Controllable Unsupervised Text Attribute Transfer via Editing Entangled Latent Representation

Introduction

The paper first points out the problems with disentanglement of text to separately model attribute and content-

1) Disentanglement undermines the integrity and readability of the generated content
2) They lack flexibility and controllability because of modelling each attribute separately

Instead the paper uses a different approach. It generates a combined entangled latent representation of the using the Transformer encoder and then uses the classifier and Fast-Gradient-Iterative-Modification (FGIM) algorithm to iteratively edit the latent representation, until the latent representation can be identified as target attribute by the classifier. Then finally decode the sentence.This model can provide some additional benefits as well like controlling the degree of style transfer and also allowing to transfer over multiple aspects.The model modifies the latent representation in the direction that highly activates the classifier.

Problem Formalisation

A dataset is given with paired sentences x and its attribute vector y, which may contain one or more attributes of the sentence like tense, sentiments, overall,etc. In general, the problem boils down to, given a source sentence x and target attribute y’, the model should generate a new sentence x’ which preserves the content of x and has the attributes close to y’.

Model Overview

As described in the paper, the model consists of an encoder network $E_{\theta e}$ which maps the input sentence x to latent space representation z. It consists of a decoder network $D_{\theta d}$ and an attribute classifier $C_{\theta c}$.

$$\boldsymbol{z} = E_{\theta_e}(\boldsymbol{x});\ \boldsymbol{y} = C_{\theta_c}(\boldsymbol{z});\ \hat{\boldsymbol{x}} = D_{\theta_d}(\boldsymbol{z}).$$

The task of finding the target sentence then is modelled as an optimisation problem in the latent space -

$$\hat{\boldsymbol{x}}' = D_{\theta_d}(\boldsymbol{z}')\ where\ \boldsymbol{z}' = argmin_{\boldsymbol{z}^*} ||\boldsymbol{z}^* - E_{\theta_e}(\boldsymbol{x})||\ s.t.\ C_{\theta_c}(\boldsymbol{z}^*) = \boldsymbol{y}'.$$

Which basically says modify z as little as possible such that classifier indicates that we have reached the target attribute.
To solve this problem the Fast-Gradient-Iterative-Modification algorithm (FGIM) has been proposed.

Transformer-based Autoencoder

The latent representation z of the input sentence is generated by the following layers -

$$\boldsymbol{z} = E_{\theta_e}(\boldsymbol{x}) = Sum(Sigmoid(GRU(\boldsymbol{U} + \boldsymbol{H}))), where\ \boldsymbol{U} = E_{transformer}(\boldsymbol{x}).$$

Where U is the output of standard transformer encoding, H are positional embeddings to add a positional element to the sentence and then we have the GRU layer with self attention, followed by sigmoid on the hidden representations and sum them to get z.
The autoencoder reconstruction loss during training would be -

$$\mathcal{L}_{ae}(D_{\theta_d}(E_{\theta_e}(\boldsymbol{x})), \boldsymbol{x}) = \mathcal{L}_{ae}(D_{\theta_d}(\boldsymbol{z}), \boldsymbol{x}) = -\sum^{|\boldsymbol{x}|}((1-\varepsilon)\sum_{i=1}^{v} \bar{p}_i \log(p_i) + \frac{\varepsilon}{v}\sum_{i=1}^{v} \log(p_i)),$$

Where v is the vocabulary size ε is the smoothing parameter to relax our confidence in the label.

Attribute Classifier of latent representation

The classifier is just two stacks of linear layer with sigmoid activation function, and the attribute classification loss is:

$$\mathcal{L}_c(C_{\theta_c}(\boldsymbol{z}), \boldsymbol{y}) = -\sum_{i=1}^{|\boldsymbol{q}|} \bar{q}_i \log q_i,$$

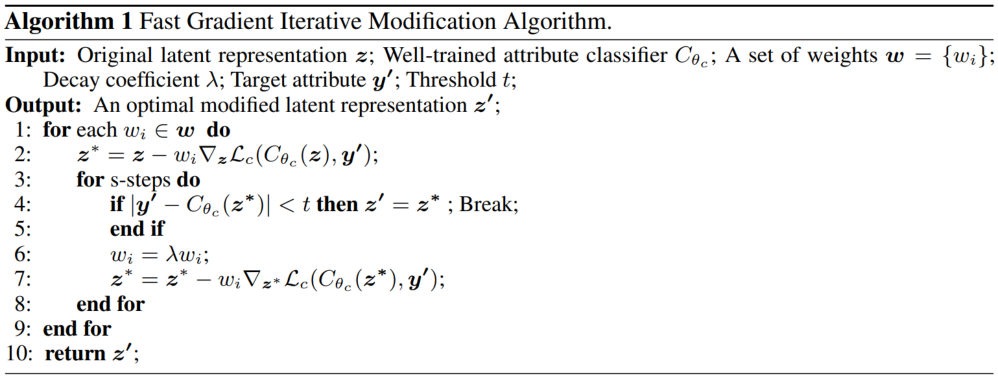
The paper suggested to train these two networks separately rather than together.

Fast Gradient Iterative Modification Algorithm

To get to the target latent representation z' with the target attribute y', the model first takes computes the attribute classification loss of C$_{\theta c}$ (z),y' and then takes the gradient of that loss with respect to z to modify z in the direction of z', and it does this iteratively -

$$\boldsymbol{z}^* = \boldsymbol{z} - w_i \nabla_{\boldsymbol{z}} \mathcal{L}_c(C_{\theta_c}(\boldsymbol{z}), \boldsymbol{y}')$$

where w_i is the modification weight used for controlling the degree of transfer. The model uses an increasing sequence of weights to optimise z and this way it prevents the function to fall into a local minimum.



Algorithm 1 Fast Gradient Iterative Modification Algorithm.

Input: Original latent representation $\boldsymbol{z}$; Well-trained attribute classifier C_{θ_c}; A set of weights $\boldsymbol{w} = \{w_i\}$; Decay coefficient λ; Target attribute $\boldsymbol{y}'$; Threshold t;

Output: An optimal modified latent representation $\boldsymbol{z}'$;

```
for each $w_i \in \boldsymbol{w}$ do
    $\boldsymbol{z}^* = \boldsymbol{z} - w_i \nabla_{\boldsymbol{z}} \mathcal{L}_c(C_{\theta_c}(\boldsymbol{z}), \boldsymbol{y}')$;
    for s-steps do
        if $|\boldsymbol{y}' - C_{\theta_c}(\boldsymbol{z}^*)| < t$ then $\boldsymbol{z}' = \boldsymbol{z}^*$ ; Break;
        end if
        $w_i = \lambda w_i$;
        $\boldsymbol{z}^* = \boldsymbol{z}^* - w_i \nabla_{\boldsymbol{z}^*} \mathcal{L}_c(C_{\theta_c}(\boldsymbol{z}^*), \boldsymbol{y}')$;
    end for
end for
return $\boldsymbol{z}'$;
```

Thus this model has two additional advantages over the existing text-style transfer models-

1) Style-transfer over multiple aspects by just expanding the y vector to include more styles.
2) Transfer degree control - which can be done by adjusting the weights $\{w_i\}$ set.

<u>Experiment</u>

Automatic Evaluation -

1) Acc: is the measure of the attribute transfer accuracy of the generated texts with a fastText classifier trained on the training data
2) BLEU : the BLEU score was used to measure the similarity between the generated text and human written text
3) PPL : the perplexity scores were measured and used as an indicator of the fluency of the generated sentence

The results show that the model performed better in almost all aspects than the baseline models and even outperformed in the human evaluation where a dataset of 100 sentences for each attribute transfer was evaluated.

<u>Multi-aspect Sentiment Transfer</u>

The model achieved high accuracy, fluency and BLEU scores even with the 5 dimension attribute vector of beer review records which means 5 sentiment values. This shows the model can effectively expand its attribute vector to include as many aspects as required.

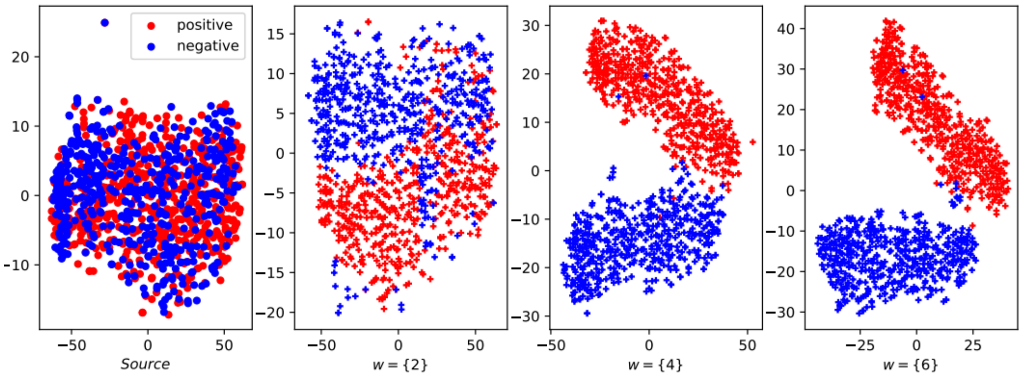
Transfer Degree Control

The value of weight variable was toggled to see the effect on the results of the experiments. It showed the following results -

1) As the value in w increases, the attribute of the generated sentence becomes more and more accurate.
2) However, the BLEU score first increases and then decreases, this maybe because the attribute of some human-written references is not obvious.
3) PPL has not changed so much, which proves the effectiveness of the autoencoder with low reconstruction bias, and the latent representation editing method does not damage the fluency and naturalness of the sentence.

Latent Space Visualisation

Initially the positive and the negative texts were mixed in the latent space. As the value of w increases, the distinction between these two becomes more and more clear



This shows the effectiveness of w in controlling the amount of attribute transfer to the latent representation.

Thus, this concludes the report.