Text Generation by Offline Reinforcement Learning

He He

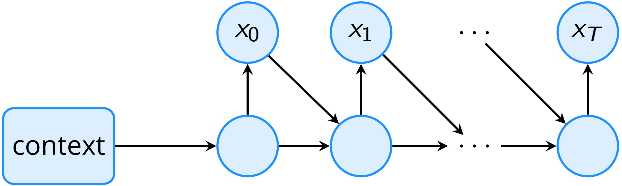
NEW YORK UNIVERSITY

Tsinghua University, IIIS, RL Reading Group

May 31, 2022

The status quo for text generation

- **Modeling**: Auto-regressive models



$$p(\text{output} \mid \text{context}) = \prod_t p(t\text{-th word} \mid \text{prefix}, \text{context})$$

The status quo for text generation

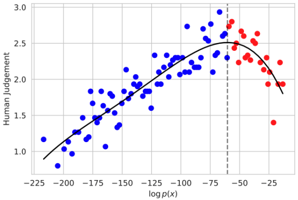
- **Learning**: Maximum likelihood estimation

$$\max_{\theta} \sum_{\text{reference}} \log p_{\theta}(\text{reference} \mid \text{context})$$

- **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



[Zhang+ 2020]

A: How about watching a movie?
B: I don't know.
A: Let's go home then.
B: I don't know.

[Li+ 2016]

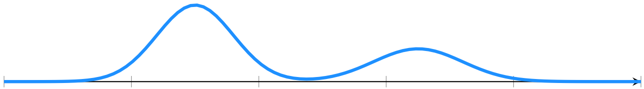
beam-1: British woman won Olympic gold in pair rowing.

beam-1000: `</s>`

[Murray+ 2018, Ott+ 2018]

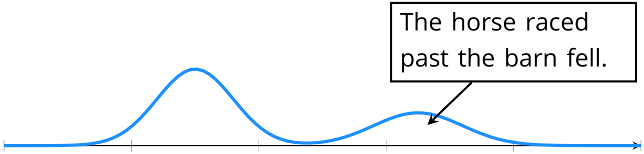
What does the model error look like?

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



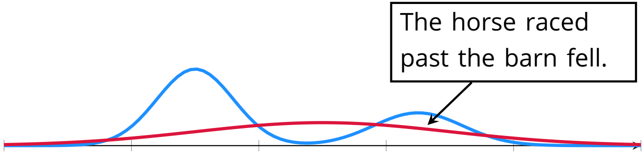
What does the model error look like?

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



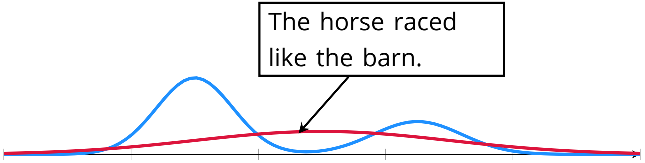
What does the model error look like?

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



What does the model error look like?

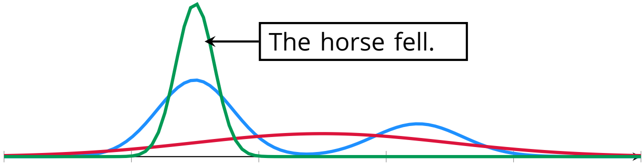
MLE tends to **over-generalize** [Huszár 2015]



MLE is " high recall",

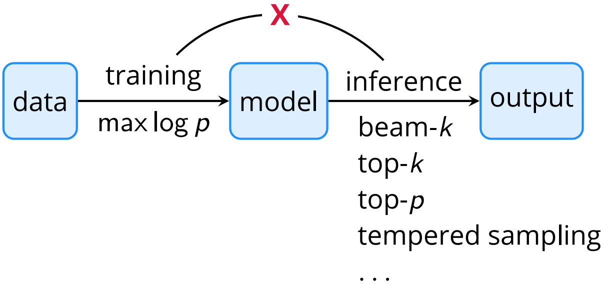
What does the model error look like?

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



MLE is " high recall", but a "high precision" solution may be preferred.

Misaligned training and evaluation objectives



 log-likelihood of the reference text

 quality of the output text (judged by humans)

Contents

Training vs evaluation losses

Training loss (NLL):

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

Evaluation loss (perceptual quality):

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

Training vs evaluation losses

Training loss (NLL):

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

Evaluation loss (perceptual quality):

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

Training vs evaluation losses

Training loss (NLL):

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

- High recall: p_θ must cover all outputs from p_{human}

Evaluation loss (perceptual quality):

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

- High precision: all output from p_θ must be scored high under p_{human}

The reinforcement learning formulation

Evaluation loss (perceptual quality):

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

The reinforcement learning formulation

Evaluation loss (perceptual quality):

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

policy reward action state

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 ...
i to me to me to me to me ...
```

degenerative solution

Optimization challenges

Obstacles:

- Gradient estimated by samples from π_θ has high variance.
- Degenerate once the reward is close to zero.

Optimization challenges

Obstacles:

- Gradient estimated by samples from π_θ 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:

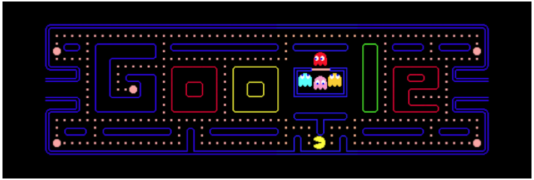
- Gradient estimated by samples from π_θ has high variance.
- Degenerate once the reward is close to zero.

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

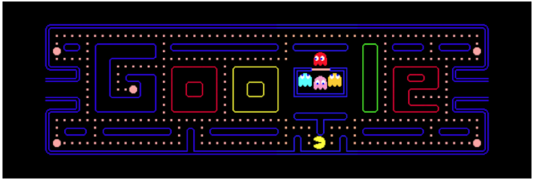
(Marginal improvement in practice [Wu+ 2018, Choshen+ 2020])

Problem: policy/generator *interacting* with the environment.

Is interaction useful?

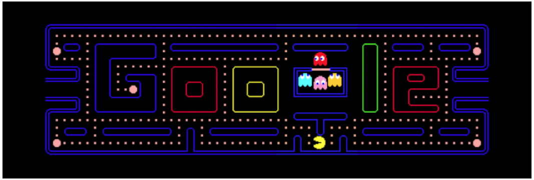


Is interaction useful?



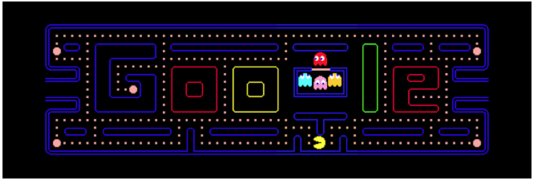
- Learn about the environment dynamics.

Is interaction useful?



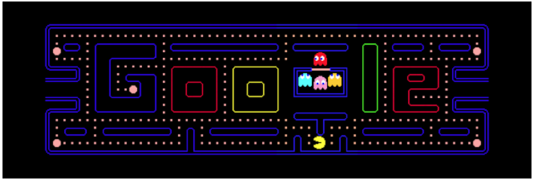
- Learn about the environment dynamics.
  - We already know the dynamics.

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?



<s> $\xrightarrow{\texttt{hello}}$ <s> hello $\xrightarrow{\texttt{world}}$ <s> hello world

- 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:

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

Existing approaches:

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

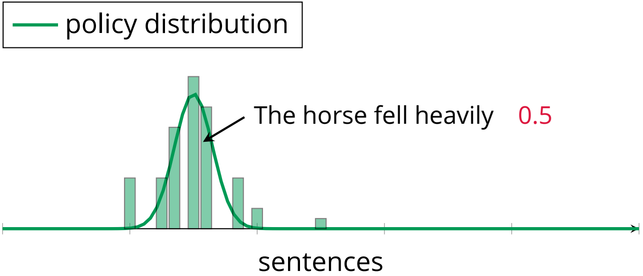
Contents

Online policy gradient

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

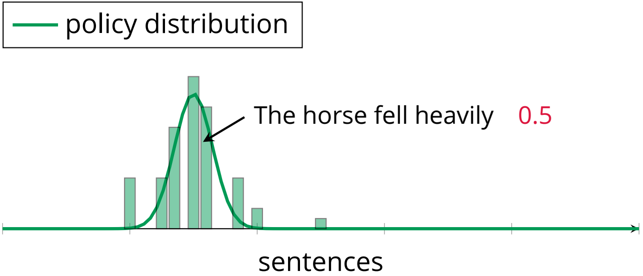
Online policy gradient

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



Online policy gradient

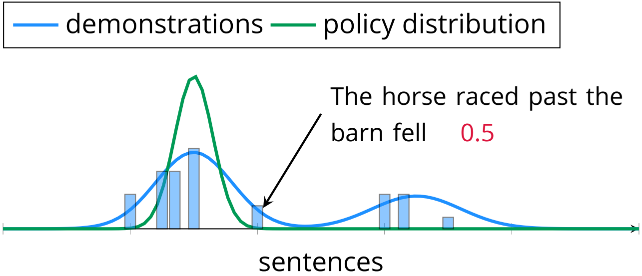
Objective: $\mathbb{E}_{\pi_\theta}\left[R(s, a)\right]$



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

Offline policy gradient

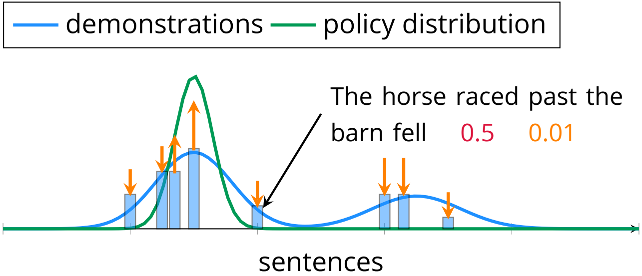
Objective: $\mathbb{E}_{\pi_\theta}\left[R(s,a)\right]$



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

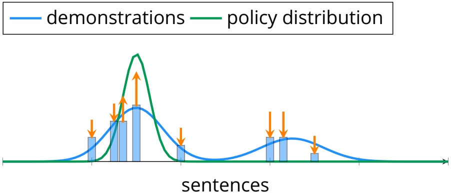
Offline policy gradient

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



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

Approximated importance weights



$$\boxed{w_t = \pi_\theta(a_t \mid s_t)}$$

- **Intuition**: up-weight actions preferred by the current policy
- Closer to model distribution

What is a good reward function

Offline policy gradient:

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

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

- 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:

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

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

- 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

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

Offline policy gradient:

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

- 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

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

Offline policy gradient:

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

- 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:

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

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

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

	naive	ideal ($R = \log p_{\text{human}}$)
The horse fell	1	0.5
The horse was in the barn	1	0.2
The horse raced past the barn fell	1	0.1

Estimate p_{human} (for the demonstrations)

$$\boxed{R_{\text{ideal}} = \log p_{\text{human}}}$$

Estimate p_{human}(for the demonstrations)

$$\boxed{R_{\text{ideal}} = \log p_{\text{human}}}$$

Approximate p_{human} using the demonstrations:

$$\hat{p}_{\text{human}} \stackrel{\text{def}}{=} \min_{q} \mathrm{KL}\left(\pi_D \| q\right) = p_{\text{MLE}}$$

Estimate p_{human} (for the demonstrations)

$$\boxed{R_{\text{ideal}} = \log p_{\text{human}}}$$

Approximate p_{human} using the demonstrations:

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

Estimate p_{human} (for the demonstrations)

$$\boxed{R_{\text{ideal}} = \log p_{\text{human}}}$$

Approximate p_{human} using the demonstrations:

$$\hat{p}_{\text{human}} \stackrel{\text{def}}{=} \min_{q} \mathrm{KL}\left(\pi_D \| q\right) = p_{\text{MLE}}$$ (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.

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

Generation by Off-policy Learning from Demonstrations

1. Learn p_{MLE} to compute the reward.

Generation by Off-policy Learning from Demonstrations

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

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

Generation by Off-policy Learning from Demonstrations

1. Learn p_{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:

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

Generation by Off-policy Learning from Demonstrations

1. Learn p_{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*.

Generation by Off-policy Learning from Demonstrations

1. Learn p_{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.

Generation by Off-policy Learning from Demonstrations

1. Learn p_{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_{MLE}.

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 ?

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 $\hat{p}_{\text{human}}$
- GOLD-s: sum of $\hat{p}_{\text{human}}$

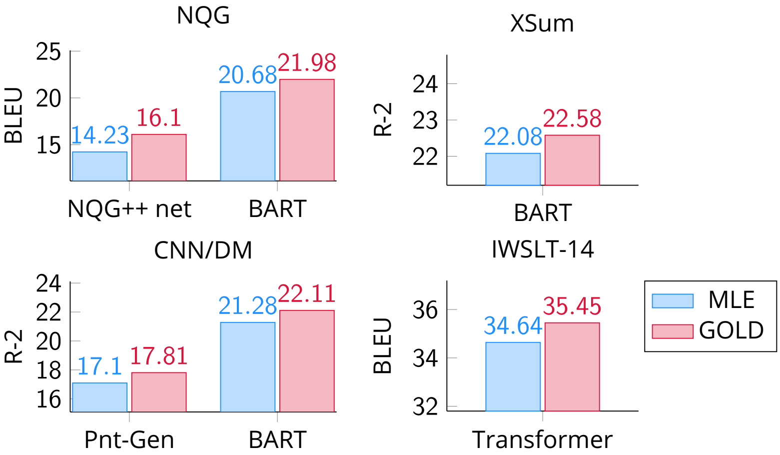
Characteristics of GOLD

- 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

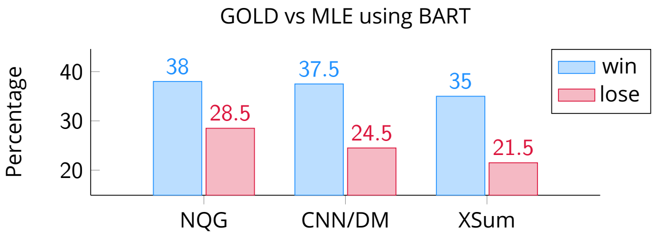
Human comparison on 200 pairs of outputs:

- Question generation

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

- Summarization

  Which summary is closer to the reference in meaning?



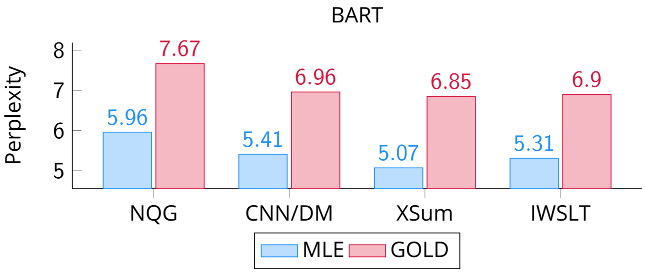
Characteristics of GOLD

- ☑ 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

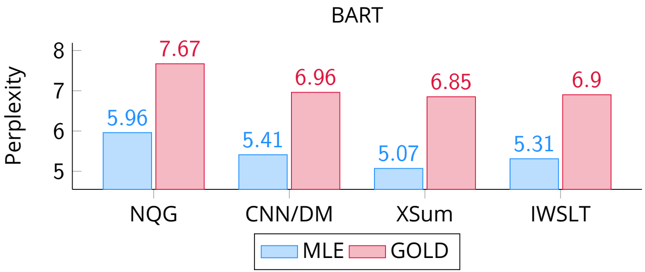
Characteristics of GOLD

- ☑ 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

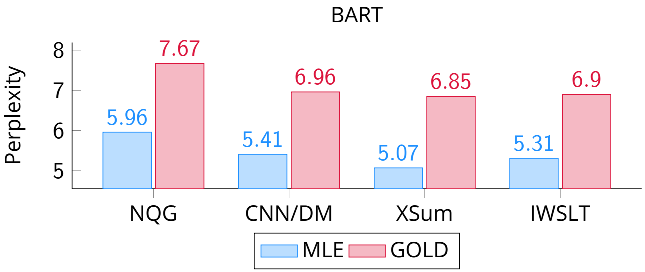
Held-out perplexity



Held-out perplexity

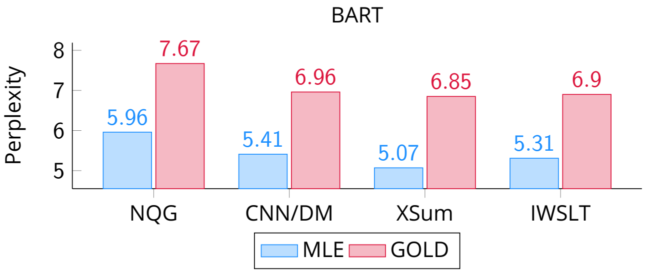


Held-out perplexity



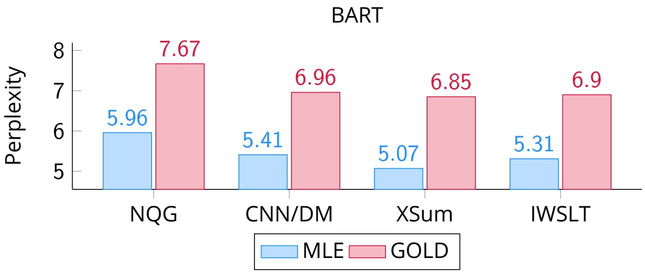
- High perplexity $\neq$ low quality

Held-out perplexity



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

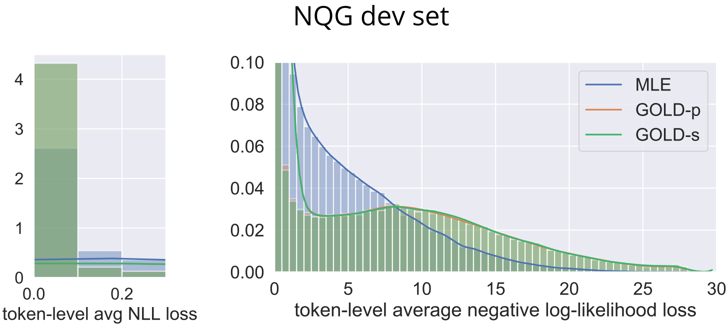
Held-out perplexity



- 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)

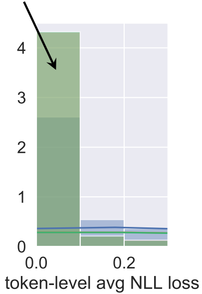
High perplexity but good BLEU/ROUGE score?

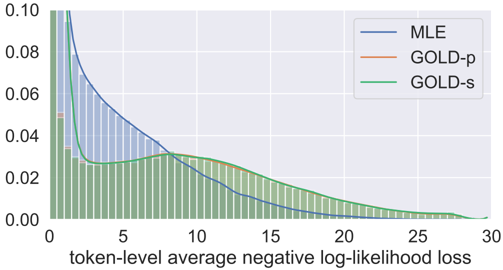


High perplexity but good BLEU/ROUGE score?

GOLD is skewed towards near-zero losses

NQG dev set



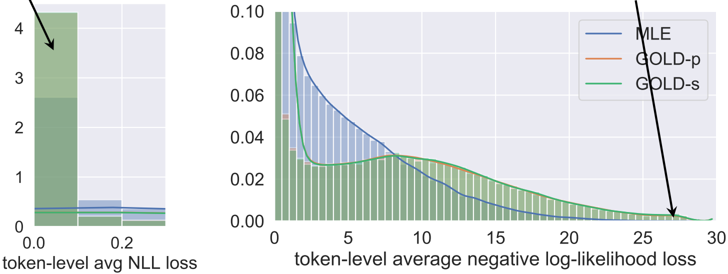


High perplexity but good BLEU/ROUGE score?

GOLD is skewed towards near-zero losses

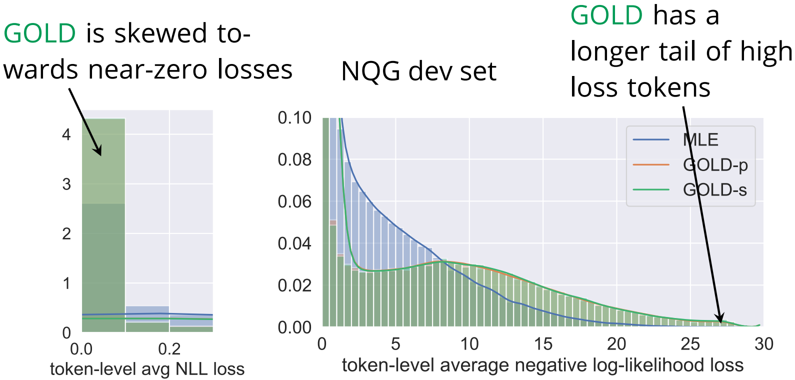
NQG dev set

GOLD has a longer tail of high loss tokens



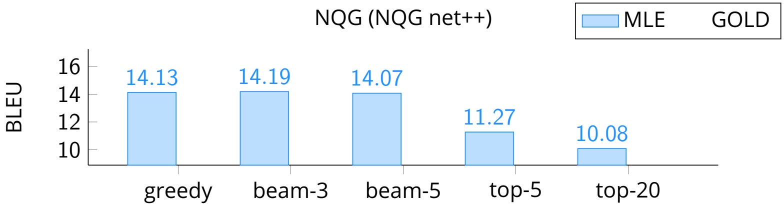
- Perplexity is sensitive to (a few) low probability tokens

High perplexity but good BLEU/ROUGE score?

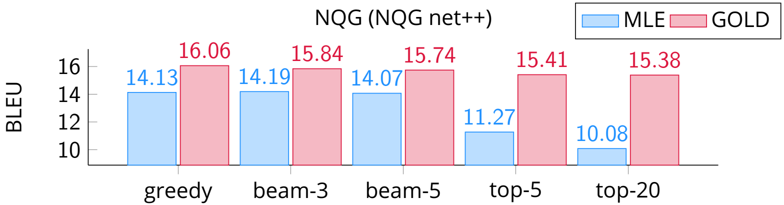


- Perplexity is sensitive to (a few) low probability tokens
- GOLD improves quality (precision) at the cost of diversity (recall)

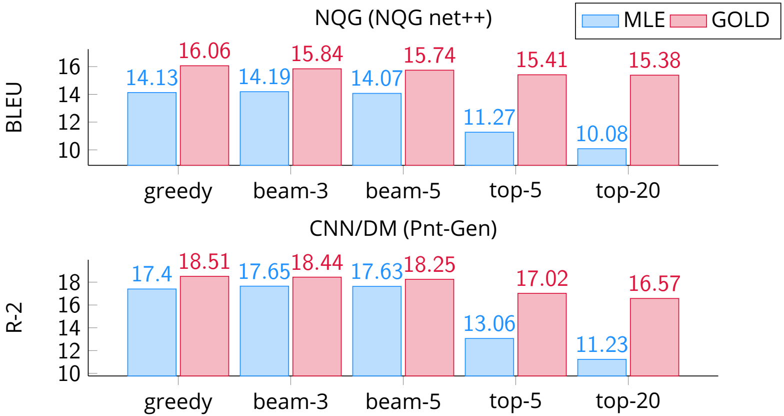
Low sensitivity to decoding algorithms



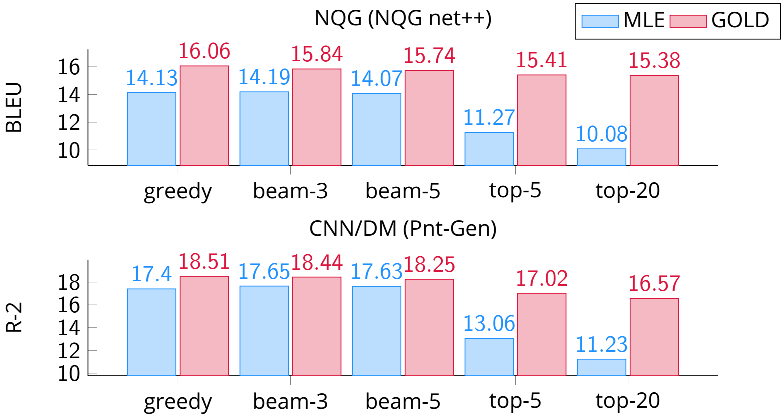
Low sensitivity to decoding algorithms



Low sensitivity to decoding algorithms



Low sensitivity to decoding algorithms



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

Characteristics of GOLD

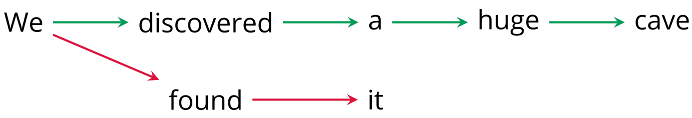
- ☑ GOLD improves generation quality
  - Better quality in terms of automatic metric and human judgment
- ☑ GOLD improves precision at the cost of recall
  - On reference: more low-ppl tokens with a long tail of high-ppl tokens
  - Generation: less sensitive to decoding algorithms
- ☐ GOLD alleviates 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**

Exposure bias

Mismatched training and inference prefix:

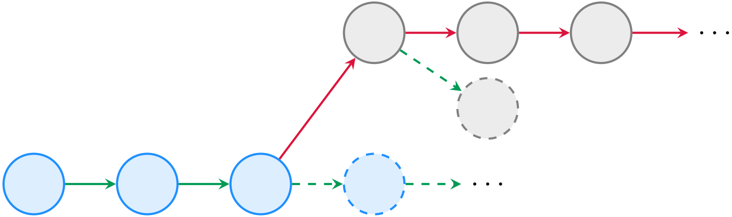


Training	$p(t\text{-th word} \mid \text{gold prefix, context})$
Inference	$p(t\text{-th word} \mid \text{generated prefix, context})$

Exposure bias

Theoretical worst case:

$O(\#\text{steps}^2)$ mistakes [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 (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Exposure bias problems in text generation

Empirical observations:

- Repetitions [Holtzman+ 2020]

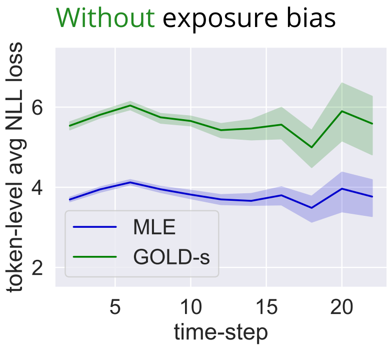
  Beam Search, *b*=32:
  "The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma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

- Hallucination [Wang+ 2020]

source	So hoře nicht auf die Ableugner.
reference	So hearken not to those who deny.
output	Do not eddrive or use machines.

GOLD alleviates exposure bias

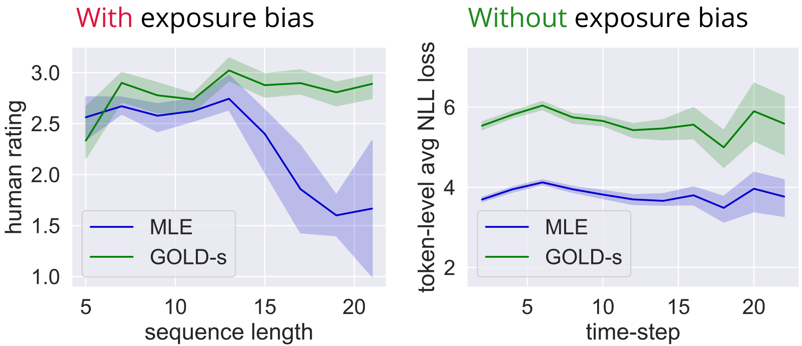
GOLD alleviates exposure bias



- Given reference prefix, both losses do not change with length

GOLD alleviates exposure bias

NQG (NQG net++)



- 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.

Contents

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

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

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

Close the gap

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

$$\downarrow$$

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

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

Close the gap

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

$$\downarrow$$

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

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

Close the gap

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

$$\downarrow$$

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

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

Close the gap

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

$$\downarrow$$

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

$$\downarrow$$

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

- Interact with the environment
- Robust reward functions

Close the gap

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

$$\downarrow$$

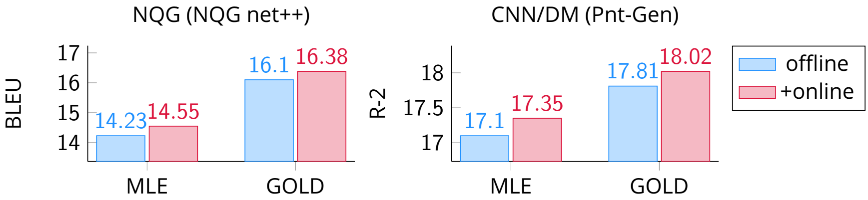
$$\mathbb{E}_{\pi_D}\left[\pi_\theta(x) Q(x)\right] \qquad \text{(GOLD)}$$

$$\downarrow$$

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

- 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

RL for alignment

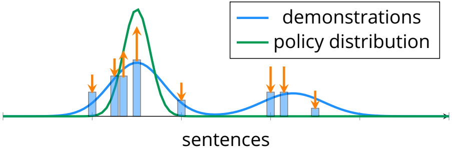
Learning from human preferences using PPO:

- 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?