Toward Controlled Generation of Text

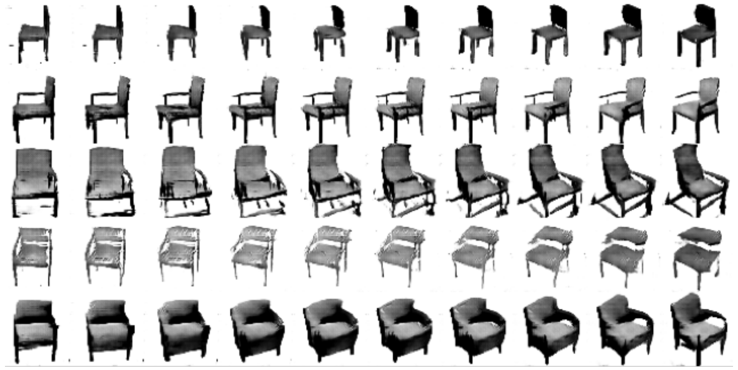
Zhiting Hu[1,2], Zichao Yang[1], Xiaodan Liang[1], Ruslan Salakhutdinov[1], Eric P. Xing[1,2]

Carnegie Mellon University[1]

Petuum Inc[2]

Recent advances in deep generative models

- Deep generative models
  - Variational autoencoders (VAEs) [Kingma & Welling, 2013]
  - Generative adversarial networks (GANs) [Goodfellow et al., 2014]
  - Auto-regressive models
- Impressive success in vision domain
  - Image generation/editing
  - Interpretable representation learning



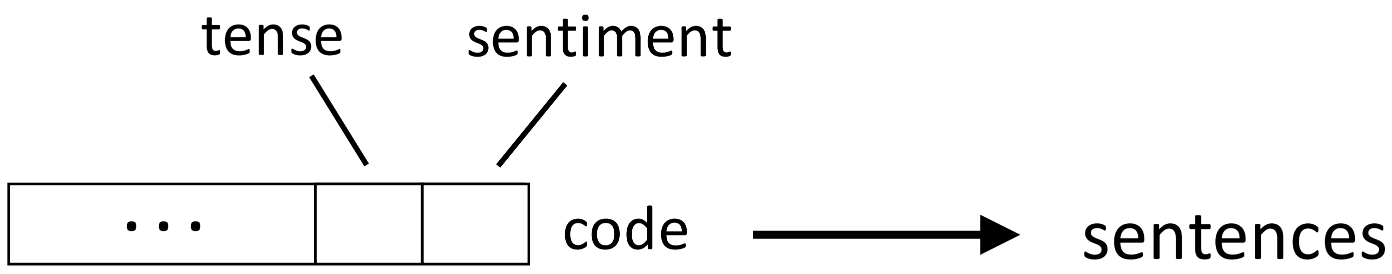
[Chen et al., 2016]

Limited success in text generation

- Task-specific supervised settings
  - Machine translation / image captioning / …
  - Seq2seq models
- Generic text generation
  - Produces realistic sentences given arbitrary hidden code
  - VAEs, GANs
- Mostly limited to randomized and uncontrollable generation

This paper: Controlled generation of text

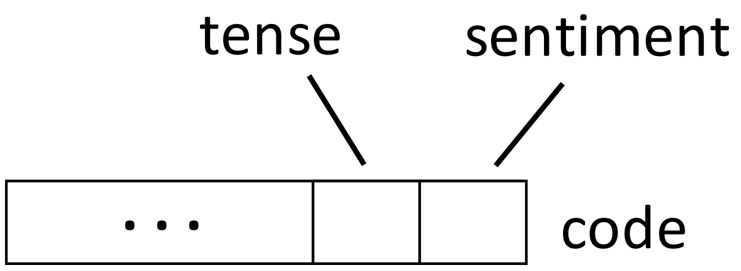
- Generation of realistic sentences
- Control of *user-specified* attributes
  - E.g., sentiment, tense, …
  - Generates sentences with sentiment (negative/positive) by simply setting the sentiment code (0/1)



Challenge 1: User-specified semantics

- Impose user-specified semantics on each part of latent code
  - Methods like conditional language models require large amount of sentences exhaustively annotated with all attributes of interest

- This work:
  - Semi-supervised learning
    - *Synthesize* (sentence, label) pairs for training
  - Independent dataset for each attribute



Exhaustively annotated data:

``I hope he'll make more movies in the future''
sentiment=positive, tense=future

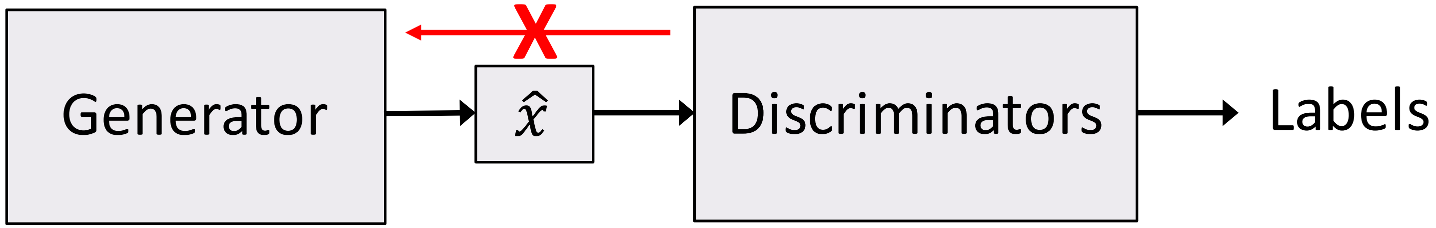
Independent data:

``The film is just great''
sentiment=positive
``I will watch the movie''
tense=future

Challenge 2: Non-differentiable text samples

- Text samples are discrete and non-differentiable
  - Disables holistic discriminators that evaluate generated whole sentences
  - Reconstruction-based methods (LM, VAEs) lose holistic view of whole sentences

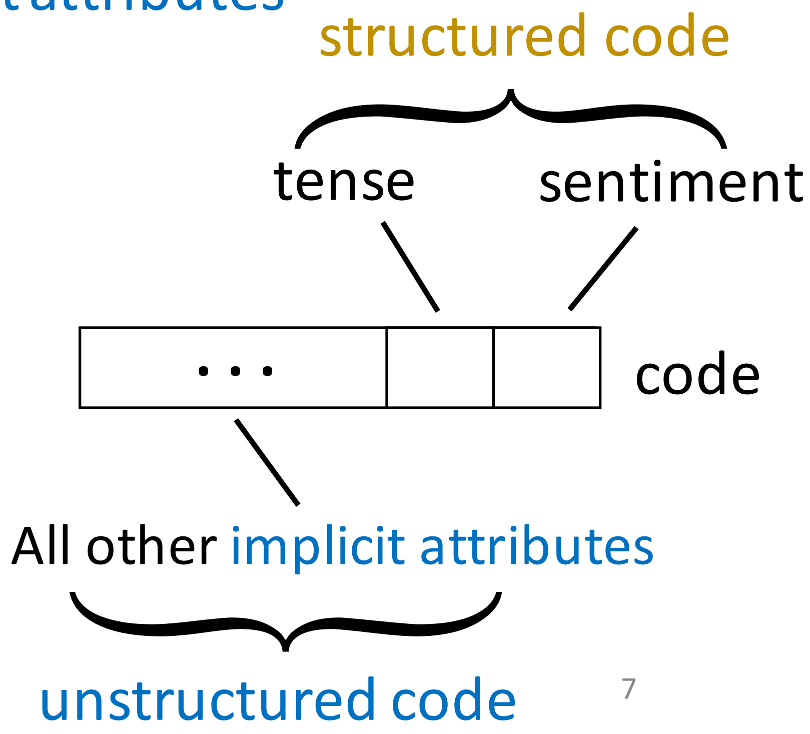
- This work:
  - Enables attribute discriminator through deterministic softmax approximation



Challenge 3: Learning fully disentangled representations

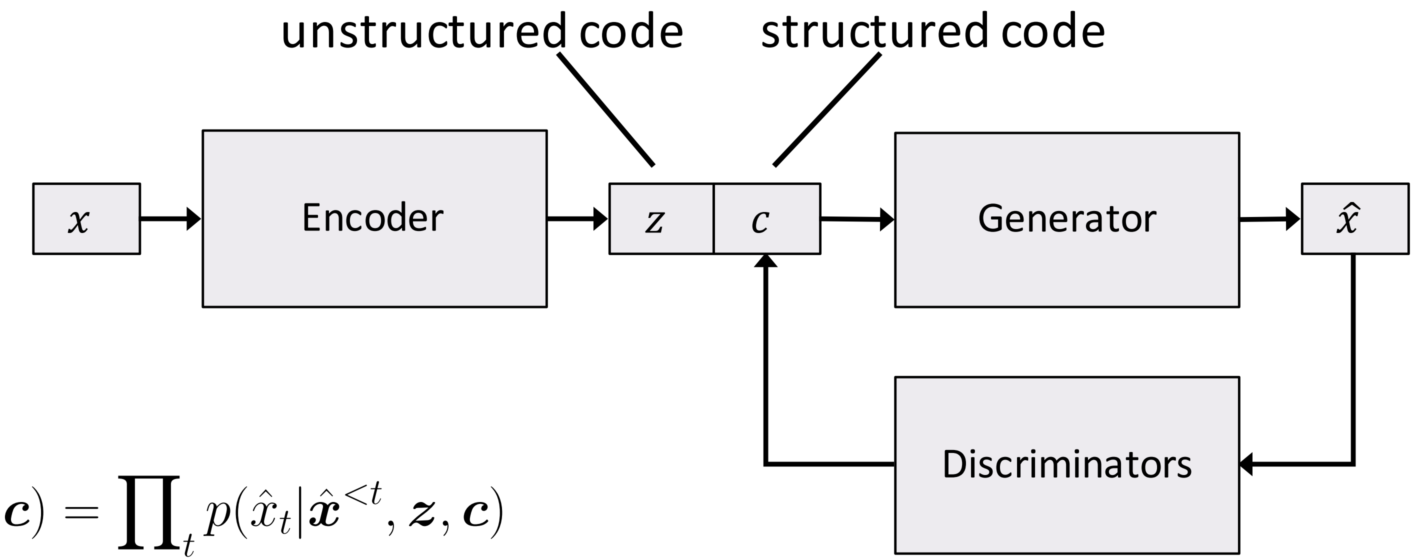
- Want each part of structured code to control one and only one attribute
  - Previous works lack necessary independence constraints
  - Especially, varying structured code can change implicit attributes
    - Toggling sentiment code change content :

      Sentiment=1: “The movie is so much fun .”
      Sentiment=0: “The acting is bad .”



- This work:
  - Explicit independence constraint

Model



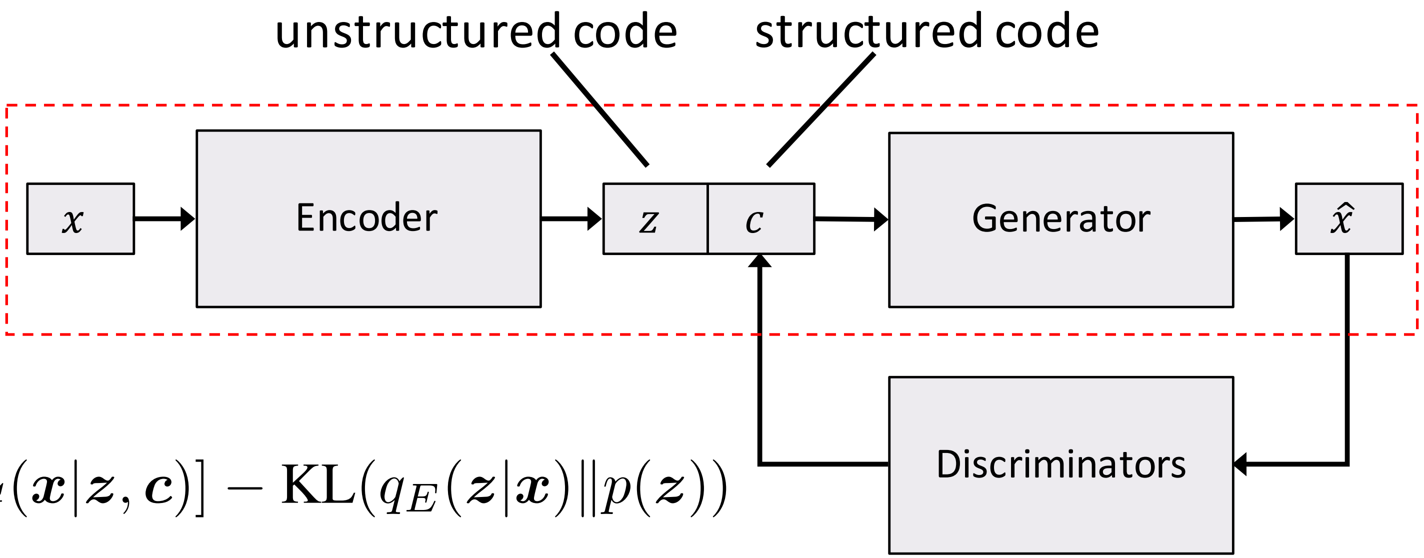
- Generator: $\hat{\boldsymbol{x}} \sim G(\boldsymbol{z}, \boldsymbol{c}) = p_G(\hat{\boldsymbol{x}}|\boldsymbol{z}, \boldsymbol{c}) = \prod_t p(\hat{x}_t|\hat{\boldsymbol{x}}^{<t}, \boldsymbol{z}, \boldsymbol{c})$

  $\hat{x}_t \sim \mathrm{softmax}(\boldsymbol{o}_t/\tau)$

- Encoder: $\boldsymbol{z} \sim E(\boldsymbol{x}) = q_E(\boldsymbol{z}|\boldsymbol{x})$

- Discriminators: $D(\boldsymbol{x}) = q_D(\boldsymbol{c}|\boldsymbol{x})$
  - One for each attribute to control
  - E.g., for sentiment, discriminator is a sentiment classifier

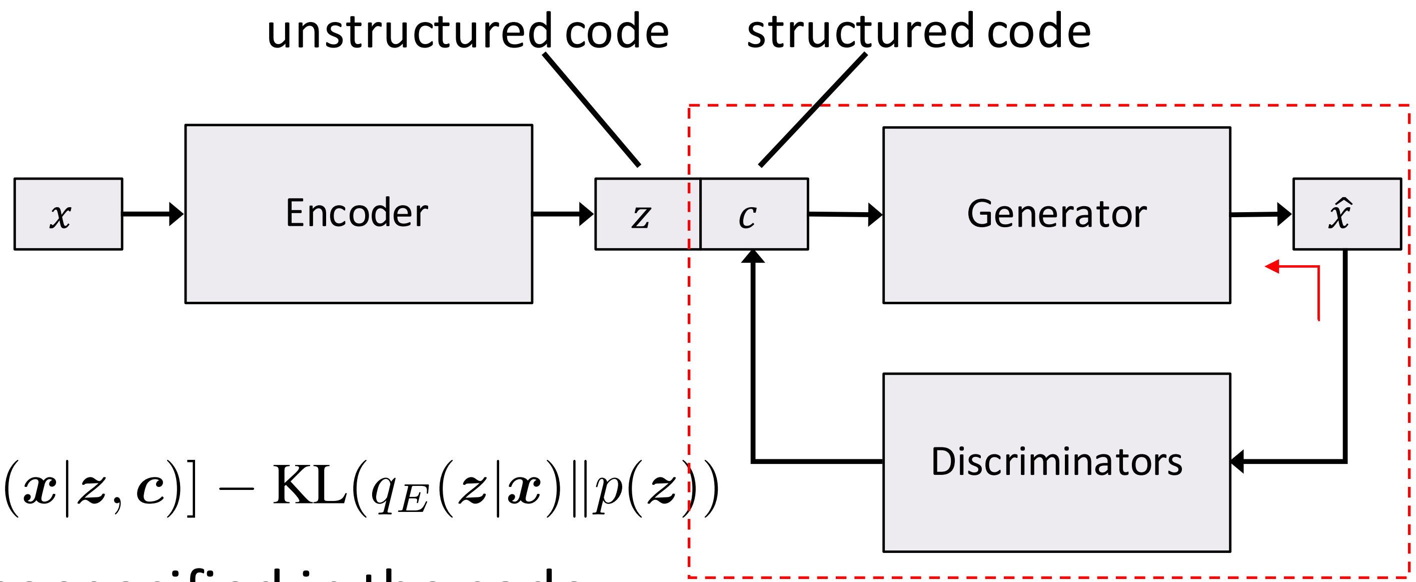
Generator Learning



- Generate *realistic* sentences

$$\mathcal{L}_{\mathrm{VAE}}(\boldsymbol{\theta}_G, \boldsymbol{\theta}_E; \boldsymbol{x}) = \mathbb{E}_{q_E(\boldsymbol{z}|\boldsymbol{x})q_D(\boldsymbol{c}|\boldsymbol{x})}\left[\log p_G(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right] - \mathrm{KL}(q_E(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z}))$$

Generator Learning



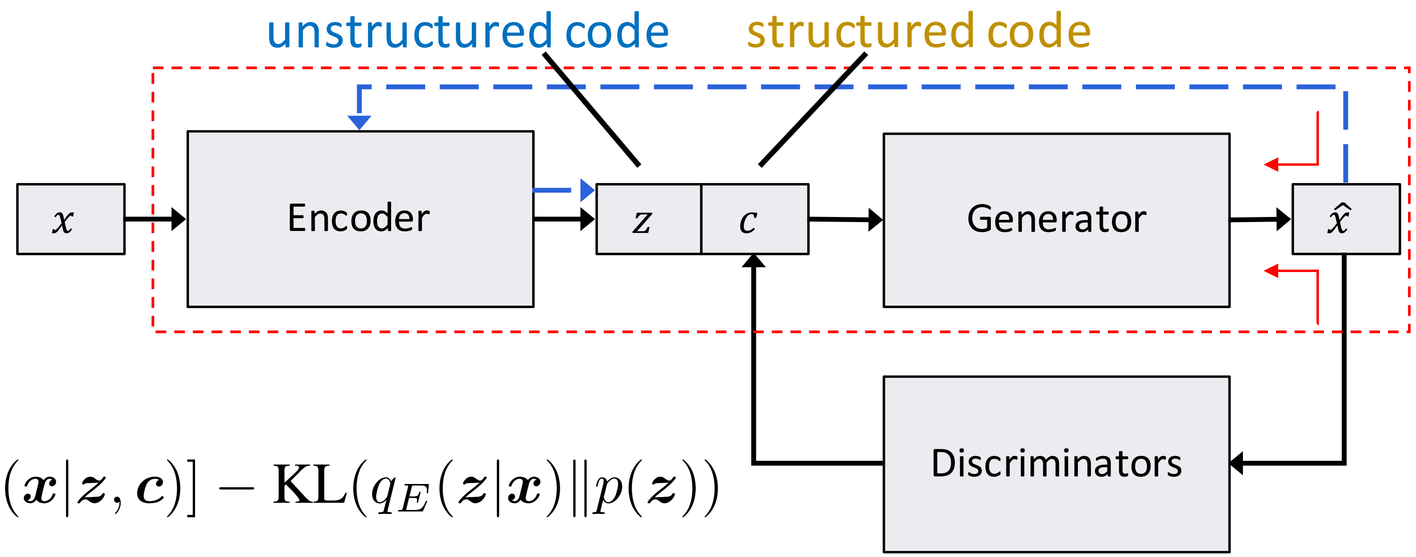
- Generate *realistic* sentences

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{\theta}_G, \boldsymbol{\theta}_E; \boldsymbol{x}) = \mathbb{E}_{q_E(\boldsymbol{z}|\boldsymbol{x})q_D(\boldsymbol{c}|\boldsymbol{x})}\left[\log p_G(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right] - \text{KL}(q_E(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$$

- Generate sentences with attributes specified in the code
  - Discriminators evaluate generated sentences and backpropagate gradients
  - Deterministic softmax approximation of discrete text sentences
    - Replace discrete token $\hat{x}_t$ (*one-hot vector*) with *probability vector* $\text{softmax}(\boldsymbol{o}_t/\tau)$

$$\mathcal{L}_{\text{Attr},c}(\boldsymbol{\theta}_G) = \mathbb{E}_{p(\boldsymbol{z})p(\boldsymbol{c})}\left[\log q_D(\boldsymbol{c}|\widetilde{G}_\tau(\boldsymbol{z}, \boldsymbol{c}))\right]$$

Generator Learning



- Generate *realistic* sentences

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{\theta}_G, \boldsymbol{\theta}_E; \boldsymbol{x}) = \mathbb{E}_{q_E(\boldsymbol{z}|\boldsymbol{x})q_D(\boldsymbol{c}|\boldsymbol{x})}\left[\log p_G(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right] - \mathrm{KL}(q_E(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$$

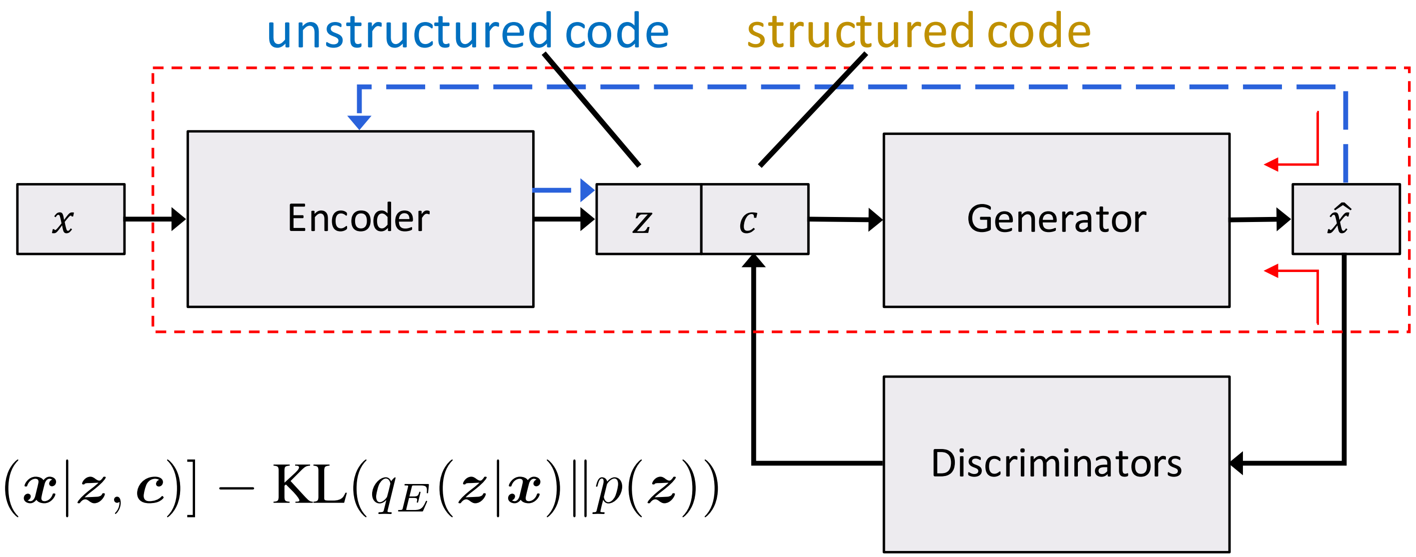
- Generate sentences with attributes specified in the code
  - Discriminators evaluate generated sentences and backpropagate gradients
  - Deterministic softmax approximation of discrete text sentences
    - Replace discrete token $\hat{x}_t$ (*one-hot vector*) with *probability vector* $\mathrm{softmax}(\boldsymbol{o}_t/\tau)$

$$\mathcal{L}_{\text{Attr},c}(\boldsymbol{\theta}_G) = \mathbb{E}_{p(\boldsymbol{z})p(\boldsymbol{c})}\left[\log q_D(\boldsymbol{c}|\widetilde{G}_\tau(\boldsymbol{z}, \boldsymbol{c}))\right]$$

- Independence constraint
  - Implicit attributes should be fully modeled in $\boldsymbol{z}$ and independent with $\boldsymbol{c}$

$$\mathcal{L}_{\text{Attr},z}(\boldsymbol{\theta}_G) = \mathbb{E}_{p(\boldsymbol{z})p(\boldsymbol{c})}\left[\log q_E(\boldsymbol{z}|\widetilde{G}_\tau(\boldsymbol{z}, \boldsymbol{c}))\right]$$

Generator Learning



- Generate *realistic* sentences

$$\mathcal{L}_{\text{VAE}}(\boldsymbol{\theta}_G, \boldsymbol{\theta}_E; \boldsymbol{x}) = \mathbb{E}_{q_E(\boldsymbol{z}|\boldsymbol{x})q_D(\boldsymbol{c}|\boldsymbol{x})}\left[\log p_G(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c})\right] - \text{KL}(q_E(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}))$$

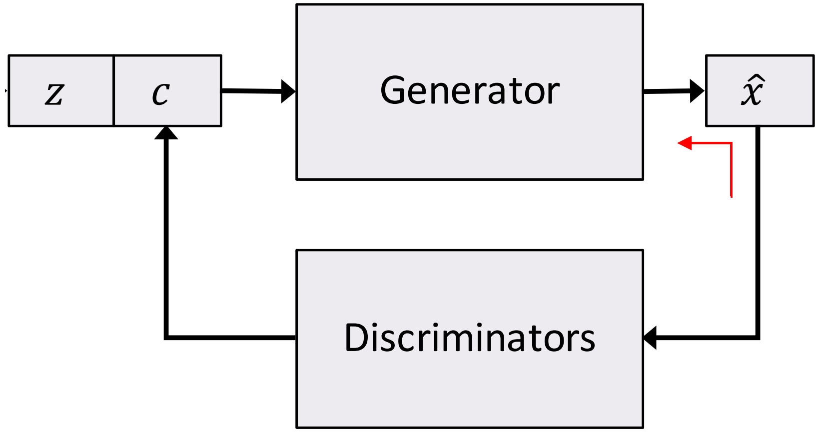
$$\min_{\boldsymbol{\theta}_G} \mathcal{L}_G = \mathcal{L}_{\text{VAE}} + \lambda_c \mathcal{L}_{\text{Attr},c} + \lambda_z \mathcal{L}_{\text{Attr},z}$$

$$\mathcal{L}_{\text{Attr},c}(\boldsymbol{\theta}_G) = \mathbb{E}_{p(\boldsymbol{z})p(\boldsymbol{c})}\left[\log q_D(\boldsymbol{c}|\widetilde{G}_\tau(\boldsymbol{z}, \boldsymbol{c}))\right]$$

- Independence constraint
  - Implicit attributes should be fully modeled in ***z*** and independent with ***c***

$$\mathcal{L}_{\text{Attr},z}(\boldsymbol{\theta}_G) = \mathbb{E}_{p(\boldsymbol{z})p(\boldsymbol{c})}\left[\log q_E(\boldsymbol{z}|\widetilde{G}_\tau(\boldsymbol{z}, \boldsymbol{c}))\right]$$

Discriminator Learning



- Supervised objective on labeled examples $\{(\boldsymbol{x}_L, c_L)\}$

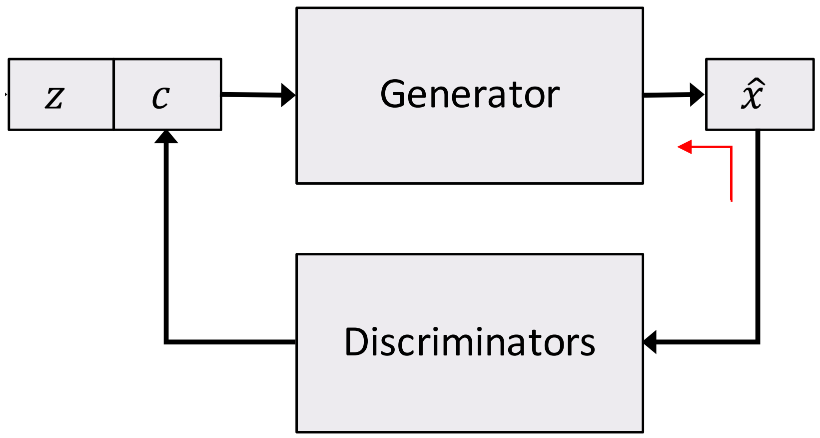
$$\mathcal{L}_s(\boldsymbol{\theta}_D) = \mathbb{E}_{\mathcal{X}_L}\left[\log q_D(\boldsymbol{c}_L|\boldsymbol{x}_L)\right]$$

  - Each attribute discriminator can be trained on *separate* labeled datasets

- Unsupervised objective on synthesized samples $\{(\hat{\boldsymbol{x}}, c)\}$ by the generator
  - Add a minimum entropy regularization to alleviate noise

$$\mathcal{L}_u(\boldsymbol{\theta}_D) = \mathbb{E}_{p_G(\hat{\boldsymbol{x}}|\boldsymbol{z},\boldsymbol{c})p(\boldsymbol{z})p(\boldsymbol{c})}\left[\log q_D(\boldsymbol{c}|\hat{\boldsymbol{x}}) + \beta\mathcal{H}(q_D(\boldsymbol{c}'|\hat{\boldsymbol{x}}))\right]$$

Discriminator Learning



- Supervised objective on labeled examples $\{(\boldsymbol{x}_L, c_L)\}$

$$\mathcal{L}_s(\boldsymbol{\theta}_D) = \mathbb{E}_{\mathcal{X}_L}\left[\log q_D(\boldsymbol{c}_L|\boldsymbol{x}_L)\right]$$

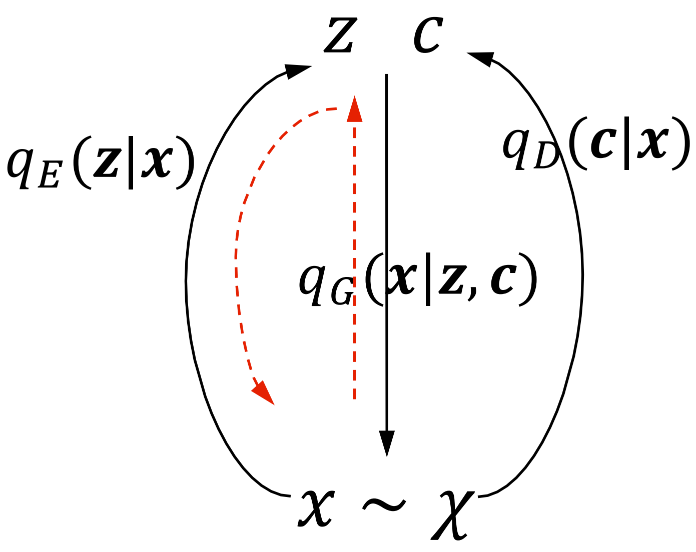
  - Each attribute discriminator can be trained on *separate* labeled datasets

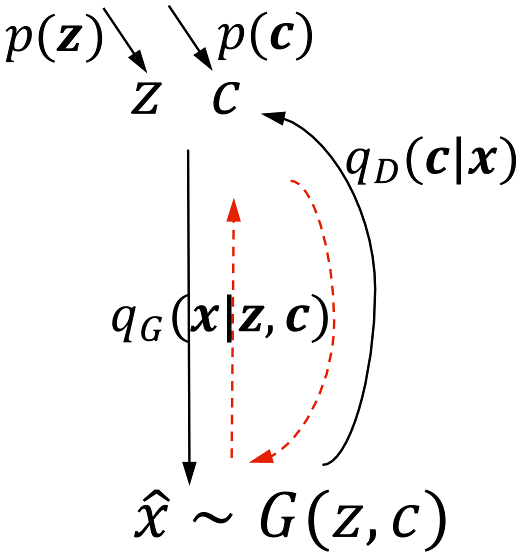
- Unsupervised objective on synthesized samples $\{(\hat{\boldsymbol{x}}, c)\}$ by the generator
  - Add a minimum entropy regularization to alleviate noise

$$\mathcal{L}_u(\boldsymbol{\theta}_D) = \mathbb{E}_{p_G(\hat{\boldsymbol{x}}|\boldsymbol{z},\boldsymbol{c})p(\boldsymbol{z})p(\boldsymbol{c})}\left[\log q_D(\boldsymbol{c}|\hat{\boldsymbol{x}}) + \beta\mathcal{H}(q_D(\boldsymbol{c}'|\hat{\boldsymbol{x}}))\right]$$

$$\min_{\boldsymbol{\theta}_D} \mathcal{L}_D = \mathcal{L}_s + \lambda_u \mathcal{L}_u$$

Alternative view: VAE + extended wake-sleep





VAE / Extended wake procedure:

- Use real data

Extended sleep procedure

- Use generated data

[Hu et al., 2017] "On unifying deep generative models"

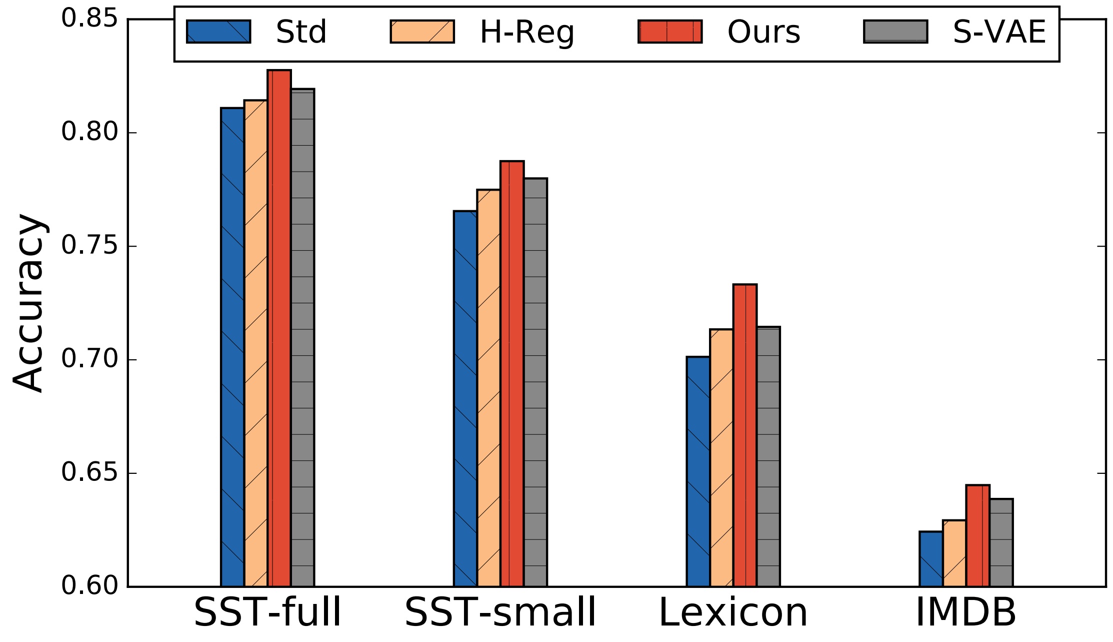
Experiments

- Sentence corpus
  - 350K IMDB movie reviews
  - Maximum sentence length = 15
- Control *sentiment* and *tense*
  - Sentiment dataset: IMDB, SST with labels ∈ {positive, negative}: 0.1-6K labels
  - Tense dataset: phrases/words with labels ∈ {past, present, future}: ~5K labels

Generation accuracy

Model	Dataset		
	SST-full	SST-small	Lexicon
S-VAE	0.822	0.679	0.660
Ours	**0.851**	**0.707**	**0.701**

Sentiment accuracy of generated sentences evaluated with a pre-trained sentiment classifier



Test-set accuracy of sentiment classifiers trained on generated sentences

Independence constraint

w/ independency constraint	w/o independency constraint
the film is strictly routine ! the film is full of imagination .	the acting is bad . the movie is so much fun .
after watching this movie , i felt that disappointed . after seeing this film , i ’m a fan .	none of this is very original . highly recommended viewing for its courage , and ideas .
the acting is uniformly bad either . the performances are uniformly good .	too bland highly watchable
this is just awful . this is pure genius .	i can analyze this movie without more than three words . i highly recommend this film to anyone who appreciates music .

Varying the code of tense	
i thought the movie was too bland and too much i guess the movie is too bland and too much i guess the film will have been too bland	this was one of the outstanding thrillers of the last decade this is one of the outstanding thrillers of the all time this will be one of the great thrillers of the all time

More examples

Varying the unstructured code *z*	
("negative", "past")	*("positive", "past")*
the acting was also kind of hit or miss .	his acting was impeccable
i wish i 'd never seen it	this was spectacular , i saw it in theaters twice
by the end i was so lost i just did n't care anymore	it was a lot of fun
("negative", "present")	*("positive", "present")*
the movie is very close to the show in plot and characters	this is one of the better dance films
the era seems impossibly distant	i 've always been a big fan of the smart dialogue .
i think by the end of the film , it has confused itself	i recommend you go see this, especially if you hurt
("negative", "future")	*("positive", "future")*
i wo n't watch the movie	i hope he 'll make more movies in the future
and that would be devastating !	i will definitely be buying this on dvd
i wo n't get into the story because there really is n't one	you will be thinking about it afterwards, i promise you

More examples

Failure cases	
the plot is not so original the plot weaves us into <unk>	it does n’t get any better the other dance movies it does n’t reach them , but the stories look
he is a horrible actor ’s most part he ’s a better actor than a standup	i just think so i just think !

Conclusions

- A new text generation model
  - Incorporates attribute discriminators for effective attribute semantic learning
  - Enables semi-supervised learning of both generator and discriminators
  - Requires only separate annotated data for each attribute
  - Imposes explicit independence constraints
- Future work
  - A general framework of **collaborative** generator-discriminator learning
  - Interpretable code representation provides an interface connecting end-to-end neural models with conventional structured methods
    - Combine structured knowledge with neural generative models [Hu et al., 2016]
    - Plug into dialog systems