

# 生成对抗网络在文本生成中的应用

Xiachong Feng

## 摘要

生成对抗网络 (Generative Adversarial Network, GAN) 于 2014 年被提出, 并广泛应用于计算机视觉领域, 例如图片生成等任务。生成对抗网络一般包括生成器与判别器两个部分。生成器用于生成数据, 判别器用于区分数据的真假。生成器尽可能使得生成的数据可以“骗过”判别器。通过对抗训练的方式, 生成器与判别器同时提高各自性能。图片天然属于连续型数据, 使得生成对抗网络可以直接进行梯度求导和反向传播。由于文本生成中存在采样过程, 导致输出离散化, 使得梯度无法直接从判别器传递到生成器, 因此一定程度上阻碍了生成对抗网络应用于文本生成领域。但是随着近几年结合强化学习方法, 一定程度上缓解了该问题, 使得生成对抗网络开始越来越多的应用在文本生成领域, 缓解了基于极大似然估计目标函数训练导致的文本生成单一、重复、无意义的问题。

## 1 简介

生成对抗网络 (Generative Adversarial Network, GAN) 最初由 Goodfellow[Goodfellow et al., 2014] 提出, 并广泛应用于计算机视觉领域。生成对抗网络通常由两个核心部分构成。生成器 (Generator) 与判别器 (Discriminator), 生成器用于完成生成任务, 例如图像生成, 句子生成。判别器用于区分真实数据与非真实数据。如图 1 所示为生成对抗网络的基本结构图。

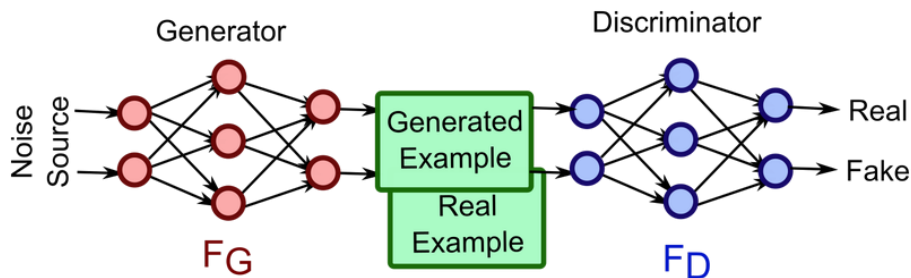


图 1: 生成对抗网络结构图

判别器的目的是尽可能区分真实数据与非真实数据, 生成器的目的是尽可能使模型的生成“骗过”判别器。通过对抗训练的方式, 生成器与判别器逐步更新, 来完成生成器与判别器的联合训练。假设  $\{x^1, x^2, \dots, x^m\}$  为真实数据,  $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$  为生成器生成数据, 生成器表示为  $G$ , 判别器表示为  $D$ , 生成器与判别器的目标函数通常表述如公式 1,2 所示。

$$\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log D(x^i) + \frac{1}{m} \sum_{i=1}^m \log(1 - D(\tilde{x}^i)) \quad (1)$$

$$\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log(D(G(z^i))) \quad (2)$$

生成对抗网络被广泛应用于计算机视觉领域, 例如图像生成任务。图像可以被视为天然的连续型数据, 因此可以使用生成对抗网络直接进行梯度回传进行训练。最近几年, 该网络才开始应用于

自然语言处理领域，一个重要的原因是在文本生成的过程中存在采样的过程，导致最终的损失无法直接进行回传。解决该问题的方式有强化学习、Gumbel-softmax[Kusner and Hernández-Lobato, 2016]、使用连续型输入。强化学习使用策略梯度 (Policy gradient) 将判别器的输出作为一种“奖励”训练生成器，Gumbel-softmax 将原本因为采样导致的离散过程转变可以微分的过程，除此以外，还可以使用连续输出代替采样过程。本文主要讨论使用强化学习的方法。由于上述方法的出现，一定程度上缓解了将生成对抗网络应用于自然语言处理领域的难题。因此，该网络开始在自然语言处理的多个领域崭露头角，并取得了一定的效果，例如：无监督机器翻译，文本风格迁移，无监督摘要等等。

本文后面小节将介绍近几年来将生成对抗网络应用于自然语言处理的一些方法，最后做出总结。

## 2 CGAN

生成对抗网络一种最常见的使用形式为条件生成对抗网络 (Conditional GAN)[Mirza and Osindero, 2014]，在图片生成和文本生成领域有着重要的应用。其与普通的生成对抗网络最主要的区别在于判别器的输入部分。判别器除了输入生成器的输出以外，往往还需要输入生成器的输入部分，构成限制条件，使得生成器的输入与输出有很强的关联。常见的条件生成对抗网络判别器的结构如图 2所示。

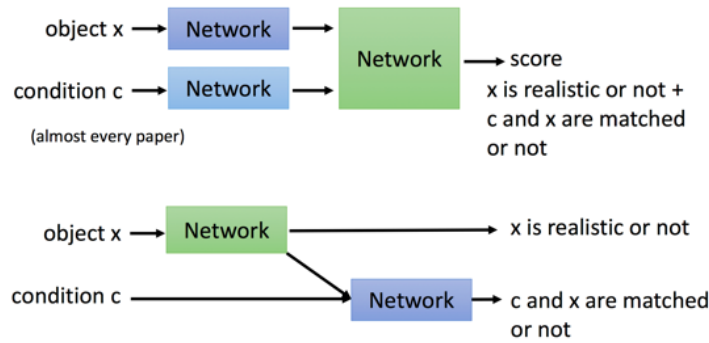


图 2: CGAN 判别器结构图

在条件生成对抗网络的情形下，生成器与判别器的目标函数形式化为公式 3和 4所示。

$$\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log D(c^i, x^i) + \frac{1}{m} \sum_{i=1}^m \log (1 - D(c^i, \tilde{x}^i)) + \frac{1}{m} \sum_{i=1}^m \log (1 - D(c^i, \hat{x}^i)) \quad (3)$$

$$\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log (D(G(c^i, z^i))) \quad (4)$$

条件生成对抗网络的训练算法如图 3所示。

## 3 SeqGAN

由于文本生成离散化输出的原因，梯度无法直接从判别器回传到生成器。SeqGAN[Yu et al., 2017] 使用强化学习中的策略梯度方法来解决这一问题。将生成器视为“代理”，当前已经生成的词语视为“状态”，下一时刻将要生成的词语视为“行为”，当生成下一个词语时，完成状态转移。使

- In each training iteration:
  - Sample  $m$  positive examples  $\{(c^1, x^1), (c^2, x^2), \dots, (c^m, x^m)\}$  from database
  - Sample  $m$  noise samples  $\{z^1, z^2, \dots, z^m\}$  from a distribution
  - Obtaining generated data  $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$ ,  $\tilde{x}^i = G(c^i, z^i)$
  - Sample  $m$  objects  $\{\hat{x}^1, \hat{x}^2, \dots, \hat{x}^m\}$  from database
  - Update discriminator parameters  $\theta_d$  to maximize
    - $\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log D(c^i, x^i)$
    - $+\frac{1}{m} \sum_{i=1}^m \log(1 - D(c^i, \tilde{x}^i)) + \frac{1}{m} \sum_{i=1}^m \log(1 - D(c^i, \hat{x}^i))$
    - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$
  - Sample  $m$  noise samples  $\{z^1, z^2, \dots, z^m\}$  from a distribution
  - Sample  $m$  conditions  $\{c^1, c^2, \dots, c^m\}$  from a database
  - Update generator parameters  $\theta_g$  to maximize
    - $\tilde{V} = \frac{1}{m} \sum_{i=1}^m \log(D(G(c^i, z^i)))$ ,  $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

图 3: CGAN 算法图

用蒙特卡洛搜索来估计状态行为价值函数。将判别器的输出作为“奖励”，通过该方法来更新生成器的参数。判别器使用真实数据和生成器生成的数据（假数据）进行训练。SeqGAN 结构如图 4 所示。

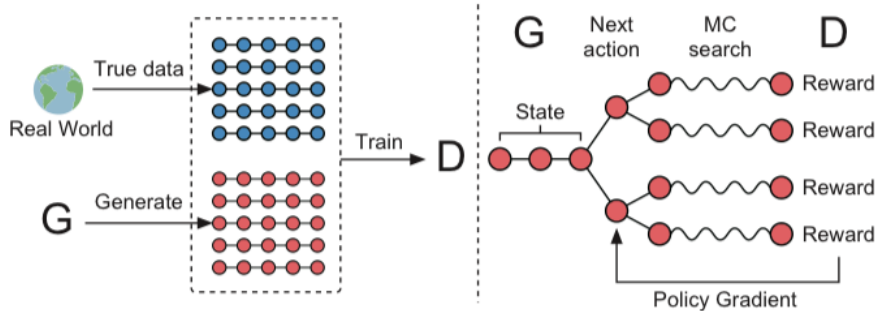


图 4: SeqGAN 结构图

生成器的目的是尽可能最大化从判别器得到的期望奖励。因此生成器的目标函数如公式 5 所示，

$$J(\theta) = E[R_T | s_0, \theta] = \sum_{y_1 \in \mathcal{Y}} G_\theta(y_1 | s_0) \cdot Q_{D_\phi}^{G_\theta}(s_0, y_1) \quad (5)$$

其中  $R_T$  是针对整个句子的奖励， $Q$  是行为价值函数，表示了在当前状态  $s$  下（目前已经生成的词语）执行行为  $a$ （生成下一个词语）的好坏。SeqGAN 使用判别器的概率输出作为行为价值函数的近似。如公式 6 所示。表示在已经生成  $T - 1$  个词的状态下，生成词语  $y_T$  的价值。

$$Q_{D_\phi}^{G_\theta}(a = y_T, s = Y_{1:T-1}) = D_\phi(Y_{1:T}) \quad (6)$$

上述奖励往往针对一个完整的句子给出，无法衡量部分生成句子的好坏。针对这一问题，SeqGAN 使用蒙特卡洛搜索来采样接下来的词语，并用来估计当前状态的好坏。因此最终的行为价值函数可以由公式 7 给出。

$$Q_{D_\phi}^{G_\theta}(s = Y_{1:t-1}, a = y_t) = \begin{cases} \frac{1}{N} \sum_{n=1}^N D_\phi(Y_{1:T}^n), Y_{1:T}^n \in \text{MC}^{G_\beta}(Y_{1:t}; N) & t < T \\ D_\phi(Y_{1:t}) & t = T \end{cases} \quad (7)$$

使用 REINFORCE 算法 [Williams, 1992] 完成梯度回传, 如公式 8。

$$\nabla_\theta J(\theta) = \sum_{t=1}^T E_{Y_{1:t-1} \sim G_\theta} \left[ \sum_{y_t \in \mathcal{Y}} \nabla_\theta G_\theta(y_t | Y_{1:t-1}) \cdot Q_{D_\phi}^{G_\theta}(Y_{1:t-1}, y_t) \right] \quad (8)$$

SeqGAN 生成器与判别器分别使用长短时神经网络 (LSTM) 与卷积神经网络 (CNN) 实现。在中文诗句生成、奥巴马演讲生成、音乐生成任务上进行了实验。该模型分别针对离散化输出导致的梯度回传问题和没有及时奖励两个问题提出了解决方法, 使用强化学习中的策略梯度方法完成梯度从判别器到生成器的回传, 使用蒙特卡洛搜索来估计行为价值函数, 提供及时奖励。SeqGAN 为之后将生成对抗网络应用于自然语言处理任务提出了一种通用框架。

## 4 DG-GAN

DG-GAN<sup>1</sup>来自于李纪为博士 2017 年的工作 [Li et al., 2017], 该工作将对抗训练用于开放域对话生成, 与 SeqGAN 相似, 使用强化学习来完成生成器与判别器的联合训练。使用判别器的输出作为奖励。目前生成任务往往使用一种被称之为序列到序列 (sequence-to-sequence, Seq2Seq) 的模型实现, 该模型使用极大似然估计 (MLE) 作为目标函数, 取得了一定的效果。但是由于该目标函数过于简单, 无法描述生成文本的真实质量, 往往导致生成重复、无意义等问题。

DG-GAN 使用对抗训练的方式, 使用判别器来区分句子是真实的还是机器生成的, 并将输出概率作为奖励传递给生成器, 期望生成器生成更加符合真实数据的句子。以此来缓解使用极大似然估计作为目标函数产生的问题。DG-GAN 的模型结构如图所示, 其本质是一种条件生成对抗网络 (Conditional Generative Adversarial Network)。主要体现在判别器的输入除了有生成器的输出以外, 还必须有生成器的输入。两者需要满足某种匹配, 判别器才会输出高的奖励。

DG-GAN 的生成器使用 Seq2Seq 模型实现, 在目标端使用 *softmax* 函数完成词语的生成。判别器使用层次化编码器实现, 作为一个二分类器输出二维概率, 分别指示该句子属于真实的概率  $Q_+(\{x, y\})$  和属于机器生成的概率  $Q_-(\{x, y\})$ 。使用强化学习中的策略梯度进行训练, 最大化来自判别器的奖励期望。如公式 9 所示。

$$J(\theta) = E_{y \sim p(y|x)} (Q_+(\{x, y\}) | \theta) \quad (9)$$

上述方法同样只针对一个完整的句子进行打分。DG-GAN 使用 REGS (reward for every generation step) 来提供及时奖励。第一种方式使用蒙特卡洛搜索来估计当前状态的价值, 该方法与 SeqGAN 类似, 这种方法由于采样过程需要消耗一定的时间。第二种方式为, 直接训练一个判别器, 可以为全部或者部分生成的句子打分, 第二种方法可以节省时间, 但是相比于第一种方法, 准确率较低。

在对抗训练的过程中, 生成器使用来自判别器的奖励进行参数更新。但是没有使用到真实的数据, 这样一旦生成器生成数据不稳定, 导致判别器打分很低, 就无法很好地更新生成器, 这种情况迭代下去, 导致训练无法收敛。因此 DG-GAN 使用 Teacher Forcing [Williams and Zipser, 1989] 的训练方式帮助训练。

<sup>1</sup>论文并没有将模型称之为 DG-GAN, 为了保持全文统一, 这里缩写为 DG-GAN (Dialogue Generation GAN)

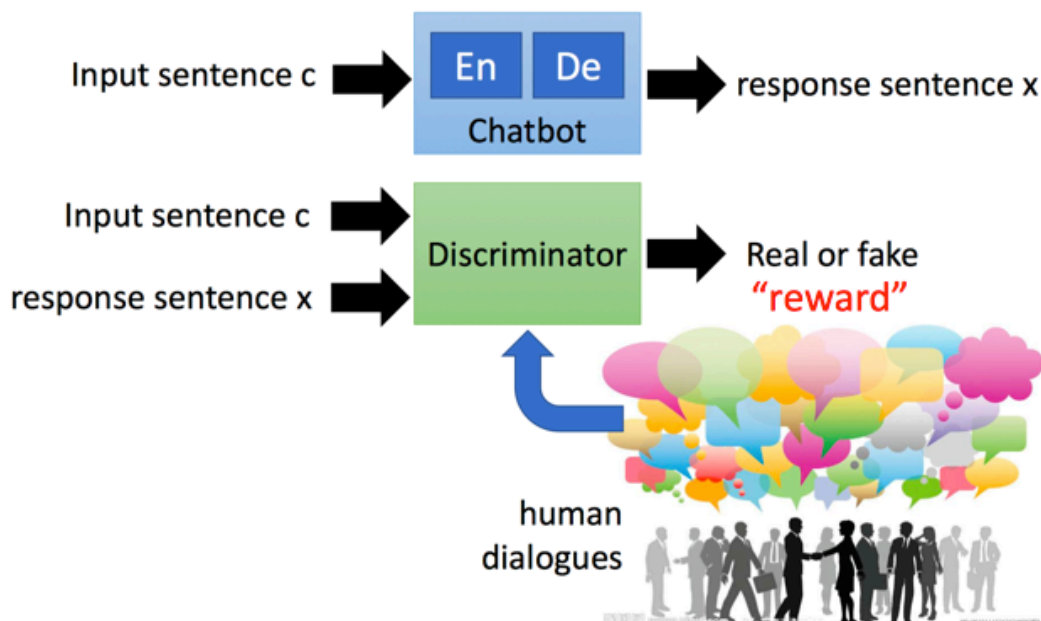


图 5: DG-GAN 结构图

## 5 LeakGAN

在对抗训练的方式下，判别器往往仅提供最终奖励给生成器用于参数更新。在长文本生成的情况下，一个标量的奖励信号由于信息不足和信息稀疏的问题，无法很好地指导生成器完成生成任务。LeakGAN[Guo et al., 2017] 将判别器的部分高层信息“泄露”给生成器，来帮助生成器更好的完成任务。该方式的出发点在于，判别器往往是人为设定的一个具体模型，例如卷积神经网络，而非一个“黑盒”系统，既然知道判别器的内在构造，因此可以使用判别器来提供更加丰富的信息。LeakGAN 的结构如图所示。

在 LeakGAN 中，判别器分为特征抽取器和 *softmax* 分类层两个部分，特征抽取器根据当前状态得到高层特征表示  $f_t$ ，并将该表示作为“泄露”信息提供给生成器。具体由一个卷积神经网络实现。 $f_t$  可以为网络高层的 feature map。生成器由层次化结构构成，分为 Manager 模块和 Worker 模块。Manager 模块接收判别器提供的“泄露”信息，在经过一定的变换后，提供给 Worker 模块完成生成任务。

以往在训练生成对抗网络时，更新生成器的参数主要依靠判别器的奖励结合强化学习实现。现在除了标量的奖励之外，在每一个时刻还可以额外得到判别器的高层特征表示，作为一种额外的“特征”帮助生成器来生成文本序列，通过这种方式，可以更好地指导长文本的生成。

## 6 MaskGAN

MaskGAN[Fedus et al., 2018] 由谷歌大脑提出。通过让生成器完成类似于完形填空的任务来提高文本生成的质量，输入是隐藏掉几个词语的一句话，输出一句补全的话，希望输出的词语尽可能的与真实词语相近。利用生成对抗网络来进行训练。序列到序列模型 (Seq2Seq) 往往使用极大似然估计和 Teacher Forcing 进行训练。但是在生成过程中，一旦出现错误，模型会基于错误的分布继续进行生成，这种问题被称之为“Exposure Bias”。因此 MaskGAN 的判别器同样使用 Seq2Seq 结构，在训练过程中模拟采样过程，来缓解上述问题。

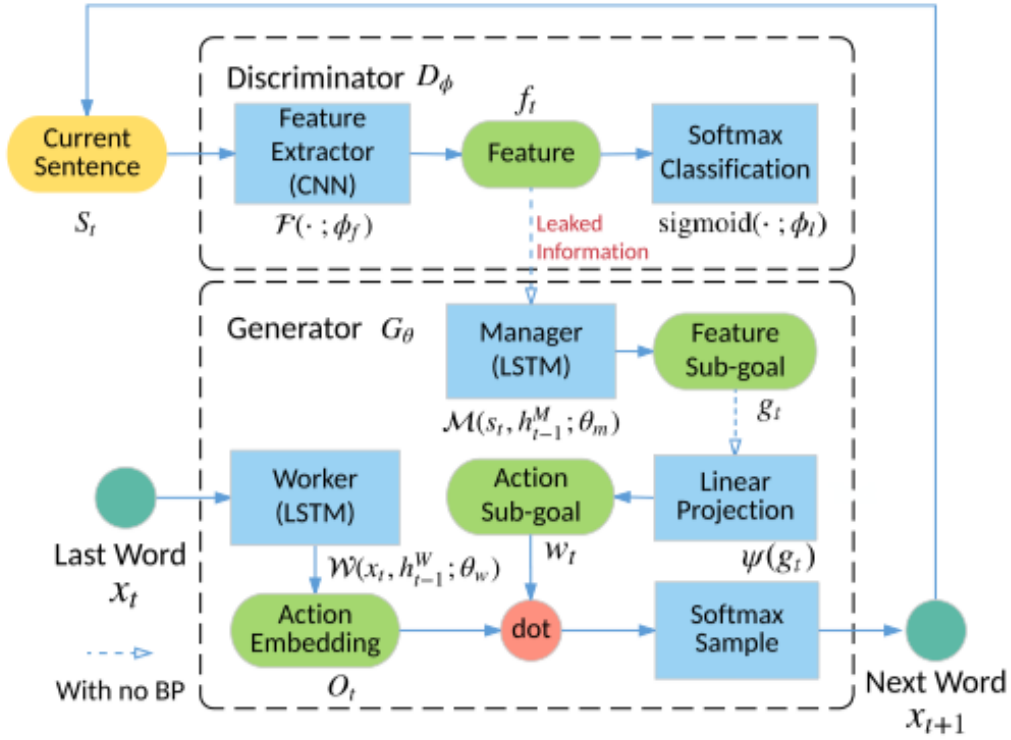


图 6: LeakGAN 结构图

生成对抗网络在训练过程中有两个典型的问题，训练不稳定（training instability）和模式下降（mode dropping），这两个问题在文本生成任务中更为明显。在文本数据中有着更复杂的模式，比如词组、短语等，然而普通生成对抗网络针对句子级别进行判别，导致生成器学习到的句子多样性较差。因此，MaskGAN 使得模型能够在词级别的生成上做判断。

MaskGAN 的生成器与判别器均为 Seq2Seq 模型，生成器输入一句经过隐藏处理的语句，输出补全之后的语句。判别器输入生成器的输入与输出，并在解码器部分的每一个位置输出标量概率，而不是在词表中进行选择。将这些概率作为奖励传递给生成器。生成器的结构如图 7 所示。

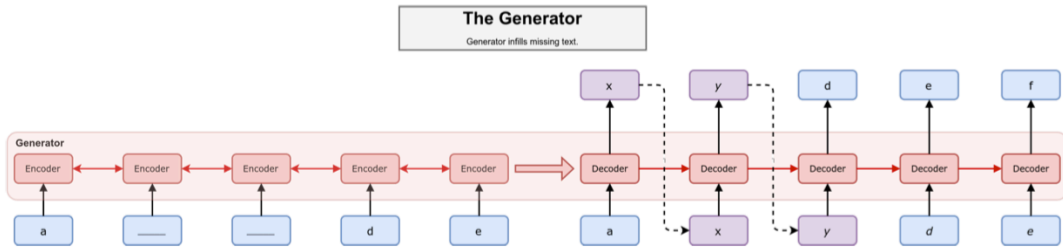


图 7: MaskGAN 生成器结构图

判别器在每一时刻的输出概率为公式 10 所示。

$$D_\phi(\tilde{x}_t | \tilde{x}_{0:T}, m(x)) = P(\tilde{x}_t = x_t^{\text{real}} | \tilde{x}_{0:T}, m(x)) \quad (10)$$

MaskGAN 更是一种训练方法，通过由 Seq2Seq 实现的判别器在每一时刻提供奖励，使得生成器的输出更加满足真实分布。



## 7 BR-CSGAN

BR-CSGAN[Yang et al., 2017] 将生成对抗网络应用于神经机器翻译任务。生成器将源语言句子翻译到目标语言，判别器基于源语言句子和翻译句子判断翻译句子是否更接近真实翻译。判别器被视为一种动态的目标函数，由于其输出会随着参数更新而改变，除此以外，BR-CSGAN 还使用句子级别的 BLEU 值作为强化学习的奖励。BR-CSGAN 的结构如图 8 所示。

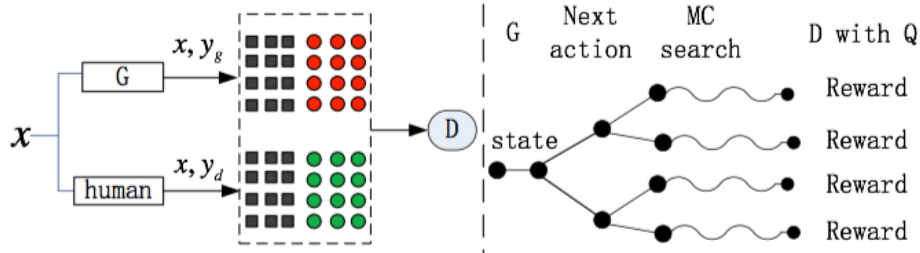


图 8: BR-CSGAN 结构图

BR-CSGAN 中，生成器使用 RNNSearch 和 Transformer 两种方式实现。判别器使用卷积神经网络实现。

## 8 DP-GAN

现有的文本生成方法往往倾向于生成一些重复的、无意义的文本。DP-GAN[Xu et al., 2018] (Diversity-Promoting Generative Adversarial Network) 通过区分重复的、无意义的文本和流畅的、有意义的文本使得生成器倾向于生成更加真实地文本。之前生成对抗网络中的判别器往往基于分类器实现，使用打分来表示句子是否来源于真实世界，但是该方法无法体现两个句子的新颖程度。不同于之前的工作，DP-GAN 的判别器由语言模型而非分类器实现，使用语言模型的输出，交叉熵作为奖励。DP-GAN 结构如图所示。

判别器使用语言模型实现，具体为一个单向的长短时神经网络。使用语言模型的输出交叉熵作为奖励。给定一个句子  $y_t$  第  $k$  个词语的奖励计算如公式 11 所示。判别器的奖励分为两个部分，句子级别的奖励和词语级别的奖励。句子级别的奖励为每个词语奖励的平均。

$$R(y_{t,k}) = -\log D_\phi(y_{t,k}|y_{t,<k}) \quad (11)$$

## 9 SentiGAN

生成对抗网络产生的文本往往存在质量差、缺乏多样性、模式下降等问题。为了解决上述问题，北京大学提出的 SentiGAN[Wang and Wan, 2018] 包含了多个生成器和一个判别器，将该网络用于不同情感文本生成。与之前的工作不同，SentiGAN 使用基于惩罚的目标函数而非基于奖励的目标函数。使得生成器可以生成更加多样化的句子。判别器为一个多分类器，使得每个生成器可以更加专注的生成相对应的情感文本。

假设一共有  $k$  类情感，SentiGAN 使用  $k$  个生成器和一个判别器。第  $i$  个生成器的目标是生成具有第  $i$  类情感类型的文本，尽可能欺骗判别器。判别器的目标是区分生成文本和  $k$  类情感真实文

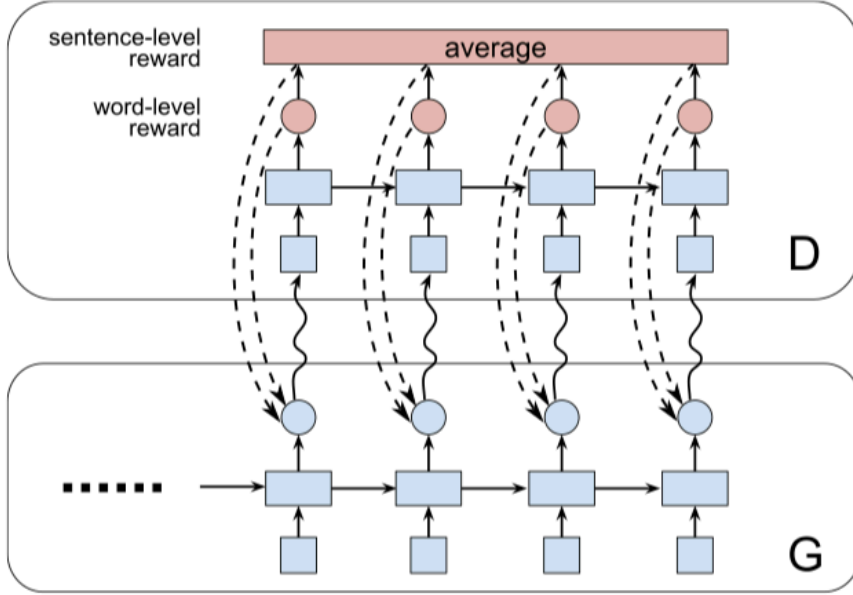


图 9: DP-GAN 结构图

本。SentiGAN 的结构如图 10所示。

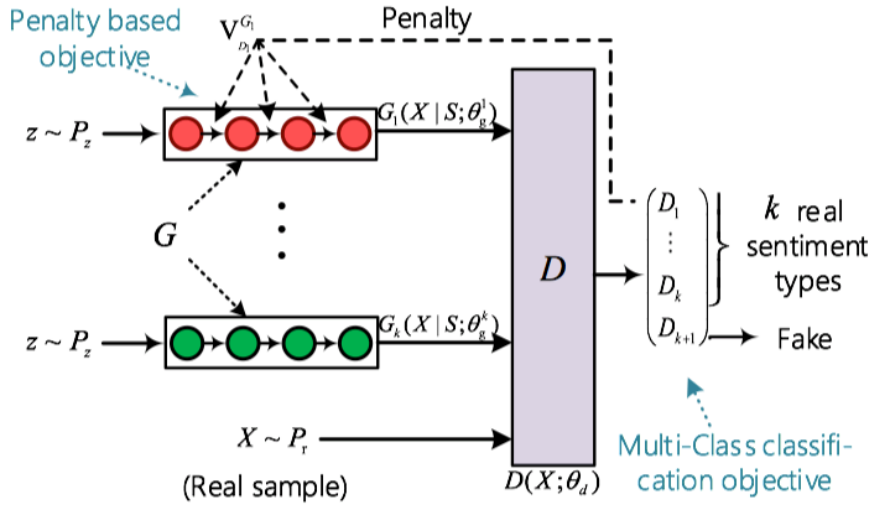


图 10: SentiGAN 结构图

SentiGAN 生成器使用基于惩罚的函数进行参数更新。其与 GAN[Goodfellow et al., 2014] 和 SeqGAN[Yu et al., 2017] 的区别如公式 12所示。

$$J_G(X) = \begin{cases} E_{X \sim P_g} [-\log(D(X; \theta_d))] & \text{GAN} \\ E_{X \sim P_g} [-\log(G(X|S; \theta_g)D(X; \theta_d))] & \text{SeqGAN} \\ E_{X \sim P_g} [G(X|S; \theta_g)V(X)] & \text{SentiGAN} \end{cases} \quad (12)$$

基于惩罚的目标函数提升主要有两个方面。首先，该目标函数可以被视为对 wasserstein 距离的衡量，可以提供有意义的梯度，而另外两个损失函数则不能做到这一点。其次，使用损失项而不是奖励项。



## 10 RankGAN

RankGAN[Lin et al., 2017] 认为判别器的二分类器对于生成多样、符合现实逻辑的文本是不够的。其通过让判别器对于一个由人类书写和机器生成构成 Reference 集中的句子进行排序，来指导生成器的生成。借鉴信息检索中的相似度的思想，其首先计算两个句子的 *cosine* 相似度，然后据此给出一个语料集上的排序分。RankGAN 结构如图 11所示。

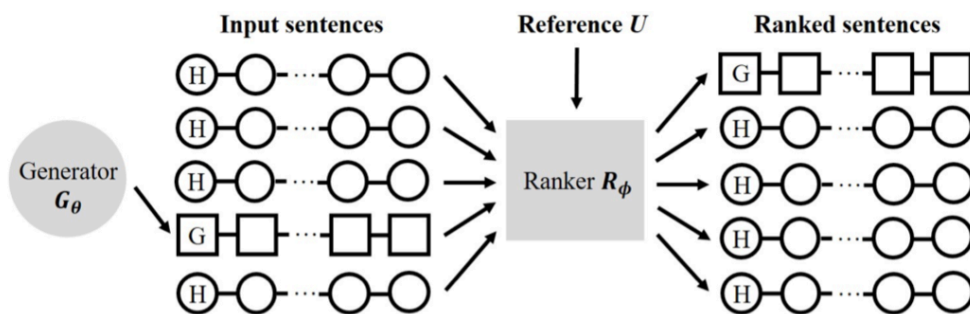


图 11: RankGAN 结构图

RankGAN 的核心就是用一个排序器来替代判别器，以提供更好地生成句子的评估，进而帮助生成器生成更为真实的句子。

## 11 ORGAN

ORGAN[Guimaraes et al., 2017] 基于 SeqGAN[Yu et al., 2017]，使用强化学习的方法，将领域特定的目标作为奖励的一部分。为了缓解 GAN 训练不稳定的问题，ORGAN 采用 Wasserstein 距离 [Arjovsky et al., 2017] 作为判别器的损失函数。

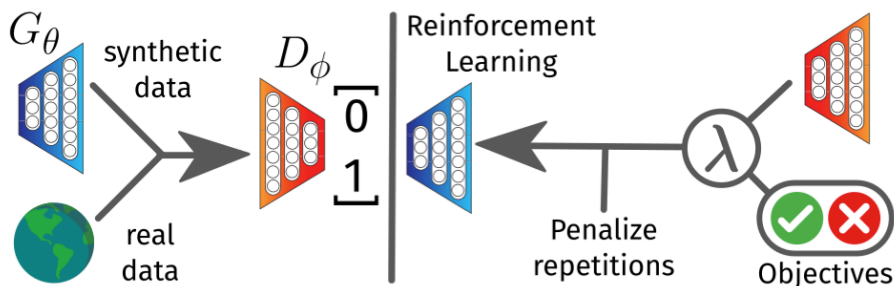


图 12: ORGAN 结构图

## 12 总结

从 SeqGAN 开始，结合强化学习等方法，一定程度上缓解了由于文本离散化输出造成的梯度回传问题。使用对抗训练的方法，可以有效的缓解序列到序列模型基于极大似然估计目标函数训练造成的生成重复、简短、无意义等问题，通过判别器来指导生成器的参数更新。但是单一的标量奖励信号容易导致信息稀疏和信息不全问题，现在越来越多的方法开始考虑充分利用判别器可以提供

的信息，来为生成器提供更加多样化的奖励。生成对抗网络已经开始越来越多的应用在文本生成领域，包括神经机器翻译、文本摘要、开放域对话生成等等，利用判别器来为生成器提供更加丰富、任务特定的奖励将会是未来的一个趋势。

## 参考文献

- Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
- William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation via filling in the \_\_. *arXiv preprint arXiv:1801.07736*, 2018.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, and Alán Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.
- Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. *arXiv preprint arXiv:1709.08624*, 2017.
- Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.
- Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.
- Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. Adversarial ranking for language generation. In *Advances in Neural Information Processing Systems*, pages 3155–3165, 2017.
- Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- Ke Wang and Xiaojun Wan. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452, 2018.
- Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.
- Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.
- Jingjing Xu, Xuancheng Ren, Junyang Lin, and Xu Sun. Diversity-promoting gan: A cross-entropy based generative adversarial network for diversified text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3940–3949, 2018.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. Improving neural machine translation with conditional sequence generative adversarial nets. *arXiv preprint arXiv:1703.04887*, 2017.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. 2017.