Text Generation by Learning from Demonstrations

Richard Yuanzhe Pang NEW YORK UNIVERSITY
yzpang.me
He He NEW YORK UNIVERSITY
hhexiy.github.io

ICLR

1 Motivation and Takeaways

The most widespread approach for supervised conditional text generation: MLE + teacher forcing

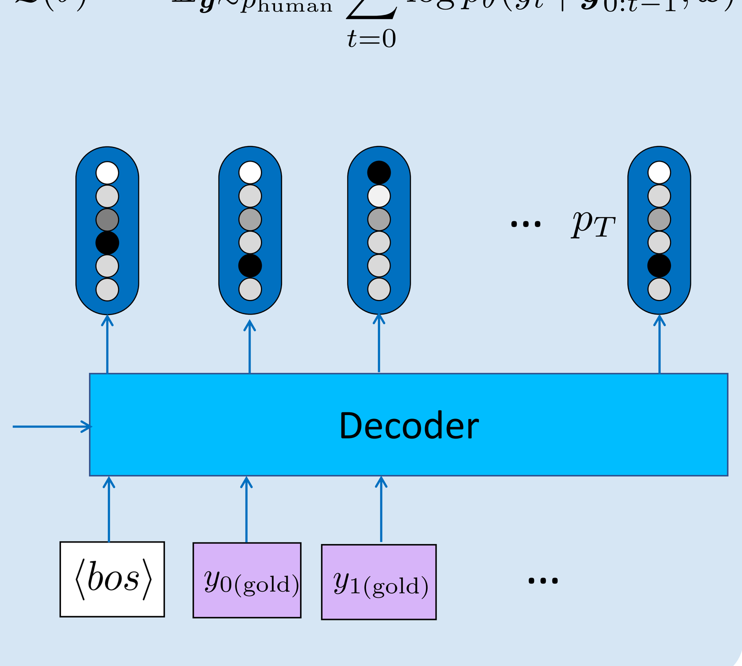
Motivations

1. Train-test mismatched history (gold vs. model-generated)
   ⇒ repetitions and hallucinations; "exposure bias"
2. Train-test mismatched objectives (high recall vs. high precision)
   High recall: encourages high probability on ***every*** reference
   High precision: model generations should be rated highly by humans

Training (usually, teacher forcing)

Loss

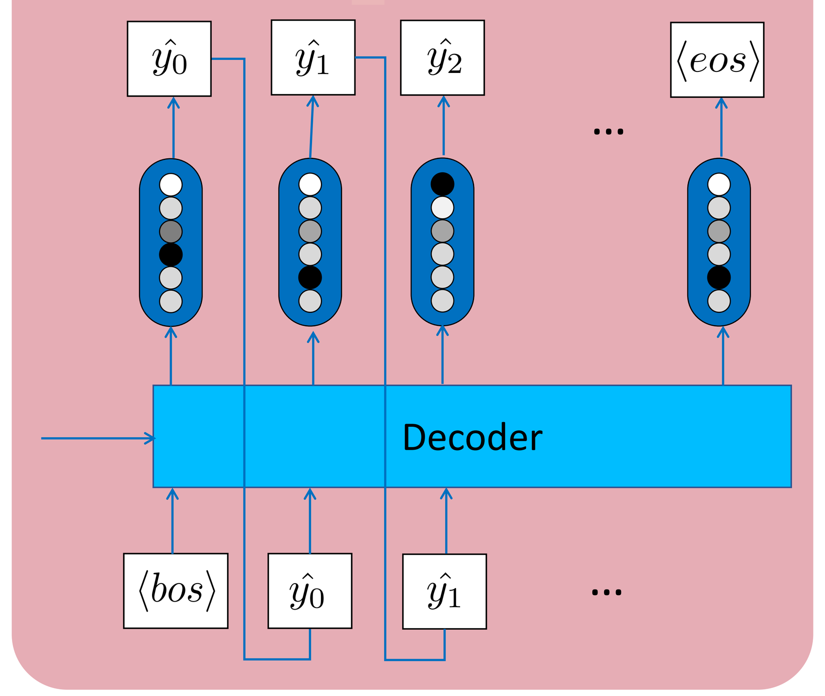
$$\mathcal{L}(\theta) = -\mathbb{E}_{\boldsymbol{y}\sim p_{\text{human}}} \sum_{t=0}^{T} \log p_\theta(y_t \mid \boldsymbol{y}_{0:t-1}, \boldsymbol{x})$$



Usually, autoregressive **inference**

Eval metric

$$\mathbb{E}_{\boldsymbol{y}\sim p_\theta} \sum_{t=0}^{T} \log p_{\text{human}}(y_t \mid \boldsymbol{y}_{0:t-1}, \boldsymbol{x})$$



TAKEAWAYS!

1. GOLD is an offline + off-policy algorithm; there's no interaction with the environment
2. GOLD's intuition: weighted MLE; upweights "confident" tokens and downweights "unconfident" ones
3. GOLD encourages high-precision generation (instead of distribution matching) for generation tasks where "one good output is sufficient"

2 Background: RL formulation for text generation

The above eval objective

$$\mathbb{E}_{\boldsymbol{y}\sim p_\theta} \sum_{t=0}^{T} \log p_{\text{human}}(y_t \mid \boldsymbol{y}_{0:t-1}, \boldsymbol{x})$$

policy reward action state

RL formulation

$$\max_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta} \sum_{t=0}^{T} R(a_t, s_t)$$

Prior approach Directly optimize a sequence-level metric like BLEU, ROUGE, etc. using policy gradient (e.g., REINFORCE)

- Pros: no exposure bias, may discover high-quality outputs outside refs
- Cons: degenerate solutions; difficult optimization

3 Offline objective: GOLD (generation by offline+off-policy learning from demonstrations)

(Traditionally:) online + on-policy policy gradient

Step 1: sample outputs from the model
Step 2: get seq-level rewards like BLEU
Step 3: use policy gradient to optimize

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim p_\theta} \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{Q}(s_t, a_t)$$

Offline + off-policy policy gradient (**NO INTERACTION** w/ environment)

Step 1: sample from demonstrations (i.e., gold supervised data)
Step 2: get token-level rewards based on p_{MLE} (discussed below)
Step 3: use policy gradient with importance weights to optimize

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_b} \sum_t w_t \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{Q}(s_t, a_t) \quad (*)$$

$\pi_b = p_{\text{human}}$ — use empirical distn

$w_t \approx \pi_\theta(a_{t'} \mid s_{t'})$ — model "confidence"

$\hat{Q}(s_t, a_t) = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ — p_{MLE} based reward (see below)

Intuition: upweights more "confident" tokens

Reward function

(1) Use *dirac-delta* function: Q is 1 for all training data, 0 for other data **GOLD-*delta***

(2) Use estimated p_{human}: find p that **min** $\text{KL}(\pi_b \,\|\, p)$

The p is p_{MLE}! Good *for demonstrations*, but not in general.

(2.1) product of estimated p_{human} (a sequence is good if all words are good) **GOLD-*p***

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

(2.2) sum of estimated p_{human} (a sequence is good if most words are good) **GOLD-*s***

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

Full algorithm: GOLD

Algorithm 1: GOLD

```
1 π_θ ← p_MLE, π̃_θ ← p_MLE
2 for step = 1, 2, ..., M do
3     Sample a minibatch B = {(x^i, y^i)}_{i=1}^{|B|}
4     foreach (s_t^i, a_t^i) do
5         Compute importance weights
          max(u, π̃_θ), and compute returns
          Q̂(s_t^i, a_t^i) − b
6     Update θ by (*) using gradient descent
7     if step % k = 0 then π̃_θ ← π_θ
8 Return: π_θ
```

Two sources of variance…

(1) from importance weights
- fix: periodic synchronization of policy
- fix: lower bound importance weights

(2) from the return Q
- fix: subtract by baseline (popular trick)
- fix: lower bound Q by lower bounding p_{MLE}

Paper + code + more info: yzpang.me

4 Experiments

Tasks Conditional text generation tasks where "one good generation is sufficient": (1) **NQG** (natural question generation); (2) **CNN/DM** (extractive summarization); (3) **XSum** (abstractive summarization); (4) **IWSLT14 De-En** (machine translation)

Discussion on "diversity" can be found in the paper

Hypothesis 1: GOLD improves generation quality

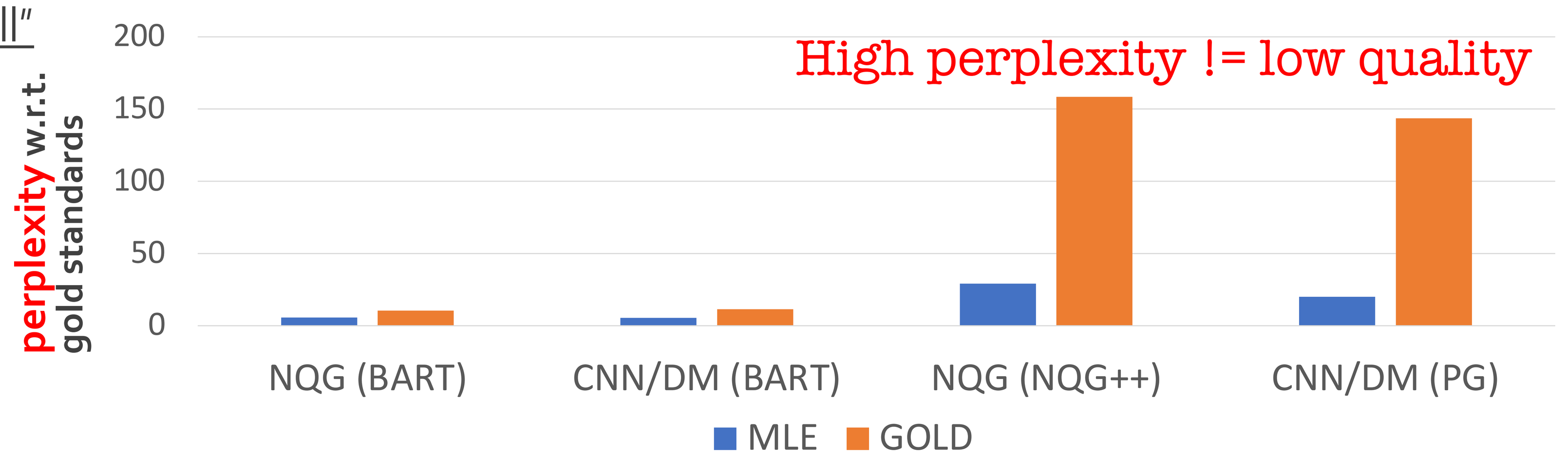
Auto evals	NQG (BART) (BLEU)	CNN/DM (BART) (ROUGE-2)	XSum (BART) (ROUGE-2)	IWSLT14 De-En (Transformer) (BLEU)
MLE	20.68	21.28	22.08	34.64
GOLD-p	21.42	22.01	22.26	35.33
GOLD-s	**21.98**	**22.09**	**22.58**	**35.45**

Human evals	NQG (BART) win/lose/tied	CNN/DM (BART) win/lose/tied	XSum (BART) win/lose/tied
GOLD-s vs. MLE	38.0/28.5/33.5	37.5/24.5/38.0	35.0/21.5/43.5

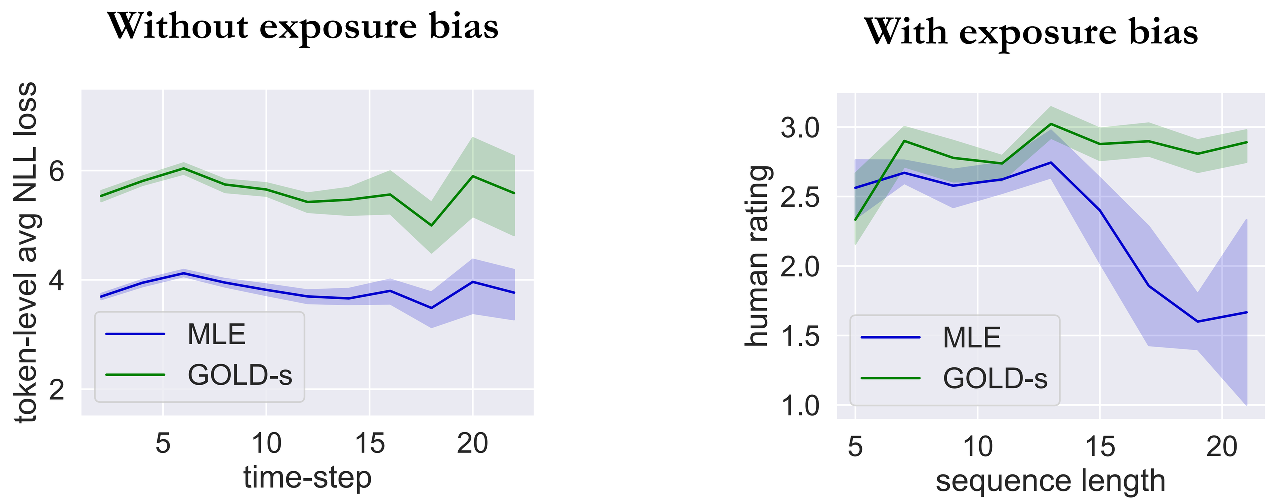
Hypothesis 2: GOLD improves precision at the cost of recall

"Precision"	NQG (BART model; BLEU metric)	CNN/DM (BART model; ROUGE-2 metric)	NQG (NQG++ model; BLEU metric)	CNN/DM (pointer generator model; R-2 metric)
MLE	20.68	21.28	14.23	17.10
GOLD-s	**21.98**	**22.09**	**16.10**	**17.81**

"Recall"



Hypothesis 3: GOLD improves precision at the cost of recall



- (Left) Given reference prefix, both losses do not change with lengths
- (Right) Given generated prefix, MLE outputs degrade with length while GOLD stays relatively stable
- More exposure bias related analysis in the paper and the appendix