2019

RL与GAN在文本序列生成的应用

Without ideal, life is a desert, without vitality; Without ideal, life is like night, without light; Without ideal, life is like a maze, without direction.

汇报人：弭晓月

汇报时间：2019.04.22

Part one

RL in sequence generation

Trust me

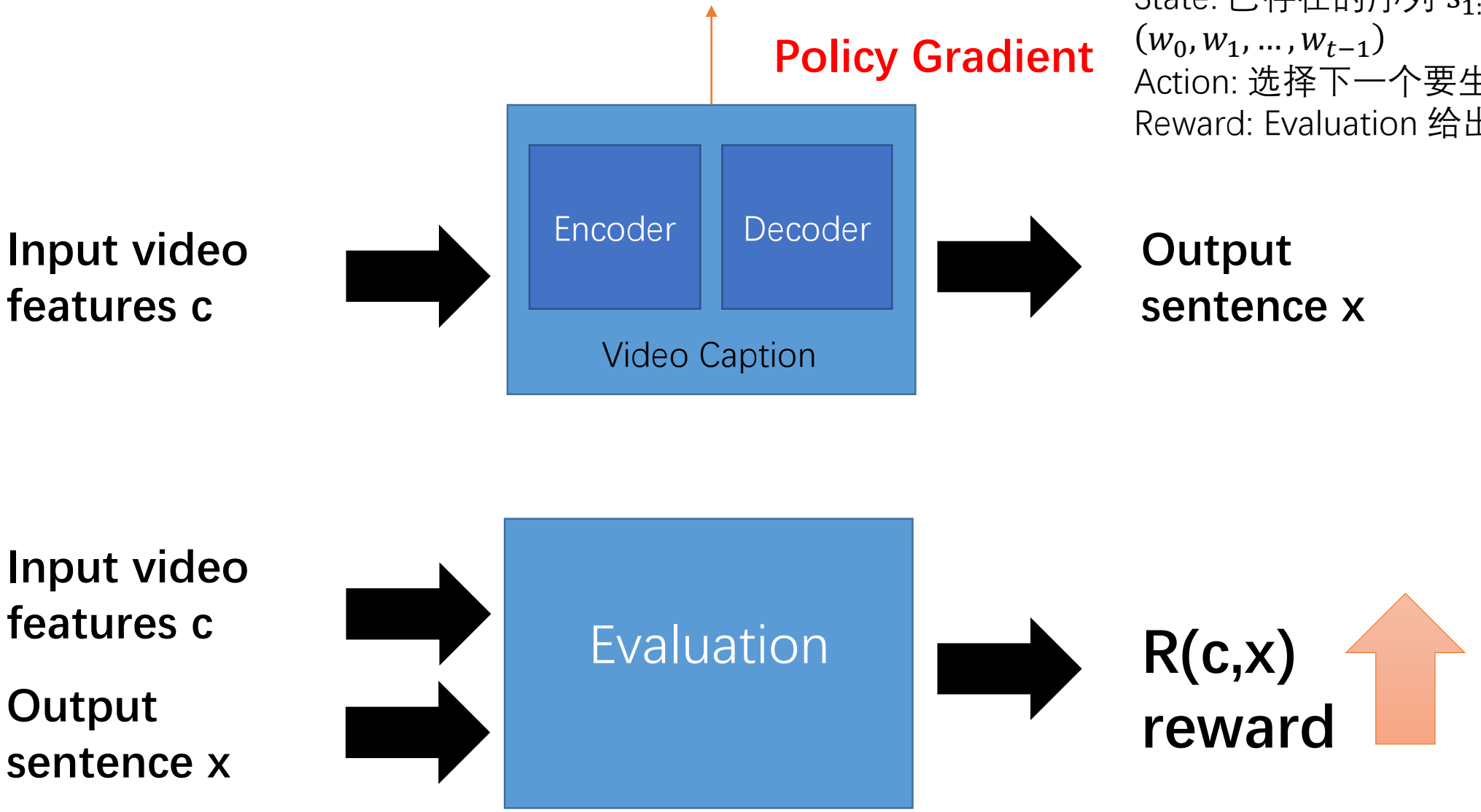
基于交叉熵损失函数的Sequence-Sequence 模型常常有两大问题:

- 训练与测试的评价机制不一致
  1. 直接以评价指标作为优化目标 →不可微

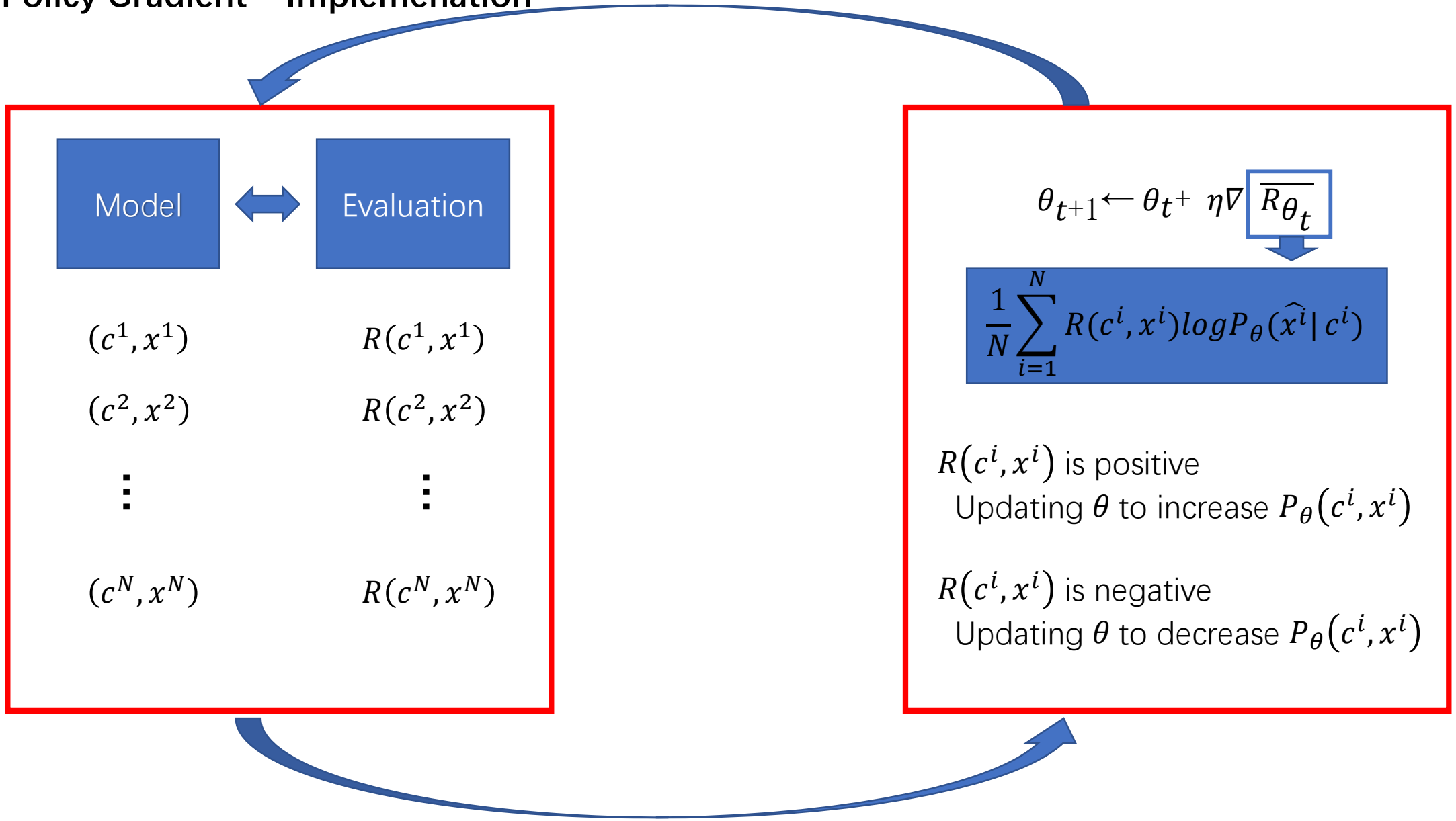
- Exposure Bias
  1. scheduled sampling→未能从根本上解决问题
  2. 在整个生成的序列上构建损失函数→合适的评价指标难找

用RL可以解决上述两个问题!

学会最大化期望奖励


Policy Gradient
State: 已存在的序列 $s_{1:t-1} = (w_0, w_1, \dots, w_{t-1})$
Action: 选择下一个要生成的token w_t
Reward: Evaluation 给出
Input video features c
Encoder
Decoder
Video Caption
Output sentence x
Input video features c
Output sentence x
Evaluation
R(c,x) reward

Policy Gradient - Implemenation



	MLE	**RL**
Objective Function	$\frac{1}{N}\sum_{i=1}^{N} logP_\theta(\widehat{x^i}\mid c^i)$	$\frac{1}{N}\sum_{i=1}^{N} R(c^i,x^i) logP_\theta(\widehat{x^i}\mid c^i)$
Gradient	$\frac{1}{N}\sum_{i=1}^{N} \nabla\, logP_\theta(\widehat{x^i}\mid c^i)$	$\frac{1}{N}\sum_{i=1}^{N} R(c^i,x^i) \nabla\, logP_\theta(\widehat{x^i}\mid c^i)$
Training Data	$\{(c^1,\widehat{x^1}),\dots,(c^N,\widehat{x^N})\}$ $R(c^i,\widehat{x^i})=1$	$\{(c^1,\widehat{x^1}),\dots,(c^N,\widehat{x^N})\}$ obtained from interaction weighted by $R(c^i,\widehat{x^i})$

MLE 可以看作 一种特殊的 RL

目前的评价指标都存在问题，不能很好的指导文本的生成

GAN!

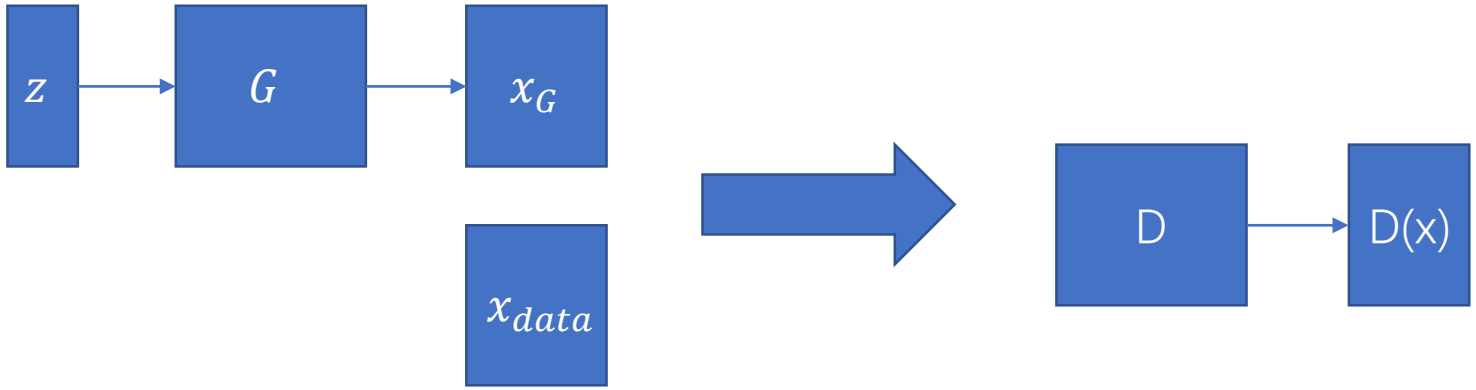
Part two

GAN in sequence generation

Trust me

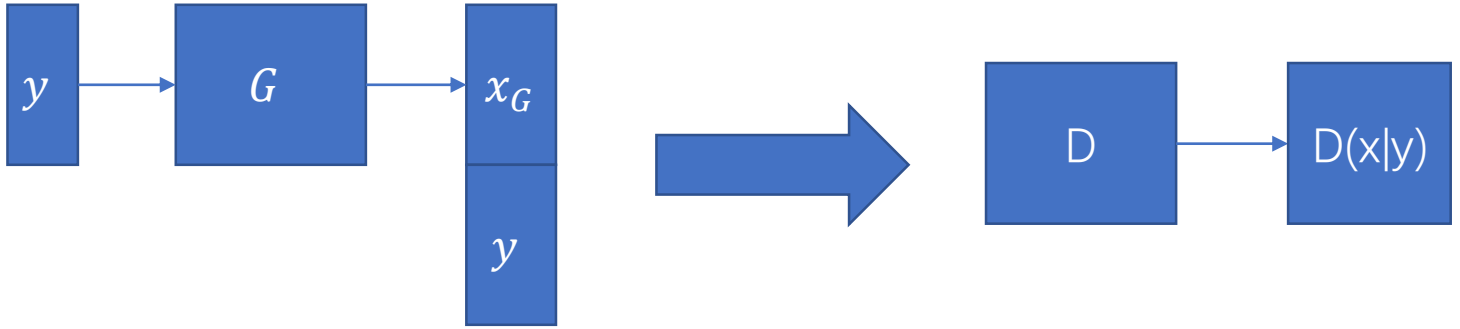
Generative Adversarial Nets

- Proposed by Ian J. Goodfellow et al. in 2014

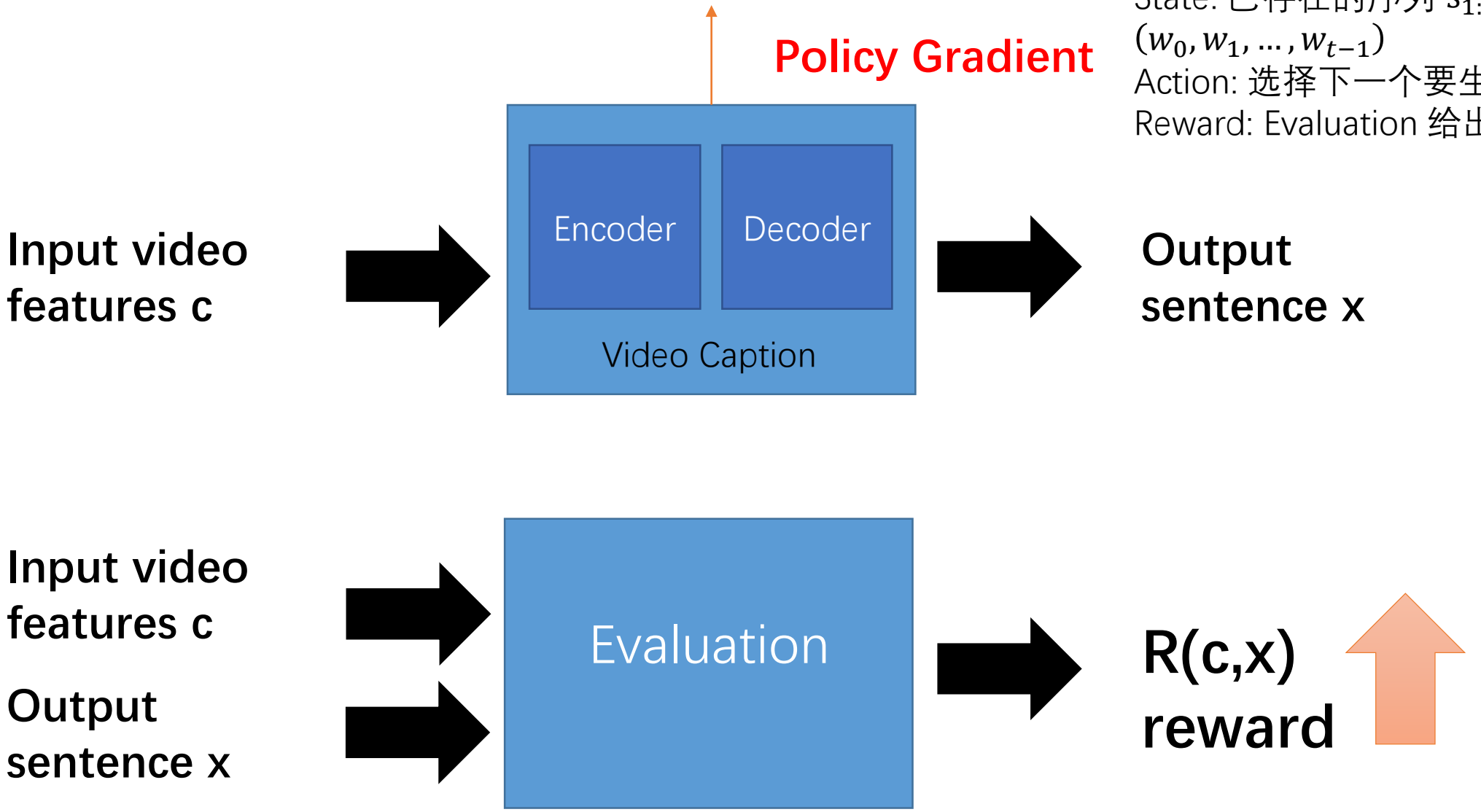


Conditional Generative Adversarial Nets

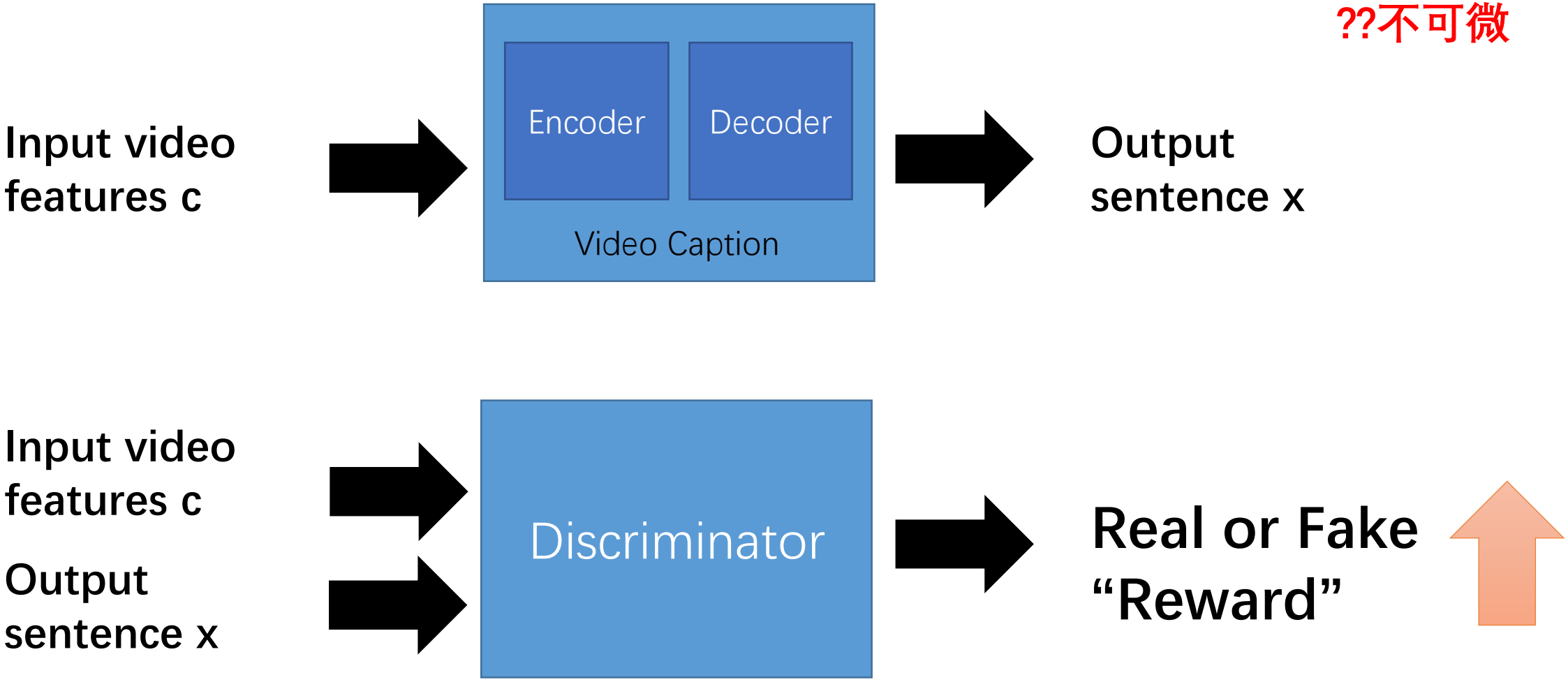
- Proposed by Mehdi Mirza et al. in 2014



学会最大化期望奖励


Policy Gradient
State: 已存在的序列 $s_{1:t-1} = (w_0, w_1, \dots, w_{t-1})$
Action: 选择下一个要生成的token w_t
Reward: Evaluation 给出
Input video features c
Encoder
Decoder
Video Caption
Output sentence x
Input video features c
Output sentence x
Evaluation
R(c,x) reward

??不可微


Input video features c
Encoder
Decoder
Video Caption
Output sentence x
Input video features c
Output sentence x
Discriminator
Real or Fake
“Reward”

Part three

SeqGAN

Trust me

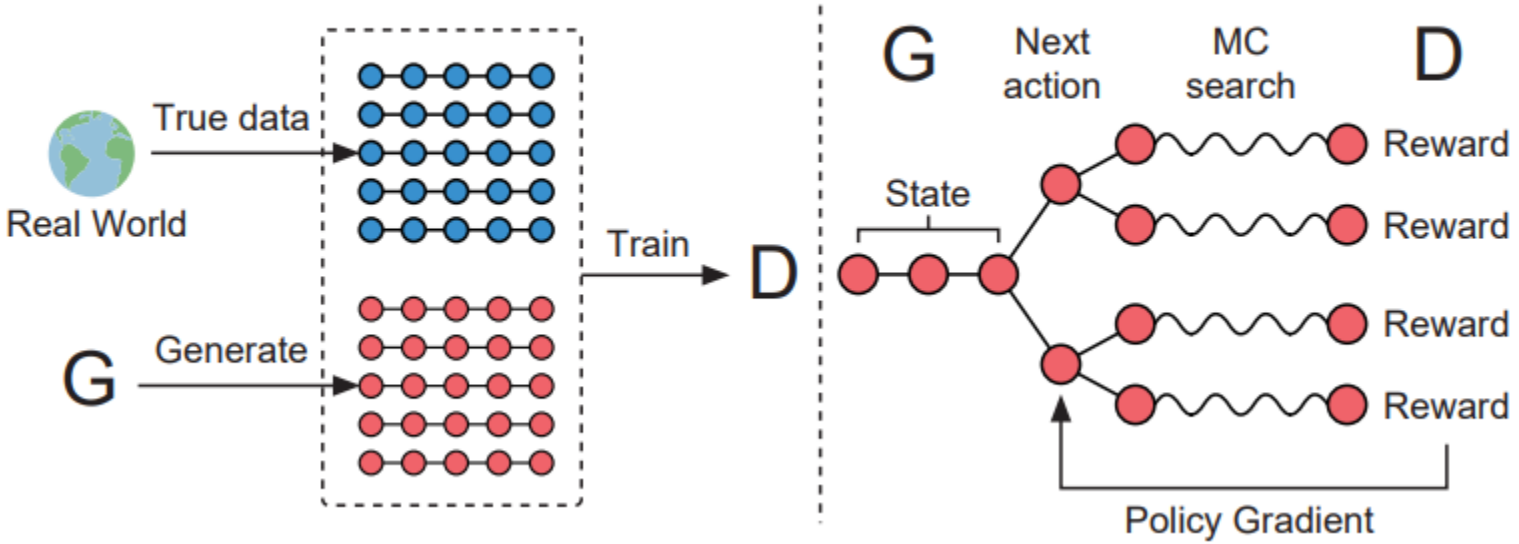
SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient

Lantao Yu[†], Weinan Zhang[†*], Jun Wang[‡], Yong Yu[†]
[†]Shanghai Jiao Tong University, [‡]University College London
{yulantao,wnzhang,yyu}@apex.sjtu.edu.cn, j.wang@cs.ucl.ac.uk

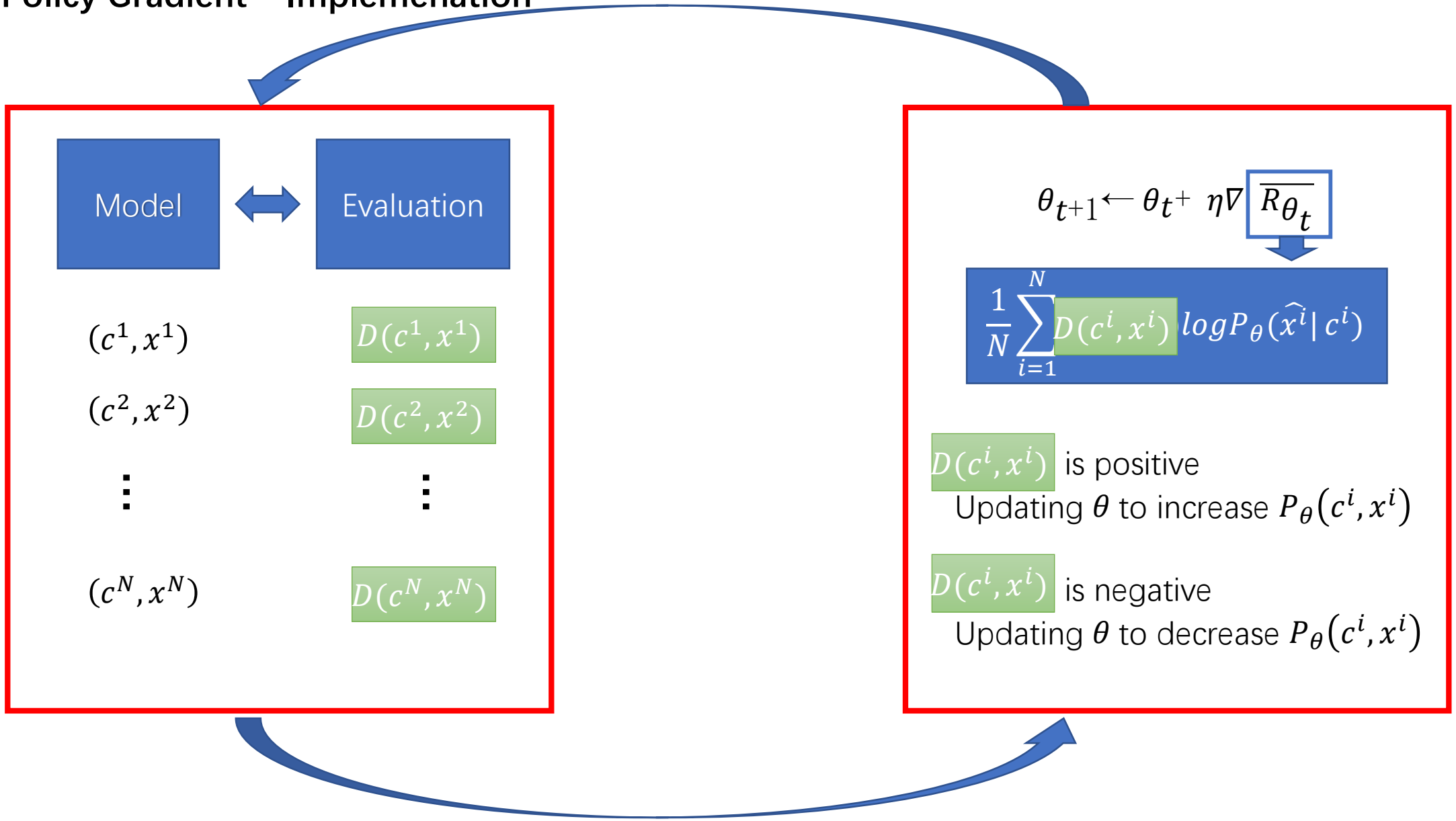
- Source: AAAI A类会议 2017
- Organization:
  Shanghai Jiao Tong University
  University College London
- Motivations:
  解决GAN在离散序列生成的不可微问题,和判别模型只能对一个完整的序列进行评价的缺陷

- GAN在离散序列生成的不可微问题
  - Policy gradient.

    State: 已存在的序列 $s_{1:t-1} = (w_0, w_1, \dots, w_{t-1})$
    Action: 选择下一个要生成的token w_t
    Reward: 判别器得分
- 判别模型只能对一个完整的序列进行评价的缺陷
  - Monte Carlo Search



Policy Gradient - Implemenation



- Monte Carlo Search
  - MCTS可以无限循环，而每一次循环都由以下4个步骤构成：

  - Selection：从根节点开始，连续选择子节点向下搜索，直至抵达一个叶节点。子节点的选择方法一般采用UCT（Upper Confidence Bound applied to trees）算法，根据节点的“胜利次数”和“游戏次数”来计算被选中的概率，保持了Exploitation和Exploration的平衡，是保证搜索向最优发展的关键。

  - Expansion：在叶节点创建多个子节点。

  - Simulation：在创建的子节点中根据roll-out policy选择一个节点进行模拟，又称为playout或者rollout。它和Selection的区别在于：Selection指的是对于搜索树中已有节点的选择，从根节点开始，有历史统计数据作为参考，使用UCT算法选择每次的子节点；Simulation是简单的模拟，从叶节点开始，用自定义的roll-out policy（可以只是简单的随机概率）来选择子节点，且模拟经过的节点并不加入树中。

  - Backpropagation：根据Simulation的结果，沿着搜索树的路径向上更新节点的统计信息，包括“胜利次数”和“游戏次数”，用于Selection做决策。

- Monte Carlo Search
  - MCTS可以无限循环，而每一次循环都由以下4个步骤构成：

  - Selection：从根节点开始，连续选择子节点向下搜索，直至抵达一个叶节点。子节点的选择方法一般采用UCT（Upper Confidence Bound applied to trees）算法，根据节点的“胜利次数”和“游戏次数”来计算被选中的概率，保持了Exploitation和Exploration的平衡，是保证搜索向最优发展的关键。

  - Expansion：在叶节点创建多个子节点。

  - Simulation：在创建的子节点中根据roll-out policy选择一个节点进行模拟，又称为playout或者rollout。它和Selection的区别在于：Selection指的是对于搜索树中已有节点的选择，从根节点开始，有历史统计数据作为参考，使用UCT算法选择每次的子节点；Simulation是简单的模拟，从叶节点开始，用自定义的roll-out policy（可以只是简单的随机概率）来选择子节点，且模拟经过的节点并不加入树中。

  - Backpropagation：根据Simulation的结果，沿着搜索树的路径向上更新节点的统计信息，包括“胜利次数”和“游戏次数”，用于Selection做决策。

Part four

RankGAN

Trust me

Adversarial Ranking for Language Generation

Kevin Lin*
University of Washington
kvlin@uw.edu

Dianqi Li*
University of Washington
dianqili@uw.edu

Xiaodong He
Microsoft Research
xiaohe@microsoft.com

Zhengyou Zhang
Microsoft Research
zhang@microsoft.com

Ming-Ting Sun
University of Washington
mts@uw.edu

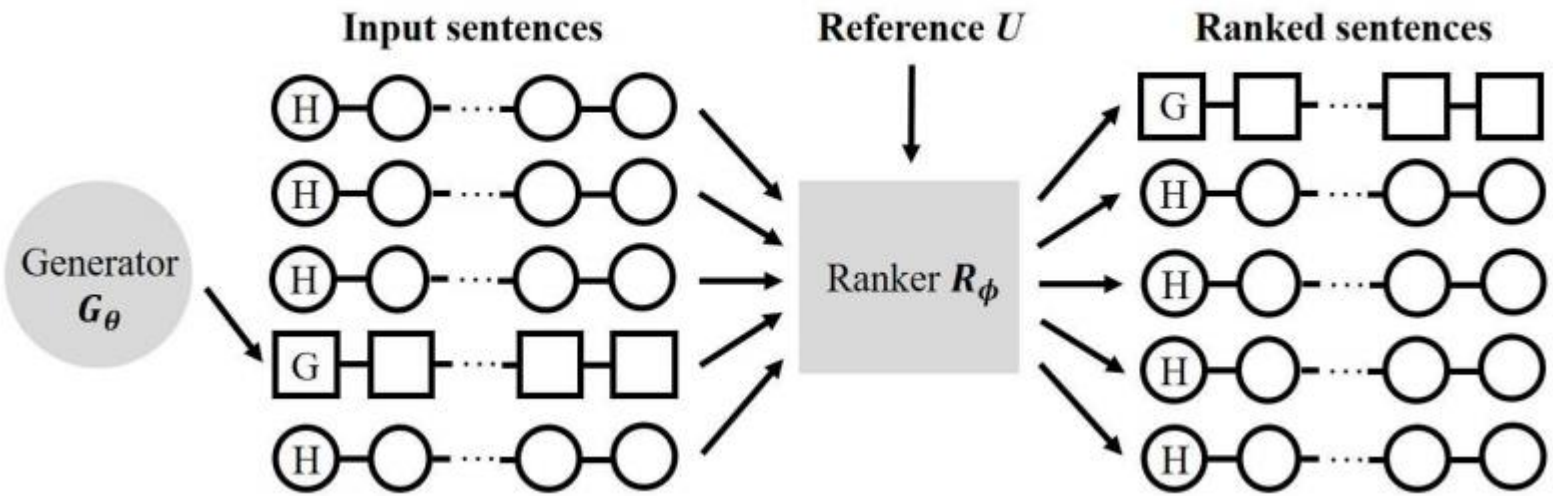
- Source: 2017 NIPS
- Organization:
  1. University of Washington
  2. Microsoft Research
- Motivations:
  1. 现有的GANs将识别器限制为二分类器，从而限制了它们对需要合成具有丰富结构(如自然语言描述)输出的任务的学习能力

发现:

- 大多数现有的GANs假设判别器的输出是一个二元谓词，表示给定的句子是由人还是机器写的.对于大量的自然语言表达式来说，这种二值化的预测方法限制太多，因为句子内部的多样性和丰富性受到二值化分类导致的退化分布的限制

  →RankGAN从机器写的句子和人写的句子之间的相对排序信息中学习了该模型，将判别器的训练简化为一个学习到rank的优化问题。

RankGAN



生成器：合成使排序者感到困惑的句子，从而使机器写的句子在引用方面的排名高于人写的句子。

排序器：训练排序器将机器写的句子相对于人写的参考句进行排序，使机器写的句子低于人写的句子。以排名分数作为学习语言生成器的奖励。

- U为估算相对排名的参考集，C^+，C^-为不同输入语句s的比较集。当输入语句s为真实数据时，从生成的数据中预采样C^-;如果输入的句子s是合成数据，那么C^+将从人工编写的数据中预先采样
- Rank score
- 排序器与卷积结构相似，首先通过一系列非线性函数F将串联序列矩阵映射到嵌入的特征向量y = F(s)中，然后用R预先提取的参考特征yu计算序列特征y的排序得分。
- 1.计算s与参考句u的相似度

$$\alpha(s|u) = cosine(y_s, y_u) = \frac{y_s \cdot y_u}{\|y_s\|\|y_u\|}$$

- 2.在给定比较集C的情况下，利用类似softmax公司计算某序列s的排序得分

$$P(s|u,\mathcal{C}) = \frac{exp(\gamma\alpha(s|u))}{\sum_{s' \in \mathcal{C}'} exp(\gamma\alpha(s'|u))} \qquad \mathcal{C}' = \mathcal{C} \cup \{s\}$$

- 3.输入句子的总体排名分数是给定在参考空间中采样的不同参考句的期望分数。（在学习过程中，我们从人写的句子中随机抽取一组参考文献，构建参考文献U，同时根据输入句子s的类型构建比较集C，即，如果s是G生成的合成句，则从人写的集合中抽取C，反之亦然。）

$$\log R_\phi(s|U,\mathcal{C}) = \mathop{\mathbb{E}}_{u \in U} \log\left[P(s|u,\mathcal{C})\right]$$

Part five

DPGAN

Trust me

DP-GAN: Diversity-Promoting Generative Adversarial Network for Generating Informative and Diversified Text

Jingjing Xu*, Xu Sun*, Xuancheng Ren, Junyang Lin, Binzhen Wei, Wei Li
School of Electronics Engineering and Computer Science, Peking University
{jingjingxu,xusun,renxc,linjunyang,weibz,liweitj47}@pku.edu.cn

- Source: EMNLP B类会议 2018
- Organization:
  Peking University
- Motivations:
  增强GAN文本生成的新颖性

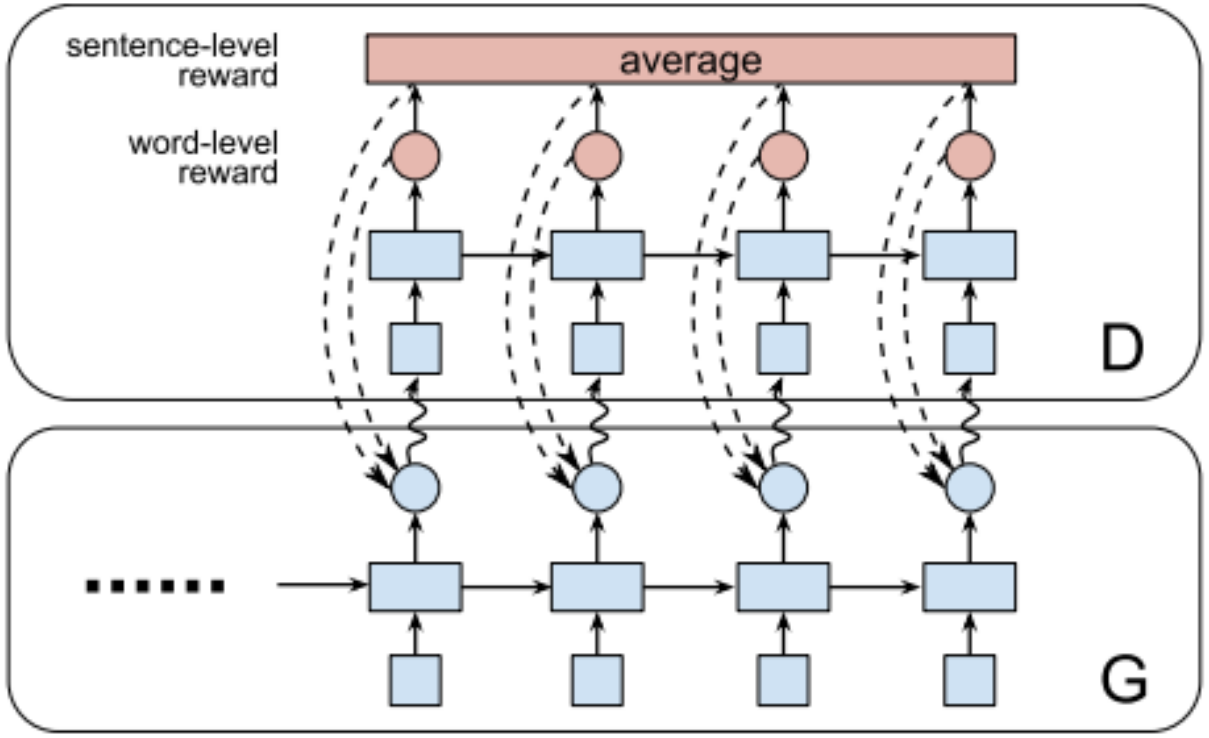


Figure 1: Illustration of DP-GAN. Lower: The generator is trained by policy gradient where the reward is provided by the discriminator. Upper: The discriminator is based on the language model trained over the real text and the generated text.

Word-Level Reward

$$R(y_{t,k}|y_{t,<k}) = -\log D_\phi(y_{t,k}|y_{t,<k})$$

Sentence-Level Reward

$$R(y_t) = -\frac{1}{K}\sum_{k=1}^{K} \log D_\phi(y_{t,k}|y_{t,<k})$$

总　　结

- RL in Sequence Generation (Policy Gradient)
  - State: 已存在的序列 $s_{1:t-1} = (w_0, w_1, \dots, w_{t-1})$
  - Action: 选择下一个要生成的token w_t
  - Reward: Evaluation 给出
  - Policy: EN-DE

- GAN in Sequence Generation
  - Policy Gradient + Monte Carlo Search
  - SeqGAN
  - RankGAN
  - DPGAN

2019

THANKS!

Without ideal, life is a desert, not angry; Without ideal, life is like night, without light; Without ideal, life is like a maze, without direction.

汇报人：弭晓月

汇报时间：2019.04.22