**Text Style Transfer using Transformers**

**Abstract**

Text Style Transfer (TST) is a 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.

**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.

**Adversarial Training**

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:

**Loss Terms**

- *Style Loss:* where $y_s$ is the Style prediction, and $t_s$ is the true style label.
- *Content Loss:* where $y_c$ is the BoW prediction, and $t_c$ is the true BoW.
- *Autoencoding Loss:* Model decoder is trained to reconstruct sentence $x$, given $z_c$ and $z_s$.
- *Adversarial Style Loss:* Model encodes content vector with least style information.
- *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**