The Science of Detecting LLM-Generated Texts

> https://arxiv.org/pdf/2303.07205.pdf

这篇文章把classifier-based approaches和text watermarking这两种划分方式总结出来了；它的叫法是Black-box detection和White-box detection；

- Black-box vs White-box detection
  - Black-box对于LLM的access是API level的，意思是它不知道LLM的logits机制，只能调用API去生成文本。其实这就是classifer-based approaches，它只能通过收集大量的machine-generated和human text，通过统计规律和sentence patterns来train classifier；
  - White-box对LLM的access是几乎全透明的，知道LLM的logits机制，它通常是植入大模型内部的
- Black-box detection
  - Step1: Data Acquisition：收集数据
  - Step2: Feature Selection：特征采集
    - statistical disparities
      1. Zipf's law，自然语言词频相关的统计规律
      2. GLTR
      3. PPL：普遍认为LLM是在特定的语料上训练的，生成的语句PPL更低，而自然文本包含的语料范围更广，PPL会高一些

      （自己用实验在GPT2上验证了，确实是这样，也应该看看更大的模型是什么情况）
    - linguistic patterns

      vocabulary features, part-of-speech（词性）, dependency parsing, sentiment analysis, stylish features

      1. vocabulary features：之前对chatgpt的研究表明chatgpt倾向于使用更多样化、但长度更小的词汇
      2. part-of-speech：研究表明chatgpt更喜欢用名词
      3. sentiment anaylsis: 研究表明chatgpt更中立，尤其是它的负面情绪比人类少很多
      4. stylish features：如重复性、可读性等

      比较局限的是我们可以通过修改prompt让chatgpt表现出不同的语气，用不同的词汇等等，轻易就能改变这些linguistic patterns，所以这类方法的对adversarial attack非常不鲁棒
    - fact verification：基于事实核查的判定方法

      大模型经常生成一些编造的文本，因此可以以此为特征去区分。但是对于事实性弱的命题，比如写一篇散文，就很难区分了。
  - Step3: The Execution of the classification model：分类器构建和使用
    - 很传统的方法：SVM、Bayes、Decision Tree
    - Deep Learning Approaches：例如fine-tuned RoBERTa、fine-tuned BERT
  - 局限性：可解释性很弱
- White-box detection （text watermarking）
  - 一个好的水印算法应该具备的特征
    - Effectiveness：高效植入，并能被高准确度地检测出来

      很重要的事情，文中没有重点强调：不能影响文本质量，也不能影响*language statistic (word distribution, etc.)*，其实*language statistic*也是文本质量的一个保证

还有很重要的事情，是应该具有可解释性，越可解释，可扩展性相对就强

- Secrecy：加密算法不能被轻易识破，其实是privacy
- Robustness：对adversarial attack鲁棒，不能人家随便改改水印就被移除了

- Post-hoc Watermarking：顾名思义，文本全生成之后再加水印，整体思路是在生成文本之后加一些隐藏的信息进去，detector通过恢复隐藏信息来甄别
  - Rule-based Approaches
    1. Format based：例如文本左移、右移等，但直接reformat就移除水印了
    2. Syntactic based：在句法上植入特殊的规则
    3. Semantic based：例如通过替换成特定的同义词植入水印
  - Neural-based Approaches

    最开始只需要两个组件：Watermark Encoder和Watermark Decoder；一个Target文本和一个Secret Message文本输给Encoder，它输出一段隐含了Secret Message的Modified Text；Decoder接收文本，看是否能还原出Secret Message，能还原出就是有水印，还原不出就是没水印

    但是如果只这样train的话，可以想象，如果只追求检测的准度高，Encoder会给train成非常奇怪的样子，modified text会和正常的文本差别很大，所以要引入一个新的组件discriminator network；encoder的另一个训练目标设置成它生成的modified text和target text应该让discriminator network不好辨别。

    所以说，最后是三个组件在训练中达到一个平衡，encoder能让decoder好识别，但不能让discriminator好识别。

    但这个训练过程比较复杂，缺乏可解释性

- Inference-time watermarking：在LLM inference的过程中，对它的decode过程做小修改，修改它的打分表

  本文重点介绍了 A Watermark for Large Language Model 那篇文章（红绿列表）

  Text Quality是一个很大的concern

- Attack 水印破解方法（指推断加密过程，而不是混淆uw/w）

  Sadasivan：parahrasing attack