直接把GAN应用到NLP领域（主要是生成序列），有两方面的问题：

1. GAN最开始是设计用于生成连续数据，但是自然语言处理中我们要用来生成离散tokens的序列。因为生成器(Generator，简称G)需要利用从判别器(Discriminator，简称D)得到的梯度进行训练，而G和D都需要完全可微，碰到有离散变量的时候就会有问题，只用BP不能为G提供训练的梯度。在GAN中我们通过对G的参数进行微小的改变，令其生成的数据更加“逼真”。若生成的数据是基于离散的tokens，D给出的信息很多时候都没有意义，因为和图像不同。图像是连续的，微小的改变可以在像素点上面反应出来，但是你对tokens做微小的改变，在对应的dictionary space里面可能根本就没有相应的tokens.

2.GAN只可以对已经生成的完整序列进行打分，而对一部分生成的序列，如何判断它现在生成的一部分的质量和之后生成整个序列的质量也是一个问题。

**(1) 原始 GAN 主要应用实数空间（连续型数据）上，在生成离散数据（texts）这个问题上并不 work。**GAN 理论的提出者 Ian Goodfellow 博士这样回答来这个问题问题：“GANs 目前并没有应用到自然语言处理（NLP）中，最初的 GANs 仅仅定义在实数领域，GANs 通过训练出的生成器来产生合成数据，然后在合成数据上运行判别器，判别器的输出梯度将会告诉你，如何通过略微改变合成数据而使其更加现实。一般来说只有在数据连续的情况下，你才可以略微改变合成的数据，而如果数据是离散的，则不能简单的通过改变合成数据**。**例如，如果你输出了一张图片，其像素值是1.0，那么接下来你可以将这个值改为1.0001。如果输出了一个单词“penguin”，那么接下来就不能将其改变为“penguin + .001”，因为没有“penguin +.001”这个单词。 因为所有的自然语言处理（NLP）的基础都是离散值，如“单词”、“字母”或者“音节”， NLP 中应用 GANs是非常困难的。一般而言，采用增强学习算法。目前据我所知，还没有人真正的开始研究利用增强算法解决 NLP 问题。”

**(2)**在生成 text 时，GAN 对整个文本序列进行建模打分。对于部分(partially)生成的序列，十分难判断其在之后生成整个 (fully) 序列时的分数。

**(3)**另一个潜在的挑战涉及 RNN 的性质（生成文本大多采用 RNN 模型）。假设我们试图从 latent codes 生成文本，error 就会随着句子的长度成指数级的累积。最开始的几个词可能是相对合理的，但是句子质量会随着句子长度的增加而不断变差。另外，句子的长度是从随机的 latent representation 生成的，所以句子长度也是难以控制。