

提示对人工智能生成文本零点检测的影响

Kaito Taguchi *、Yujie Gu † 和 Kouichi Sakurai ‡

日本福岡九州大学

摘要

近年来，大型语言模型（LLM）的开发取得了重大进展。虽然 LLM 的实际应用现已十分广泛，但其潜在的滥用问题，如生成假新闻和剽窃等，也引起了人们的极大关注。为了解决这个问题，人们开发了检测器来评估给定文本是人类生成的还是人工智能生成的。其中，零镜头检测器是一种有效的方法，它不需要额外的训练数据，通常基于似然法。在基于聊天的应用中，用户通常会输入提示并使用人工智能生成的文本。然而，零镜头检测器通常是孤立地分析这些文本，忽略了原始提示的影响。可以想象，这种方法可能会导致文本生成阶段和检测阶段的可能性评估出现差异。迄今为止，关于提示的存在与否如何影响零镜头检测器的检测准确性，仍是一个未经验证的空白。在本文中，我们引入了一个评估框架，以实证分析提示对人工智能生成文本检测准确性的影响。我们使用白盒检测（利用提示信息）和黑盒检测（在没有提示信息的情况下运行）对各种零点检测器进行了评估。我们的实验揭示了提示对检测准确性的重要影响。值得注意的是，与不使用提示信息的黑盒检测相比，使用提示信息的白盒检测方法的 AUC 至少提高了 0.1。代码见：<https://github.com/kaito25atugich/Detector>。

1 引言

近年来，大型语言模型（LLMs）的开发取得了重大进展[1, 2, 3]，其实际应用也变得越来越广泛。与此同时，其潜在的误用问题也引起了人们的极大关注。例如，利用 LLM 生成假新闻和剽窃就是一个值得注意的问题。检测器可以评估给定文本是人类生成的还是人工智能生成的，是防止此类滥用的一种防御机制。针对人工智能生成的文本的检测器大致可分为三类：零镜头检测器和人工智能检测器。

利用统计特性的探测器[4, 5, 6, 7, 8, 9, 10, 11]，采用监督学习的探测器[12, 13, 14, 15]，以及利用水标记的探测器[16, 17]。

许多方法都使用基于似然法的分数来设计零点检测器，如 DetectGPT [5]，这种检测器不需要额外的训练。零点检测器的总和如表 1 所示。换句话说，零点检测是通过在生成阶段复制似然值来实现的。在使用 LLM 时，我们通常输入提示并利用生成的输出。然而，在检测阶段，预计复制似然

表 1：零短路探测器概述

方法概要

使用给定文本的对数似然进行检测。

等级 计算给定文本的可能性，并根据整个词汇量将每个标记的可能性转换为等级，然后以此进行检测。

Log-Rank 计算给定文本的可能性，并根据整个词汇量将每个标记的可能性转化为等级，然后对这些等级应用对数进行检测。

熵检测 利用词汇中的词块可能性计算熵。

DetectGPT [5] 使用屏蔽语言模型，随机替换文本中的词语。使用评分模型观察替换后的文本与原始文本的相似度，并利用这种变化来检测篡改。

FastDetectGPT [6] 用类似于评分模型的自动回归模型取代 DetectGPT 中的掩码模型。从词汇表中随机抽样替换单词。以与 DetectGPT 相同的方式计算分数。

LRR [7] 使用对数概率与对数秩的比值进行检测。

NPR [7] 与 DetectGPT 类似，利用对数等级而非对数可能性计算得分。

双筒望远镜 [8] 利用略微不同的数据量训练模型，并计算每个模型的易错性。然后利用复杂度的差异进行检测。

由于缺乏提示提供的文本信息，这种方法变得具有挑战性。这有可能导致文本生成和检测阶段的可能性评估出现差异。在本文中，我们将评估这一现象在多大程度上影响基于似然法的零点检测器。本研究的贡献如下：

- 我们提出了两种使用零镜头检测器检测人工智能生成文本的方法：白盒检测和黑盒检测，前者利用生成文本的提示语，后者则不依赖提示语检测人工智能生成的文本。

- 大量实验证明，现有黑盒检测中的零点检测器的检测精度有所下降。

- 表明样本量及其比率对快速序列向量稳健性的重要性。

2 相关工作

在利用提示故意破坏检测准确性方面，可以确定两大类研究。第一类涉及

第二类研究则是故意制作带有恶意的提示语，以故意降低检测准确率。与此相反，第二类研究采用的是没有恶意的良性提示任务。

2.1 恶意提示

首先，我们将深入探讨专门针对故意创建恶意提示的研究。在 [19] 中，Koike 等人提出了 OUTFOX，利用问题陈述 P 、人类生成的文本 H 和人工智能生成的文本 A 进行上下文学习。通过构建诸如 " $p_i \in P \rightarrow h_i \in H$ 是人类的正确标签， $p_i \in P \rightarrow a_i \in A$ 是人工智能的正确标签"这样的提示，他们旨在为给定的问题陈述生成文本，使生成的文本与人类撰写的内容保持一致。这种方法使得检测人工生成的内容具有挑战性。Shi 等人通过使用教学提示对 OpenAI 的 Detector [22] 进行了攻击，结果发现检测准确率有所下降 [18]。结构性提示包括添加一个参考文本 X_{ref} 和一个指导文本 X_{ins} ，这两个文本的特征会降低原始输入 X 的检测准确性，从而破坏检测准确性。在 [20] 中，Lu 等人提出了 SICO，这是一种通过指示提示中的模型模仿人类撰写的文本的写作风格并更新结构中的内容来降低检测准确率的方法。Kumarage 等人提出了一种名为 "软提示" (Soft Prompt) 的攻击方法，利用强化学习生成一个向量，诱导检测器进行错误分类。这种软提示向量随后被用作 DetectGPT 和 RoBERTa-base 检测器的输入向量 [12]，结果表明检测准确率有所下降 [21]。

2.2 良性提示

我们在此回顾了涉及良性提示任务的案例。

Liu 等人使用基于监督学习的 Check-GPT 模型进行了实验。他们的研究表明，在使用不同的提示时，虽然检测准确率都超过了 90%，但实验证明检测准确率下降了约 7% [15]。Dou 等人 [14] 进行了学生使用 LLM 的实验。在他们的研究中，当使用提示时，DetectGPT 的检测精度有所下降。Hans 等人 [8] 利用 "写一个水豚天文学家" 这样的独特提示，指出了根据提示的有无来重现可能性的困难。针对水豚问题，他们提出了双筒望远镜。我们假设执行的是良性任务，如总结。因此，与恶意提示不同的是，在构建提示时，不需要刻意选择会降低使用检测器准确率的提示，也不需要收集成对的数据进行上下文学习。另一方面，Dou 等人 [14] 在实验中证明了检测准确率的意外下降。不过，他们并没有深入研究精度下降的原因，也没有提到其他基于似然法的零点检测器。此外，Hans 等人 [8] 没有具体验证检测器知道或不知道提示对检测精度的影响。因此，"双筒望远镜" 对提示引起的相似性变化的适应能力还没有得到充分的研究。在这种情况下，基于监督学习的方法 [15] 被排除在我们的实验之外。在本研究中，我们证明了即使在总结等普通任务中，使用基于似然法的零点检测器时，提示的存在或不存在都会无意中导致准确率的下降。

3 初步

3.1 语言模式

捕捉生成单词或句子概率的模型被称为语言模型。

模型。让 V 代表词汇量。长度为 n 的词序列的语言模型表示为 x_1, x_2, \dots, x_n ，其中 $x_i \in V$ ，其定义如下 (1)。

$$p(x_1, x_2, \dots, x_n) = \prod_{t=1}^n p(x_t | x_1, \dots, x_{t-1}) \quad (1)$$

3.2 现有的零点探测器

表 1 简要介绍了现有的零镜头检测器。这里的 $P_{T\theta}$ 指的是用于检测的语言模型。词汇表 V 由 C 个词组组成。输入文本 S 由 N 个标记符组成，表示为 $S = \{S_1, S_2, \dots, S_N\}$ ，从 S_1 开始的标记符序列为 S_N 。从 S_1 到 S_{i-1} 的标记序列表示为 S

3.2.1 对数似然法

对数似然法是一种利用组成文本的标记的相似性进行检测的方法。计算公式如 (2) 所示。对数似是构成给定文本的词组的对数似然的平均值。

$$\text{Log-likelihood} = \frac{1}{N-1} \sum_{i=2}^N \log P_{T\theta}(S_i | S_{<i}). \quad (2)$$

3.2.2 熵

熵法是一种利用词汇熵进行检测的方法。计算公式如 (3) 所示。熵的计算是利用词汇的可能性，取每个上下文的平均值。

$$\text{Entropy} = \frac{-1}{N-1} \sum_{i=2}^N \sum_{j=1}^C P_{T\theta}(j | S_{<i}) \log P_{T\theta}(j | S_{<i}). \quad (3)$$

3.2.3 等级

排序法是一种利用词汇中的词块在排序时的可能性大小顺序的方法。计算公式如 (4) 所示。等级是构成给定文本的词组的平均位置。排序

函数 sort 是一个按降序对给定数组排序的函数，而函数 index 则是一个在给定数组和元素作为输入时，返回元素在给定数组中的索引的函数。

$$\text{rank} = \frac{-1}{N-1} \sum_{i=2}^N \text{index}(\text{sort}(\log P_{T\theta}(S_i | S_{<i})), S_i). \quad (4)$$

3.2.4 DetectGPT

语言模型的目标是在文本生成过程中最大限度地提高可能性，而人类创建文本则与可能性无关。DetectGPT 关注这一现象，并提出了一个假设：通过改写某些词语，人工智能生成内容的文本可能性会降低，而人类生成内容的文本可能性则会增加或降低[5]。DetectGPT 的概览见图 1。替换过程是通过给定文本 S 中包含的部分单词使用掩码模型 P_M （如 T5 [24]）来实现的。这一操作重复进行 k 次迭代，然后计算得到的 k 个替换文本的平均对数似然值。(5) 表示得分，计算原始文本的对数可能性与获得的替换文本的平均对数可能性之间的差值。可以通过除以替换文本的对数似然的标准偏差来进行标准化。如果得分高于阈值 ϵ ，则视为人工智能生成的文本。

$$\text{DetectGPT} = \frac{\log P_{T\theta}(S) - \tilde{m}}{\tilde{\sigma}_S} \quad (5)$$

其中

$$\tilde{m} = \frac{1}{k} \sum_{i=1}^k \log P_{T\theta}(\tilde{S}_i)$$

$$\tilde{\sigma}_S = \frac{1}{k-1} \sum_{i=1}^k (\log P_{T\theta}(\tilde{S}_i) - \tilde{u})^2$$

和 $\tilde{S}_i \sim P_M(S_i)$ 分别代表平均值、样本变量和 $P_M(S_i)$ 的样本。

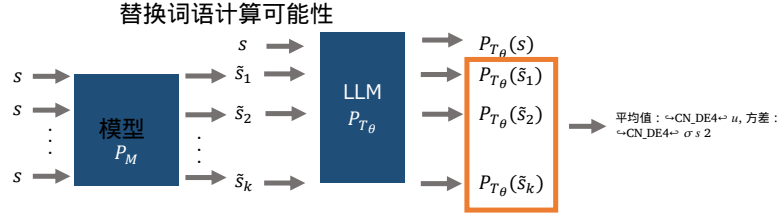


图 1 : DetectGPT 概述

3.2.5 FastDetectGPT

在[6]中，Bao 等人强调了 Detect- GPT 使用不同模型进行替换和分数计算的挑战，以及每次替换迭代都需要访问模型的成本相关问题。为此，FastDetectGPT 是一种改进的检测器，它减少了对模型的访问，在解决成本问题的同时实现了替换。虽然该方法涉及的假设设置与 DetectGPT 类似，但并没有根本性的改变。它仍然基于以下假设运行："人工智能生成的文本很可能是最大似然值，而人类生成的文本则不是"。我们在图 2 中展示了 FastDetect- GPT 的整体架构。在 FastDetectGPT 中，替代过程被一种不依赖掩码模型的替代方法所取代。与检测模型类似，它采用自回归模型， P_{T_θ} 和 P_{U_θ} 可以相同。第 i 个单词的替换包括从下一单词列表中随机抽取一个单词，考虑上下文直至输入文本中的第 $(i - 1)$ - 个单词，并用所选单词替换该单词。换句话说，进行 N 次替换后会得到替换文本 \tilde{S} ，通过在选词过程中进行抽样，替换过程会在一次访问中生成 k 个替换文本。由于后续的分数计算过程与 DetectGPT 相同，因此省略。

3.2.6 LRR 和 NPR

LLR (L ikelihood L og-Rank r atio) 和 NPR (N ormalized p erturbed log r ank) 是经典的对数

Su 等人提出的秩增强技术[7]。这两种方法的配置都很简单。LLR 是指对数概率与对数秩的比值，如 (6) 所示。这里， r_θ 表示使用 P_{T_θ} 时的秩。

$$LLR = - \frac{\sum_{i=1}^t \log P_{T_\theta}(S_i | S_{<i})}{\sum_{i=1}^t \log r_\theta(S_i | S_{<i})} \quad (6)$$

另一方面，NPR 和 DetectGPT 一样，会对文本中的词语进行 k 次替换。它取的是被替换文本的平均对数秩与原文对数秩的比值。其定义见 (7)。

$$NPR = \frac{\frac{1}{k} \sum_{p=1}^k \log r_\theta(\tilde{S}_p)}{\log r_\theta(S)} \quad (7)$$

3.2.7 双筒望远镜

Hans 等人提出了一种利用两个密切相关的语言模型 Falcon- 7b [26] 和 Falcon-7b-instruct 的检测方法 Binoculars，该方法采用了一种称为交叉复杂度的指标 [8]。整体框架如图 3 所示。让第一个模型称为 M_1 (如 Falcon-7b)，第二个模型称为 M_2 (如 Falcon-7b-instruct)。在这种情况下，使用 M_1 ，我们可以计算对数复杂度，如 (8) 所示。

$$\log PPL_{M_1}(S) = - \frac{1}{N} \sum_{i=1}^N \log(M_1(S_i | S_{<i})) \quad (8)$$

接下来，我们利用 M_1 和 M_2 计算交叉困惑度，如 (9) 所示。这里，符号 \cdot 代表点积。

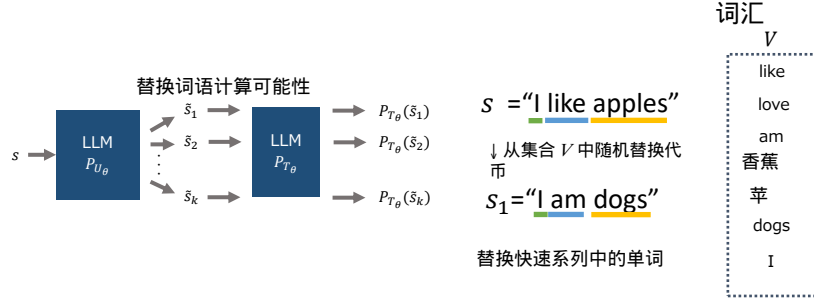


图 2：FastDetectGPT 和采样概述

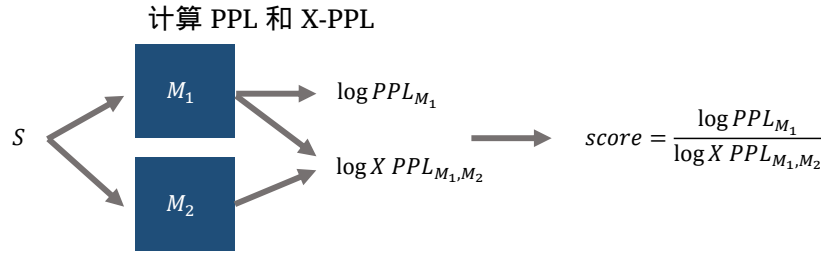


图 3：望远镜概览

4.1 FastNPR

NPR 中的单词替换是通过屏蔽模型来实现的。在这项研究中，为了降低成本，我们还采用了 FastNPR，这是一种用采样代替单词替换的方法，类似于 FastDetectGPT。

4.2 检测方法

我们将解释检测方法。为了便于解释，让 x 表示要检测的文本，如果 x 是人工智能生成的文本，则让 p 表示生成文本时使用的提示。检测可分为两种模式：黑盒检测和白盒检测。图 4 介绍了这两种模式的概况。黑盒检测发生在检测器不知道提示信息的情况下，本质上与现有的检测方法类似。在这种情况下，只向检测器提供 x 的内容。

$$\log X-PPL_{M_1, M_2}(S) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C M_1(j|S_{<i}) \cdot \log(M_2(j|S_{<i})) \quad (9)$$

双筒望远镜的得分由 (10) 决定。

$$B_{M_1, M_2}(S) = \frac{\log PPL_{M_1}(S)}{\log X-PPL_{M_1, M_2}(S)} \quad (10)$$

4 建议

在本研究中，我们提出了一种检测流程，以研究提示对可能性的影响。在介绍实验设置之前，我们先介绍一种额外的检测方法。

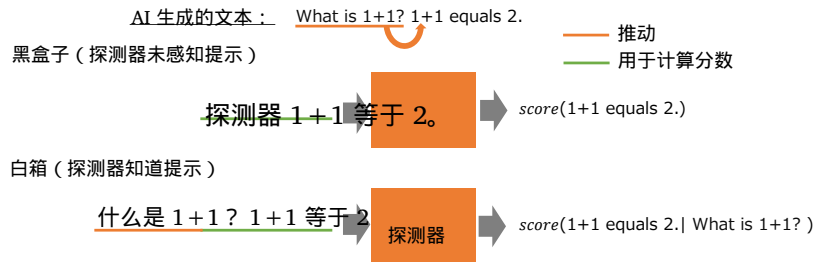


图 4：拟议的检测方法概览

另一方面，白盒检测涉及检测器对提示信息的了解。对于人类生成的文本，只有 x 是输入信息。值得注意的是，在白盒检测中，提示信息仅用于可能性计算，不包括在得分计算中。

5 实验

5.1 Configuration

首先，我们使用 GPT2-XL [23] 作为除保护模型，不包括 Binoculars。由于 GPU 的限制，Binoculars 使用了经过预训练和指导调整的 Phi1.5 [27]，而不是 Falcon。对于 DetectGPT 和 NPR，我们为整个文本的 10% 生成 5 个替换句子，而 Fast 系列则生成 10,000 个替换句子。在 DetectGPT 和 NPR 中，我们使用 T5-Large [24] 进行单词替换，而 Fast 系列则使用 GPT2-XL，即相同的检测模型。此外，我们还使用了 XSum 数据集 [28]。对于人类生成的文本，我们从 XSum 数据集中提取了 200 个样本；对于人工智能生成的文本，我们采用了 Llama2 7B 聊天模型 [25]，最多生成 200 个标记。使用的提示语是 "请您总结一下下面的句子，好吗？"

5.2 结果

从表 2 的结果可以看出，白盒检测的准确度更高，而黑盒检测的准确度更低。

表 2：检测生成的摘要：有提示和无提示案例之间的差异

| 方法 | 黑箱 | 白箱 |
|----------------|-------|-------|
| DetectGPT | 0.453 | 1.000 |
| FastDetectGPT | 0.819 | 0.958 |
| LRR | 0.532 | 0.995 |
| NPR | 0.560 | 0.934 |
| FastNPR | 0.768 | 0.993 |
| Entropy | 0.330 | 0.978 |
| Log-likelihood | 0.474 | 0.998 |
| Rank | 0.432 | 0.977 |
| Log-Rank | 0.485 | 0.999 |
| Binoculars | 0.877 | 0.999 |

检测准确率较低。正如预期的那样，通过提示修改可能性会导致基于可能性的检测器的检测准确率下降。值得注意的是，所有方法的准确度都一致下降了 0.1 或更多，这是一个重要的观察结果。与其他方法相比，双筒望远镜和快速系列检测器表现出稳健性。特别是快速系列检测器与传统方法保持了相同的评分计算方法，这表明在采样过程中存在稳健性因素。为了进一步验证，我们进行了额外的实验。在这个实验中，我们研究了不同替换率（表示替换文本中标记的程度）和样本大小（表示替换句子的数量）时检测精度的差异。DetectGPT 和

NPR 需要使用屏蔽语言模型来替换可信的标记，这使得替换并不可行，尤其是在替换率较高的情况下。因此，我们主要在 Fast 系列中改变替换率来进行研究。

DetectGPT 的结果见表 3，NPR 的结果见表 4。从这些结果可以看出，增加替换率和样本量有助于缓解检测准确率的下降。这一观察结果与 Chakraborty 等人的论断相似，即如果分布略有不同，增加样本量也能实现检测[29]。

然而，在我们的验证中，准确率的提高在 10 个样本左右就趋于平稳，最大 AUC 约为 0.8，这并不算高。特别是近年来，实际应用的趋势是强调高真阳性率和低假阳性率，这表明至少需要 0.9s 以上的 AUC [31, 8]。此外，DetectGPT 和 NPR 的检测准确率没有提高可能是由于可替换标记的数量有限。

表 3：替代率（SR）和样本量（SS）变化对 AUC（DetectGPT）的影响

| Method | SR | SS | AUC |
|---------------|------|-------|-------|
| FastDetectGPT | 10% | 5 | 0.640 |
| FastDetectGPT | 20% | 5 | 0.697 |
| FastDetectGPT | 100% | 5 | 0.779 |
| FastDetectGPT | 10% | 10 | 0.704 |
| FastDetectGPT | 20% | 10 | 0.739 |
| FastDetectGPT | 100% | 10 | 0.821 |
| FastDetectGPT | 100% | 10000 | 0.819 |
| DetectGPT | 10% | 5 | 0.453 |
| DetectGPT | 20% | 5 | 0.522 |
| DetectGPT | 30% | 5 | 0.490 |
| DetectGPT | 10% | 10 | 0.446 |
| DetectGPT | 30% | 10 | 0.446 |

表 4：替代率（SR）和样本量（SS）变化对 AUC（NPR）的影响

| Method | SR | SS | AUC |
|---------|------|-------|-------|
| FastNPR | 10% | 5 | 0.628 |
| FastNPR | 20% | 5 | 0.661 |
| FastNPR | 100% | 5 | 0.747 |
| FastNPR | 10% | 10 | 0.647 |
| FastNPR | 20% | 10 | 0.715 |
| FastNPR | 100% | 10 | 0.750 |
| FastNPR | 100% | 10000 | 0.763 |
| NPR | 10% | 5 | 0.560 |
| NPR | 20% | 5 | 0.590 |
| NPR | 30% | 5 | 0.577 |
| NPR | 10% | 10 | 0.589 |
| NPR | 30% | 10 | 0.588 |

6 讨论

6.1 零发探测器的假设

虽然我们的研究只关注提示语，但其他元素也可能出现类似现象。例如，生成和检测阶段之间温度或惩罚重复的变化可能会导致所选词块的差异，从而使基于可能性的检测变得困难。根据上述观察结果，我们推测，任何在语言生成过程中无法复制可能性的行为，都可能会影响零镜头探测器的检测准确性，而零镜头探测器的检测准确性则依赖于下一个单词预发音的可能性。

6.2 共同任务

虽然我们的研究侧重于摘要文本的生成，但还有其他一些潜在的任务需要考虑，例如意译生成、故事生成和翻译文本生成。在这些常见任务中，检测准确率也有可能下降。由于这些任务可能在没有恶意的情况下被使用，因此对它们进行类似的评估至关重要。

6.3 与转述攻击的相关性

正如前一节所简要讨论的，意译生成假定是单一行为。然而，目前已知的意译攻击[30, 31, 18, 13]是为每个句子生成意译并将结果组合起来。虽然使用掩码语言模型的意译攻击可能结构略有不同，因为它们利用前语境和后语境进行词语替换，但可以说，在检测过程中再现可能性变得具有挑战性。因此，意译攻击可被视为本研究中验证的任务的更复杂版本。

6.4 文本长度

在当前实验中，生成的文本固定为 200 个词块。词块的长度可能会影响重现可能性的难易程度。因此，使用更长的文本进行进一步验证是有益的。像叙事文生成这样的任务，文本的长度不是问题，可能适合进行此类研究。

6.5 Number of parameters

在这项研究中，每种检测方法都使用了一个约有 10 亿个参数的语言模型。如果使用更大的语言模型进行实验，是否能观察到更强的鲁棒性，这将是一个值得研究的问题。相反，也有实验研究表明，在更广泛的语言模型范围内，较小的语言模型能够为人工智能生成的文本实现更高的可能性[32]。考虑到这些发现，使用较小的语言模型进行实验并验证鲁棒性是否存在差异，也能提供有价值的见解。

6.6 与监督学习探测器的关系

即使使用监督学习，人们也注意到，基于提示的任务所生成的文本

可能会降低检测精度[15]。不过，这些模型有可能比零镜头检测器更稳健。例如，RADAR[13]在本实验所使用的任务中达到了 0.939 的 AUC。相比之下，RoBERTa-large 检测器[12] 的 AUC 为 0.767。这表明，针对仿写攻击的鲁棒检测器在其他任务中也可能表现出类似的鲁棒结果。

6.7 与水印的关系

水印技术利用统计方法进行验证 [16]。由于这些方法在生成和验证过程中都基于似然法，如果在验证阶段不能再现似然法，可能会导致准确性下降。另一方面，出现了针对仿写攻击的鲁棒水印技术[17]。这些方法也可能对提示表现出鲁棒性。

6.8 实现弹性零发探测器

目前，许多方法都是基于似然法进行检测。将这些方法与其他方法相结合，可能会带来更稳健的检测。本征维度就是这样一种方法[11]。本征维度指的是表示给定文本所需的最小维度。Tulchinskii 等人提出了一种基于持久同源性 (Persistent Homology) 的检测器来估算本征维度并将其作为得分。不过，这种方法要求文本长度恒定，不适用于我们的实验。在涉及较长文本的实验中探索这种方法的应用会很有意义。利用遮蔽语言模型获得的表征的方法（包括本征维度）计算似然的方式与我们实验中使用的检测器不同，后者基于自回归语言模型。将这些元素结合起来，可以开发出更强大的零点检测器。

鸣谢

本研究部分得到了日本和印度之间的国际科学交流、双边项目 DTS-JSP (资助编号 JPJSBP120227718) 以及 Kayamori 信息科学发展基金会的支持。

参考资料

- [1] OpenAI. (2023). GPT-4 Technical Report, arXiv e-prints.
- [2] Microsoft. Microsoft Copilot, Retrieved October 31, 2023, from <https://adoption.microsoft.com/ja-jp/copilot/>.
- [3] Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J. B., Yu, J., ... & Ahn, J. (2023). Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.
- [4] Gehrmann, S., Strobel, H., & Rush, A. (2019). GLTR: Statistical Detection and Visualization of Generated Text. In M. R. Costajussà & E. Alfonseca (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (pp. 111–116). Association for Computational Linguistics.
- [5] Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). DetectGPT: Zero-shot machine-generated text detection using probability curvature. In Proceedings of the 40th International Conference on Machine Learning (ICML'23) (Vol. 202, pp. 24950–24962). JMLR.org
- [6] Bao, G., Zhao, Y., Teng, Z., Yang, L., & Zhang, Y. (2023). Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature. arXiv preprint arXiv:2310.05130.
- [7] Su, J., Zhuo, T. Y., Wang, D., & Nakov, P. (2023). DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection
- arXiv preprint arXiv:2306.05540.
- [8] Hans, A., Schwarzschild, A., Cherepanova, V., Kazemi, H., Saha, A., Goldblum, M., ... & Goldstein, T. (2024). Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text. arXiv preprint arXiv:2401.12070.
- [9] Liu, S., Liu, X., Wang, Y., Cheng, Z., Li, C., Zhang, Z., ... & Shen, C. (2024). DoesDETECTGPT Fully Utilize Perturbation? Selective Perturbation on Model-Based Contrastive Learning Detector would be Better. arXiv preprint arXiv:2402.00263.
- [10] Sasse, K., Barham, S., Kayi, E. S., & Staley, E. W. (2024). To Burst or Not to Burst: Generating and Quantifying Improbable Text. arXiv preprint arXiv:2401.15476.
- [11] Tulchinskii, E., Kuznetsov, K., Kushnareva, L., Cherniavskii, D., Barannikov, S., Piontkovskaya, I., ... & Burnaev, E. (2023). Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts. arXiv preprint arXiv:2306.04723.
- [12] Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., ... & Wang, J. (2019). Release strategies and the social impacts of language models. arXiv preprint arXiv:1908.09203.
- [13] Hu, X., Chen, P. Y., & Ho, T. Y. (2023). RADAR: Robust AI-Text Detection via Adversarial Learning. arXiv preprint arXiv:2307.03838.
- [14] Dou, Z., Guo, Y., Chang, C. C., Nguyen, H. H., & Echizen, I. (2024). Enhancing Robustness of LLM-Synthetic Text Detectors for Academic Writing: A Comprehensive Analysis. arXiv preprint arXiv:2401.08046.
- [15] Liu, Z., Yao, Z., Li, F., & Luo, B. (2023). Check Me If You Can: Detecting ChatGPT-Generated Academic Writing using CheckGPT. arXiv preprint arXiv:2306.05524.

- [16] Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I. & Goldstein, T. (2023). A Watermark for Large Language Models. Proceedings of the 40th International Conference on Machine Learning, in Proceedings of Machine Learning Research 202:17061-17084.
- [17] Ren, J., Xu, H., Liu, Y., Cui, Y., Wang, S., Yin, D., & Tang, J. (2023). A Robust Semantics-based Watermark for Large Language Model against Paraphrasing. arXiv preprint arXiv:2311.08721.
- [18] Shi, Z., Wang, Y., Yin, F., Chen, X., Chang, K. W., & Hsieh, C. J. (2023). Red Teaming Language Model Detectors with Language Models. arXiv preprint arXiv:2305.19713.
- [19] Koike, R., Kaneko, M., & Okazaki, N. (2023). Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. arXiv preprint arXiv:2307.11729.
- [20] Lu, N., Liu, S., He, R., & Tang, K. (2023). Large Language Models can be Guided to Evade AI-Generated Text Detection. arXiv preprint arXiv:2305.10847.
- [21] Kumarage, T., Sheth, P., Moraffah, R., Garland, J., & Liu, H. (2023). How reliable are ai-generated-text detectors? an assessment framework using evasive soft prompts. arXiv preprint arXiv:2310.05095.
- [22] OpenAI. (2023). New AI classifier for indicating AI-written text, Retrieved November 30, 2023.
- [23] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. Retrieved October 31, 2023, from https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
- [24] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a 统一文本到文本转换器。机器学习研究期刊》, 21 (1) , 5485-5551.
- [25] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
- [26] Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., ... & Penedo, G. (2023). The falcon series of open language models. arXiv preprint arXiv:2311.16867.
- [27] Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., & Lee, Y. T. (2023). Textbooks are all you need ii: phi-1.5 technical report. arXiv preprint arXiv:2309.05463.
- [28] Narayan, S., Cohen, S. B., & Lapata, M. (2018). Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 1797–1807). Association for Computational Linguistics
- [29] Chakraborty, S., Bedi, A. S., Zhu, S., An, B., Manocha, D., & Huang, F. (2023). On the possibilities of ai-generated text detection. arXiv preprint arXiv:2304.04736.
- [30] Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). Can ai-generated text be reliably detected? arXiv preprint arXiv:2303.11156.
- [31] Krishna, K., Song, Y., Karpinska, M., Wieting, J., & Iyyer, M. (2023). Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. arXiv preprint arXiv:2303.13408.
- [32] Mireshghallah, F., Mattern, J., Gao, S., Shokri, R., & Berg-Kirkpatrick, T. (2023). Smaller Language Models are Better Black-box Machine-Generated Text Detectors. arXiv preprint arXiv:2305.09859.