Robust Distortion-free Watermarks for Language Models

> https://arxiv.org/abs/2307.15593

这篇文章很明显是受到Undetectable Watermarks for Language Models的启发，首先Distortion-free就是Undetectable的一个替换词，都代表不改变token的distribution，都是在softmax之后只影响token sampling的过程。 这篇文章通过准备n簇备选随机数来代替仅仅一簇，再引入编辑距离，提升了鲁棒性

- 准备阶段
  - 用一个key生成624个int_32的随机数序列W
  - 根据这个随机数序列生成一个[n, vocab_size]的tensor：xi，tensor的每一个元素都是从随机序列W中随机挑选的一个值
    - 和prob做操作的备选tensor集合，prob做操作的时候就是从n簇随机数中选择一个做操作
  - 从[0, n)中随机一个shift
- watermarking process
  - LLM生成文本阶段
    - logits过softmax得到probs，shape：[1, vocab_size]
    - 从[n, vocab_size]中取一簇[1, vocab_size]形状的向量u，具体取哪簇根据(shift+i)%n

      注意：虽然因为shift的随机性，这个文本第一个选的簇是随机的；但是后面选的簇都是连续的；比如第一簇选的是i，第二簇就是i+1，一直往后排
    - vocab_size的每个维度用u**(1/prob)计算，取argmax决定到底选哪个token
- watermark detection process
  - levenshtein函数：（**编辑距离**）
    - 编码后的文本和xi取k簇（连续的k簇）挨个算levenshtein distance，取最低值

      其实是遍历一遍备选，希望找到generator里到底是用的哪连续k簇
  - 随机修改xi n_runs次，即生成n_runs（default=500）的xi_alternative，再用levenshtein计算；因为这次是完全随机生成，所以按道理来讲，加了水印的话应该完全随机生成xi之后编辑距离变大了。

```
# assuming lower test values indicate presence of watermark
p_val += null_result <= test_result
```