Style Transfer from Non-Parallel Text by Cross-Alignment

Shen et al 2017 Arxiv: 1705.09655

Presented by Leon Yin
ML^2 Reading Group 2017-10-31

Maintain content *and* change style?

View a sentence (x) of some distribution function of of style (y) and content (z).

Style is sentiment between positive yelp reviews (3+ reviews) and negative.

The two datasets are assumed to be talking about the same restaurants.

Taco is z

x_1

These tacos are cold!

y_1= :(

x_2

These tacos are the bomb!

y_2= :)

Variational Auto-Encoder (VAE)

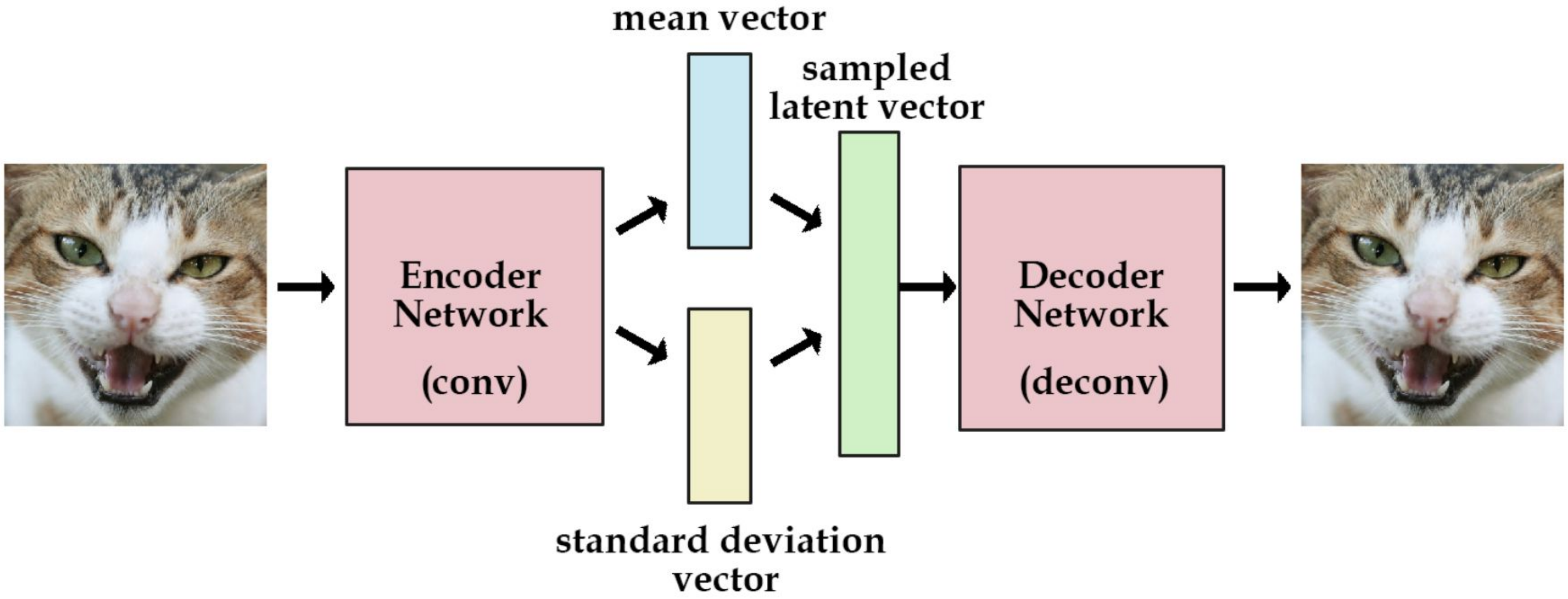


Image coutesty of: http://kvfrans.com/variational-autoencoders-explained/

Pros and Cons of VAE?

“The fact that VAEs basically optimize likelihood while GANs optimize something else can be viewed both as an advantage or a disadvantage for either one.”

- Yoshua Bengio via Quora

Two step solution

Encoder infers content (z) given sentence (x) and style (y).

$$E : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Z}$$

Generator returns sentence (x’) given style (y) from latent rep for content (z).

$$G : \mathcal{Y} \times \mathcal{Z} \rightarrow \mathcal{X}$$

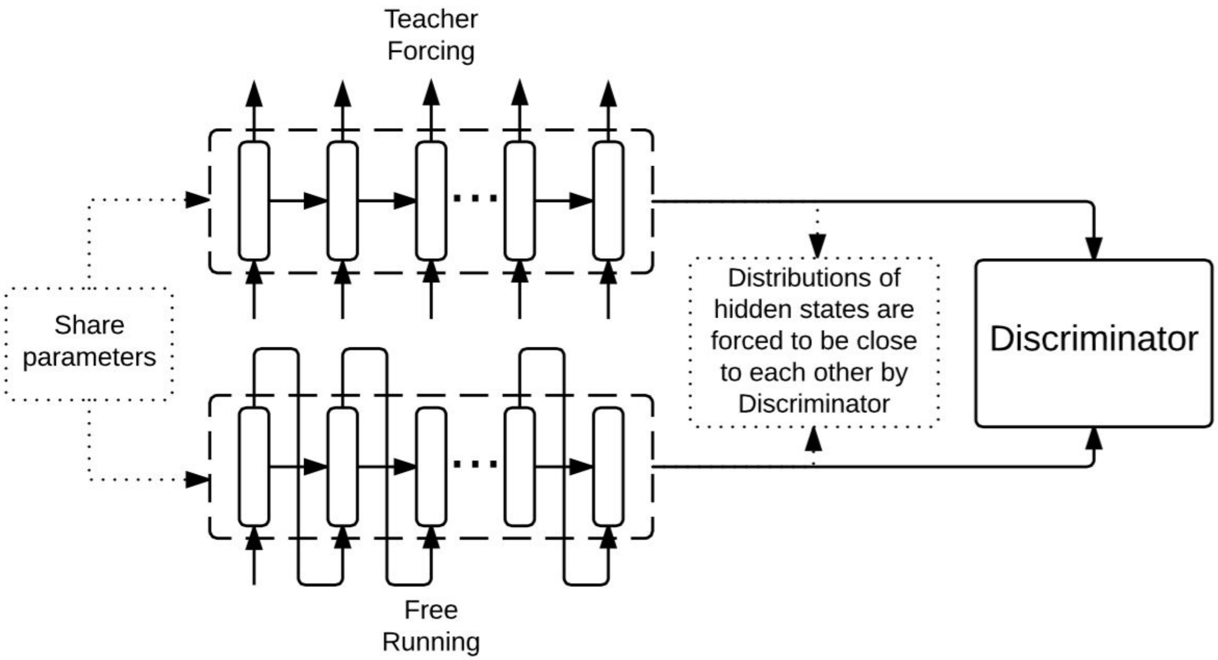
This system can be trained using a GAN!

Pros and Cons of VAE?

“The fact that VAEs basically optimize likelihood while GANs optimize *something else* can be viewed both as an advantage or a disadvantage for either one.”

- Yoshua Bengio via Quora

Professor Forcing (Lamb et al 2016)



Teacher Forcing

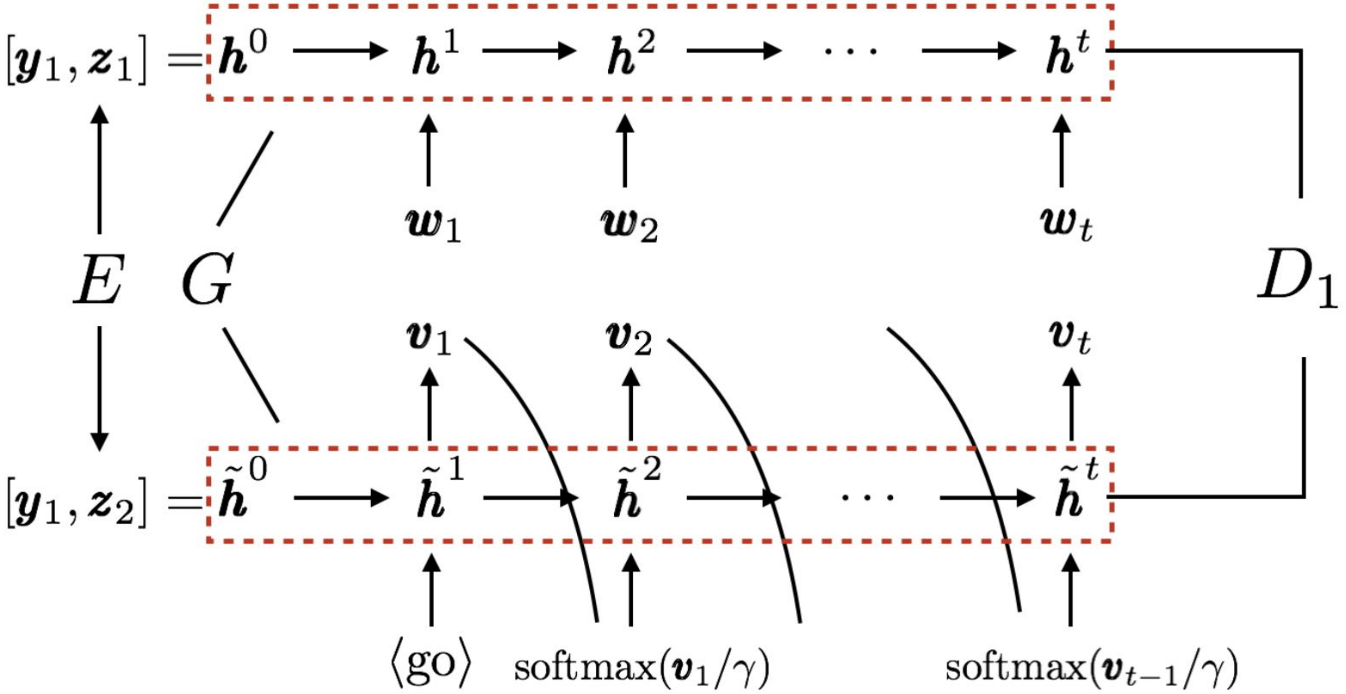
Share parameters

Distributions of hidden states are forced to be close to each other by Discriminator

Discriminator

Free Running

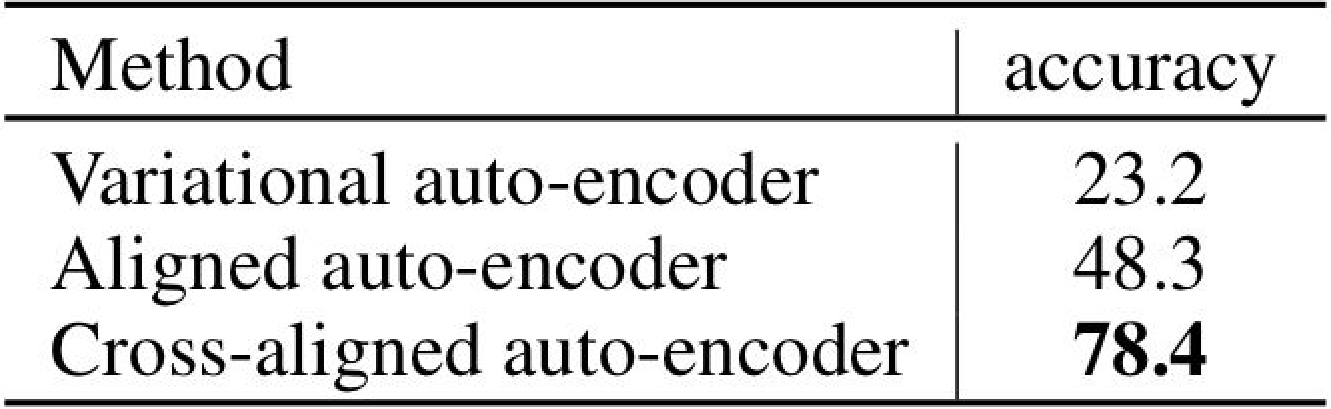
Cross-Aligned Auto-Encoder (Shen et al 2017)



$[\boldsymbol{y}_1, \boldsymbol{z}_1] = \boldsymbol{h}^0$
$\boldsymbol{h}^1$
$\boldsymbol{h}^2$
$\cdots$
$\boldsymbol{h}^t$
$\boldsymbol{w}_1$
$\boldsymbol{w}_2$
$\boldsymbol{w}_t$
E
G
D_1
$\boldsymbol{v}_1$
$\boldsymbol{v}_2$
$\boldsymbol{v}_t$
$[\boldsymbol{y}_1, \boldsymbol{z}_2] = \tilde{\boldsymbol{h}}^0$
$\tilde{\boldsymbol{h}}^1$
$\tilde{\boldsymbol{h}}^2$
$\cdots$
$\tilde{\boldsymbol{h}}^t$
$\langle \mathrm{go} \rangle$
$\mathrm{softmax}(\boldsymbol{v}_1/\gamma)$
$\mathrm{softmax}(\boldsymbol{v}_{t-1}/\gamma)$

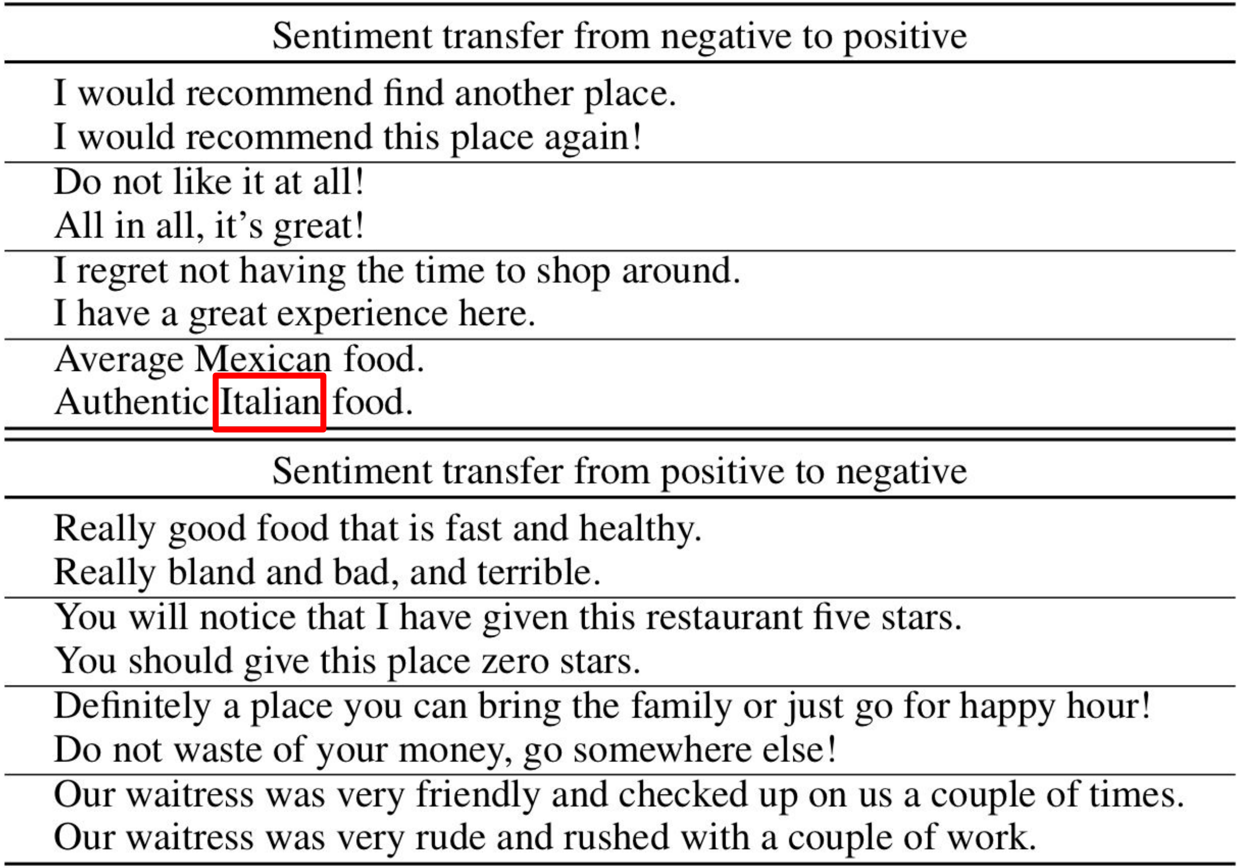
Evaluation

Used pre-trained sentiment classifier with a prediction accuracy of 85.4%.



Method	accuracy
Variational auto-encoder	23.2
Aligned auto-encoder	48.3
Cross-aligned auto-encoder	**78.4**

Table 1: Sentiment accuracy on transferred data



Sentiment transfer from negative to positive
I would recommend find another place. I would recommend this place again!
Do not like it at all! All in all, it’s great!
I regret not having the time to shop around. I have a great experience here.
Average Mexican food. Authentic Italian food.

Sentiment transfer from positive to negative
Really good food that is fast and healthy. Really bland and bad, and terrible.
You will notice that I have given this restaurant five stars. You should give this place zero stars.
Definitely a place you can bring the family or just go for happy hour! Do not waste of your money, go somewhere else!
Our waitress was very friendly and checked up on us a couple of times. Our waitress was very rude and rushed with a couple of work.

Taco is z?

x_1

These tacos are cold!

y_1= :(

x_2

These tacos are the bomb!

y_2= :)

What is z?

x_1

These tacos are cold!

y_1= :(

x_2

This spaghetti is sooo Italian!

y_2= :)

Open Questions

Is sentiment a good example of style?

Other training systems like Professor Forcing?

Emerging methods of evaluating and comparing GANs?

How much time do you spend picking or exploring the data you feed into a model?

Thanks!

“Translation is a matter of compromises.”

- Ken Liu Reddit AMA

Extra Slides For Questions...

Data set for Pos X_1 (n=250k) and Neg X_2 (n=350k)

2 datasets w/ same content distro (Yelp reviews) and styles y_1 (pos) and y_2 (neg).

- 3+ star reviews == positive.
- Filter out reviews if
  - +10 sentences
  - +15 words / sentence.

Used to estimate the style transfer functions between X_1 and X_2

$p(x_1|x_2;y_1,y_2)$ and $p(x_2|x_1;y_1,y_2)$.

Reconstruction Loss

$$\begin{aligned}\mathcal{L}_{\text{rec}}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G) = \; & \mathbb{E}_{\boldsymbol{x}_1 \sim \boldsymbol{X}_1}[-\log p_G(\boldsymbol{x}_1 | \boldsymbol{y}_1, E(\boldsymbol{x}_1, \boldsymbol{y}_1))] + \\ & \mathbb{E}_{\boldsymbol{x}_2 \sim \boldsymbol{X}_2}[-\log p_G(\boldsymbol{x}_2 | \boldsymbol{y}_2, E(\boldsymbol{x}_2, \boldsymbol{y}_2))]\end{aligned}$$

Algorithm 1 Cross-aligned auto-encoder training. The hyper-parameters are set as $\lambda = 1, \gamma = 0.001$ and learning rate is 0.0001 for all experiments in this paper.

```
Input: Two corpora of different styles $\boldsymbol{X}_1, \boldsymbol{X}_2$. Lagrange multiplier $\lambda$, temperature $\gamma$.
  Initialize $\boldsymbol{\theta}_E, \boldsymbol{\theta}_G, \boldsymbol{\theta}_{D_1}, \boldsymbol{\theta}_{D_2}$
  repeat
    for $p = 1, 2; q = 2, 1$ do
      Sample a mini-batch of $k$ examples $\{\boldsymbol{x}_p^{(i)}\}_{i=1}^{k}$ from $\boldsymbol{X}_p$
      Get the latent content representations $\boldsymbol{z}_p^{(i)} = E(\boldsymbol{x}_p^{(i)}, \boldsymbol{y}_p)$
      Unroll $G$ from initial state $(\boldsymbol{y}_p, \boldsymbol{z}_p^{(i)})$ by feeding $\boldsymbol{x}_p^{(i)}$, and get the hidden states sequence $\boldsymbol{h}_p^{(i)}$
      Unroll $G$ from initial state $(\boldsymbol{y}_q, \boldsymbol{z}_p^{(i)})$ by feeding previous soft output distribution with temperature $\gamma$, and get the transferred hidden states sequence $\tilde{\boldsymbol{h}}_p^{(i)}$
    end for
    Compute the reconstruction $\mathcal{L}_{\text{rec}}$ by Eq. (3)
    Compute $D_1$'s loss $\mathcal{L}_{\text{adv}_1} = -\frac{1}{k}\sum_{i=1}^{k} \log D_1(\boldsymbol{h}_1^{(i)}) - \frac{1}{k}\sum_{i=1}^{k} \log(1 - D_1(\tilde{\boldsymbol{h}}_2^{(i)}))$
    Compute $D_2$'s loss $\mathcal{L}_{\text{adv}_2} = -\frac{1}{k}\sum_{i=1}^{k} \log D_2(\boldsymbol{h}_2^{(i)}) - \frac{1}{k}\sum_{i=1}^{k} \log(1 - D_2(\tilde{\boldsymbol{h}}_1^{(i)}))$
    Update $\{\boldsymbol{\theta}_E, \boldsymbol{\theta}_G\}$ by gradient descent on loss $\mathcal{L}_{\text{rec}} - \lambda(\mathcal{L}_{\text{adv}_1} + \mathcal{L}_{\text{adv}_2})$
    Update $\boldsymbol{\theta}_{D_1}$ and $\boldsymbol{\theta}_{D_2}$ by gradient descent on loss $\mathcal{L}_{\text{adv}_1}$ and $\mathcal{L}_{\text{adv}_2}$ respectively
  until convergence
```