Text Style Transfer using Transformers

Eric Sharkey

Dr. Gordon Erlebacher

Dr. Nathan Crock

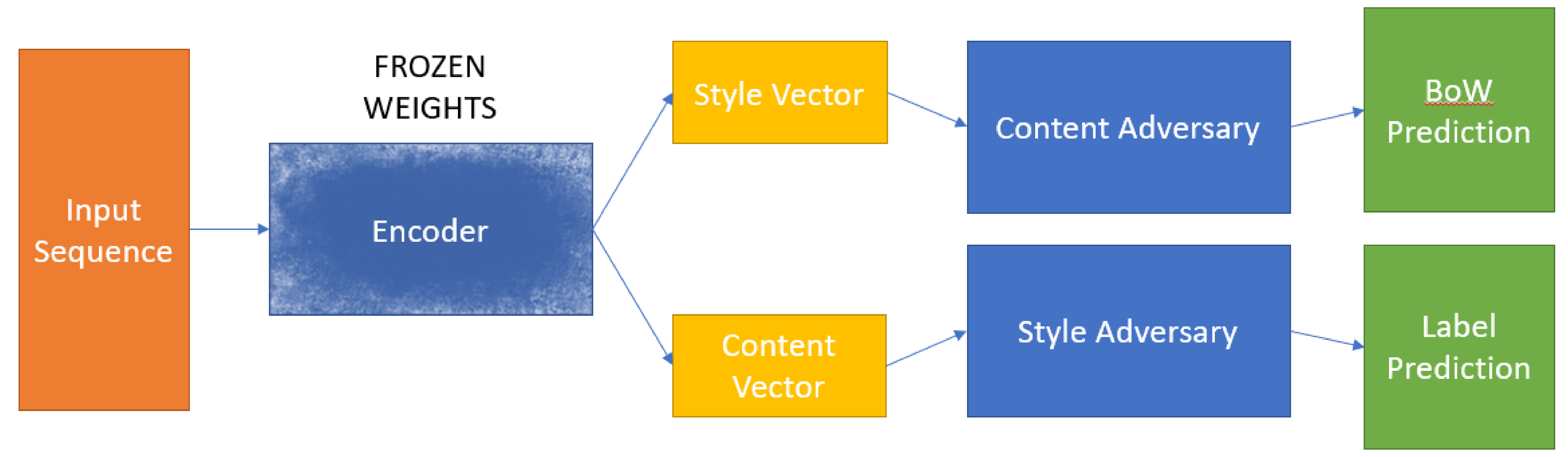
Abstract

Text Style Transfer (TST) is Natural Language Generation task that attempts to transform a sentence into a new sentence with the same content but a different style. The content of a sentence dictates what the sentence is about, while the style is commonly interpreted as the sentiment and emotions used to convey the information. One may think of TST as disentangling the style and content of a sentence to allow for style control. Text Style Transfer has many applications, such as controlling the opinions in product reviews, and creating more reactive chatbots. One difficulty in training TST models is the absence of parallel corpora, containing sentences with identical content but different styles. Thus, many TST models use adversarial training methods with non-parallel corpora, which are more available. Another difficulty is the presence of long-term word dependencies present in sentences meaningful for text style transfer. Existing Recurrent Neural Network based models do not perform well because of this. This poster presents a transformer based variational autoencoder model that effectively disentangles style from content in text. Transformers have been shown to perform well in capturing long term dependencies in text using its attention mechanisms. Its architecture is also largely parallelizable, allowing for faster training. The encoder memory is separated into distinct style and content spaces using convolutional layers, and disentanglement is encouraged in these spaces using adversarial training methods. This model improves on existing Style Transfer models that use implicit disentanglement methods by leveraging Transformers' attention mechanisms to better capture important text dependencies.

Adversarial Training

Step 1: Adversaries are optimized to predict BoW given style vector, style label given content vector

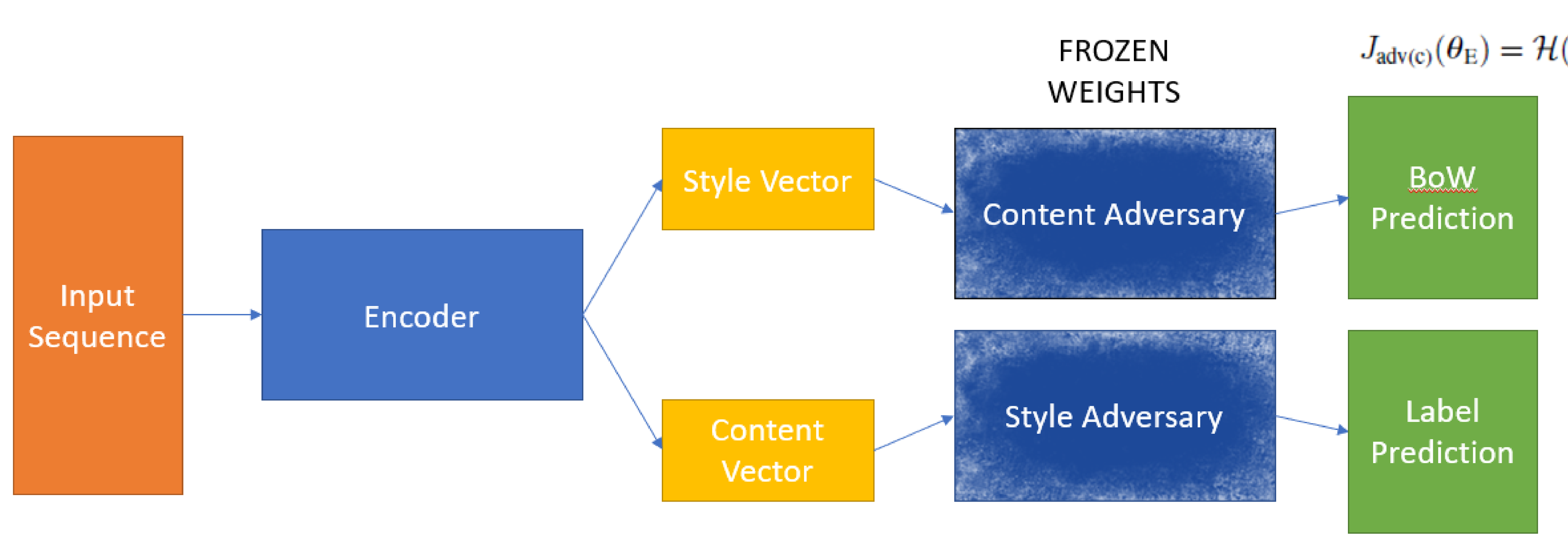
$$y_c = \mathrm{softmax}({W_{\mathrm{dis(c)}}}^\top s + b_{\mathrm{dis(c)}})$$
$$J_{\mathrm{dis(c)}}(\theta_{\mathrm{dis(c)}}) = -\sum\nolimits_{w \in \mathrm{vocab}} t_c(w) \log y_c(w)$$



$$y_s = \mathrm{softmax}(W_{\mathrm{dis(s)}} c + b_{\mathrm{dis(s)}})$$
$$J_{\mathrm{dis(s)}}(\theta_{\mathrm{dis(s)}}) = -\sum\nolimits_{l \in \mathrm{labels}} t_c(l) \log y_s(l)$$

Step 2: Encoder is optimized to maximize the entropy of the output of the adversaries, given the style and content vectors

$$J_{\mathrm{adv(c)}}(\theta_{\mathrm{E}}) = \mathcal{H}(y_c | s; \theta_{\mathrm{dis(c)}})$$



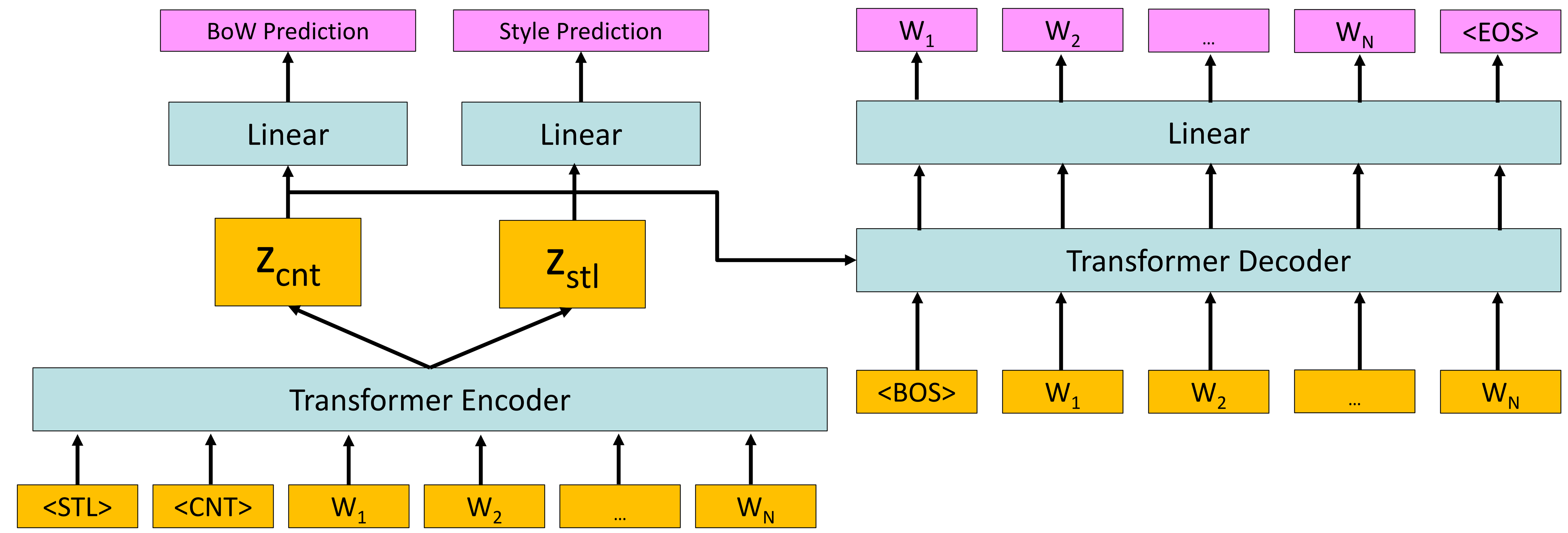
$$J_{\mathrm{adv(s)}}(\theta_{\mathrm{E}}) = \mathcal{H}(y_s | c; \theta_{\mathrm{dis(s)}})$$

During Step 1 of training, the Encoder weights are frozen. The adversarial layers each take one optimization step, the first layer to better predict the BoW prediction given the style vector, and the second to predict the style classification given the content vector. In step 2, the adversaries' weights are frozen. The encoder then takes an optimization step wrt. the overall loss function:

$$\begin{aligned} J_{\mathrm{ovr}} = & J_{\mathrm{AE}}(\boldsymbol{\theta}_{\mathrm{E}}, \boldsymbol{\theta}_{\mathrm{D}}) \\ & + \lambda_{\mathrm{mul(s)}} J_{\mathrm{mul(s)}}(\boldsymbol{\theta}_{\mathrm{E}}, \boldsymbol{\theta}_{\mathrm{mul(s)}}) - \lambda_{\mathrm{adv(s)}} J_{\mathrm{adv(s)}}(\boldsymbol{\theta}_{\mathrm{E}}) \\ & + \lambda_{\mathrm{mul(c)}} J_{\mathrm{mul(c)}}(\boldsymbol{\theta}_{\mathrm{E}}, \boldsymbol{\theta}_{\mathrm{mul(c)}}) - \lambda_{\mathrm{adv(c)}} J_{\mathrm{adv(c)}}(\boldsymbol{\theta}_{\mathrm{E}}) \end{aligned}$$

Model Architecture

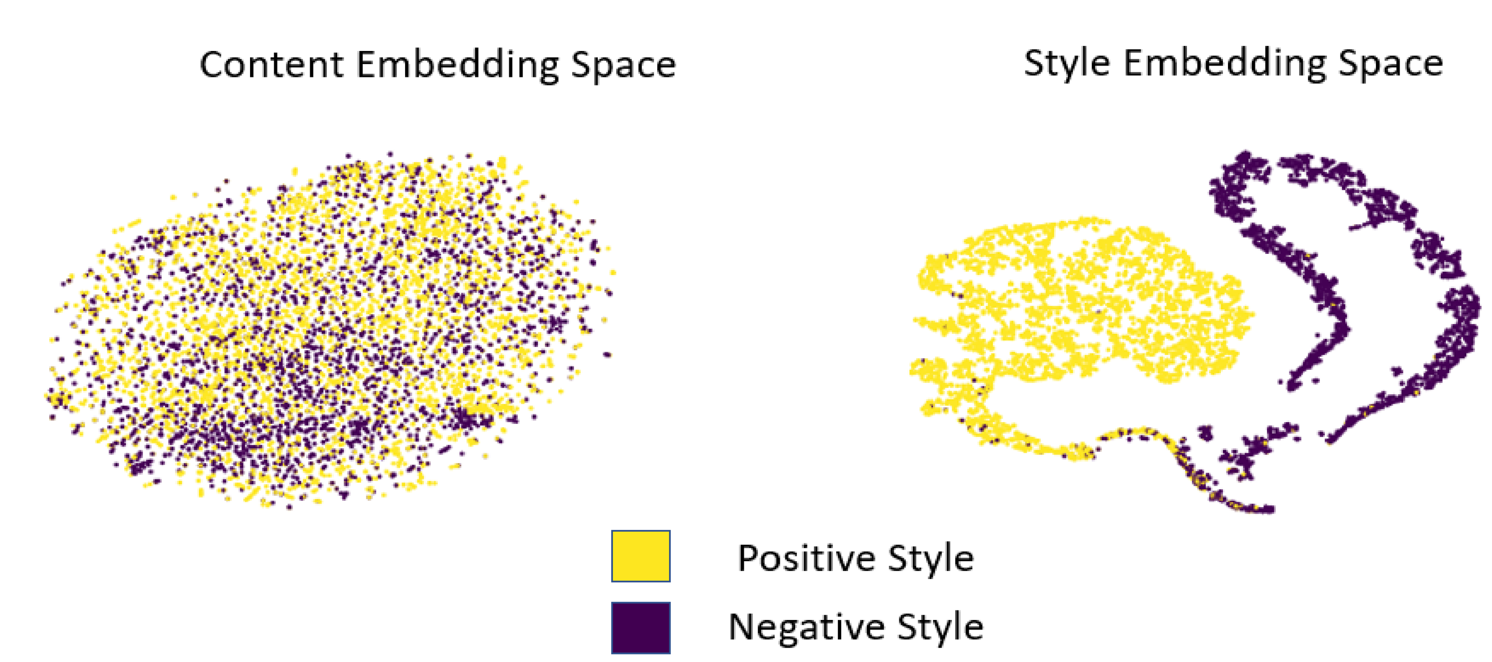
The input to the style transfer model is a sentence prepended with a STL and CNT token, representing the style and content of the sentence, respectively. The transformer encoder then generates context aware embeddings for the STL and CNT tokens. The content token is used to predict the Bag of Words representation of the sentence, while the style token is used to predict the style class. The model also contains two linear adversarial layers shown in Adversarial Training, which remove style information from the content vector, and vice versa.



Loss Terms

$J_{\mathrm{mul(s)}}(\boldsymbol{\theta}_{\mathrm{E}}; \boldsymbol{\theta}_{\mathrm{mul(s)}}) = -\sum_{l \in \mathrm{labels}} t_{\boldsymbol{s}}(l) \log y_{\boldsymbol{s}}(l)$	Style Loss: where y_s is the Style prediction, and t_s is the true style label.
$J_{\mathrm{mul(c)}}(\boldsymbol{\theta}_{\mathrm{E}}; \boldsymbol{\theta}_{\mathrm{mul(c)}}) = -\sum_{w \in \mathrm{vocab}} t_{\boldsymbol{c}}(w) \log y_{\boldsymbol{c}}(w)$	Content Loss: where y_c is the BoW prediction, and t_c is the true BoW
$J_{\mathrm{AE}}(\boldsymbol{\theta}_{\mathrm{E}}, \boldsymbol{\theta}_{\mathrm{D}}) = -\mathbb{E}_{q_E(\boldsymbol{h}\vert \mathrm{x})}[\log p(\mathrm{x}\vert\boldsymbol{h})] + \lambda_{\mathrm{kl}} \mathrm{KL}(q_E(\boldsymbol{h}\vert\mathrm{x}) \Vert p(\boldsymbol{h}))$	Autoencoding Loss: Model decoder is trained to reconstruct sentence x, given z_c and z_s
$J_{\mathrm{adv(s)}}(\boldsymbol{\theta}_{\mathrm{E}}) = \mathcal{H}(\boldsymbol{y}_{\boldsymbol{s}} \vert \boldsymbol{c}; \boldsymbol{\theta}_{\mathrm{dis(s)}})$	Adversarial Style Loss: Model encodes content vector with least style information
$J_{\mathrm{adv(c)}}(\boldsymbol{\theta}_{\mathrm{E}}) = \mathcal{H}(\boldsymbol{y}_{\boldsymbol{c}} \vert \boldsymbol{s}; \boldsymbol{\theta}_{\mathrm{dis(c)}})$	Adversarial Content Loss: Model encodes style vector with least content information

Results



Successful Style Transfer Examples

Positive to Negative

I think this place is great
I think this place is pretty bad
the sauce was lite and tasted great
the sauce was tangy and tasted bland

Negative to Positive

It was ridiculous how loud it was
It was super cool how good it was
The service was extremely slow
The service was extremely attentive

The Content and Style Embedding spaces are visualized using TSNE, and color coded according to their style classification. The model learns to encode style information in the style embedding, while minimizing the style information present in the content embedding.

References

John, Vineet, et al. "Disentangled representation learning for non-parallel text style transfer." *arXiv preprint arXiv:1808.04339* (2018).

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

Hu, Zhiqiang, Roy Ka-Wei Lee, and Charu C. Aggarwal. "Text Style Transfer: A Review and Experiment Evaluation." arXiv preprint arXiv:2010.12742 (2020).