Fine Grained Text Style Transfer using Diffusion Final Presentation

CS6420 - Topics in Deep Learning

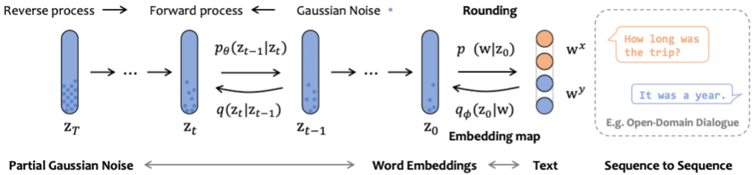
Anudeep Rao Perala (CS21BTECH11043)
Asli Nitej Reddy Busireddy (CS21BTECH11011)

Contents

Problem Statement

- We target the Seq2Seq text generation task for fine grained control over text-to-text modification.
- Given a m-length source sequence $w^x = w_1^x, ..., w_m^x$ and style tokens $s = s_1, , ..., s_k$ we aim to learn a diffusion model that can produce a n-length target sequence $w_y = w_1^y, ..., w_n^y$ conditioning on the source sequence and the style tokens.

Recap DiffuSeq



- We have an embedding function EMB(w).
- During the reverse process the generated vectors are mapped back to the embedding space.
- The variational lower bound(L_{vlb}) is formulated as:

$$\mathcal{L}_{\text{vlb}}(\mathbf{w}) = \mathbb{E}_{q_\phi(z_0|\mathbf{w})}\left[\mathcal{L}_{\text{vlb}}(z_0) + \log q_\phi(z_0|\mathbf{w}) - \log p(\mathbf{w}|z_0)\right] \tag{1}$$

Where $L_{\text{vlb}}(z_0)$ corresponds to the standard variational lower bound in diffusion.

Recap DiffuSeq Contd.

- On modifying eq(1) more we get:

$$
\begin{aligned}
\mathcal{L}_{\text{vlb}}(\mathbf{w}) = \mathbb{E}_{q_\phi(z_{0:T}|\mathbf{w})}\Bigg[& \underbrace{\log \frac{q(z_T|z_0)}{p_\theta(z_T)}}_{L_T} \\
& + \sum_{t=2}^{T} \underbrace{\log \frac{q(z_{t-1}|z_0, z_t)}{p_\theta(z_{t-1}|z_t)}}_{L_{t-1}} \\
& + \underbrace{\log \frac{q_\phi(z_0|\mathbf{w})}{p_\theta(z_0|z_1)}}_{L_0} - \underbrace{\log p(\mathbf{w}|z_0)}_{L_{\text{round}}} \Bigg]
\end{aligned} \tag{2}
$$

Recap DiffuSeq contd.

- After doing all the approximations like how we do in standard diffusion models we get:

$$\begin{aligned}\mathcal{L}_{\text{vlb}}(\mathbf{w}) = \min_{\theta} \Bigg[& \|\mu(z_T)\|^2 + \sum_{t=2}^{T} \|z_0 - f_\theta(z_t, t)\|^2 \\ & + \|\text{EMB}(\mathbf{w}^{x \oplus y}) - f_\theta(z_1, 1)\|^2 \\ & + \mathcal{R}(\|\mathbf{z}_0\|^2) \Bigg]\end{aligned} \tag{3}$$

- During training the model estimates the z_0 via $f_\theta(z_t, t)$.
- The term $\mathcal{R}(\|\mathbf{z}_0\|^2)$ is introduced to learn regularize the embedding learning.

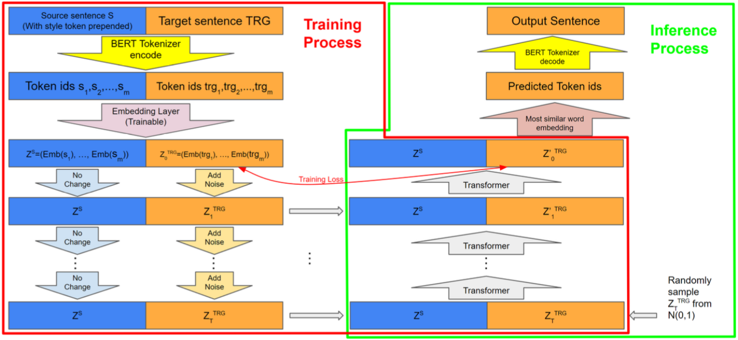
Recap DiffuSeq contd. Inference

- Given the condition EMB(w^x), we randomly sample $y_T \sim N(0, I)$ and concatenate y_T with EMB(w^x) to obtain z_T. We now repeat the reverse process until we arrive at z_0 by calculating z_0^{temp}.
- We sample $\mathbf{z}_{t-1}$ from $q(\mathbf{z}_{t-1} \mid f_\theta(\mathbf{z}_t, t), \mathbf{z}_t)$, which is fed as input to the next diffusion step.
- The equation for obtaining $\mathbf{z}_{t-1}$:

$$\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} f_\theta(\mathbf{z}_t, t) + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon, \text{where } \bar{\alpha}_t \quad = \prod_{i=0}^{t}(1 - \beta_i)$$

- At each sampling step anchoring function is executed towards the obtained z_{t-1} which does:
  - Rounds the obtained z_{t-1} back to word embedding space.
  - Replaces the part of recovered z_{t-1} that belongs to w_x with the original x_0.

Recap State of the Art for FG-TST



Recap State of the Art for FG-TST contd. Training

- Both the diffusion transformer and the token embeddings are initialized randomly and jointly optimized.
- Z^S are source embeddings and Z_0^{TRG} are target embeddings.
- We then apply the partial noise in forward process until $t \sim U(1, T)$, after which we get Z_t^{TRG}. We then concatenate Z^S and Z_t^{TRG} input that to the diffusion transformer.
- We then follow the loss in DiffuSeq for minimization.

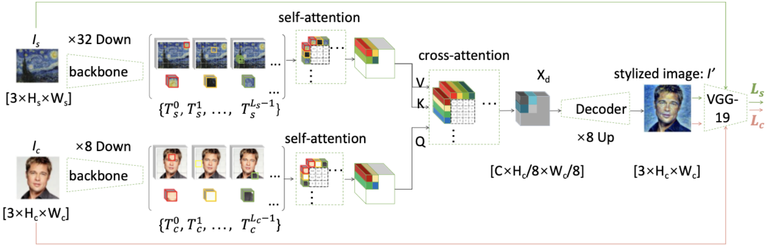
Recap State of the Art for FG-TST contd. Inference

- We randomly initialize ${Z^*_T}^{TRG} \sim N(0, 1)$, and encode the condition (source sentence and style tokens) into Z^S.
- Then we concatenate them and use the transformer to predict a temporary ${Z^*_{0_{temp}}}^{TRG}$, then we add $\mathbf{T} - \mathbf{1}$ steps of noise to obtain ${Z^*_{T-1}}^{TRG}$. Now for each embedding in finally obtained Z_0^{TRG}, we find the closest embedding in our token embedding layer by cosine distance, and decode the embedding to that token.
- Then we combine the tokens to form the output sentence in natural language.

Fine Grained Style Transfer in Vision: STTR

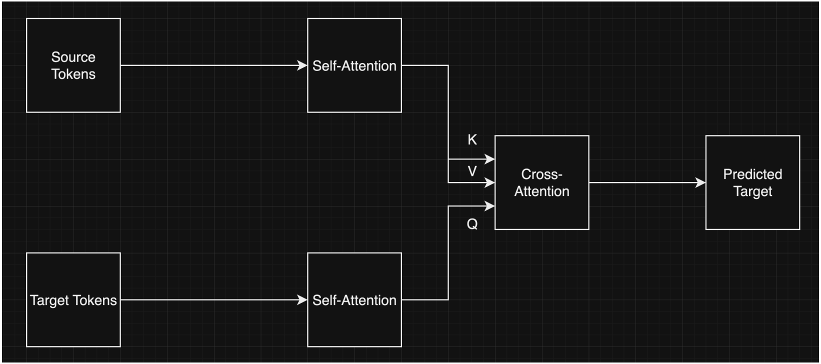
- STTR(Style Transformer) is built by taking the transformer reference used in NLP.
- Develops a Transformer based network to first break content and style images into visual tokens then learns the global context between them.
- This architecture mainly uses the two self-attention modules to summarize the style and content features and a cross attention module to match the style patterns into content patches.
- Transformer is used since its encoder consists of self-attention module while the decoder has cross-attention module to compute the correlations between content and style tokens.

Architecture



- Main components of the architecture:
  - Tokenizer
  - Encoder: Learns self-attention between the style features.
  - Decoder: Learns the relationship between content and style tokens by cross-attention. Learns self-attention between content and style features.

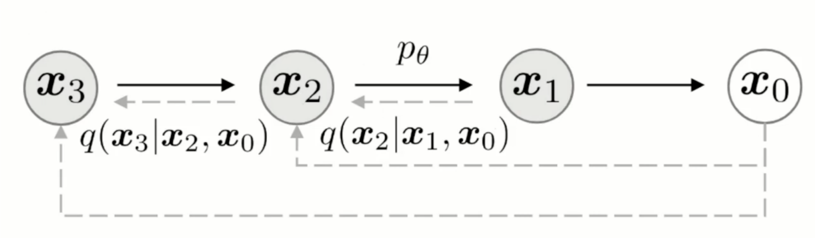
Our Proposal for the Architecture



- We will be using the eq(3) as our loss function.

Making Sampling faster

- We have included DDIM sampling for sampling faster.
- The DDPMs are generalised via a class of non-Markovian diffusion processes that lead to the same training objective which correspond to generative processes that are deterministic, giving rise to implicit models that produce high quality samples much faster.



Making Sampling faster contd.

- Now we consider the family of forward processes defined as :

$$q_\sigma(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathbf{I}\right) \tag{4}$$

- All the distributions are indexed by the real vector σ_t.
- The property of $q(x_t \mid x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$ is still valid.

Making Sampling faster contd.

- Now we model the forward process as:

$$q_\sigma(x_t \mid x_{t-1}, x_0) = \frac{q_\sigma(x_{t-1} \mid x_t, x_0) q_\sigma(x_t \mid x_0)}{q_\sigma(x_{t-1} \mid x_0)}, \qquad (5)$$

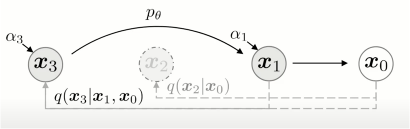
- Clearly the above is no longer Markovian.
- The σ_t maintains the stochaticity of the process, when $\sigma_t = 0$ the forward process is deterministic. This is called as DDIM.
- After doing the required math for calculating the training objective we find that irrespective of the σ_t the training objective always simplifies to the training objective used in DDPMs.

Making Sampling faster contd.

- So this means that a model trained in the original DDPM process can be used by any of the process in the family for inference.
- For sampling:

$$
\begin{aligned}
x_{t-1} = & \sqrt{\alpha_{t-1}} \left(\frac{x_t - \sqrt{1-\alpha_t}\, \epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right) \\
& + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t) \\
& + \sigma_t \, \epsilon_t,
\end{aligned} \tag{6}
$$

Making Sampling faster contd.



- Now defining only a subset of steps from $\{x_{\tau_1}, \ldots, x_{\tau_S}\}$.
- The backward process is defined as:

$$p_\theta(x_{0:T}) := p_\theta(x_T) \prod_{i=1}^{S} p_\theta^{(\tau_i)}(x_{\tau_i - 1} \mid x_{\tau_i}) \times \prod_{t \in \bar{\tau}} p_\theta^{(t)}(x_0 \mid x_t) \quad (7)$$

- After doing the required math the training objective for the above backward process is found to be equivalent with the original DDPM.

BLEU Metric

- We have precision and recall defined as:

$$\text{Precision} = \frac{\#\text{overlapping words}}{\#\text{predicted words}} \tag{8}$$

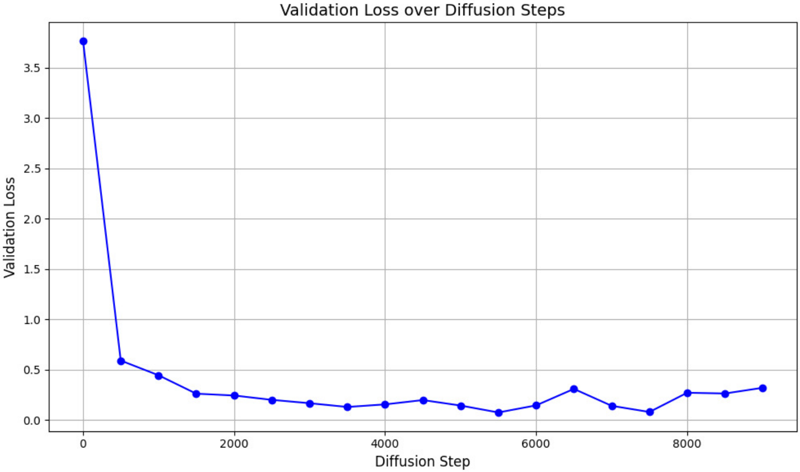
- BLEU uses precision for measuring the quality.

$$\text{BLEU} = \text{BP} \cdot \prod_{n=1}^{N} p_n^{w_n} \tag{9}$$

Results so far

- We have only trained on the PPR(Pre-Position Removal) task. This is a Medium level task.
- Some hyperparams:
  - lr : 10^{-4}
  - batch size : 64
  - noise schedular : linear
  - sequence length : 128

Results so far



References

- Understanding Diffusion Models: A Unified Perspective
- DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models
- Fine-grained Text Style Transfer with Diffusion-Based Language Models
- Fine-Grained Image Style Transfer with Visual Transformers
- Denoising Diffusion Implicit Models
- StylePTB: A Compositional Benchmark for Fine-grained Controllable Text Style Transfer