Ailab Seminar #13

- SeqGAN -

최원혁

- Task
  - ✓ Text Generation

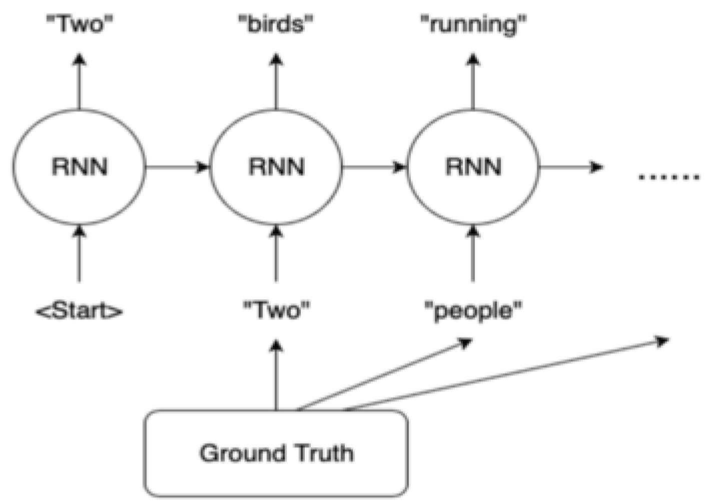
- Contribution
  - ✓ Generation task에서 GAN이 좋은 성능을 보이지만, 기존의 GAN 방식으로는 Text Generation에 적용할 수 없었음.
  - ✓ Reinforcement Learning 알고리즘을 GAN을 적용하여 Text Generation을 함

SeqGAN

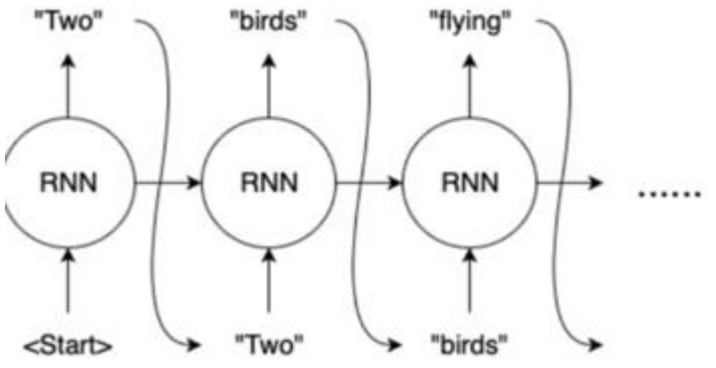
Text Generation Problem

- 기존의 MLE 방법의 문제점 <Exposure Bias>
  - ✓ Training 시 Teaching Force 방법으로 학습
  - ✓ Inference 시 단어 Token이 한 번 잘못 예측되면,

    계속 잘못된 값이 전파되는 것



<Teaching Force>

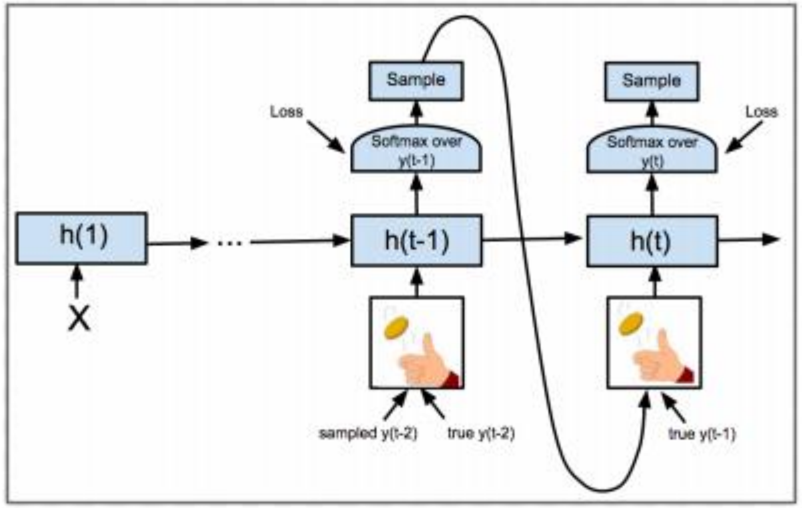


<Without Teaching Force>

- MLE 방식으로 훈련시킨 모델 – Inference 예시

    ✓ The first county of the county was the county of County County County County County County and the county of Cheadle County County County County County County Route 1.

- Exposure Bias 해결방법 1 : Scheduled Sampling (SS)
  - ✓ Training 시 Sampled Token과 Ground Truth Token을 확률적으로 입력
  - ✓ Exposure Bias에 강건한 model을 훈련
  - ✓ 근본적인 해결방법은 아님



<Scheduled Sampling>

Bengio, Samy, et al. "Scheduled sampling for sequence prediction with recurrent neural networks." *Advances in Neural Information Processing Systems*. 2015.

- Exposure Bias 해결방법 2 : Loss Function
  - ✓ 생성된 문장의 성능(ex. BLEU score)을 평가할 수 있는 metric을 Loss에 추가적으로 적용
  - ✓ Task 마다 Loss 선택에 대한 성능차이가 존재

- Generative Adversarial Network(GAN)은 Image Generation에서 좋은 성능을 보임
- 하지만 Text Generation에 적용하기에 두 가지 문제점이 있음

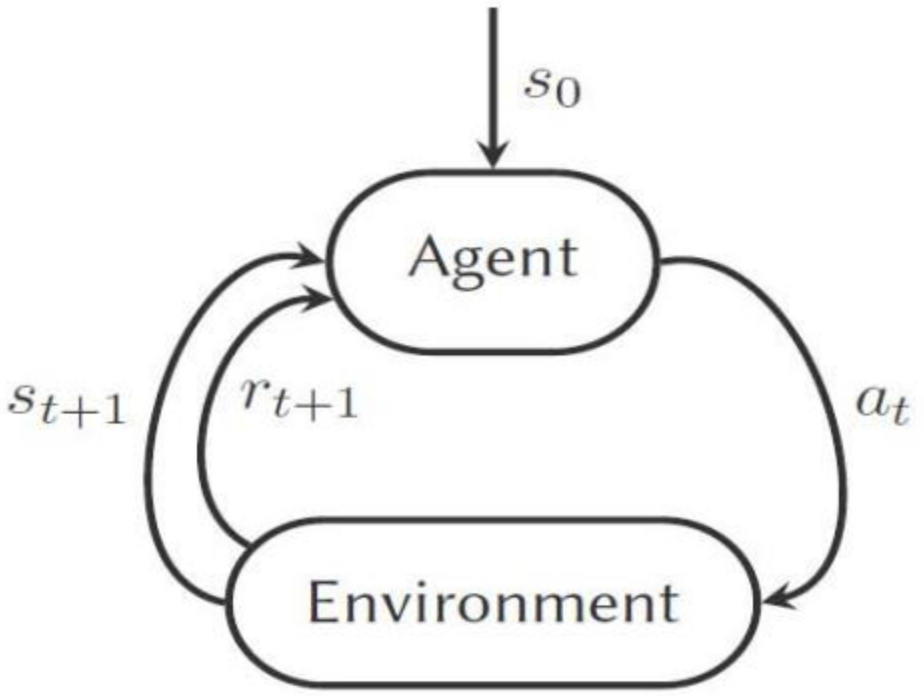
1. GAN은 continuous data를 생성하도록 Design 됨
   ✓ Text는 분포가 discrete한 token들의 sequence
2. GAN의 Discriminator는 전체 완성된 문장에 대해서 score/loss를 줄 수 있음
   ✓ Time step 마다 생성되는 token에 대한 score/loss가 필요

SeqGAN

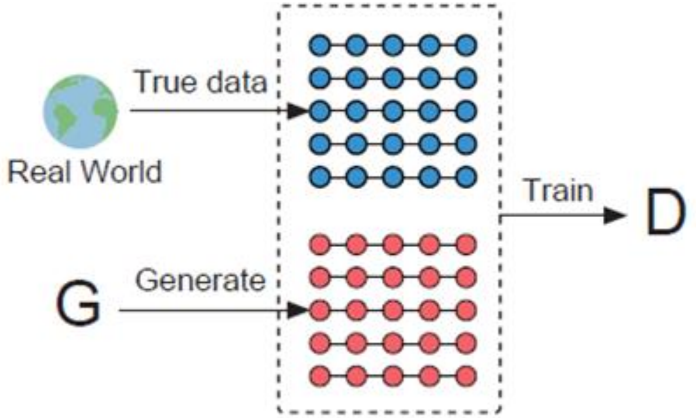
Proposal

- SeqGAN은 Reinforcement 알고리즘을 적용하여 text Generation에 GAN을 적용

- State
  - ✓ 현재 상태(게임 화면)
- Action
  - ✓ 취하는 행동(점프, 이동)
- Reward
  - ✓ 행동에 대한 결과(+1,-1,0)



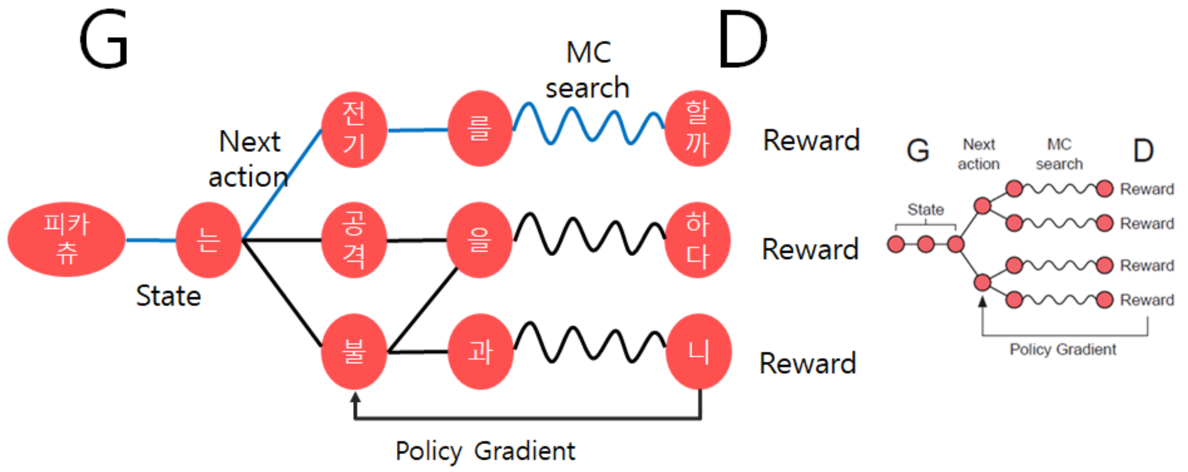
- Generator model : $G_\theta(y_t|Y_{1:t-1})$
  - ✓ State = $(y_1, y_2, \dots, y_{t-1})$, current provided tokens
  - ✓ Action = next token y_t 선택
  - ✓ $G_\theta(y_t|Y_{1:t-1})$은 stochastic model
  - ✓ LSTM 사용
- Discriminator model : $D_\emptyset$
  - ✓ $D_\emptyset(Y_{1:T})$는 생성된 문장이 real인지 확률을 생성
  - ✓ CNN 구조 (Zhang and LeCun 2015, Text Understanding from Scratch)



SeqGAN

MC search with roll-out policy

- Monte Carlo search with roll-out policy : $\left\{Y_{1:T}^{1}, \ldots, Y_{1:T}^{N}\right\} = \mathrm{MC}^{G_\beta}(Y_{1:t}; N),$
  - ✓ $y_1 \sim y_t$가 주어지고, $y_{t+1} \sim y_T$까지를 G_β를 따라서 N번 sampling
  - ✓ G_β는 Generator를 의미
  - ✓ MC search로 sampling 한 후 Discriminator로 판별 후 Immediate Reward를 계산



SeqGAN

MC search with roll-out policy

- action-value function

✓ $$Q_{D_\phi}^{G_\theta}(s = Y_{1:t-1}, a = y_t) = \tag{4}$$
$$\begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\phi(Y_{1:T}^n), \; Y_{1:T}^n \in \mathrm{MC}^{G_\beta}(Y_{1:t}; N) & \text{for} \quad t < T \\ D_\phi(Y_{1:t}) & \text{for} \quad t = T, \end{cases}$$

- Objective function

✓ $$\nabla_\theta J(\theta) \simeq \sum_{t=1}^{T} \sum_{y_t \in \mathcal{Y}} \nabla_\theta G_\theta(y_t|Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t) \tag{7}$$
$$= \sum_{t=1}^{T} \sum_{y_t \in \mathcal{Y}} G_\theta(y_t|Y_{1:t-1}) \nabla_\theta \log G_\theta(y_t|Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t)$$
$$= \sum_{t=1}^{T} \mathbb{E}_{y_t \sim G_\theta(y_t|Y_{1:t-1})}[\nabla_\theta \log G_\theta(y_t|Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t)],$$

- Update

✓ $$\theta \leftarrow \theta + \alpha_h \nabla_\theta J(\theta), \tag{8}$$

SeqGAN

Algorithm

Algorithm 1 Sequence Generative Adversarial Nets

Require: generator policy G_θ; roll-out policy G_β; discriminator D_ϕ; a sequence dataset $\mathcal{S} = \{X_{1:T}\}$

Initialize G_θ, D_ϕ with random weights θ, ϕ.
Pre-train G_θ using MLE on $\mathcal{S}$
$\beta \leftarrow \theta$
Generate negative samples using G_θ for training D_ϕ
Pre-train D_ϕ via minimizing the cross entropy
repeat
    for g-steps **do**
        Generate a sequence $Y_{1:T} = (y_1, \ldots, y_T) \sim G_\theta$
        for t in $1 : T$ **do**
            Compute $Q(a = y_t; s = Y_{1:t-1})$ by Eq. (4)
        end for
        Update generator parameters via policy gradient Eq. (8)
    end for
    for d-steps **do**
        Use current G_θ to generate negative examples and combine with given positive examples $\mathcal{S}$
        Train discriminator D_ϕ for k epochs by Eq. (5)
    end for
    $\beta \leftarrow \theta$
until SeqGAN converges

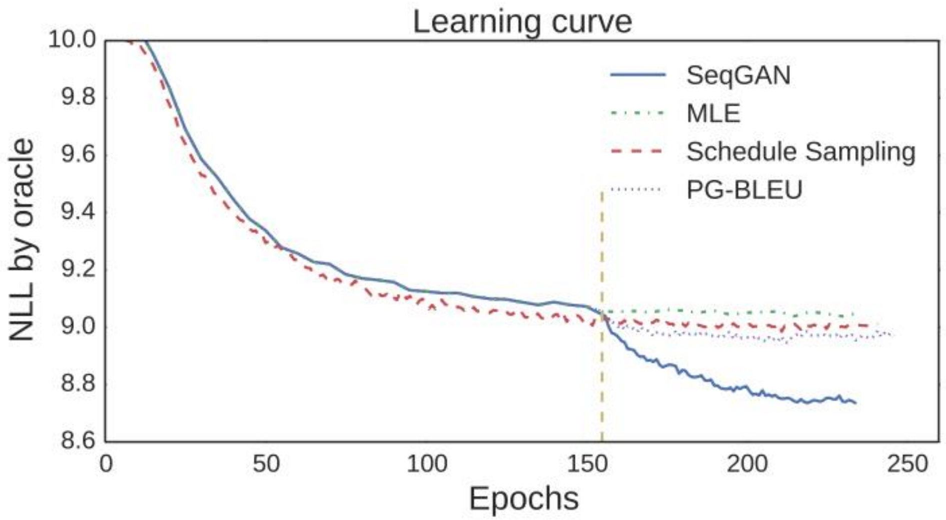
- Oracle로 불리는 랜덤 초기화 LSTM을 True model로 사용
  - ✓ 이 모델이 만드는 데이터가 진짜 data라고 가정
  - ✓ 이 데이터로 model이 학습하여 잘 모방한다면 제대로 학습이 된다는 것
  - ✓ 장점 : 학습 데이터세트를 제공, 정확한 성능 측정 가능
- Evaluation Metric
  - ✓ $\mathrm{NLL}_{\mathrm{oracle}} = -\mathbb{E}_{Y_{1:T} \sim G_\theta}\left[\sum_{t=1}^{T} \log G_{\mathrm{oracle}}(y_t | Y_{1:t-1})\right]$
  - ✓ NLL이 낮으면 model이 oracle을 잘 모방한다는 것

SeqGAN

Result

Table 1: Sequence generation performance comparison. The p-value is between SeqGAN and the baseline from T-test.

Algorithm	Random	MLE	SS	PG-BLEU	SeqGAN
NLL	10.310	9.038	8.985	8.946	**8.736**
p-value	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	



SeqGAN – MLE로 150 epoch train 이후 GAN 방식으로 학습

SeqGAN

Result

D-step과 G-step에 따라서 SeqGAN의 안전성이 결정된다

실험적으로 G-step이 크면 수렴은 빠르지만, 안정성이 떨어진다

D-step이 G-step 이상이어야 SeqGAN이 안정적이다

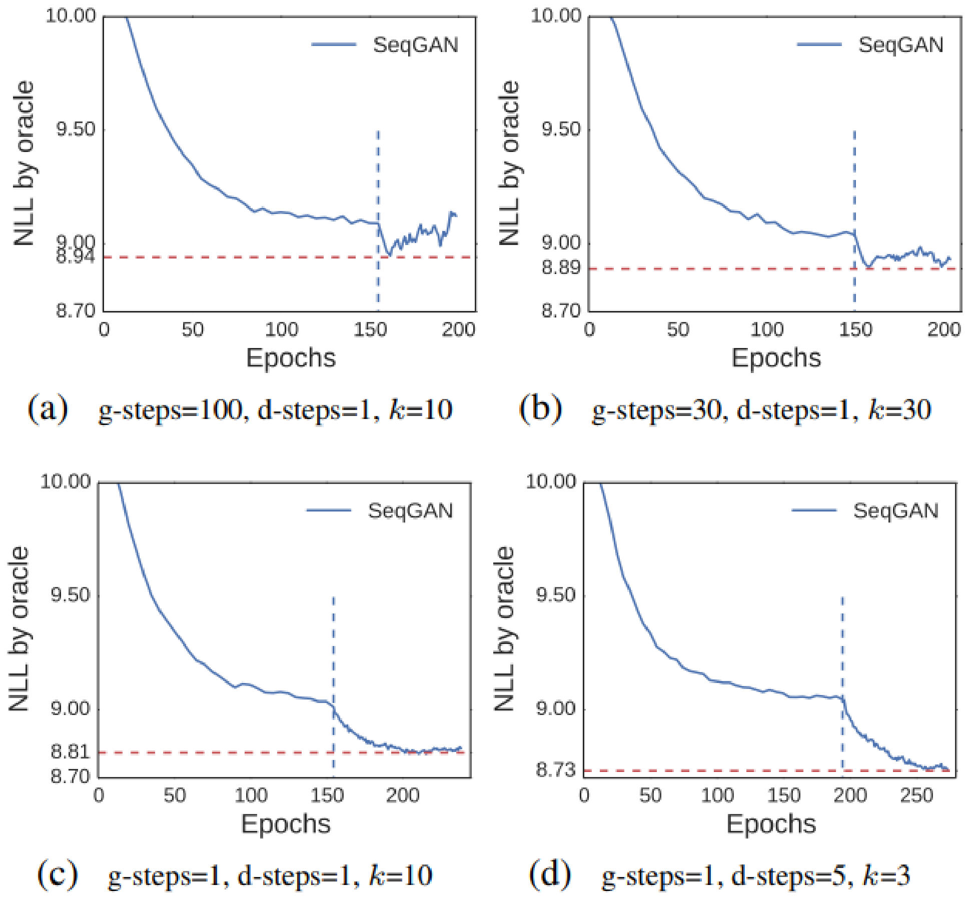


Figure 3: Negative log-likelihood convergence performance of SeqGAN with different training strategies. The vertical dashed line represents the beginning of adversarial training.

End