Automatic Detection of Machine Generated Text: A Critical Survey (COLING 2020)

> https://aclanthology.org/2020.coling-main.208.pdf

这篇文章是2020年的一篇关于机器文本检测的一个survey。文章比较早，是主要针对classifier-based approaches做的总结（当时text watermarking还没诞生）。

- 一个比较好的detector应该具备的特点：
  - accurate：精准检测
  - data-efficient：训练detector的数据越精简越好，不能训练成本太高

    补充：其实可以说是resource-efficient，不仅要考虑数据，也要考虑time cost和gpu memory cost
  - generalizable：泛化性强，不能太专用了
  - interpretable：可解释
  - robust：鲁棒，抗攻击，例如刻意修改成另一类，也要能识别出来

这篇文章前面一直在讲关于TGMs（Text Generative Models）的背景知识，这部分跳过，我们聚焦detector的部分

- Detectors归类
  - Classifiers trained from Scratch：从头开始训练分类器（没有任何先验知识）
    - Logistic Regression

      下面这篇文章用最基础的logistic regression训练了一个分类器，去甄别gpt2生成的文本和自然文本；他们分析了不同size的GPT2、不同的Sampling方式、以及GPT2是否在特定自然文本上finetune这几个不同场景下的甄别难度

      Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release Strategies and the Social Impacts of Language Models. CoRR, abs/1908.09203.
    - 扩展任务

      下面这篇文章提出了一个相关的扩展任务：设计detector去检测一段machine-generated text是在什么配置属性下生成的（prompt length, decoding strategies, TGM model size, etc.）

      他们的实验表明这个任务的难度小于识别一段文本是否由机器生成

      Yi Tay, Dara Bahri, Che Zheng, Clifford Brunk, Donald Metzler, and Andrew Tomkins. 2020. Reverse Engineering Configurations of Neural Text Generation Models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 275–279.
  - Zero-shot classifier：需要一个pretrained TGM去做detector，这个TGM一般就是负责生成机器文本的TGM，或者和他很像的TGM
    - Total log probablity：例如用pretrained GPT2做detector，它会给出一段文本的likelihood，如果这个likelihood更接近机器文本的平均likelihood（而不是human text的平均likelihood），就认为是机器文本。

      槽点：准确度很低

        - Giant Language model Test Room (GLTR) tool：去精细地分析GPT2生成的文本的token概率分布，以此来进行甄别。

          劣势：专用性太强，每新出一个TGM都得重新分析一遍，而且如果未来的TGM更像人类，在token distribution上很可能就分析不出来区别了
    - Fine-tuning NLM：需要一个pretrained LM去做detector（如BERT RoBERTa），且额外需要训练数据去train这个detector
        - GROVER detector
        - RoBERTa detector
- 现有Detector的主要问题（针对当时最好的RoBERTa detector来分析）
    - 需要太多训练数据（其实感觉这个不是很大的问题，200K也可接受）
    - 可扩展性差，finetune太专用了，对每个TGM都得重新finetune对应的detector；这个我觉得是最大的问题。从现在的眼光看，各种LLM层出不穷，其它的LLM用roberta当detector可能效果也不太好。如果用实际生成文本的LLM本身做detector，那各种LLM体量比GPT2大太多，finetune的GPU memory cost和time cost都太高了，不太可取；
    - 鲁棒性差：这个也是非常严重的问题，之前的方法都没太考虑对抗攻击，稍微改改就检测不准了。