Text Generation with No (Good) Data: New Reinforcement Learning and Causal Frameworks

Zhiting Hu
Assistant Professor, UC San Diego
Text Generation with (Clean) Supervised Data

Inspirational success

Machine Translation

Summarization

Description Generation

Captioning

Speech Recognition

OpenAI’s text-generating system GPT-3 is now spewing out 4.5 billion words a day

Robot-generated writing looks set to be the next big thing

By James Vincent | Mar 29, 2021, 8:24am EDT

[Image of text generation with supervised data]

Speak easy

Human scorers’ rating* of Google Translate and human translation

Translation method

<table>
<thead>
<tr>
<th>Phrase-based?</th>
<th>Neural-network?</th>
<th>Human</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Input sentence</th>
<th>Perfect translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pour l’ancienne secrétaire d’Etat, il s’agit de faire oublier un mois de caucholages et de convaincre l’auditoire que M. Trump n’a pas l’étoffe d’un président</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Phrase-based?
For the former secretary of state, this is to forget a month of bungling and convince the audience that Mr Trump has not the makings of a president

Neural-network?
For the former secretary of state, it is a question of forgetting a month of muddles and convincing the audience that Mr Trump does not have the stuff of a president

Human
The former secretary of state has to put behind her a month of setbacks and convince the audience that Mr Trump does not have what it takes to be a president

Source: Google

*0=completely nonsense translation, 6=perfect translation *Machine translation

[The Economist]
Text Generation with No (Good) Data?

Adversarial text examples

"entailment"  "neutral"  "contradiction"

Entailment classifier

The Old One always comforted Ca'daan, except today.
Your gift is appreciated by each and every student …
At the other end of Pennsylvania Avenue, people …
The person saint-pierre-et-saint-paul is ..
Automatically generating prompts to steer pretrained LMs
Controlling sentiment

Pos: The film is full of imagination!

Neg: The film is strictly routine!

Controlling writing style

Plain: LeBron James contributed 26 points, 8 rebounds, 7 assists.

Elaborate: LeBron James rounded out the box score with an all around impressive performance, scoring 26 points, grabbing 8 rebounds and dishing out 7 assists.

[Hu et al., 2017]  [Lin et al., 2020]
Biased data

Gender - occupation

She previously worked as a nurse practitioner

He went to law school and became a plaintiffs’ attorney
Text Generation with No (Good) Data?

Adversarial text examples

Controllable text generation

Prompt generation

Biased data

Gender - occupation

- She previously worked as a nurse practitioner
- He went to law school and became a plaintiffs’ attorney
Experiences of all kinds

Data examples

Type-2 diabetes is 90% more common than type-1

Constraints

Rewards

Auxiliary agents

Adversaries

... And all combinations of that ...
Experiences of all kinds

Learning from ALL Experiences: A Unifying ML Perspective

KDD2020 Tutorial

Zhiting Hu, Qirong Ho, and Eric Xing
Carnegie Mellon & Petuum

https://sites.google.com/view/kdd2020unified/home
Text Generation with Efficient (Soft) $Q$-Learning

Han Guo  Bowen Tan  Zhengzhong Liu  Eric P. Xing  Zhiting Hu
Reinforcement Learning (RL)

• Plug in arbitrary reward functions to drive learning
• Fertile research area for robotic and game control
• But … limited success for training text generation
• Challenges:
  • Large sequence space: \((\text{vocab-size})^{\text{text-length}} \sim (10^6)^{20}\)
  • Sparse reward: only after seeing the whole text sequence
• Impossible to train from scratch, usually initialized with MLE
• Unclear improvement vs MLE
RL for Text Generation: Background

- (Autoregressive) text generation model:

Sentence $y = (y_0, \ldots, y_T)$

In RL terms:

- trajectory, $\tau$
- action, $a_t$
- state, $s_t$
- policy $\pi_\theta(a_t | s_t)$

$$\pi_\theta(y_t | y_{<t}) = \frac{\exp f_\theta(y_t | y_{<t})}{\sum_y \exp f_\theta(y' | y_{<t})}$$
RL for Text Generation: Background

• (Autoregressive) text generation model:

Sentence $y = (y_0, ..., y_T)$

$$\pi_\theta(y_t | y_{<t}) = \frac{\exp f_\theta(y_t | y_{<t})}{\sum_{y'} \exp f_\theta(y' | y_{<t})}$$

In RL terms:

- Reward $r_t = r(s_t, a_t)$
  - Often **sparse**: $r_t = 0$ for $t < T$
- The general RL objective: maximize cumulative reward
  $$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[ \sum_{t=0}^{T} \gamma^t r_t \right]$$
- $Q$-function: expected future reward of taking action $a_t$ in state $s_t$
  $$Q^\pi(s_t, a_t) = \mathbb{E}_\pi \left[ \sum_{t'=t}^{T} \gamma^{t'} r_{t'} | s_t, a_t \right]$$
RL for Text Generation: Background

• On-policy RL
  • Most popular, e.g., Policy Gradient (PG)

\[ \nabla_\theta J(\pi_\theta) = -\mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{T} \hat{Q}(s_t, a_t) \nabla_\theta \log \pi_\theta (a_t \mid s_t) \right] \]

Generate text samples from the current policy \( \pi_\theta \) itself
• On-policy exploration to maximize the reward directly

Extremely low data efficiency: most samples from \( \pi_\theta \) are gibberish with zero reward
RL for Text Generation: Background

• Off-policy RL
  • e.g., $Q$-learning
  • Implicitly learns the policy $\pi$ by approximating the $Q^\pi(s_t, a_t)$
  • Bellman temporal consistency: $Q^*(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$

• Learns $Q_\theta$ with the regression objective:

\[
\mathcal{L}(\theta) = \mathbb{E}_{\pi'} \left[ \frac{1}{2} \left( r_t + \gamma \max_{a_{t+1}} Q_{\theta}(s_{t+1}, a_{t+1}) - Q_{\theta}(s_t, a_t) \right)^2 \right]
\]

• After learning, induces the policy as $a_t = \arg\max_a Q_\theta^*(s_t, a)$
RL for Text Generation: Background

- Off-policy RL
  - e.g., Q-learning
    - Implicitly learns the policy $\pi$ by approximating the $Q^\pi(s_t, a_t)$
    - Bellman temporal consistency: $Q^*(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})$
  - Learns $Q_\theta$ with the regression objective:
    \[
    \mathcal{L}(\theta) = \mathbb{E}_{\pi'} \left[ \frac{1}{2} \left( r_t + \gamma \max_{a_{t+1}} Q_\theta(s_{t+1}, a_{t+1}) - Q_\theta(s_t, a_t) \right)^2 \right]
    \]

- Regression target is unstable
  - Bootstrapped $Q_\theta$
  - Sparse reward $r_t = 0 \ (t < T)$: no “true” training signal

- After learning, induces the policy as $a_t = \arg\max_a Q_\theta^*(s_t, a)$

- Slow updates: gradient involves only $Q_\theta$-value of one action $a_t$ (vs $10^6$ vocab size)
RL for Text Generation: Background

• On-policy RL, e.g., *Policy Gradient (PG)*
  • Exploration to maximize reward directly
  
  red Extremely low data efficiency

• Off-policy RL, e.g., *Q-learning*
  red Unstable training due to bootstrapping & sparse reward
  red Slow updates due to large action space
  red Sensitive to training data quality; lacks on-policy exploration
New RL for Text Generation: Soft $Q$-Learning (SQL)

(Hard) $Q$-learning

- Goal

$$J(\pi) = \mathbb{E}_{r \sim \pi} \left[ \sum_{t=0}^{T} \gamma^t r_t \right]$$

- Induced policy

$$a_t = \text{argmax}_a Q_{\theta^*}(s_t, a)$$

SQL

- Goal: entropy regularized

$$J_{\text{MaxEnt}}(\pi) = \mathbb{E}_{r \sim \pi} \left[ \sum_{t=0}^{T} \gamma^t r_t + \alpha \mathcal{H}(\pi(\cdot | s_t)) \right]$$

- Induced policy

$$\pi_{\theta^*}(a_t | s_t) = \frac{\exp Q_{\theta^*}(a_t | s_t)}{\sum_a \exp Q_{\theta^*}(a | s_t)}$$

Generation model’s “logits” now act as $Q$-values!
New RL for Text Generation: Soft \( Q \)-Learning (SQL)

(Hard) \( Q \)-learning

- Goal
  \[ J(\pi) = \mathbb{E}_{r \sim \pi} \left[ \sum_{t=0}^{T} \gamma^t r_t \right] \]

- Induced policy
  \[ a_t = \text{argmax}_a Q_{\theta^*}(s_t, a) \]

- Training objective:
  - Based on temporal consistency
    🦇 Unstable training / slow updates

SQL

- Goal: entropy regularized
  \[ J_{\text{MaxEnt}}(\pi) = \mathbb{E}_{r \sim \pi} \left[ \sum_{t=0}^{T} \gamma^t r_t + \alpha \mathcal{H}(\pi(\cdot | s_t)) \right] \]

- Induced policy
  \[ \pi_{\theta^*}(a_t | s_t) = \frac{\exp Q_{\theta^*}(a_t | s_t)}{\sum_a \exp Q_{\theta^*}(a | s_t)} \]

- Training objective:
  - Based on **path consistency**
    😞 Stable / efficient
Efficient Training via Path Consistency

- (Single-step) path consistency
  \[ V^* (s_t) - \gamma V^* (s_{t+1}) = r_t - \log \pi^* (a_t \mid s_t) \]

- Objective
  \[ \mathcal{L}_{SQL, PCL}(\theta) = \mathbb{E}_{\pi'} \left[ \frac{1}{2} \left( -V_{\theta} (s_t) + \gamma V_{\theta} (s_{t+1}) + r_t - \log \pi_{\theta} (a_t \mid s_t) \right) \right] \]
  \[ \approx A_{\theta} (s_t, a_t), \text{ advantage} \]

Fast updates: gradient involves \( Q_{\theta} \) values of **all** tokens in the vocab

\[ V^* (s) = \log \sum_{a'} \exp Q^* (s, a') \]

\[ \pi^* (a \mid s) = \frac{\exp Q^* (s, a)}{\sum_{a'} \exp Q^* (s, a')} \]

SQL matches log probability of token \( a_t \) with its advantage v.s.

MLE increases log probability of token \( a_t \) blindly
Efficient Training via Path Consistency

- (Single-step) path consistency
  \[ V^*(s_t) - \gamma V^*(s_{t+1}) = r_t - \log \pi^*(a_t | s_t) \]
  
  **Objective**
  \[
  \mathcal{L}_{\text{SQL, PCL}}(\theta) = \mathbb{E}_{\pi'} \left[ \frac{1}{2} \left( -V_{\tilde{\theta}}(s_t) + \gamma V_{\tilde{\theta}}(s_{t+1}) + r_t - \log \pi_\theta(a_t | s_t) \right) \right]
  \]

- (Multi-step) path consistency
  \[ V^*(s_t) - \gamma^{T-t} V^*(s_{T+1}) = \sum_{l=0}^{T-t} \gamma^l (r_{t+l} - \log \pi^*(a_{t+l} | s_{t+l})) \]
  
  **Objective**
  \[
  \mathcal{L}_{\text{SQL, PCL-ms}}(\theta) = \mathbb{E}_{\pi'} \left[ \frac{1}{2} \left( -V_{\tilde{\theta}}(s_t) + \gamma^{T-t} r_T \sum_{l=0}^{T-t} \gamma^l \log \pi_\theta(a_{t+l} | s_{t+l}) \right)^2 \right]
  \]

\[ V^*(s) = \log \sum_{a'} \exp Q^*(s, a') \]
\[ \pi^*(a | s) = \frac{\exp Q^*(s, a)}{\sum_{a'} \exp Q^*(s, a')} \]

- **Fast updates:** gradient involves \( Q_\theta \) values of all tokens in the vocab
- **Stable updates:** Non-zero reward signal \( r_T \) as regression target
Efficient Training via Path Consistency

- (Single-step) path consistency
  \[ V^*(s_t) - \gamma V^*(s_{t+1}) = r_t - \log \pi^*(a_t | s_t) \]

- Objective
  \[ \mathcal{L}_{\text{SQL, PCL}}(\theta) = \mathbb{E}_{\pi'} \left[ \frac{1}{2} \left( -V_{\theta}(s_t) + \gamma V_{\theta}(s_{t+1}) + r_t \right) - \log \pi_{\theta}(a_t | s_t) \right] \]

  - Regression target
  - Fast updates: gradient involves \( Q_\theta \) values of all tokens in the vocab
  - Stable updates: Non-zero reward signal \( r_T \) as regression target

- Arbitrary policy:
  - Training data (if available) → off-policy updates
  - Current policy → on-policy updates
  - We combine both for the best of the two
Implementation is easy

```python
def multi_step_SQL_objective(
    Q_values, Q_values_target, actions, rewards):
    V = Q_values.logsumexp(dim=-1)
    A = Q_values[actions] - V

    V_target = Q_values_target.logsumexp(dim=-1)

    A2 = masked_reverse_cumsum(
        A, lengths=actions.sequence_length,
        dim=-1)

    return F.mse_loss(
        A2, rewards.view(-1, 1) - V_target,
        reduction="none")
```

```python
model = TransformerLM(...)

for iter in range(max_iters):
    if mode == "off-policy":
        batch = dataset.sample_batch()
        sample_ids = batch.text_ids
    if mode == "on-policy":
        sample_ids = model.decode()

    Q_values = model.forward(sample_ids)
    Q_values_target = target_model.forward(sample_ids)

    rewards = compute_rewards(sample_ids)

    sql_loss = multi_step_SQL_objective(
        Q_values,
        Q_values_target,
        actions=sample_ids,
        rewards=rewards)

    # gradient descent over sql_loss
    # ...
```
Applications & Experiments
Application (I): Learning from Noisy (Negative) Text

• Entailment generation
  • Given a premise, generates a hypothesis that entails the premise
  • “Sophie is walking a dog outside her house” -> “Sophie is outdoor”
  • Negative sample: “Sophie is inside her house”

• Training data:
  • Subsampled 50K (premise, hypothesis) noisy pairs from SNLI
  • Average entailment probability: 50%
  • 20K examples have entailment probability < 20% (≈ negative samples)

• Rewards:
  • Entailment classifier
  • Pretrained LM for perplexity
  • BLEU w.r.t input premises (which effectively prevents trivial generations)
Application (I): Learning from Noisy (Negative) Text

- **MLE** and pure off-policy RL (GOLD-s) do not work ← rely heavy on data quality
- SQL (full) > MLE+PG (PG alone does not work)
- SQL (single-step only) does not work: the multi-step SQL objective is crucial

Entailment-rate and language-quality vs diversity (top-$p$ decoding w/ different $p$)
Application (II): Universal Adversarial Attacks

• Attacking entailment classifier
  • Generate **readable** hypotheses that are classified as “entailment” for all premises
  • **Unconditional** hypothesis generation model

• Training data:
  • No direct supervision data available
  • “Weak” data: all hypotheses in MultiNLI corpus

• Rewards:
  • Entailment classifier to attack
  • Pretrained LM for perplexity
  • BLEU w.r.t input premises
  • Repetition penalty

Previous adversarial algorithms are not applicable here:
• only attack for specific premise
• not readable
Application (II): Universal Adversarial Attacks

- SQL (full) > MLE+PG (PG alone does not work)
- MLE+PG collapses: cannot generate more diverse samples

<table>
<thead>
<tr>
<th>Model</th>
<th>Generation</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE+PG</td>
<td>it’s.</td>
<td>90.48</td>
</tr>
<tr>
<td>SQL (ours)</td>
<td>the person saint-pierre-et-saint-paul is saint-pierre-et-saint-paul.</td>
<td>97.40</td>
</tr>
</tbody>
</table>

Samples of highest attack rate
Application (III): Prompt Generation for Controlling LMs

- Generate prompts to steer pretrained LM to produce topic-specific sentences

Existing gradient-based prompt tuning methods are not applicable due to discrete components
Application (III): Prompt Generation for Controlling LMs

- Steered decoding: PPLM, GeDi
- **SQL** achieves best accuracy-fluency trade-off
- Prompt control by **SQL, MLE+PG > PPLM, GeDi**
  - and much faster at inference!
- **SQL (off-policy only) > MLE**

<table>
<thead>
<tr>
<th></th>
<th>PPLM</th>
<th>GeDi</th>
<th>MLE (5)</th>
<th>SQL (off, 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12.69</td>
<td>123.88</td>
<td>25.70</td>
<td>25.77</td>
</tr>
<tr>
<td>MLE+PG (5/10/15)</td>
<td>25.52/28.16/28.71</td>
<td>25.94/26.95/29.10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>PPLM</th>
<th>GeDi</th>
<th>SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seconds</td>
<td>5.58</td>
<td>1.05</td>
<td>0.07</td>
</tr>
</tbody>
</table>
Promising results on standard supervised tasks

- **SQL** from scratch is competitive with **MLE** in terms of performance and stability
  - Results on E2E dataset
  - **PG** from scratch fails

<table>
<thead>
<tr>
<th>Model</th>
<th>MLE</th>
<th>PG</th>
<th>MLE+PG</th>
<th>SQL (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>val</td>
<td>45.67</td>
<td>0.00</td>
<td>49.08</td>
<td>47.04</td>
</tr>
<tr>
<td>test</td>
<td>41.75</td>
<td>0.00</td>
<td>42.26</td>
<td>41.70</td>
</tr>
</tbody>
</table>

![Training curves](image)
Promising results on standard supervised tasks

- **SQL** from scratch is competitive with **MLE** in terms of performance and stability
  - Results on E2E dataset
  - **PG** from scratch fails
- **SQL** is less sensitive to hyperparameters than **MLE+PG**

Training curves of different reward scales
Summary of SQL for Text Generation

• On-policy RL, e.g., Policy Gradient (PG)
  😡 Extremely low data efficiency
• Off-policy RL, e.g., $Q$-learning
  😡 Unstable training; slow updates; sensitive to training data quality

• SQL
  • Objectives based on path consistency
    😎 Combines the best of on-/off-policy, while solving the difficulties
  🧡 Stable training from scratch given sparse reward
  😋 Fast updates given large action space
  • Opens up enormous opportunities for integrating more advanced RL for text generation
Text Generation with No (Good) Data?

Biased data

Gender - occupation

She previously worked as a nurse practitioner

He went to law school and became a plaintiffs’ attorney
A Causal Lens for Controllable Text Generation

Zhiting Hu

Erran Li
Controllable Text Generation

- Generates text $x$ that contains desired properties $a$
  - Attributes, e.g., sentiment, tense, politeness, formality, ...
  - Structures, e.g., conversation strategies
- Two core tasks:
  - Attribute-conditional generation
    - Sentiment = negative $\Rightarrow$ “The film is strictly routine.”
  - Text attribute (style) transfer
    - “The film is strictly routine.” $\Rightarrow$ “The film is full of imagination.”
- Applications:
  - Emotional chatbot [e.g. Rashkin et al., 2018; Zhou et al., 2018]
  - Generating text adversarial examples [e.g. Zhao et al., 2018]
  - Data augmentation [e.g. Verma et al., 2018; Malandrakis et al., 2019]
Common Methods of Controllable Text Generation

- Separate solutions for the two tasks
  - Attribute-conditional generation: $p(x|a)$
  - Text attribute transfer: $p(x'|x,a')$

- ML-based models that learn **correlations** in the data
  - Joint/marginal/conditional distributions
  - Also inherits bias from data

- Limited generalization

Causal ladder [Pearl 2000]
Controllable Text Generation from Causal Perspective

- A unified framework for the two tasks
  - Models causal relationships, not spurious correlations
  - Generates unbiased text using rich causality tools

Causal ladder [Pearl 2000]
Controllable Text Generation from Causal Perspective

- A unified framework for the two tasks
  - Models causal relationships, not spurious correlations
  - Generates unbiased text using rich causality tools

- Attribute-conditional generation: \( p(x|\text{do}(a)) \)
  - Intervention
  - \( \text{do} \)-operation: removes dependence b/w \( a \) and confounders
Controllable Text Generation from Causal Perspective

- A unified framework for the two tasks
  - Models causal relationships, not spurious correlations
  - Generates unbiased text using rich causality tools

- Attribute-conditional generation: $p(x|do(a))$
  - Intervention
  - $do$-operation: removes dependence b/w $a$ and confounders

- Text attribute transfer: $p(x'|x, a(x), a')$
  - Counterfactual
  - “What would the text be if the attribute had taken a different value?“

Causal ladder [Pearl 2000]
The Basis: Structural Causal Model (SCM)

- Describes causal relationships between variables

(Latent) confounders: any factors correlating w/ both treatment and outcome

treatment: attributes of interest, e.g., sentiment

proxy: observed information of confounders, e.g., food type

outcome: text, e.g., restaurant reviews

$\text{outcome: text, e.g., restaurant reviews}$

$p_\theta(x, a, z, c) = p_\theta(x|a, z)p_\theta(a|z)p_\theta(c|z)p_0(z)$

Variational distribution $q_\phi(z|x, a, c)$

Often available for only a small subset of data, e.g., by asking humans to annotate.

- Previous unbiased generation work essentially assumes full unbiased proxy labels
Inference (I): **Intervention** for Attribute-Conditional Generation

- Association (correlation): $p(x|a)$

\[
p(x|a) = \sum_z p_\theta(x|a,z)p_\theta(z|a)
\]

- Intervention: $p(x|do(a))$
  - Sets $a$ to a given value independently of $z$

\[
p(x|do(a)) = \sum_z p_\theta(x|a,z)p_\theta(z)
\]
Inference (I): **Intervention** for Attribute-Conditional Generation

- Association (correlation): $p(x|a)$

\[ p(x|a) = \sum_z p_\theta(x|a, z)p_\theta(z|a) \]

- Intervention: $p(x|do(a))$
  - Sets $a$ to a given value independently of $z$

\[ p(x|do(a)) = \sum_z p_\theta(x|a, z)p_\theta(z) \]
Inference (II): Counterfactual for Text Attribute Transfer

• What would the text be if the attribute had taken a different value?

• Counterfactuals as a standard three-step procedure [Pearl 2000]

1) **Abduction**: predicts \( z \) given \( x \): \( z \sim q_\phi(z|x, a, c) \)

2) **Action**: performs intervention, \( do(a = a') \)

3) **Prediction**: generates \( x' \) given \( z \) and \( a' \) following the SCM: \( x' \sim p_\theta(x'|a', z) \)
Inference (III): Propensity Reweighting for Debiasing Pretrained LMs

• Given (biased) pretrained LM $p_{LM}(x|a)$
• Can we convert it to unbiased $p(x|do(a))$?

$$p(x|do(a)) = \sum_z p(x|a, z)p(z)$$
Inference (III): Propensity Reweighting for Debiasing Pretrained LMs

• Given (biased) pretrained LM $p_{LM}(x|a)$
• Can we convert it to unbiased $p(x|do(a))$?

\[
p(x|do(a)) = \sum_z p(x|a, z)p(z)
= \sum_z p(x|a, z)p(z|a) \frac{p(a)}{p(a|z)}
= \sum_z p(x|a)p(z|x, a) \frac{p(a)}{p(a|z)}
\]

Propensity score: the probability of the $z$ being assigned to the treatment $a$
Inference (III): **Propensity Reweighting** for Debiasing Pretrained LMs

- Given (biased) pretrained LM \( p_{LM}(x|a) \)
- Can we convert it to unbiased \( p(x|do(a)) \) ?

\[
p(x|do(a)) = \sum_z p(x|a, z)p(z)
= \sum_z p(x|a, z)p(z|a) \frac{p(a)}{p(a|z)}
= \sum_z p(x|a)p(z|x, a) \frac{p(a)}{p(a|z)} = \sum_z p_{LM}(x|a)q_\phi(z|x, a, c) \frac{p(a)}{p_\theta(a|z)}
\]

**Propensity score:** the probability of the \( z \) being assigned to the treatment \( a \)
Inference (III): **Propensity Reweighting** for Debiasing Pretrained LMs

- Given (biased) pretrained LM $p_{LM}(x|a)$
- Can we convert it to unbiased $p(x|do(a))$?

\[
p(x|do(a)) = \sum_z p(x|a, z)p(z)
= \sum_z p(x|a, z)p(z|a) \frac{p(a)}{p(a|z)}
= \sum_z p(x|a)p(z|x, a) \frac{p(a)}{p(a|z)} = \sum_z p_{LM}(x|a)q_\phi(z|x, a, c) \frac{p(a)}{p_\theta(a|z)}
\]

- Sampling-importance-resampling (SIR):
  - Biased samples ~ $p_{LM}(x|a)$
  - Compute sample weights
  - Resampling proportional to the weights
Learning of the SCM

\[ p_\theta(x, a, z, c) = p_\theta(x|a, z)p_\theta(a|z)p_\theta(c|z)p_0(z) \]

Variational distribution \( q_\phi(z|x, a, c) \)

- Variational autoencoder (VAE) objective

\[ \mathcal{L}_{vae}(\theta, \phi) = \mathbb{E}_{z \sim q_\phi} \left[ \log p_\theta(x|a, z) + \lambda_a \log p_\theta(a|z) + \lambda_c \log p_\theta(c|z) \right] - \lambda_{kl} \text{KL} (q_\phi || p_0) \]

- Counterfactual objectives
  - Draws inspirations from causality, disentangled representations & controllable generation
  - Intuition: counterfactual \( x' \) must entail \( a' \) and preserve the original \( z \) and \( c \)
Experiments

• Two datasets with strong spurious correlations
  • Yelp customer reviews:
    • Attribute $\alpha$: sentiment (1:positive, 0:negative)
    • Confounding proxy $\gamma$: category (1:restaurant, 0:others)
    • Correlation: 90% data have the same sentiment and category labels
    • Size: 510K for training, wherein 10K have category labels
  • Bios: online biographies
    • Attribute $\alpha$: gender (1:female, 0:male)
    • Confounding proxy $\gamma$: occupation (1:nurse etc, 0:rapper etc)
    • Correlation: 95%
    • Size: 43K for training, wherein 3K have occupation labels
  • Models:
    • Based on GPT-2 (117M)

\[
\begin{align*}
\alpha &= 1, \gamma &= 1 \\
\text{Soup and salad came out quickly!}
\end{align*}
\]

\[
\begin{align*}
\alpha &= 0, \gamma &= 0 \\
I texted and called Phil several times and he never responded
\end{align*}
\]

\[
\begin{align*}
\alpha &= 1, \gamma &= 1 \\
\text{She previously worked as a nurse practitioner}
\end{align*}
\]

\[
\begin{align*}
\alpha &= 0, \gamma &= 0 \\
\text{He went to law school and became a plaintiffs’ attorney}
\end{align*}
\]
(I) Attribute-Conditional Generation

- Causal model improves control accuracy and reduces bias

<table>
<thead>
<tr>
<th>Methods</th>
<th>Control accuracy (↑)</th>
<th>Bias (↓)</th>
<th>Fluency (↑)</th>
<th>Diversity (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditional LM</td>
<td>79.1</td>
<td>78.7</td>
<td>-50.4</td>
<td>41.4</td>
</tr>
<tr>
<td>Conditional LM (full)</td>
<td>80.3</td>
<td>78.9</td>
<td>-50.8</td>
<td>41.9</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>96.3</strong></td>
<td><strong>59.8</strong></td>
<td>-51.3</td>
<td>39.1</td>
</tr>
</tbody>
</table>
(I) Attribute-Conditional Generation

- Causal model improves control accuracy and reduces bias

<table>
<thead>
<tr>
<th></th>
<th>Methods</th>
<th>Control accuracy (↑)</th>
<th>Bias (↓)</th>
<th>Fluency (↑)</th>
<th>Diversity (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YELP</td>
<td>Conditional LM</td>
<td>79.1</td>
<td>78.7</td>
<td>-50.4</td>
<td>41.4</td>
</tr>
<tr>
<td></td>
<td>Conditional LM (full)</td>
<td>80.3</td>
<td>78.9</td>
<td>-50.8</td>
<td>41.9</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td><strong>96.3</strong></td>
<td><strong>59.8</strong></td>
<td>-51.3</td>
<td>39.1</td>
</tr>
<tr>
<td>BIOS</td>
<td>Conditional LM</td>
<td>95.51</td>
<td>84.73</td>
<td><strong>-17.0</strong></td>
<td>46.5</td>
</tr>
<tr>
<td></td>
<td>Conditional LM (full)</td>
<td>93.28</td>
<td>72.34</td>
<td><strong>-18.5</strong></td>
<td>48.5</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td><strong>99.2</strong></td>
<td><strong>62.4</strong></td>
<td><strong>-32.0</strong></td>
<td><strong>40.6</strong></td>
</tr>
</tbody>
</table>

Automatic evaluation
(I) Attribute-Conditional Generation

- Causal model improves control accuracy and reduces bias

<table>
<thead>
<tr>
<th></th>
<th>Methods</th>
<th>Control accuracy (↑)</th>
<th>Bias (↓)</th>
<th>Fluency (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YELP</td>
<td>Conditional LM (full)</td>
<td>80.0</td>
<td>73.0</td>
<td>3.90</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>97.0</td>
<td>56.0</td>
<td>3.85</td>
</tr>
<tr>
<td>BIOS</td>
<td>Conditional LM (full)</td>
<td>96.0</td>
<td>82.0</td>
<td>4.43</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>99.0</td>
<td>60.0</td>
<td>4.25</td>
</tr>
</tbody>
</table>

Human evaluation
(I) Attribute-Conditional Generation

restaurant

<table>
<thead>
<tr>
<th>Conditional LM (FULL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = 0$ (sentiment negative)</td>
</tr>
<tr>
<td>this was the worst experience i 've ever had at a glazier .</td>
</tr>
<tr>
<td>i even asked him if they could play on the tv channel .</td>
</tr>
<tr>
<td>this was pretty fun the first time i went . &quot;</td>
</tr>
<tr>
<td>waited in line once but almost never reached the floor .</td>
</tr>
<tr>
<td>if you are ever up in chandler , tony will stop by .</td>
</tr>
<tr>
<td>$a = 1$ (sentiment positive)</td>
</tr>
<tr>
<td>very good and long wait time .</td>
</tr>
<tr>
<td>we loved our favorite harrah 's night ! &quot;</td>
</tr>
<tr>
<td>i would love to try this restaurant again when they open . &quot;</td>
</tr>
<tr>
<td>this place is great .</td>
</tr>
<tr>
<td>everything you will find in this restaurant !</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = 0$ (sentiment negative)</td>
</tr>
<tr>
<td>no , it 's obvious that they were overcooked .</td>
</tr>
<tr>
<td>the seats were poorly done and basically sucked up .</td>
</tr>
<tr>
<td>it was n’t enough to ask us if it was okay .</td>
</tr>
<tr>
<td>very disappointed with my food order yesterday .</td>
</tr>
<tr>
<td>i declined to replace it tho they were bad .</td>
</tr>
<tr>
<td>$a = 1$ (sentiment positive)</td>
</tr>
<tr>
<td>great for a relaxed evening out .</td>
</tr>
<tr>
<td>i ’m beyond impressed with the passion fruit and unbeatable service</td>
</tr>
<tr>
<td>it ’s a true pleasure to meet andrew .</td>
</tr>
<tr>
<td>jacksville became my go-to spot for dessert</td>
</tr>
<tr>
<td>thank you for the technique , i am quite impressed .</td>
</tr>
</tbody>
</table>
(II) Text Attribute Transfer

- Previous methods tend to fail on the challenging dataset: low control accuracy
- Causal model obtains much higher accuracy, and keeps bias low

<table>
<thead>
<tr>
<th>Methods</th>
<th>Control accuracy (↑)</th>
<th>Bias (↓)</th>
<th>Preservation (↑)</th>
<th>Fluency (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hu et al. [22]</td>
<td>44.1</td>
<td>68.4</td>
<td>77.7</td>
<td>-132.7</td>
</tr>
<tr>
<td>He et al. [20]</td>
<td>35.3</td>
<td>60.2</td>
<td>80.1</td>
<td>-57.7</td>
</tr>
<tr>
<td>Ablation: Ours w/o $c.f-z/c$</td>
<td>75.0</td>
<td>67.8</td>
<td>36.3</td>
<td>-34.2</td>
</tr>
<tr>
<td>Ours</td>
<td>77.0</td>
<td>61.4</td>
<td>42.3</td>
<td>-29.6</td>
</tr>
</tbody>
</table>

Results on *biased* Yelp dataset
(II) Text Attribute Transfer

- Previous methods tend to fail on the challenging dataset: low control accuracy
- Causal model obtains much higher accuracy, and keeps bias low
- Also gets improvement on unbiased data

<table>
<thead>
<tr>
<th>Methods</th>
<th>Control accuracy (↑)</th>
<th>Preservation (↑)</th>
<th>Fluency (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>self-BLEU</td>
<td>ref-BLEU</td>
</tr>
<tr>
<td>Hu et al. [22]</td>
<td>86.7</td>
<td>58.4</td>
<td>-</td>
</tr>
<tr>
<td>Shen et al. [65]</td>
<td>73.9</td>
<td>20.7</td>
<td>7.8</td>
</tr>
<tr>
<td>He et al. [20]</td>
<td>87.9</td>
<td>48.4</td>
<td>18.7</td>
</tr>
<tr>
<td>Dai et al. [7]</td>
<td>87.7</td>
<td>54.9</td>
<td>20.3</td>
</tr>
<tr>
<td>Ablation: Ours w/o $c_f-z/c$</td>
<td>87.1</td>
<td>57.2</td>
<td>24.3</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>91.9</strong></td>
<td><strong>57.3</strong></td>
<td><strong>25.5</strong></td>
</tr>
</tbody>
</table>

Results on unbiased Yelp dataset (commonly used in previous study)
(III) Debiasing Pretrained LMs

- Resampling 2K out of 10K biased samples
- Substantially reduced bias

<table>
<thead>
<tr>
<th>Methods</th>
<th>Control accuracy (↑)</th>
<th>Bias (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YELP</td>
<td>Conditional LM</td>
<td>79.1</td>
</tr>
<tr>
<td></td>
<td>Debiased (Ours)</td>
<td>77.3</td>
</tr>
</tbody>
</table>

Debiasing results on Yelp
Summary of Causal Lens for Controllable Generation

- Causality + ML for unified unbiased controllable generation
  - Intervention
  - Counterfactual
  - Propensity reweighting
- Causal modeling for more text generation problems?
  - Dialog, summarization, ...

Causal ladder [Pearl 2000]
Thanks!