Text Generation by Offline Reinforcement Learning

He He

Tsinghua University, IIIS, RL Reading Group

May 31, 2022
The status quo for text generation

- **Modeling**: Auto-regressive models

\[ p(\text{output} \mid \text{context}) = \prod_t p(t\text{-th word} \mid \text{prefix, context}) \]
The status quo for text generation

- **Learning**: Maximum likelihood estimation
  \[ \max_{\theta} \sum_{\text{reference}} \log p_\theta(\text{reference} | \text{context}) \]

- **Inference**: focus on the high-likelihood region
  - **Search** for the highest-likelihood output:
    - greedy decoding, beam search
  - **Sample** from the learned distribution:
    - top-\(p\), top-\(k\), tempered sampling
Likelihood vs quality

High log-likelihood $\not\rightarrow$ high quality

A: How about watching a movie?
B: I don’t know.
A: Let’s go home then.
B: I don’t know.
beam-1: British woman won Olympic gold in pair rowing.
beam-1000: </s>

[Zhang+ 2020] [Li+ 2016] [Murray+ 2018, Ott+ 2018]
What does the model error look like?

MLE tends to **over-generalize** [Huszár 2015]
What does the model error look like?

MLE tends to **over-generalize** [Huszár 2015]

The horse raced past the barn fell.
What does the model error look like?

MLE tends to **over-generalize** [Huszár 2015]

The horse raced past the barn fell.
What does the model error look like?

MLE tends to **over-generalize** [Huszár 2015]

![Graph showing model error]

MLE is “high recall”,

The horse raced like the barn.
What does the model error look like?

MLE tends to **over-generalize** [Huszár 2015]

The horse fell.

MLE is “high recall”, but a “high precision” solution may be preferred.
Misaligned training and evaluation objectives

- **Training:** \( \text{max log } p \)
- **Inference:** beam-\( k \), top-\( k \), top-\( p \), tempered sampling, ...

- **Data** → **Model** → **Output**

- **Log-likelihood** of the reference text
- **Quality** of the output text (judged by humans)
Training vs evaluation losses

**Training** loss (NLL): 

\[ E_{p_{\text{human}}} \left[ - \log p_{\theta}(\text{output} \mid \text{context}) \right] \]

**Evaluation** loss (perceptual quality): 

\[ E_{p_{\theta}} \left[ - \log p_{\text{human}}(\text{output} \mid \text{context}) \right] \]
Training vs evaluation losses

**Training** loss (NLL):

\[
\mathbb{E}_{p_{\text{human}}} [- \log p_{\theta}(\text{output} \mid \text{context})]
\]

**Evaluation** loss (perceptual quality):

\[
\mathbb{F}_{p_{\theta}} [- \log p_{\text{human}}(\text{output} \mid \text{context})]
\]
Training vs evaluation losses

**Training** loss (NLL):

\[ \mathbb{E}_{p_{human}} [- \log p_{\theta}(output \mid context)] \]

- High recall: \( p_{\theta} \) must cover all outputs from \( p_{human} \)

**Evaluation** loss (perceptual quality):

\[ \mathbb{E}_{p_{\theta}} [- \log p_{human}(output \mid context)] \]

- High precision: all output from \( p_{\theta} \) must be scored high under \( p_{human} \)
The reinforcement learning formulation

Evaluation loss (perceptual quality):

$$-\mathbb{E}_{p_\theta} \left[ \sum_t \log p_{human}(t\text{-th word} \mid \text{prefix, context}) \right]$$
The reinforcement learning formulation

**Evaluation loss (perceptual quality):**

\[ -\mathbb{E}_{p_{\theta}} \left[ \sum_t \log p_{\text{human}}(t-\text{th word} \mid \text{prefix, context}) \right] \]

**The RL objective: expected return**

\[ J(\theta) = \mathbb{E}_{\pi_{\theta}} \left[ \sum_t R(a_t, s_t) \right] \]

**Aligned training and evaluation losses**
Existing RL approaches for text generation

Directly optimize a sequence-level metric (reward), e.g., BLEU, ROUGE, using policy gradient.
Existing RL approaches for text generation

Directly optimize a sequence-level metric (reward), e.g., BLEU, ROUGE, using policy gradient.

Pros:

- Aligned training and evaluation goals
- May discover high-quality outputs outside the references.
Existing RL approaches for text generation

Directly optimize a sequence-level metric (reward), e.g., BLEU, ROUGE, using policy gradient.

Pros:

▶ Aligned training and evaluation goals
▶ May discover high-quality outputs outside the references.

Cons:

we have the the the the the the ...  
i to me to me to me to me ...  

degenerative solution
Optimization challenges

Obstacles:

- Gradient estimated by samples from $\pi_\theta$ has high variance.
- Degenerate once the reward is close to zero.
Optimization challenges

Obstacles:
- Gradient estimated by samples from $\pi_\theta$ has high variance.
- Degenerate once the reward is close to zero.

Current solution: Stay close to the reference by MLE regularization, but this defeats the purpose of RL!

(Marginal improvement in practice [Wu+ 2018, Choshen+ 2020])
Optimization challenges

**Obstacles:**
- Gradient estimated by samples from $\pi_\theta$ has high variance.
- Degenerate once the reward is close to zero.

**Current solution:** Stay close to the reference by MLE regularization, but this defeats the purpose of RL!

(Marginal improvement in practice [Wu+ 2018, Choshen+ 2020])

**Problem:** policy/generator interacting with the environment.
Is interaction useful?

Learn about the environment dynamics.

We already know the dynamics.

Explore novel actions that may lead to higher reward.

We don't have good reward functions (evaluation) yet.
Is interaction useful?

- Learn about the environment dynamics.
Is interaction useful?

- Learn about the environment dynamics.
  - We already know the dynamics.
Is interaction useful?

- Learn about the environment dynamics.
  - We already know the dynamics.
- Explore novel actions that may lead to higher reward.
Is interaction useful?

- Learn about the environment dynamics.
  - We already know the dynamics.
- Explore novel actions that may lead to higher reward.
  - We don’t have good reward functions (evaluation) yet.
Summary so far

Desired loss:

\[-\mathbb{E}_{p_{\theta}} \log p_{\text{human}}(\text{output} \mid \text{context})\] (high precision)

Existing approaches:

- MLE: misaligned losses, easy to optimize
- RL: aligned losses, hard to optimize
Online policy gradient

Objective: $\mathbb{E}_{\pi_\theta} [R(s, a)]$
Online policy gradient

Objective: $\mathbb{E}_{\pi_\theta} [R(s, a)]$

The horse fell heavily $0.5$
Online policy gradient

Objective: $\mathbb{E}_{\pi_{\theta}} [R(s, a)]$

\[
\nabla_{\theta} J(\theta) = \mathbb{E}_{\pi_{\theta}} \left[ \sum_{t} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{Q}(s_t, a_t) \right]
\]
Offline policy gradient

Objective: $\mathbb{E}_{\pi_\theta} \left[ R(s, a) \right]$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_D} \left[ \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{Q}(s_t, a_t) \right]$$

The horse raced past the barn fell 0.5
Objective: $\mathbb{E}_{\pi_\theta} [R(s, a)]$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_D} \left[ \sum_t w_t \nabla_\theta \log \pi_\theta(a_t | s_t) \hat{Q}(s_t, a_t) \right]$$
Approximated importance weights

\[ w_t = \pi_\theta(a_t | s_t) \]

- **Intuition**: up-weight actions preferred by the current policy
- Closer to model distribution
What is a good reward function

Offline policy gradient:

\[ \nabla_{\theta} J(\theta) \approx E_{\pi_D} \left[ \sum_t \pi_\theta(a_t \mid s_t) \nabla_{\theta} \log \pi_\theta(a_t \mid s_t) \hat{Q}(s_t, a_t) \right] \]

\[ \sum_{t'=t}^T R(s_{t'}, a_{t'}) \]

- Finding a good \( R \) is hard in general (the evaluation problem).
- But we only need to score the demonstrations.
What is a good reward function

Offline policy gradient:

\[ \nabla_{\theta} J(\theta) \approx \mathbb{E}_{\pi_D} \left[ \sum_t \pi_\theta(a_t \mid s_t) \nabla_{\theta} \log \pi_\theta(a_t \mid s_t) \hat{Q}(s_t, a_t) \right] \]

\[ \sum_{t'=t}^{T} R(s_{t'}, a_{t'}) \]

- Finding a good \( R \) is hard in general (the evaluation problem).
- But we only need to score the demonstrations.
What is a good reward function

Offline policy gradient:

\[ \nabla_{\theta} J(\theta) \approx E_{\pi_D} \left[ \sum_{t} \pi_{\theta}(a_{t} \mid s_{t}) \nabla_{\theta} \log \pi_{\theta}(a_{t} \mid s_{t}) \hat{Q}(s_{t}, a_{t}) \right] \]

\[ \sum_{t'=t}^{T} R(s_{t'}, a_{t'}) \]

- Finding a good \( R \) is hard in general (the evaluation problem).
- But we only need to score the demonstrations.
What is a good reward function

Offline policy gradient:

\[ \nabla_{\theta} J(\theta) \approx \mathbb{E}_{\pi_D} \left[ \sum_t \pi_{\theta}(a_t \mid s_t) \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t) \hat{Q}(s_t, a_t) \right] \]

\[
\sum_{t' = t}^{T} R(s_{t'}, a_{t'})
\]

- Finding a good \( R \) is hard in general (the evaluation problem).
- But we only need to score the demonstrations.

| naive |  
|---|---|
| The horse fell | 1 |
| The horse was in the barn | 1 |
| The horse raced past the barn fell | 1 |
What is a good reward function

Offline policy gradient:

\[ \nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_D} \left[ \sum_t \pi_\theta(a_t \mid s_t) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{Q}(s_t, a_t) \right] \sum_{t'=t}^T R(s_{t'}, a_{t'}) \]

- Finding a good $R$ is hard in general (the evaluation problem).
- But we only need to score the demonstrations.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>naive</th>
<th>ideal ($R = \log p_{\text{human}}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The horse fell</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>The horse was in the barn</td>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>The horse raced past the barn</td>
<td>1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Estimate $p_{\text{human}}$ (for the demonstrations)

\[ R_{\text{ideal}} = \log p_{\text{human}} \]
Estimate $p_{\text{human}}$ (for the demonstrations)

Approximate $p_{\text{human}}$ using the demonstrations:

$$\hat{p}_{\text{human}} \overset{\text{def}}{=} \min_q \text{KL}(\pi_D \parallel q) = p_{\text{MLE}}$$

$$R_{\text{ideal}} = \log p_{\text{human}}$$
Estimate $p_{\text{human}}$ (for the demonstrations)

Approximate $p_{\text{human}}$ using the demonstrations:

$$\hat{p}_{\text{human}} \overset{\text{def}}{=} \min_q \text{KL} (\pi_D \parallel q) = p_{\text{MLE}}$$

(Good enough for training examples.)

$$R_{\text{ideal}} = \log p_{\text{human}}$$
Estimate $p_{\text{human}}$ (for the demonstrations)

Approximate $p_{\text{human}}$ using the demonstrations:

$$\hat{p}_{\text{human}} \overset{\text{def}}{=} \min_q \text{KL} (\pi_D \| q) = p_{\text{MLE}}$$  (Good enough for training examples.)

Reward functions:

1. **Product** of $\hat{p}_{\text{human}}$: a sequence is good if all words are good.

   $$\hat{Q}(s_t, a_t) = \sum_{t'=t}^{T} \log \hat{p}_{\text{human}}(a_t \mid s_t)$$

2. **Sum** of $\hat{p}_{\text{human}}$: a sequence is good if most words are good.

   $$\hat{Q}(s_t, a_t) = \sum_{t'=t}^{T} \hat{p}_{\text{human}}(a_t \mid s_t)$$
1. Learn $p_{\text{MLE}}$ to compute the reward.
1. Learn $p_{\text{MLE}}$ to compute the reward.

2. Update with MLE gradient for a few epochs:

$$\sum_{a_1:T, s_1:T \sim D} \sum_{t} \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t)$$
1. Learn $p_{\text{MLE}}$ to compute the reward.

2. Update with MLE gradient for a few epochs:

$$
\sum_{a_1:T, s_1:T \sim D} \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)
$$

3. Update with off-policy policy gradient until convergence:

$$
\sum_{a_1:T, s_1:T \sim D} \sum_t \pi_{\theta}(a_t | s_t) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \sum_{t' = t}^T \log p_{\text{MLE}}(a_t | s_t)
$$
Generation by Off-policy Learning from Demonstrations

1. Learn $p_{\text{MLE}}$ to compute the reward.
2. Update with MLE gradient for a few epochs:

$$
\sum_{a_1:T, s_1:T \sim D} \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t)
$$

3. Update with off-policy policy gradient until convergence:

$$
\sum_{a_1:T, s_1:T \sim D} \sum_t \pi_{\theta}(a_t \mid s_t) \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t) \sum_{t' = t}^{T} \log p_{\text{MLE}}(a_t \mid s_t)
$$

▶ No interaction: all updates are on training examples.
1. Learn $p_{\text{MLE}}$ to compute the reward.

2. Update with MLE gradient for a few epochs:

$$\sum_{a_{1:T},s_{1:T}\sim D} \sum_{t} \nabla_{\theta} \log \pi_{\theta}(a_{t} \mid s_{t})$$

3. Update with off-policy policy gradient until convergence:

$$\sum_{a_{1:T},s_{1:T}\sim D} \sum_{t} \pi_{\theta}(a_{t} \mid s_{t}) \nabla_{\theta} \log \pi_{\theta}(a_{t} \mid s_{t}) \sum_{t'=t}^{T} \log p_{\text{MLE}}(a_{t} \mid s_{t})$$

- No interaction: all updates are on \textit{training examples}.
- Up-weight examples \textit{preferred} by the model.
Generation by Off-policy Learning from Demonstrations

1. Learn $p_{\text{MLE}}$ to compute the reward.
2. Update with MLE gradient for a few epochs:
   $$\sum_{a_1:T, s_1:T \sim D} \sum_t \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t)$$
3. Update with off-policy policy gradient until convergence:
   $$\sum_{a_1:T, s_1:T \sim D} \sum_t \pi_{\theta}(a_t \mid s_t) \nabla_{\theta} \log \pi_{\theta}(a_t \mid s_t) \sum_{t' = t}^{T} \log p_{\text{MLE}}(a_t \mid s_t)$$

- No interaction: all updates are on training examples.
- Up-weight examples preferred by the model.
- Up-weight examples with high probability under $p_{\text{MLE}}$. 
Datasets:

- Question generation (NQG) [Zhou+ 2017]
  
  **Input:** Some members of this community emigrated to the United States in the 1980s.
  
  **Output:** In what era did some members of this community emigrate to the US?
Experiment setup

Datasets:

- Question generation (NQG) [Zhou+ 2017]
  
  Input: Some members of this community emigrated to the United States in the 1980s.
  
  Output: In what era did some members of this community emigrate to the US?

- Summarization (CNN/DM, XSum) [Hermann+ 2015, Narayan+ 2018]
Experiment setup

Datasets:

- **Question generation (NQG) [Zhou+ 2017]**
  
  Input: Some members of this community emigrated to the United States in the 1980s.
  
  Output: In what era did some members of this community emigrate to the US?

- **Summarization (CNN/DM, XSum) [Hermann+ 2015, Narayan+ 2018]**

- **Machine translation (IWSLT14 De-En) [Cettolo+ 2014]**
Experiment setup

Datasets:

- **Question generation (NQG) [Zhou+ 2017]**
  
  **Input:** Some members of this community emigrated to the United States in the **1980s**.
  
  **Output:** In what era did some members of this community emigrate to the US?

- **Summarization (CNN/DM, XSum) [Hermann+ 2015, Narayan+ 2018]**

- **Machine translation (IWSLT14 De-En) [Cettolo+ 2014]**

Variations of GOLD:

- **GOLD-$p$: product of $\hat{p}_{human}$**
- **GOLD-$s$: sum of $\hat{p}_{human}$**
Characteristics of GOLD

- GOLD improves generation quality
- GOLD improves precision at the cost of recall
- GOLD alleviates exposure bias
Characteristics of GOLD

- GOLD improves generation quality
- GOLD improves precision at the cost of recall
- GOLD alleviates exposure bias
GOLD on standard vs advanced models

GOLD improve both standard and Transformer-based models.
Human evaluation

Human comparison on 200 pairs of outputs:

- Question generation

Which question is better given the paragraph and the intended answer?
Human evaluation

Human comparison on 200 pairs of outputs:

▶ Question generation
  
  *Which question is better given the paragraph and the intended answer?*

▶ Summarization

  *Which summary is closer to the reference in meaning?*
Human evaluation

Human comparison on 200 pairs of outputs:

- **Question generation**
  
  *Which question is better given the paragraph and the intended answer?*

- **Summarization**

  *Which summary is closer to the reference in meaning?*

GOLD vs MLE using BART

<table>
<thead>
<tr>
<th></th>
<th>win</th>
<th>lose</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQG</td>
<td>38</td>
<td>28.5</td>
</tr>
<tr>
<td>CNN/DM</td>
<td>37.5</td>
<td>24.5</td>
</tr>
<tr>
<td>XSum</td>
<td>35</td>
<td>21.5</td>
</tr>
</tbody>
</table>
Characteristics of GOLD

- GOLD improves generation quality
  - Better quality in terms of automatic metric and human judgment
- GOLD improves precision at the cost of recall
- GOLD alleviates exposure bias
Characteristics of GOLD

✓ GOLD improves generation quality
  ➤ Better quality in terms of automatic metric and human judgment

□ GOLD improves precision at the cost of recall

□ GOLD alleviates exposure bias
Held-out perplexity

- BART
- GOLD

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MLE</th>
<th>GOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQG</td>
<td>5.96</td>
<td>7.67</td>
</tr>
<tr>
<td>CNN/DM</td>
<td>5.41</td>
<td>6.96</td>
</tr>
<tr>
<td>XSum</td>
<td>5.07</td>
<td>6.85</td>
</tr>
<tr>
<td>IWSLT</td>
<td>5.31</td>
<td>6.9</td>
</tr>
</tbody>
</table>

High perplexity $\neq$ low quality

GOLD improves quality at the cost of diversity (recall)

Using better models alleviate the quality-diversity tradeoff

(NQG++ net ppl: GOLD/158 vs MLE/29)
Held-out perplexity

- NQG: 7.67
- CNN/DM: 6.96
- XSum: 6.85
- IWSLT: 6.9

**BART**

Perplexity

- NQG: 5.96
- CNN/DM: 5.41
- XSum: 5.07
- IWSLT: 5.31

**MLE vs GOLD**

- GOLD improves quality at the cost of diversity (recall)
- Using better models alleviates the quality-diversity tradeoff

(NQG++ net ppl: GOLD/158 vs MLE/29)
Held-out perplexity

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MLE Perplexity</th>
<th>GOLD Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>NQG</td>
<td>5.96</td>
<td>7.67</td>
</tr>
<tr>
<td>CNN/DM</td>
<td>5.41</td>
<td>6.96</td>
</tr>
<tr>
<td>XSum</td>
<td>5.07</td>
<td>6.85</td>
</tr>
<tr>
<td>IWSLT</td>
<td>5.31</td>
<td>6.90</td>
</tr>
</tbody>
</table>

- High perplexity ≠ low quality
Held-out perplexity

- High perplexity $\neq$ low quality
- GOLD improves quality at the cost of diversity (recall)
Held-out perplexity

High perplexity $\neq$ low quality

GOLD improves quality at the cost of diversity (recall)

Using better models alleviate the quality-diversity tradeoff

(NQG++ net ppl: GOLD/158 vs MLE/29)
High perplexity but good BLEU/ROUGE score?

NQG dev set

GOLD is skewed towards near-zero losses.
GOLD has a longer tail of high loss tokens.

Perplexity is sensitive to (a few) low probability tokens.
GOLD improves quality (precision) at the cost of diversity (recall).
High perplexity but good BLEU/ROUGE score?

**GOLD** is skewed towards near-zero losses

NQG dev set

GOLD is skewed towards near-zero losses
High perplexity but good BLEU/ROUGE score?

- **GOLD** is skewed towards near-zero losses
- **GOLD** has a longer tail of high loss tokens
- **NQG** dev set
- Perplexity is sensitive to (a few) low probability tokens
High perplexity but good BLEU/ROUGE score?

- Perplexity is sensitive to (a few) low probability tokens
- **GOLD** improves quality (precision) at the cost of diversity (recall)
Low sensitivity to decoding algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>BLEU 10</th>
<th>BLEU 12</th>
<th>BLEU 14</th>
<th>BLEU 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLE</td>
<td>14.13</td>
<td>14.19</td>
<td>14.07</td>
<td>11.27</td>
</tr>
<tr>
<td>GOLD</td>
<td>10.08</td>
<td>10.08</td>
<td>10.08</td>
<td>10.08</td>
</tr>
</tbody>
</table>

NQG (NQG net++)

High-precision models are less sensitive to decoding algorithms. Greedy decoding works just fine.
Low sensitivity to decoding algorithms

- Greedy decoding works well.
- Beam search algorithms (beam-3, beam-5) also perform well.
- Top-k (top-5, top-20) are less effective.

![BLEU scores for NQG (NQG net++)](chart)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MLE</th>
<th>GOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy</td>
<td>14.13</td>
<td>16.06</td>
</tr>
<tr>
<td>Beam-3</td>
<td>14.19</td>
<td>15.84</td>
</tr>
<tr>
<td>Beam-5</td>
<td>14.07</td>
<td>15.74</td>
</tr>
<tr>
<td>Top-5</td>
<td>11.27</td>
<td>15.41</td>
</tr>
<tr>
<td>Top-20</td>
<td>10.08</td>
<td>15.38</td>
</tr>
</tbody>
</table>

The chart illustrates the BLEU scores for different decoding algorithms on the NQG (NQG net++) model. The greedy algorithm consistently outperforms the beam search and top-k algorithms.
Low sensitivity to decoding algorithms

**NQG (NQG net++)**

- **BLEU**
  - Greedy: 14.13, 16.06
  - Beam-3: 14.19, 15.84
  - Beam-5: 14.07, 15.74
  - Top-5: 11.27, 10.08
  - Top-20: 15.41, 15.38

**CNN/DM (Pnt-Gen)**

- **R-2**
  - Greedy: 17.4, 18.51
  - Beam-3: 17.65, 18.44
  - Beam-5: 17.63, 18.25
  - Top-5: 17.02, 16.57
  - Top-20: 13.06, 11.23

High-precision models are less sensitive to decoding algorithms. Greedy decoding works just fine.
Low sensitivity to decoding algorithms

- High-precision models are less sensitive to decoding algorithms
- Greedy decoding works just fine
Characteristics of GOLD

- ✓ GOLD improves generation quality
  - Better quality in terms of automatic metric and human judgment

- ✓ GOLD improves precision at the cost of recall
  - On reference: more low-ppl tokens with a long tail of high-ppl tokens
  - Generation: less sensitive to decoding algorithms

- □ GOLD alleviates exposure bias
Characteristics of GOLD

- GOLD improves generation quality
  - Better quality in terms of automatic metric and human judgment

- GOLD improves precision at the cost of recall
  - On reference: more low-ppl tokens with a long tail of high-ppl tokens
  - Generation: less sensitive to decoding algorithms

- GOLD alleviates exposure bias

- GOLD alleviates exposure bias
Exposure bias

Mismatched training and inference prefix:

We discovered a huge cave

found it

Training $p(t\text{-th word} \mid \text{gold prefix, context})$
Inference $p(t\text{-th word} \mid \text{generated prefix, context})$
Exposure bias

Theoretical worst case:

\[ O(#\text{steps}^2) \] mistakes \cite{Ross+2011}

Once off the gold path, a mistake is made in all following steps.
Exposure bias problems in text generation

Empirical observations:

▶ Repetitions [Holtzman+ 2020]

**Beam Search, $b=32$:**

"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/..."
Exposure bias problems in text generation

Empirical observations:

▶ Repetitions [Holtzman+ 2020]

**Beam Search, $b=32$:** "The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ...”

▶ Hallucination [Wang+ 2020]

| source | So höre nicht auf die Ableugner. |
| reference | So hearken not to those who deny. |
| output | Do not eddrive or use machines. |
GOLD alleviates exposure bias

With exposure bias

NQG (NQG net++)

Without exposure bias

▶ Given reference pre/uniFB01x, both losses do not change with length
▶ Given generated pre/uniFB01x, MLE outputs degrade with length while GOLD outputs is stable
GOLD alleviates exposure bias

- Given reference prefix, both losses do not change with length
GOLD alleviates exposure bias

NQG (NQG net++)

With exposure bias

- Human rating
- Token-level avg NLL loss

Without exposure bias

- MLE
- GOLD-s

Given reference prefix, both losses do not change with length

Given generated prefix, MLE outputs degrade with length while GOLD outputs is stable
Characteristics of GOLD

- GOLD improves generation quality
  - Better quality in terms of automatic metric and human judgment
- GOLD improves precision at the cost of recall
- GOLD alleviates exposure bias
  - Generation quality is stable across output lengths.
When to use GOLD?

When it’s good enough to have one good answer (high precision)
▶ Machine translation
▶ Summarization
▶ Code generation

Not suitable when multiple diverse answers are desired (high recall)
▶ Creative writing assistant
▶ Story generation
Close the gap

\[ \mathbb{E}_{\pi_D} [\log \pi_\theta(x)] \quad \text{(MLE)} \]

\[ \mathbb{E}_{\pi_\theta} [\log p_{\text{human}}(x)] \]
Close the gap

\[
\mathbb{E}_{\pi_D} [\log \pi_\theta(x)] \quad \text{(MLE)}
\]

\[
\downarrow
\]

\[
\mathbb{E}_{\pi_D} [\pi_\theta(x)Q(x)] \quad \text{(GOLD)}
\]

\[
\mathbb{E}_{\pi_\theta} [\log p_{\text{human}}(x)]
\]
Close the gap

\[
\mathbb{E}_{\pi_D}[\log \pi_\theta(x)] \quad \text{(MLE)}
\]

\[
\downarrow
\]

\[
\mathbb{E}_{\pi_D}[\pi_\theta(x)Q(x)] \quad \text{(GOLD)}
\]

\[
\mathbb{E}_{\pi_\theta}[\log p_{\text{human}}(x)]
\]
Close the gap

\[ \mathbb{E}_{\pi_D} [\log \pi_\theta(x)] \quad \text{(MLE)} \]

\[ \downarrow \]

\[ \mathbb{E}_{\pi_D} [\pi_\theta(x) Q(x)] \quad \text{(GOLD)} \]

\[ \mathbb{E}_{\pi_\theta} [\log p_{\text{human}}(x)] \]
Close the gap

$\mathbb{E}_{\pi_D}[\log \pi_\theta(x)]$ (MLE)

$\mathbb{E}_{\pi_D}[\pi_\theta(x)Q(x)]$ (GOLD)

$\mathbb{E}_{\pi_\theta}[\log p_{\text{human}}(x)]$

- Interact with the environment
- Robust reward functions
Close the gap

\[ \mathbb{E}_{\pi_D} \left[ \log \pi_\theta(x) \right] \quad \text{(MLE)} \]

\[ \downarrow \]

\[ \mathbb{E}_{\pi_D} \left[ \pi_\theta(x) Q(x) \right] \quad \text{(GOLD)} \]

\[ \downarrow \]

\[ \mathbb{E}_{\pi_\theta} \left[ \log p_{\text{human}}(x) \right] \]

- Interact with the environment (RL algorithms)
- Robust reward functions (key challenge)
Averaging over model distribution: additional interaction

- Additional on-policy training yields *marginal* improvement
- Reward function may not be useful on model outputs
Better reward function: human in the loop

Failed attempt:
- Learn a reward function from human-annotated translations
- Use the reward function in online/offline RL
- Only helpful with small data

Pitfall with learned reward function:
- Model can exploit shortcuts in the learned reward model, e.g., length, specific phrases
Learning from human preferences using PPO:

- Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. Anthropic.
- Aligning Language Models to Follow Instructions. OpenAI.

What made it work?

- Periodically update the preference function
- Quality control (reward signal from human can be sparse and noisy)
Parting remarks

- RL is a great framework for aligning task objective and learning objective.
- Offline RL helps with scaling (reducing to supervised learning).
- For text generation, the key is to find the right reward function.
  - How to best represent human preference which can be ambiguous?