Inverse reinforcement learning

- Toward Diverse Text Generation with Inverse Reinforcement Learning
- No metrics are perfect: Adversarial Reward Learning for Visual Storytelling
- Counterfactual Multi-Agent Policy Gradient
- Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments

Toward Diverse Text Generation with Inverse Reinforcement Learning

- **Reward Approximator**

Following the framework of maximum entropy IRL [Ziebart *et al.*, 2008], we assume that the texts in training set are sampled from the distribution $p_\phi(\tau)$,

Boltzmann exploration：

$$p_\phi(\tau) = \frac{1}{Z}\exp(R_\phi(\tau)),$$
$$Z = \int_\tau \exp(R_\phi(\tau))d\tau$$

$$q_\theta(x_{1:T}) = q_\theta(\tau) = \prod_{t=1}^{T-1} \pi_\theta(a_t = x_{t+1}|s_t = x_{1:t}),$$

Objective of Reward Approximator

$$\mathcal{J}_r(\phi) = \frac{1}{N}\sum_{n=1}^{N} \log p_\phi(\tau_n) = \frac{1}{N}\sum_{n=1}^{N} R_\phi(\tau_n) - \log Z, \quad (4)$$

where τ_n denotes the n_{th} sample in the training set D_{train}.

$$\nabla_\phi \mathcal{J}_r(\phi) = \frac{1}{N}\sum_n \nabla_\phi R_\phi(\tau_n) - \frac{1}{Z}\int_\tau \exp(R_\phi(\tau))\nabla_\phi R_\phi(\tau)\mathrm{d}\tau$$
$$= \mathbb{E}_{\tau\sim p_{data}} \nabla_\phi R_\phi(\tau) - \mathbb{E}_{\tau\sim p_\phi(\tau)} \nabla_\phi R_\phi(\tau). \quad (5)$$

Importance Sampling

$$\nabla_\phi \mathcal{J}_r(\phi) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\phi R_\phi(\tau_i) - \frac{1}{\sum_j w_j}\sum_{j=1}^{M} w_j \nabla_\phi R_\phi(\tau'_j),$$

$$w_j \propto \frac{\exp(R_\phi(\tau_j))}{q_\theta(\tau_j)}$$

Toward Diverse Text Generation with Inverse Reinforcement Learning

- **Text Generator**

The text generator uses a policy $\pi_\theta(a|s)$ to predict the next word one by one. The current state s_t can be modeled by LSTM neural network as shown in Figure 2. For $\tau = \{s_1, a_1, s_2, a_2 ..., s_T, a_T\}$,

$$\mathbf{s}_t = \text{LSTM}(\mathbf{s}_{t-1}, \mathbf{e}_{a_{t-1}}), \quad (7)$$

$$\pi_\theta(\mathbf{a}_t|s_t) = \text{softmax}(\mathbf{W}\mathbf{s}_t + \mathbf{b}), \quad (8)$$

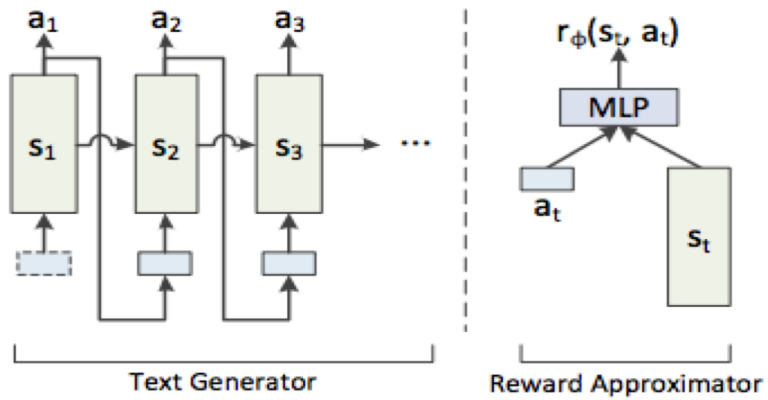


Figure 2: Illustration of text generator and reward approximator.

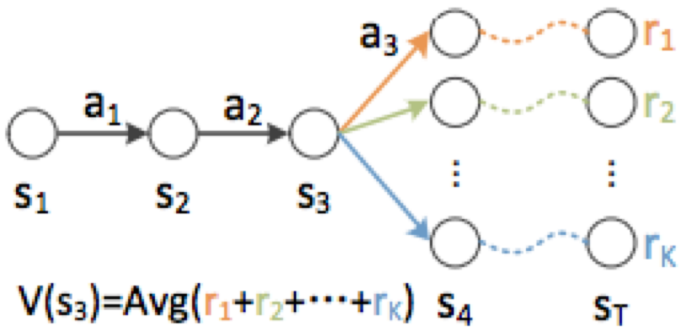
Objective of Text Generator

$$\mathcal{J}_g(\theta) = \mathbb{E}_{\tau \sim q_\theta(\tau)}[R_\phi(\tau)] + H(q_\theta(\tau))$$

$$\mathcal{J}_g(\theta) = -\text{KL}(q_\theta(\tau)||p_\phi(\tau)) + \log Z,$$

$$\nabla_\theta \mathcal{J}_g(\theta) = \sum_t \mathbb{E}_{\pi_\theta(a_t|s_t)} \nabla_\theta \log \pi_\theta(a_t|s_t) \quad [R_\phi(\tau_{t:T}) - \log \pi_\theta(a_t|s_t) - 1].$$

$$R_\phi(\tau_{t:T}) \approx r_\phi(s_t, a_t) + V(s_{t+1}),$$



Toward Diverse Text Generation with Inverse Reinforcement Learning

2.3 Why Can IRL Alleviate Mode Collapse?

GANs often suffer from mode collapse, which is partially caused by the use of Jensen-Shannon (JS) divergence. There is a reverse KL divergence $\mathrm{KL}(q_\theta(\tau)\|p_{data})$ in JS divergence. Since the p_{data} is approximated by training data, the reverse KL divergence encourages $q_\theta(\tau)$ to generate safe samples and avoid generating samples where the training data does not occur. In our method, the objective is $\mathrm{KL}(q_\theta(\tau)\|p_\phi(\tau))$. Different from GANs, we use $p_\phi(\tau)$ in IRL framework instead of p_{data}. Since $p_\phi(\tau)$ never equals to zero due to its assumption, IRL can alleviate the model collapse problem in GANs.

Toward Diverse Text Generation with Inverse Reinforcement Learning

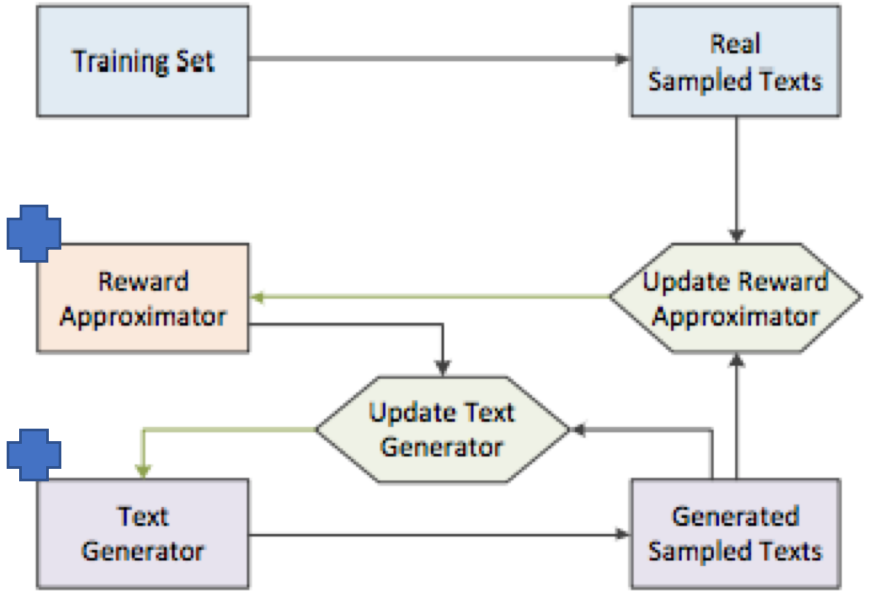


Figure 1: IRL framework for text generation.

Algorithm 1 IRL for Text Generation

repeat
  Pretrain π_θ on D_{train} with MLE
  for n_r epochs in r-step **do**
    Drawn $\tau^{(1)}, \tau^{(2)}, \cdots, \tau^{(i)}, \cdots, \tau^{(N)} \sim p_{data}$
    Drawn $\tau'^{(1)}, \tau'^{(2)}, \cdots, \tau'^{(j)}, \cdots, \tau'^{(M)} \sim q_\theta$
    Update $\phi \leftarrow \phi + \alpha \nabla_\phi \mathcal{J}_r(\phi)$
  end for
  for n_g batches in g-step **do**
    Drawn $\tau^{(1)}, \tau^{(2)}, \cdots, \tau^{(i)}, \cdots, \tau^{(N)} \sim q_\theta$
    Calculate expected reward $R_\phi(\tau_{t:T})$ by MCMC
    Update $\theta \leftarrow \theta + \beta \nabla_\theta \mathcal{J}_g(\theta)$
  end for
until Convergence

$$\nabla_\phi \mathcal{J}_r(\phi) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\phi R_\phi(\tau_i) - \frac{1}{\sum_j w_j}\sum_{j=1}^{M} w_j \nabla_\phi R_\phi(\tau'_j), \quad w_j \propto \frac{\exp(R_\phi(\tau_j))}{q_\theta(\tau_j)}$$

$$\nabla_\theta \mathcal{J}_g(\theta) = \sum_t \mathbb{E}_{\pi_\theta(a_t|s_t)} \nabla_\theta \log \pi_\theta(a_t|s_t) \quad [R_\phi(\tau_{t:T}) - \log \pi_\theta(a_t|s_t) - 1].$$

No metrics are perfect: Adversarial Reward Learning for Visual Storytelling

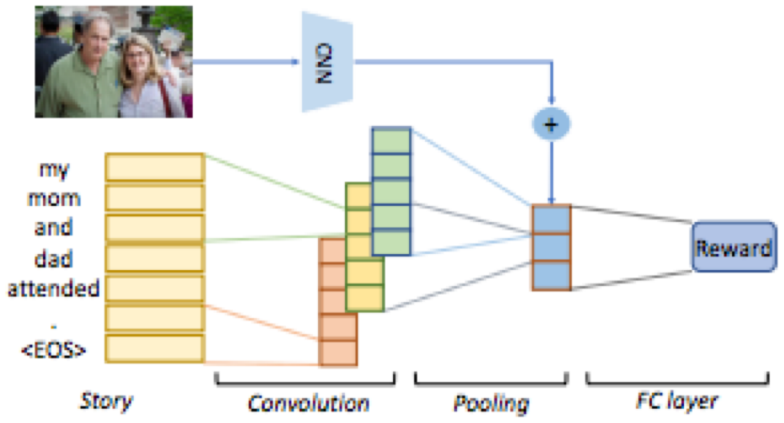


Figure 4: Overview of the reward model. Our reward model is a CNN-based architecture, which utilizes convolution kernels with size 2, 3 and 4 to extract bigram, trigram and 4-gram representations from the input sequence embeddings. Once the sentence representation is learned, it will be concatenated with the visual representation of the input image, and then be fed into the final FC layer to obtain the reward.

$$R_\theta(W) = W_r(f_{conv}(W) + W_i I_{CNN}) + b_r,$$

Reward Boltzmann Distribution

$$p_\theta(W) = \frac{\exp(R_\theta(W))}{Z_\theta},$$

In order to approximate the Reward Boltzmann distribution towards the "real" data distribution $p^*(W)$, we design a min-max two-player game, where the Reward Boltzmann distribution p_θ aims at maximizing the its similarity with empirical distribution p_e while minimizing that with the "faked" data generated from policy model π_β. On the contrary, the policy distribution π_β tries to maximize its similarity with the Boltzmann distribution p_θ. Formally, the adversarial objective function is defined as

$$\max_\beta \min_\theta KL(p_e(W)||p_\theta(W)) - KL(\pi_\beta(W)||p_\theta(W)).$$

W

tion. According to the energy-based model (LeCun et al., 2006), the optimal reward function $R^*(W)$ is achieved when the Reward-Boltzmann distribution equals to the "real" data distribution $p_\theta(W) = p^*(W)$.

No metrics are perfect: Adversarial Reward Learning for Visual Storytelling

$$\max_{\beta} \min_{\theta} KL(p_e(W)||p_\theta(W)) - KL(\pi_\beta(W)||p_\theta(W)) .$$

We further decompose it into two parts. First, because the objective J_β of the story generation policy is to minimize its similarity with the Boltzmann distribution p_θ, the optimal policy that minimizes KL-divergence is thus $\pi(W) \sim \exp(R_\theta(W))$, meaning if R_θ is optimal, the optimal $\pi_\beta = \pi^*$. In formula,

$$\begin{aligned} J_\beta &= -KL(\pi_\beta(W)||p_\theta(W)) \\ &= \underset{W\sim\pi_\beta(W)}{E}[R_\theta(W)] + H(\pi_\beta(W)) , \end{aligned} \tag{6}$$

$$\begin{aligned} J_\theta =& KL(p_e(W)||p_\theta(W)) - KL(\pi_\beta(W)||p_\theta(W)) \\ =& \sum_W [p_e(W)R_\theta(W) - \pi_\beta(W)R_\theta(W)] \\ & - H(p_e) + H(\pi_\beta) , \end{aligned}$$

$$J_\theta = \underset{W\sim p_e(W)}{E}[R_\theta(W)] - \underset{W\sim\pi_\beta(W)}{E}[R_\theta(W)] + C .$$

Algorithm 1 The AREL Algorithm.

```
for episode ← 1 to N do
    collect story W by executing policy π_θ
    if Train-Reward then
        θ ← θ − η × ∂J_θ/∂θ (see Equation 9)
    else if Train-Policy then
        collect story W̄ from empirical p_e
        β ← β − η × ∂J_β/∂β (see Equation 9)
    end if
end for
```

$$\frac{\partial J_\theta}{\partial \theta} = \underset{W\sim p_e(W)}{E} \frac{\partial R_\theta(W)}{\partial \theta} - \underset{W\sim\pi_\beta(W)}{E} \frac{\partial R_\theta(W)}{\partial \theta} ,$$

$$\frac{\partial J_\beta}{\partial \beta} = \underset{W\sim\pi_\beta(W)}{E} (R_\theta(W) + \log \pi_\theta(W) - b) \frac{\partial \log \pi_\beta(W)}{\partial \beta} ,$$

where $p(|mathbf s_1)$ and $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{y}_t)$ are the true initial state distribution and dynamics, and $\pi(\mathbf{y}_t|\mathbf{s}_t)$ is the parameterized policy that we wish to learn. We can t this distribution by maximizing the evidence lower bound (ELBO):

$$\begin{aligned}\log p(\mathcal{O}_{1:T}) \geq &- \mathrm{KL}(q(\tau)||p(\mathcal{O}_{1:T}, \tau)) \\ =&\mathbb{E}_{\tau\sim\rho_\pi(\tau)}\left[\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{y}_t) - \mathrm{KL}(\pi(\mathbf{y}_t|\mathbf{s}_t)||p(\mathbf{y}_t))\right]\end{aligned} \tag{7}$$

Variational Latent Policy Learning

$$\begin{aligned}\mathcal{J} = \mathbb{E}_{\tau\sim\rho_\pi(\tau)}\Bigg[&\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{y}_t) - \mathrm{KL}(\pi(\mathbf{y}_t|\mathbf{s}_t, \mathbf{c})||p(\mathbf{y}_t)) \\ &- \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{y}, \mathbf{c})||p_\theta(\mathbf{z}|\mathbf{c}))\Bigg]\end{aligned} \tag{8}$$

Counterfactual Multi-Agent Policy Gradient