WaterMax: breaking the LLM watermark detectability-robustness-quality trade-off

> https://arxiv.org/abs/2403.04808

- 总结过往算法的limitations：
  - token entropy limit：检测成功率和文本的熵关系很大
  - quality of the generated text：
    - Christ & Aaronson不改变token distribution，但是in practice失真
    - KGW家族：由于modify了sample distribution，所以会带来一个watermark strength和text quality的trade-off
  - text size limit：现在的方法都是基于统计来检测的，它们的有效性取决于文本的长度，太短的文本检测不出来
  - robustness: 大部分算法没办法从理论上保证detector在经过text editing attack之后的power
- WaterMax
  - 核心思想1：Watermark by generating multiple texts
    - 这个technique的好处是可以泛化到几乎现在所有的水印算法上
    - 现在的水印算法是现在generate的过程中加点signal，然后detect手段基本都是给定一个文本计算出一个p-value，然后规定一个阈值来区分
    - 本文的方法只取这些算法的detect的部分，也就是说只取p-value计算的部分，而在生成的时候，加水印就是让大模型生成n条文本，依次计算p-value，sample最小的那个出来；而不加水印就是random-sample，或者可以理解为只generate一个文本。这样有水印和无水印大模型生成的文本它们的p-value分布是不同的，可以被检测出来。
    - 在本文的setting中，最终检测的方法是，给定一个文本，设它算出来的p-value是p；计算$\Lambda(p) = (n-1)\log(1-p) + \log n$，然后和一个规定好的阈值比较一下
  - 核心思想2：Watermarking chunks of text

    因为涉及生成n个text，肯定不是全都生成完再选，用类似beam search的思想，把generate的过程切分为N个步骤，每个步骤生成一个chunk。这样如果不做筛选应该会生成n^N种，但是这样就复杂度爆炸了，所以每迭代一步要选择m个最好的（选m个p-value最小的）