大模型为什么会重复输出

作者：纽约的自行车

日期：2024.03.03

1 存在的问题

尽管当前大模型生成的文本可读性已经接近人类交流水平，但生成内容的重复问题任然困扰着大家，尚未有很好的方案能够在保证精度的条件下完美解决这些问题。现阶段重复问题可以简单分成三类：

1. 字符级别重复，指大模型针对一个字或一个词重复不断的生成，例如“steckdose steckdose steckdose steckdose steckdose...”；
2. 语句级别重复，针对一句话重复不断的生成，例如“这是一个杯子，这是一个杯子...”；
3. 章节级别重复，多次相同的 prompt 输出完全相同或十分近似的内容，没有一点创新性的内容。除此之外，对不同的 prompt 也可能会生成类似的内容，且有效信息很少、信息熵偏低；

关于生成式大模型的输出重复问题仍然是现在关注的重点之一。重复生成现象是一个外在表象，有可能有很多种原因，然后造成了一个共同结果。

2 现象解释

Jin Xu[https://github.com/Jxu-Thu/DITTO] 等认为当模型生成一个高概率的句子时，它接下来更有可能继续生成相同的句子。当模型生成了一个重复的句子后，由于有更多重复句子共享相同的句子级上下文，所以重复句子的概率会进一步增加。这种上下文共享为复制这个句子提供了支持，模型会更倾向于重复生成这个句子，从而导致重复句子的生成。

Yixuan Su[https://arxiv.org/abs/2202.06417] 等认为神经语言模型的退化源于单词表示的各向异性分布，即它们的表示位于全体表示空间的一个狭窄子集中。这意味着这些表示彼此接近，自然地导致模型在不同的步骤中容易生成重复的单词。

我在实验中发现，两种情况下容易出现重复生成现象：

1. 第一种是条件文本过长，生成的文本过短。原因也很好理解，大模型建模概率是一个条件概率：p(x_t|x_1, x_2,...,x_t-1) ，当前面的条件文本过长时，大模型输出的短文本会被条件文本淹没，继续预测下一个 token 的话，在模型看来条件仍差不多，模型极大可能会将前面已经生成的短文本重新预测成概率最大的文本，以此类推，会一直重复下去。
2. 第二种是微调时在一个 epoch 中重复采样数据。最好不要重复采样，当然这是针对微调时的建议，当直接使用 LLM 时，预训练已经完成，对于重复现象只能是找方法缓解，而不是改变预训练策略。

3 解决之道

解决重复输出现象的方法分为两种：基于训练的方法和基于解码的方法。

3.1 基于解码的方法

基于解码的方法有 n-gram blocking、top-k、top-p、typical_p 等。这些方法并没有改变模型，而是基于规则直接修改模型计算的置信度，虽然能减少重复，但会对文本的流畅度和语义有比较大的影响。

n-gram blocking 是一种最暴力抑制重复的方法，通过限制设置的 ngram 不能出现重复，如果重复，就选概率次大的一个，来强制模型不生成重复的 token。

top-k 增添了随机性，相当于同样的输入，多次经过 top-k 采样生成的结果大概率不会一样，是一种行之有效，当然它最大问题是不能保证句子的通顺度以及对 prompt 的忠诚度。

top-p 是对简单暴力的 top-k 修改后的方法，该方法生成的句子通顺度以及对 prompt 的忠诚度更佳。

如果模型开始自我重复，则表明 softmax 温度设置过低。高温意味着更多的随机性，这可以帮助模型给出更有创意的输出。提高温度配合 top-k/top-p 算法，可以达到生成更激进创新性回答的需求，但生成句子的稳定性不可控。

假设在自然语言上下文中，在上文给定的情况下，每个字符包含的信息量应该和下文内容的平均信息量是基本一致的，那么，在模型生成采样时，我们就应该只采样那些与条件熵对应概率接近的字符，这就是 typical_p。与 top-k 和 top-p 抽样相比，locally typical sampling 在质量方面具有竞争力，同时可以持续减少重复度低的内容。

3.2 基于训练的方法

基于训练的方法有 SimCTG、DITTO。这种方法会改变模型权重，可能对模型能力造成损失。模型训练后可结合基于解码的方法一起使用。

3.2.1 SimCTG

SimCTG 是为了解决最大化生成方式解码时出现解码退化的问题，即生成的文本不自然的并包含文本重复而提出的一种解决方案。

作者认为神经语言模型的退化源于单词表示的各向异性分布，即它们的表示位于全体表示空间的一个狭窄子集中。这意味着这些表示彼此接近，自然地导致模型在不同的步骤中容易生成重复的单词。理想情况下，模型输出的单词表示应遵循各向同性分布，即单词的相似度矩阵是稀疏的。此外，在解码过程中，应尽可能保持文本的标记相似度矩阵的稀疏性，从而避免模型退化。基于上述动机，作者提出“对比训练 + 对比搜索”的方法，鼓励模型学习各向同性的单词表示，避免模型退化。

作者引入“对比学习”的思想，对于文本中的每一个单词，选取该单词作为锚点和正例，其他单词作为负例，以余弦相似度为距离度量，构建对比学习的三元损失。对比学习的目标在于

拉近锚点和正例的表示距离，拉远锚点和负例的表示距离，这样就可以构造一个具有稀疏分布特性的良好表示空间。

光有对比训练还不足以保证解码过程中单词表示的稀疏性。于是作者设计了这么一套解码方案：在每个解码步骤中，从模型置信度最高的候选单词集合中进行选择，从而确保生成文本是流畅、可靠的；同时，计算得到的新单词表示要和前文相似度越低越好，从而相对于先前的语境有足够的区分度。这样，生成的文本可以更好地保持与前文的语义一致性，同时避免模型退化。

对比训练和对比搜索在训练和推理两个阶段限制生成 token 间的相似度，有效降低了模型对一些特别常见表达的依赖，让模型尝试生成不一样的表达，整体上提升模型的创造性。

3.2.2 DITTO

伪重复惩罚法 (Pseudo-Repetition Penalization，DITTO)。通过重复从训练语料库中采样句子来手动构建伪数据，通过将句子 s 重复 N+1 次，构造一个伪重复样本 x。在这些伪数据上设计了一个重复惩罚损失函数，让模型学会随着句子重复次数的增加，以指数方式衰减重复概率。训练时将伪数据和正常数据混合训练，正常数据中掺杂 token 或者 phrase 级别的重复文本，这样模型既能避免句子级别重复，又能正常输出 token/phrase 重复，符合人类书写习惯。

4 最后

去重策略可节省计算资源，提高输出质量。本文探讨了多种去重方法和策略，从 TopP，TopK 采样到 SimCTG，DITTO 算法，每一种都有其适用的场景和特点，选择合适的去重策略是优化模型性能的关键。

5 参考文献

https://huggingface.co/blog/zh/how-to-generate

https://zhuanlan.zhihu.com/p/659961396

https://github.com/yxuansu/SimCTG

https://github.com/Jxu-Thu/DITTO

https://arxiv.org/abs/2202.00666