Style Transformer: Unpaired Text Style Transfer without Disentangled Latent Representation

Introduction

The paper starts by pointing out the problems related to existing models which use disentangled latent representation of a sentence to separate out content/meaning and style. Hence although disentangled latent representation brings better interpretability, it faces the following problems -

1) It is not easy to judge the quality of disentanglement and to clearly separate out semantics and style.
2) Overwriting may be a better option rather than disentanglement.
3) Latent representation does not efficiently capture the rich semantics of long texts and also, due to the fixed size vector in the latent space, attention mechanisms cannot be directly used to preserve the information of the input sequence.

Finally the problem of text-style transfer itself is a bit complicated since we don't have a large database of paired sentences with opposite styles and identical semantics. Hence the model itself needs to develop a self-supervision mechanism to test the correctness.

Hence the paper proposes a Transformer based fully connected, self attention network for text-style transfer.

Problem Formalisation

The goal of style transfer is that: given a arbitrary natural language sentence x and a desired style $\hat{s} \in \{s^{(i)}\}^{K}_{i=1}$, where K are the number of different style databases from which we get our x, rewrite this sentence to a new one $\hat{x}$ which has the style $\hat{s}$ and preserve the information in original sentence x as much as possible.

Model Overview

The goal of the model is to learn a mapping function $f_{\Theta}(x,s)$ where x is a natural language sentence and s is a style control variable.The output of this function is the transferred sentence $\hat{x}$ for the input sentence x.

To tackle the problem of lack of paired sentences available for supervision, the paper implements a discriminator network to create supervision from non-parallel corpora.

Style Transformer Network

This network is a standard transformer encoder-decoder network, except for one addition, the encoder instead of just getting the input sentence x , also takes in the style control variable s in the input. Hence the encoder Enc(x,s; θ_E) maps sentence x and s to a sequence of continuous representations z = (z1,z2,….,zn). And the Transformer decoder Dec(z; θ_D) estimates the conditional probability for the output sentence y = (y1, y2, ..., yn) by auto-regressively factored its as:

$$p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{s}) = \prod_{t=1}^{m} p_\theta(y_t|\mathbf{z}, y_1, ..., y_{t-1}).$$

At each time step t, the probability of the next token is computed by a softmax classifier:

$$p_\theta(y_t|\mathbf{z}, y_1, ..., y_{t-1}) = \mathrm{softmax}(\mathbf{o}_t)$$

The predicted output sentence of this network by $f_\theta(x, s)$.

Discriminator Network

So as we only have supervision for the case $f_\theta(x, s)$ where x and s are from the same dataset, the model should regenerate the same sentence when these are input to the network.

For the case when supervision is required for $f_\theta(x,\hat{s})$, where s != ŝ, the paper takes the help of discriminator network to do supervision by -

1) For content preservation when $\hat{y}$ = $f_\theta(x,\hat{s})$ is given as the input to the network with the original style of x = s. It should regenerate the same sentence x.
2) For style controlling, the discriminator network is trained to better control the style of the sentence. It learns to distinguish between different styles of sentences and thus helps in supervision.

The two types of discriminator networks proposed are -

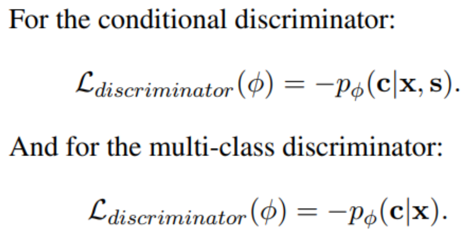
1) Conditional - In this type, a sentence and a style is given as input to the discriminator $d_\varphi(x, s)$, and the network returns a bool value

if the sentence is of the given style or not. In discriminator training phase, the network is trained to give positive and negative values for same input sentence x and s and transformed sentence $\hat{y}$ = $f_\theta(x,\hat{s})$ and s respectively. During the style transformer training phase the network f_θ is trained to maximise the probability of positive when fed $f_\theta(x,\hat{s})$ and ŝ to the discriminator.

2) Multi-class- In this case, the discriminator $d_\varphi(x)$ is just fed with the input sentence and it predicts the style of the sentence by classifying it into one of the K classes or the last stand class for the generated data $f_\theta(x,\hat{s})$ (fake sample or transformed sentence). Similar to the previous case, during the discriminator training phase, the model trains to predict the input, reconstructed sentences and transformed sentences correctly to the corresponding class. And during the style network training phase, the model is trained to maximise the probability that the style for generated sequence is class stand.

Training algorithm

1) Discriminator Learning - As specified earlier, the network is trained for better classification of sentences. The loss functions will be -



For the conditional discriminator:

$$\mathcal{L}_{discriminator}(\phi) = -p_\phi(\mathbf{c}|\mathbf{x}, \mathbf{s}).$$

And for the multi-class discriminator:

$$\mathcal{L}_{discriminator}(\phi) = -p_\phi(\mathbf{c}|\mathbf{x}).$$

2) Style Transformer Learning - Depending upon various cases of $f_\theta(x,\hat{s})$ when ŝ = s or not, it is trained differently.
   a) Self reconstruction - when x and s are input, the model should return x itself. The loss function will simply be-

$$\mathcal{L}_{self}(\theta) = -p_\theta(\mathbf{y} = \mathbf{x}|\mathbf{x}, \mathbf{s}).$$

   b) Cycle-reconstruction - when transformed sentence and s are input to the network, it should return x. Loss function will be-

$$\mathcal{L}_{cycle}(\theta) = -p_\theta(\mathbf{y} = \mathbf{x}|f_\theta(\mathbf{x}, \widehat{\mathbf{s}}), \mathbf{s}).$$

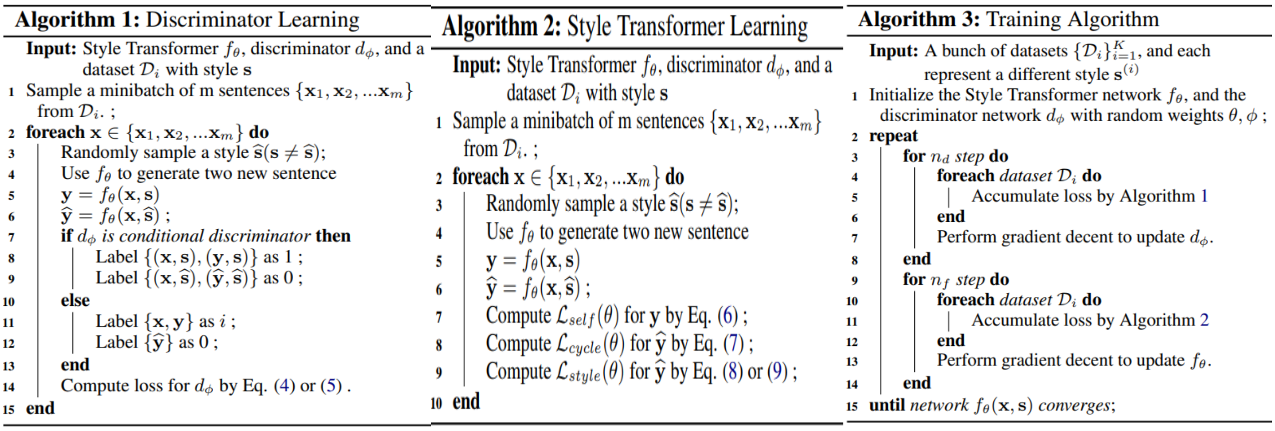
c) Style Controlling - To avoid the model to let it learn just to regenerate the input sentence x, the paper introduced one more loss function to correctly predict the style of the transformed sentence. The loss function depending upon the discriminator network used is-

$$\mathcal{L}_{style}(\theta) = -p_\phi(\mathbf{c} = 1 | f_\theta(\mathbf{x}, \widehat{\mathbf{s}}), \widehat{\mathbf{s}}). \quad \text{(conditional)}$$

$$\mathcal{L}_{style}(\theta) = -p_\phi(\mathbf{c} = \widehat{\mathbf{s}} | f_\theta(\mathbf{x}, \widehat{\mathbf{s}})). \quad \text{(multi-class)}$$

A final subtle point was written in the paper, about the difficulty in backpropagation. . Because of the discrete nature of the natural language, for the generated sentence $\hat{y} = f_\theta(x,\hat{s})$ we can’t directly propagate gradients from the discriminator through the discrete samples.To solve this problem the model feeds the entire softmax of the generated output downstream. And instead of taking a greedy decoding, it took a weighted average embedding to finally train the network.

The pseudo code as given in the paper for training the networks are-



Algorithm 1: Discriminator Learning

```
Input: Style Transformer $f_\theta$, discriminator $d_\phi$, and a
       dataset $\mathcal{D}_i$ with style $\mathbf{s}$
Sample a minibatch of m sentences $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$
  from $\mathcal{D}_i$. ;
foreach $\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$ do
    Randomly sample a style $\widehat{\mathbf{s}}(\mathbf{s} \neq \widehat{\mathbf{s}})$;
    Use $f_\theta$ to generate two new sentence
    $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$
    $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ ;
    if $d_\phi$ is conditional discriminator then
        Label $\{(\mathbf{x}, \mathbf{s}), (\mathbf{y}, \mathbf{s})\}$ as 1 ;
        Label $\{(\mathbf{x}, \widehat{\mathbf{s}}), (\widehat{\mathbf{y}}, \widehat{\mathbf{s}})\}$ as 0 ;
    else
        Label $\{\mathbf{x}, \mathbf{y}\}$ as $i$ ;
        Label $\{\widehat{\mathbf{y}}\}$ as 0 ;
    end
    Compute loss for $d_\phi$ by Eq. (4) or (5) .
end
```

Algorithm 2: Style Transformer Learning

```
Input: Style Transformer $f_\theta$, discriminator $d_\phi$, and a
       dataset $\mathcal{D}_i$ with style $\mathbf{s}$
Sample a minibatch of m sentences $\{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$
  from $\mathcal{D}_i$. ;
foreach $\mathbf{x} \in \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_m\}$ do
    Randomly sample a style $\widehat{\mathbf{s}}(\mathbf{s} \neq \widehat{\mathbf{s}})$;
    Use $f_\theta$ to generate two new sentence
    $\mathbf{y} = f_\theta(\mathbf{x}, \mathbf{s})$
    $\widehat{\mathbf{y}} = f_\theta(\mathbf{x}, \widehat{\mathbf{s}})$ ;
    Compute $\mathcal{L}_{self}(\theta)$ for $\mathbf{y}$ by Eq. (6) ;
    Compute $\mathcal{L}_{cycle}(\theta)$ for $\widehat{\mathbf{y}}$ by Eq. (7) ;
    Compute $\mathcal{L}_{style}(\theta)$ for $\widehat{\mathbf{y}}$ by Eq. (8) or (9) ;
end
```

Algorithm 3: Training Algorithm

```
Input: A bunch of datasets $\{\mathcal{D}_i\}_{i=1}^{K}$, and each
       represent a different style $\mathbf{s}^{(i)}$
Initialize the Style Transformer network $f_\theta$, and the
  discriminator network $d_\phi$ with random weights $\theta, \phi$ ;
repeat
    for $n_d$ step do
        foreach dataset $\mathcal{D}_i$ do
            Accumulate loss by Algorithm 1
        end
        Perform gradient decent to update $d_\phi$.
    end
    for $n_f$ step do
        foreach dataset $\mathcal{D}_i$ do
            Accumulate loss by Algorithm 2
        end
        Perform gradient decent to update $f_\theta$.
    end
until network $f_\theta(\mathbf{x}, \mathbf{s})$ converges;
```

Experimental Evaluation

1) Automatic Evaluation -
    a) Style Control - Target sentiment accuracy of the transferred sentences
    b) Content Preservation - Two types of BLEU scores were calculated self-BLEU (with respect to input sentence) and ref-BLEU(with respect to human reference). Higher the BLEU score, better is the content preservation

c) Fluency - measured by the perplexity of transferred sentence.

Although the perplexity scores were not so good, the BLEU and accuracy scores were good.

2) Human Evaluation - The model performed better than all others in almost all metrics of accuracy, content and fluency.

Ablation Study

By disabling various loss functions and doing the analysis, the results show that -

1) The cycle reconstruction loss is able to encourage the model to preserve the information from the input sentence.
2) When the discriminator loss is not used, the model quickly degenerates to a model which is only copying the input sentence to output without any style modification since it only trains on reconstruction loss.
3) The self-reconstruction loss guides the model to generate readable natural language sentences

Also it's necessary to use both input real sentences and generated sentences for the correctness of the model.

This concludes the report.