

摘要

本文系统地介绍了神经文本生成模型的最新发展。具体来说，我们从循环神经网络语言模型和传统的极大似然估计训练方案出发，指出其在文本生成方面的不足。因此，我们介绍了最近提出的基于强化学习、重参数化技巧和生成对抗网(GAN)技术的文本生成方法。我们比较了这些模型的不同性质以及相应的处理它们常见问题如梯度消失和生成多样性的技术。最后，我们在两个知名的数据集上使用不同类型的神经文本生成模型进行了基准测试实验，并讨论了经验结果以及上述模型的属性。

1. 引言

自然语言处理(NLP)问题，特别是自然语言生成(NLG)，长期以来一直被认为是最具挑战性的计算任务之一[Murty 和 Kabadi, 1987]。NLG 技术作为关键模块被广泛应用于各种任务中，包括无控制的句子或诗歌生成[Zhang and Lapata, 2014]和输入条件文本生成如图像字幕[Karpathy and feifei, 2015]和情绪/时态控制的句子生成[Hu et al., 2017]等。由于一些原因，这个问题具有挑战性。一般来说，这些任务的输入和输出之间存在着信息的不平衡[Shapiro, 1992]，特别是在非文本输入的情况下。输入的语义通常是明确和明确的，而自然语言的组成部分往往是模糊的。这一事实迫使神经文本生成(NTG)模型找到目标语言的公共模式，并通过构造适当的上下文消除歧义来表达输入信息。在这个过程中，主要有两个困难。其一是自然语言的语法复杂性。另一个是关于输入信息提取、简化和转换过程中的难点。后一种方法过于具体，因此不是本文的重点。为了解决语法问题，研究人员开发了一般的方法来构建复杂的基于知识的系统，如在[Reiter 和 Dale, 2000]中所讨论的。需要指出的是，尽管这种范式需要大量的人工努力，但它仍然广泛应用于今天的许多商业产品中，因为如果设计得好，它是可解释的和健壮的。

近十年来，神经网络及其变体在许多任务中都显示出了良好的应用前景。对于文本生成，首先提出了神经网络语言模型(NNLM) [Bengio 等人, 2003]，利用神经网络在文本生成任务中的优势。NNLM 可以看作是 n-gram 范式的直接扩展，具有神经网络的泛化能力。给定 ground truth 序列 $s_n = [x_0, x_1, \dots, x_{n-1}]$ ， θ -参数化语言模型 $G_\theta(x|\text{context}) = \hat{P}(x|\text{context})$ (下同)，典型的 NNLM 采用近似为

$$\hat{P}(x_t|\text{context}) \approx P(x_t|x_{t-n+1}, x_{t-n+2}, \dots, x_{t-1}). \quad (1)$$

然而，根据之前的一些批评，从理论上讲，n-gram 范式不可能捕获长期依赖关系[Rosenfeld, 2000]。为了解决这一问题，开发了循环神经网络语言模型(RNNLM) [Mikolov 等人, 2010]，它是具有马尔可夫性质的语言模型的更一般的实现。典型的 RNNLM 使用循环神经网络(RNN)自回归地将先前的变长输入编码为一个“隐藏”向量，然后在推理下一个令牌时使用该向量。本程序可表述为

$$\hat{P}(x_t|\text{context}) \approx P(x_t|\text{RNN}(x_0, x_1, \dots, x_{t-1})). \quad (2)$$

自 RNNLM 以来，神经文本生成已经得到了很好的研究。例如，通过采用 RNN 的改进变量，如长短期记忆(LSTM) [Hochreiter 和 Schmidhuber, 1997]和门控循环单元(GRU)

[Choi 等人, 2014], RNNLMs 在捕获文本中的长期依赖关系方面显示出可接受的结果。然而, 正如[Bengio et al., 2015]所指出的, 由于暴露偏倚的存在, 拟合观测数据的分布并不意味着生成满意的文本。后来, 人们提出了各种各样的解决方案, 包括基于强化学习(RL)的模型, 生成式对抗网(GANs) [Goodfellow 等人, 2014;Yu 等人, 2017]框架和端到端重新参数化技术[Kusner 和 Hernández-Lobato, 2016]。

在本文中, 我们对最近提出的神经文本生成(NTG)模型进行了系统的研究。我们详细讨论了这些模型的不同性质以及相应的处理方法, 如训练过程中的梯度消失和生成多样性。与以往主要研究序列到序列(sequence-to-sequence, Seq2Seq)模型的工作[Xie, 2017]相比, 本文更关注最近提出的基于 RL 和 GAN 的方法, 而 Seq2Seq 是基本 MLE 方法的一种特殊情况。最后, 我们在两个知名的数据集上使用不同类型的神经文本生成模型进行了基准测试实验, 并讨论了经验结果以及上述模型的属性。希望本文能为该领域的进一步研究提供有益的方向。

2. RLNM 的训练范式研究

在本节中, 结合时间顺序和技术分类的介绍, 我们主要介绍三种 RNNLM 的训练模式, 即监督学习、强化学习技术和对抗训练方案。

2.1 有监督学习的 NTG

尽管文本生成实际上是一种无监督的学习任务, 但确实存在一些监督度量, 它们在某些约束条件下很好地近似于基本事实。这些算法专注于直接优化一些监督指标。其中一些可能包括一些有用的技巧, 以帮助缓解一些特定的问题。

最大似然估计

通常, 经典的神经语言模型是通过最大似然估计(MLE, 又称教师强迫)来训练的 [Williams 和 Zipser, 1989]。对于 RNNLM, MLE 是一种自然的方法, 它将生成问题看作是一个顺序的多标签分类, 然后直接优化多标签交叉熵。通过 MLE 训练的语言模型 G_{θ} 的目标可以表述为

$$J_{\theta}(s_n) = - \sum_{t=0}^{n-1} \log \hat{P}(x_t | s_t), \quad (3)$$

其中 s_0 表示空字符串。

值得注意的是, 到目前为止, 大多数应用最先进的 NTG 模型都采用 MLE 作为他们的训练目标[Karpathy 和 feifei - fei, 2015;胡等, 2017]。与其他算法相比, MLE 具有更好的收敛速度和训练鲁棒性。

但从理论上讲, MLE 存在所谓的暴露偏差[Huszár, 2015], 这是由于 MLE 训练的语言模型在训练阶段和推理阶段存在内在的差异。也就是说, 对语言模型进行训练, 在训练时间内生成适当的给定 ground truth 前缀的后续 token。然而, 在自由生成的推理阶段, 模型需要进行预测随后给了一个生成的前缀。当前缀与训练数据中的前缀稍有不同时, 不能保证模型仍能正常运行。暴露偏差的影响随着序列的延长而变得更加明显和严重, 使得 MLE 模型在长文本生成任务中的应用变得不那么有效。

安排抽样

schedule Sampling (SS) [Bengio et al., 2015]是为了缓解这一问题而提出的。它引入了一个随机变量 ϵ 调和监督强迫与自由生成，弥合训练阶段与推理阶段的差异。在 SS 培训过程的每一步中，对 ϵ 决定模型是执行监督强制还是自由生成。SS 在直观上看起来是合理的，并且与普通 MLE 相比确实有明显的改善。但很快就被证明是前后矛盾的[Huszár, 2015]。

2.2 NTG 与强化学习

利用 RNNLMs 生成文本可以看作是一个马尔可夫决策过程(MDP)，其局部最优策略可以通过强化学习(RL)找到[Sutton 和 Barto, 1998]。

PG-BLEU

一个简单的想法是使用 RL 策略梯度算法(例如，REINFORCE) [Sutton 等人，1999]来优化一些不可微指标。一个经典的选择是优化 BLEU [Papineni 等人，2002]，一个用于语言模型评估的 n-gram 级度量。这个想法就是 PG-BLEU。PG-BLEU 通过 REINFORCE 优化 BLEU，其最小化目标可以表述为

$$J_{\theta}(\hat{s}_n) = - \sum_{t=0}^{n-1} R_t \log \hat{P}(x_t | s_t), \quad (4)$$

其中 $\hat{s}_n \sim G_{\theta}(\cdot)$ 是从生成器 G_{θ} 中采样的完整序列， $R_t = E_{s \sim G_{\theta}(\cdot | s_{t+1})}[\text{BLEU}(s)]$ 是给定前缀 s_{t+1} 和随后的生成策略 G_{θ} 下的期望 BLEU 得分。

该算法存在一些问题。首先，BLEU 在计算上并不便宜，这使得 PGBLEU 在实践中几乎毫无用处。其次，BLEU 不是一个完美的度量，甚至不是一个强大的度量，因为它只是计算生成的文本和参考文本(语料库)之间的 n-gram 统计相似性。因此，它给模型引入了很多不必要的偏差。

MLE 的思考

从 RL 的角度来看，MLE 可以看作是策略外的模仿学习，从重放缓冲区中采样的片段由地面真实数据组成。在训练过程的每一步中，式(11)中的奖励 R_t 固定为 1.0。由于模仿学习在许多 RL 场景中也有助于减少方差，这解释了为什么像 PG-BLEU 这样的算法通常包含使用 MLE 进行预训练[Yu et al., 2017]。

2.3 通过对抗训练改善 NTG

生成式对抗网(GANs)的成功[Goodfellow et al., 2014]启发了研究者通过对抗训练来解决问题。

专业强迫:对抗训练作为正规化

作为一项早期工作，教授强迫[Lamb 等人，2016]使用对抗模块作为一种正规化机制，缩小监督强迫和自由生成程序之间的差距。Lamb 等人声称这将有助于减轻暴露偏差。

表示发生器 G_{θ} 的参数为 θ ，鉴别器 D_{ϕ} 的参数为 ϕ 。用输入 x_t, s_t 表示连接的隐藏状态和存储单元为 $B(x_t, s_t)$ 。教授强迫可以表示为：

$$\text{NLL}(\theta) = \mathbb{E}_{(x_t, s_t) \sim p_{\text{data}}} [-\log \hat{P}(x_t | s_t)] \quad (5)$$

$$C_f(\theta | \phi) = \mathbb{E}_{s_t \sim p_{\text{data}}, \hat{x}_t \sim G_\theta(x | s_t)} [-\log(D_\phi(B(\hat{x}_t, s_t)))] \quad (6)$$

$$C_t(\theta | \phi) = \mathbb{E}_{(x_t, s_t) \sim p_{\text{data}}} [-\log(1 - D_\phi(B(x_t, s_t)))] \quad (7)$$

$$J_G(\theta) = \text{NLL}(\theta) + C_f(\theta | \phi) + (\text{optional}) C_t(\theta | \phi) \quad (8)$$

$$J_D(\phi) = \mathbb{E}_{s_t \sim p_{\text{data}}, \hat{x}_t \sim G_\theta(x | s_t)} [-\log(1 - D_\phi(B(\hat{x}_t, s_t)))] \\ + \mathbb{E}_{(x_t, s_t) \sim p_{\text{data}}} [-\log(D_\phi(B(x_t, s_t)))] \quad (9)$$

在训练过程中，对模型参数进行更新，使方程。(8)和(9)交替优化。

序列生成对抗网络:对抗强化学习

更直接、更自然的方法是序列生成对抗网络 (Sequence Generative Adversarial Network, SeqGAN) [Yu 等人, 2017]。SeqGAN 由两部分组成。一种是发生器，通常采用 θ 参数化的 RNNLM G_θ 来实现。另一种是判别器，它是一个由 ϕ 参数化的二分类器 D_ϕ ，训练它将生成的序列与地面真值区分开来。SeqGAN 使用 REINFORCE (policy gradient) 对原 GAN 目标进行优化：

$$\min_{\theta} \max_{\phi} \mathbb{E}_{s \sim p_{\text{data}}} [\log(D_\phi(s))] + \mathbb{E}_{s \sim G_\theta(\cdot)} [\log(1 - D_\phi(s))] \quad (10)$$

SeqGAN 的目标可以表述为

$$J_\theta(\hat{s}_n) = - \sum_{t=0}^{n-1} R_t \log \hat{P}(x_t | s_t), \quad (11)$$

其中 $\hat{s}_n \sim G_\theta(\cdot)$ 为从生成器中采样的完整序列，如式(11)所示，期望回报 $R_t = \mathbb{E}_{s \sim G_\theta(\cdot | s_{t+1})} [D_\phi(s)]$ 是基于对鉴别器的评价。在实践中，如 SeqGAN [Yu et al., 2017] 所述， R_t 可以通过蒙特卡罗搜索来估计。SeqGAN 显示了很有前景的实证结果，并有一系列后续工作 [Guo et al., 2017; Fedus 等, 2018]。

关于 SeqGAN 的进一步讨论

对抗学习起源于 GAN [Goodfellow et al., 2014]，最近发展成为一种新的无监督学习范式。对抗学习在许多非监督学习任务中显示出了很有前景的结果，如互信息消除 [Liu 等人, 2017] 和超分辨率 [Ledig 等人, 2016]。另一个有趣的例子是生成式对抗模仿学习 (generative adversarial imitation learning) [Ho and Ermon, 2016]，其目的是通过观察 ground-truth 事件来提供对环境的估计，从而训练 agent 去模仿所提供的 ground truth 的行为模式。SeqGAN 中也存在这样的过程。

Ranking List		Case A (Gradient Vanishes)	Case B (No Gradient Vanishing)
Generated Samples	I have a pen which writes well.	1.0	0.9
	I have a pen.	0.99999	0.6
	I have pen a.	1e-10	0.5
	pen have a I am.	1e-15	0.4
	pen a pen pen I am.	1e-20	0.3

图 1:不同的分数列表可能会导致相同的排名表

在对抗性训练过程中，鉴别器逐步学习估计环境，这实际上是当前代理不能很好执行的真实环境的残余。然而，**discriminator** 的输出值没有任何可见的物理意义。解决这个问题一个尝试是对鉴别器执行 **Lipschitz** 约束，就像 **Wasserstein GAN** [Arjovsky 等人，2017]及其改进的变体[Gulrajani 等人，2017]。

尽管有这些优势，**SeqGAN** 仍然存在两个主要问题。一个是梯度消失问题，这意味着当鉴别器被训练得比生成器强得多时，生成器将很难有任何实际的更新，因为生成器的任何输出实例都会被评分为接近 0。这可能会导致训练在达到真正的收敛或纳什均衡之前过早地停止。另一个是由 **REINFORCE** 算法引起的模态坍塌问题，它增加了采样特定令牌的估计概率，使鉴别器对其评价较高。因此，生成器只能模拟目标分布的一个有限部分，这大大降低了输出的多样性。这些问题激发了对 **SeqGAN** 的进一步改进研究。提出了几种扩展的变体。

关于消失梯度的缓解

一个可观察到的事实是，不同的分数列表可能会导致相同的排名列表，如图 1 所示。从图中可以看出，虽然 **case A** 和 **case B** 两个分类器的 **AUC** 分数相同，但是对于通过 **policy gradient** 训练的生成模型来说，**case B** 提供的奖励使得模型收敛的更快更稳定。从直观上看，这是奇怪的，因为一个训练得更好的鉴别器应该能够更好地估计数据的潜在分布，而不是相反。

要解决这个问题，基本上有两种主要的方法。第一种方法是使用调整后的分数作为奖励信号。最大似然增强离散生成对抗网(**MaliGAN**) [Che et al., 2017]是由该方法衍生出来的典型工作。它将从鉴别器获得的分数作为奖励

$$R(s) = \frac{D(s)}{1 - D(s)} . \quad (12)$$

此外，为了加快收敛速度，**MaliGAN** 包括奖励基线方法，即保持一个运行平均值和计算出奖励的方差 $R(s)$ ，然后用它们对奖励进行线性缩放。实验结果表明，该方法不仅缓解了梯度消失问题，而且在一定程度上改善了模型的多样性。

第二种方法是用排名分数代替二元分类分数。语言生成的对抗性排序 [Lin et al., 2017] (RankGAN) 提出了一种对抗性文本生成模型，将该模型的判别器替换为对抗性排序器 S_ϕ ，训练 S_ϕ 来优化成对排序损失：

$$J_\phi(\cdot) = \mathbb{E}_{s \sim p_{\text{data}}} [\log S_\phi(s|U, C^-)] - \mathbb{E}_{s \sim G_\theta} [\log S_\phi(s|U, C^+)] \quad (13)$$

其中，

$$S_\phi(s|U, C) = \frac{\exp(\gamma \alpha(s|u))}{\sum_{s' \in C} \exp(\gamma \alpha(s'|u))}$$

$$\alpha(s|u) = \cos(y_s, y_u),$$

$u \sim U$, U 是参考集， C^+ 是来自数据集的序列， C^- 是生成序列。 γ 是可以作为超参数调节的逆玻耳兹曼温度因子。在许多情况下，RankGAN 在提高 SeqGAN 的收敛性能方面表现出了良好的效果。但是，由于它需要对原始数据进行额外的采样，因此其计算成本也高于其他模型。

受这两种想法的启发，研究者提出了 Bootstrapped Ranking Activation (BRA) [Guo et al., 2017]。BRA 不需要修改 SeqGAN 的架构。它只是使用每一批中的排名信息来调整奖励，可以用公式表示

$$R^t(\cdot) = \sigma\left(\delta \cdot \left(0.5 - \frac{\text{rank}(i)}{B}\right)\right), \quad (14)$$

式中 $\text{rank}(i)$ 表示第 i 个序列在批次中的排序由高到低， δ 为激活平滑度超参数， B 为批次大小， $\sigma(\cdot)$ 为非线性函数，将原来的等差分布进行了重新设计，使 BRA 更有效、更普遍。在 LeakGAN [Guo et al., 2017] 的实验中，研究者选择了 sigmoid 函数为 $\sigma(\cdot)$ ，这在我们的实验中也实际证明是有效的。BRA 不需要太多的额外计算，并且易于纳入其他模型，是一种具有竞争力的梯度稳定器。

关于多样性的增强

模式崩溃问题的一个特殊情况是 SeqGAN 总是倾向于生成较短的序列，因为这些序列很容易被学习，从而获得较高的分数。为了增强对长期依赖的建模能力，提出了 LeakGAN [Guo et al., 2017] 等分层方法。LeakGAN 在提高长文本生成鲁棒性方面已经显示出了很有希望的结果，然而，根据我们的实验，引入了其他类型的模式崩溃。

同时，受 Wasserstein GAN [Arjovsky et al., 2017] 等 GAN 变体的启发，提出了多样性促进 GAN (DPGAN) [Xu et al., 2018] 来缓解崩溃模式问题。DPGAN 使用对抗语言模型 $D_\phi(x_t|s_t)$ 来替代原来的二分类器 $D_\phi(s)$ 。DPGAN 没有使用饱和二元分类分数 $D_\phi(s)$ ，而是设计了一种层次奖励机制优化非饱和目标，即对抗语言模型 $D_\phi(x_t|s_t)$ 估计的对抗 NLL。

Re-parametrization

除了基于 RL 的方法外，也有人尝试将重参数化技巧应用到 RNNLMs 中，以绕过离散令牌的梯度计算问题。一个典型的例子是，使用 Gumbel softmax trick 生成文本的 GAN [Kusner and Hernández-Lobato, 2016] 使用了 Gumbel 分布来避免显式采样，使得通过反向传播进行联合训练成为可能。自然语言的对抗性生成 [Rajeswar et al., 2017] 通过在每一步中解码一个高斯噪声来引入随机性，以避免显式采样。但根据我们的实验，通过重参数化实现的上述工作的可靠性明显低于 SeqGAN 变体。生成的文本很少可读，并且具有严重的模式崩溃。这种问题是由一个标杆平台——Texygen [Zhu et al., 2018] 揭示出来的，该平台进行了标准模型设定，并对不同的 NTG 模型进行了公平的比较。

其他方法

MaskGAN [Fedus et al., 2018] 是第一个通过序列到序列(Seq2Seq)学习的无条件生成模型。MaskGAN 的基本版本与计划采样(SS)有类似的想法[Bengio 等人, 2015]，但通过对抗性训练来解决 SS 的不一致性。然而，通过其他有用的附加功能，如注意机制[Bahdanau 等人, 2014]，MaskGAN 有潜力远远超越 SS 的性能。

文本生成对抗特征匹配(TextGAN) [Zhang et al., 2017] 通过最小化生成文本的对抗特征估计的重建成本，引入对抗学习。

2.4 论 rnnlm 的局限性及其超越

尽管上面提到了成功的发展，但目前大多数方法都有一些共同的特性，在某些情况下，这些特性限制了它们的有效性。

首先，它们都遵循将文本生成问题视为顺序分类过程的 RNNLM 公式，该公式的每一步都具有形式

$$s_{t-1} \rightarrow s_t | x_t. \quad (15)$$

需要注意的是，这种产生规则的马尔氏范式表明，虽然 LSTM 等 RNN 变体被证明是图林完备的[Siegelmann and Sontag, 1995]，但采用 Eq.(15) 的语言模型仍然可以看作是正规语言的广义版本。在乔姆斯基层次结构的四种语言中，正则语言是最简单的一种，如果有超出规则语法能力的情况发生，则 rnnlm 在建模自然语言时的所有有效性都将依赖于 RNNs 的泛化能力。神经网络的泛化能力并不总是可靠的，这使得这些模型很难得到很好的训练。此外，由于每一步从多项分布中采样引入了随机性[Y u 等人, 2017]，该过程中无法获得显式潜码。

另一方面，一些有效的体系结构，如深度卷积神经网络(DCNN 或简称 CNN) [LeCun 等人, 1998] 在神经文本生成方面还没有得到有效的研究，特别是在无条件文本生成方面。然而，正如 WGAN-GP 的论文[Gulrajani et al., 2017] 所示，CNN 可以通过对抗式训练来应用于这个任务。

出于几个原因，我们特别期待在这方面取得突破。首先，cnn 利用语言的自然局部性，同时处理长期依赖关系，将全局依赖关系转换为网络上层的局部依赖关系。这就像通过上下文无关的语法来解析语言的逆向过程，这在许多经典的 NLP 任务中已经被证明是有用的。第二，cnn 是并行友好的。神经网络的训练、收敛和推理通常被认为比神经网络快十倍以上，这对于实际应用尤为重要。

3. 实证研究

在本节中，我们对典型的神经文本生成模型进行了实证研究。其中大多数都是基于 Texygen [Zhu 等人，2018] 进行评估的，这是一个基准测试平台，特别适用于文本生成任务，有许多执行良好的基线模型和不同的评估指标。除了 Texygen 已经集成的模型外，我们还使用作者提供的程序对 MaskGAN 进行了评估。

3.1 数据集

Image COCO3 数据集被提议用于图像标题任务。在实验中，我们只使用它的图像标题注释，其中我们采样了 10,000 个句子作为训练集，另外 10,000 个句子作为测试集。它包含 4682 个不同的单词，句子的最大长度为 37。本数据集中的句子具有相对简短的句型。

EMNLP2017 WMT News4 数据集包含新闻文章句子。考虑到大多数句子包含利基词，我们只保留包含最常用的 5700 个词的句子。经过预处理，我们选择 20 万句作为训练集，10,000 句作为测试集。最大句子长度为 51，可视为一个较长的文本生成数据集。

3.2 度量

BLEU [Papineni et al., 2002] 和 NLLtest [Zhu et al., 2018] 用于评估文档之间的相似性或生成器对真实数据的拟合能力。此外，Self-BLEU [Zhu et al., 2018] 计算生成的句子之间的 BLEU 得分，用于监测模式崩溃的严重程度。

3.3 培训细节

基准线

实验中选择 SeqGAN、RankGAN、MaliGAN、TextGAN、LeakGAN 和 MaskGAN 作为对比算法。此外，还加入了标准 MLE 的结果作为参考。

表 1：对 Image COCO 测试数据进行 BLEU 评分

Models	BLEU2	BLEU3	BLEU4	BLEU5
SeqGAN	0.745	0.498	0.294	0.180
MaliGAN	0.673	0.432	0.257	0.159
RankGAN	0.743	0.467	0.264	0.156
LeakGAN	0.744	0.517	0.327	0.205
MaskGAN	0.539	0.328	0.209	0.143
TextGAN	0.593	0.463	0.277	0.207
MLE	0.731	0.497	0.305	0.189

表 2：EMNLP2017 WMT 测试数据的 BLEU 评分

	BLEU2	BLEU3	BLEU4	BLEU5
SeqGAN	0.724	0.416	0.178	0.086
MaliGAN	0.755	0.436	0.168	0.077
RankGAN	0.686	0.387	0.178	0.086
LeakGAN	0.835	0.648	0.437	0.271
MaskGAN	0.265	0.165	0.094	0.057
TextGAN	0.205	0.173	0.153	0.133
MLE	0.771	0.481	0.249	0.133

我们也对 PG-BLEU 进行了实验，但在实践中，我们发现使用 PG-BLEU 训练的模型生成的文本具有单一的模式，这是因为模型在策略梯度训练过程中容易收敛于局部最优。因此 PGBLEU 的结果不包括在本节中。

实验参数

实验中，所有 GAN 模型的参数都按照标准的高斯分布 $N(0,1)$ 进行初始化。在进行对抗训练之前，我们先使用 80 个 epoch 的 MLE 训练对每个模型的生成器和鉴别器进行预训练，然后再进行 100 个 epoch 的对抗训练。在 LeakGAN 的训练过程中，我们采用了作者提出的交错训练方案，每隔 10 个敌对 epoch 进行 5 个 MLE epoch 的训练。

3.4 实验结果

Image COCO 和 EMNLP2017 WMT 测试数据的 BLEU 评分分别如表 1 和表 2 所示。LeakGAN 在这个度量中显示了很大的优势，特别是当任务是长文本生成时。在其他模型中，SeqGAN 在生成短文本时性能优于其他模型，而 MaliGAN 在生成长文本时稍有不足，但性能相当好。MaskGAN 和 TextGAN 在这个指标上表现不佳。

图 2 和图 3 描述了训练过程中的 NLLtest 曲线。垂直虚线表示训练前过程的结束。由于 NLLtest 无法对 MaskGAN 进行评估，因此在此部分中不包含 MaskGAN。由于 MLE 直接对 NLLtest(对训练数据)进行了优化，所以除 LeakGAN 外，通常在训练前结束时都取得了最好的成绩。LeakGAN 的 NLLtest loss 在 pre-training 结束时达到最小值后，趋于收敛到更低的值，这可能是其交错训练过程的结果。TextGAN 在所有模型中提供了最低的性能，因为它的训练目标不是关于可能性而是关于特征分布距离。

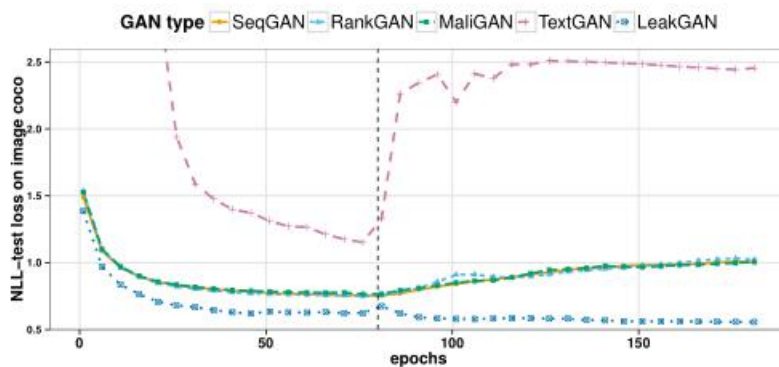


图 2: 图像 COCO 的 nll 测试损失

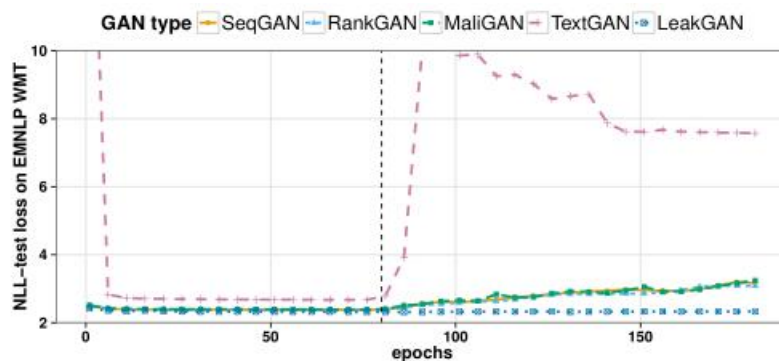


图 3: nll 测试损失在 EMNLP2017 WMT News 上

表 3: Self-BLEU 在 Image COCO 上的得分

Models	BLEU2	BLEU3	BLEU4	BLEU5
SeqGAN	0.950	0.840	0.670	0.489
MaliGAN	0.918	0.781	0.606	0.437
RankGAN	0.959	0.882	0.762	0.618
LeakGAN	0.934	0.818	0.663	0.510
MaskGAN	0.752	0.516	0.378	0.293
TextGAN	0.942	0.931	0.804	0.746
MLE	0.916	0.769	0.583	0.408

其他模型在训练前的学习曲线几乎相同，因为它们在这个过程中都使用了标准的 MLE。

Self-BLEU 得分如表 3 和表 4 所示。我们可以观察到，当生成短文本时，MaskGAN 具有最不严重的模式崩溃，正如作者在论文中所声称的那样。另一方面，TextGAN 存在非常严重的模式折叠问题，特别是当训练集是长文本时。考虑到当 n-gram 较大时，TextGAN 的 BLEU 分数较高，这可能是因为它模态崩溃，这意味着它会大量重复地生成高频短语。在其他模型中，MaliGAN 具有最低水平的模式崩溃，但当涉及到生成长文本时，它的优势就会减弱。

4. 总结

本文概述了经典的和最近提出的神经文本生成模型。本文从监督学习、强化学习和对抗训练三种训练模式详细讨论了神经网络神经网络的发展。有监督学习方法是最广泛采用的一种学习方法

表 4: 在 EMNLP2017 WMT News 上的 Self-BLEU 得分

Models	BLEU2	BLEU3	BLEU4	BLEU5
SeqGAN	0.907	0.704	0.463	0.265
MaliGAN	0.909	0.718	0.470	0.252
RankGAN	0.897	0.677	0.448	0.298
LeakGAN	0.938	0.821	0.668	0.510
MaskGAN	0.448	0.244	0.140	0.091
TextGAN	0.999	0.975	0.967	0.962
MLE	0.851	0.572	0.316	0.171

但它们可能会导致暴露偏差问题。基于 RLbased 和对抗性训练方法可以解决暴露偏差问题，但通常存在梯度消失和模式崩溃问题。因此，各种技术，包括奖励缩放和分层架构，被提出来缓解这些问题。本文还对基于最大似然法和基于最小似然法的模型提出了统一的观点，这也解释了为什么在基于最小似然法的模型中，通常需要使用最大似然法进行预训练。文章还对 rnnlm 的有效性是否仍然有限提出了质疑，并提出了自己的观点和理由。希望本文的研究能为神经文本生成景观及其未来的研究提供新的思路。

