shen et al. “Style transfer from non-parallel text by cross-alignment”. In: *Advances in neural information processing systems* 30 (2017).

[17] John Wieting et al. “Simple and effective paraphrastic similarity from parallel translations”. In: *arXiv preprint arXiv:1909.13872* (2019).

[18] Zichao Yang et al. “Unsupervised text style transfer using language models as discriminators”. In: *Advances in Neural Information Processing Systems* 31 (2018).