The Impact of Prompts on Zero-Shot Detection of AI-Generated Text

Kaito Taguchi*, Yujie Gu†, and Kouichi Sakurai‡

Kyushu University, Fukuoka, Japan

Abstract

In recent years, there have been significant advancements in the development of Large Language Models (LLMs). While their practical applications are now widespread, their potential for misuse, such as generating fake news and committing plagiarism, has posed significant concerns. To address this issue, detectors have been developed to evaluate whether a given text is human-generated or AI-generated. Among others, zero-shot detectors stand out as effective approaches that do not require additional training data and are often likelihood-based. In chat-based applications, users commonly input prompts and utilize the AI-generated texts. However, zero-shot detectors typically analyze these texts in isolation, neglecting the impact of the original prompts. It is conceivable that this approach may lead to a discrepancy in likelihood assessments between the text generation phase and the detection phase. So far, there remains an unverified gap concerning how the presence or absence of prompts impacts detection accuracy for zero-shot detectors. In this paper, we introduce an evaluative framework to empirically analyze the impact of prompts on the detection accuracy of AI-generated text. We assess various zero-shot detectors using both white-box detection, which leverages the prompt, and black-box detection, which operates without prompt information. Our experiments reveal the significant influence of prompts on detection accuracy. Remarkably, compared with black-box detection without prompts, the white-box methods using prompts demonstrate an increase in AUC of at least 0.1 across all zero-shot detectors tested. Code is available: `https://github.com/kaito25atugich/Detector`.

1 Introduction

Recent years have seen significant advancements in the development of Large Language Models (LLMs) [1, 2, 3], and their practical applications have become widespread. Meanwhile, their potential misuse have raised significant concerns. For example, the generation of fake news and plagiarism using LLMs is a notable issue. Detectors that evaluate whether a given text is human-generated or AI-generated serve as a defense mechanism against such misuse.

Detectors for AI-generated text can be broadly classified into three categories: a zero-shot detector leveraging statistical properties [4, 5, 6, 7, 8, 9, 10, 11], a detector employing supervised learning [12, 13, 14, 15], and a detector utilizing watermarking [16, 17].

Zero-shot detectors, such as DetectGPT [5], which do not require additional training, are designed in many methods using likelihood-based scores. A summary of zero-shot detectors is illustrated in Table 1. In other words, the zero-shot detection is carried out by replicating the likelihood at the generation phase. When using LLMs, we usually input prompts and utilize the generated output. However, at the detection phase, it is anticipated that reproducing likelihood

提示对AI生成文本零样本检测的影响

田口海斗 *，顾宇杰†，桜井耕一‡

日本福冈九州大学

摘要

近年来，大型语言模型（LLMs）的发展取得了显著进展。虽然它们的实际应用现在已经广泛，但其潜在的误用，例如生成假新闻和抄袭，已引发了重大担忧。为了解决这个问题，已经开发出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。在众多检测器中，零样本检测器作为一种有效的方法脱颖而出，它们不需要额外的训练数据，通常基于可能性。在基于聊天的应用中，用户通常输入提示并利用AI生成的文本。然而，零样本检测器通常孤立地分析这些文本，忽视了原始提示的影响。可以想象，这种方法可能导致文本生成阶段和检测阶段之间的可能性评估出现差异。到目前为止，关于提示的存在或缺失如何影响零样本检测器的检测准确性仍然存在未验证的差距。在本文中，我们引入了一个评估框架，以实证分析提示对AI生成文本检测准确性的影响。我们使用白盒检测（利用提示）和黑盒检测（在没有提示信息的情况下操作）评估各种零样本检测器。我们的实验揭示了提示对检测准确性的显著影响。值得注意的是，与没有提示的黑盒检测相比，使用提示的白盒方法在所有测试的零样本检测器中显示出至少0.1的AUC增加。代码可用： https://github.com/kaito25atugich/Detector。

1 介绍

近年来，大型语言模型（LLMs）的发展取得了显著进展[1, 2, 3]，其实际应用已变得广泛。同时，它们潜在的误用引发了重大担忧。例如，使用LLMs生成假新闻和抄袭是一个显著问题。评估给定文本是人类生成还是AI生成的检测器作为防御机制，以应对这种误用。

AI生成文本的检测器可以大致分为三类：零样本检测器利用统计特性[4, 5, 6, 7, 8, 9, 10, 11]的检测器，采用监督学习[12, 13, 14, 15]的检测器，以及利用水印技术[16, 17]的检测器。

零样本检测器，例如 DetectGPT [5]，不需要额外的训练，采用多种方法设计，使用基于似然的分数。零样本检测器的总结如表1所示。换句话说，零样本检测是在生成阶段通过复制似然来进行的。在使用大型语言模型（LLMs）时，我们通常输入提示并利用生成的输出。然而，在检测阶段，预计会重现似然。

Table 1: Summary of Zero-short Detectors

Method	Summary
Log-likelihood	Detect using the log likelihood of the given text.
Rank	Calculate the likelihood of the given text and convert the likelihood of each token into ranks based on the entire vocabulary, then use this to detect.
Log-Rank	Calculate the likelihood of the given text and transform the likelihood of each token into ranks based on the entire vocabulary, then apply logarithm to these ranks for detection.
Entropy	Detect by calculating entropy using the likelihood of tokens in the vocabulary.
DetectGPT [5]	Using a masked language model, randomly replace words in the text. Observe the likelihood of the replaced text and the original text using a scoring model, and utilize the change to detect alterations.
FastDetectGPT [6]	Replace the mask model in DetectGPT with a auto-regressive model similar to the scoring model. Sample words randomly from the vocabulary to replace words. Calculate scores in the same manner as DetectGPT.
LRR [7]	Detect using the ratio of log-likelihood to log-rank.
NPR [7]	Similar to DetectGPT, utilize logarithmic ranks rather than logarithmic likelihood for scoring calculation.
Binoculars [8]	Utilize models trained with slightly different amounts of data and calculate the perplexity of each model. Then leverage the difference in perplexity for detection.

becomes challenging due to the absence of the contextual information provided by prompts. It may potentially result in differences in likelihood evaluations between the text generation and detection stages.

In this paper, we assess to what extent this phenomenon affects likelihood-based zero-shot detectors. The contributions of this study are as follows:

- We propose two methods for detecting AI-generated text using zero-shot detectors: white-box detection, which leverages the prompts used to generate the text, and black-box detection, which detects AI-generated text without relying on a prompt.
- Extensive experiments demonstrate a decrease in detection accuracy for existing zero-shot detectors in black-box detection.
- Indication of the significance of sample size and its ratio for the robustness of the Fast series detectors.

2 Related work

In the context of intentionally undermining detection accuracy using prompts, two main categories of studies can be identified. The first category involves the

表1：零样本检测器摘要

方法	摘要
对数似然	使用给定文本的对数似然进行检测。
排名	计算给定文本的可能性，并根据整个词汇表将每个词元的可能性转换为排名，然后使用此信息进行检测。
对数秩检验	计算给定文本的可能性，并根据整个词汇表将每个标记的可能性转换为排名，然后对这些排名应用对数以进行检测。
熵	通过计算词汇中标记的可能性来检测熵。
DetectGPT [5]	使用掩码语言模型，随机替换文本中的单词。观察替换文本和原始文本的可能性，使用评分模型，并利用变化来检测更改。
快速检测GPT [6]	将DetectGPT中的掩码模型替换为类似于评分模型的自回归模型。随机从词汇表中抽取单词以替换单词。以与DetectGPT相同的方式计算分数。
LRR [7]	使用对数似然比与对数秩的比率进行检测。
NPR [7]	类似于DetectGPT，使用对数排名而不是对数似然进行评分计算。
双筒望远镜 [8]	利用训练数据量略有不同的模型，并计算每个模型的困惑度。然后利用困惑度的差异进行检测。

由于缺乏提示所提供的上下文信息，这变得具有挑战性。这可能导致文本生成和检测阶段之间的可能性评估出现差异。

在本文中，我们评估了这一现象在多大程度上影响基于似然的零-shot 检测器。本研究的贡献如下：

- 我们提出了两种使用零样本检测器检测AI生成文本的方法：白盒检测，利用生成文本所使用的提示，以及黑盒检测，检测AI生成的文本而不依赖于提示。
- 大量实验表明，在黑箱检测中，现有的零样本检测器的检测准确性有所下降。
- 样本大小及其比例对快速系列探测器稳健性的意义指示。

2 相关工作

在故意削弱检测准确性的提示背景下，可以识别出两大类研究。第一类涉及到

deliberate crafting of prompts with malicious intent to deliberately reduce detection accuracy. In contrast, the second category encompasses research that employs tasks with benign prompts, devoid of malicious intent.

2.1 Malicious prompts

First, we delve into studies that specifically concentrate on the deliberate creation of malicious prompts.

In [19], Koike et al. proposed OUTFOX, utilizing in-context learning with the problem statement P, human-generated text H, and AI-generated text A. By constructing prompts such as "$p_i \in P \rightarrow h_i \in H$ is the correct label by humans, and $p_i \in P \rightarrow a_i \in A$ is the correct label by AI," they aim to generate text for a given problem statement in such a way that the generated text aligns with human-authored content. This approach makes the detection of artificially generated content challenging.

Shi et al. conducted an attack on OpenAI's Detector [22] by employing an Instructional Prompt, confirming a decrease in detection accuracy [18]. The Instructional Prompt involves adding a reference text X_{ref} and an instructional text X_{ins} with characteristics that reduce the detection accuracy to the original input X, thereby undermining the detection accuracy.

In [20], Lu et al. proposed SICO, a method that lowers detection accuracy by instructing the model within prompts to mimic the writing style of human-authored text and updating the content of the instructions to reduce detection accuracy.

Kumarage et al. proposed an attack named Soft Prompt, which generates a vector using reinforcement learning to induce misclassification by detectors. This Soft Prompt vector is then used as input for the DetectGPT and RoBERTa-base detectors [12], demonstrating a decrease in detection accuracy [21].

2.2 Benign prompts

We review cases involving tasks with benign prompts here.

Liu et al. conducted experiments using the Check-GPT model, an approach based on supervised learning. Their findings indicate that when using different prompts, although all surpass 90%, there is an experimental demonstration of approximately a 7% decrease in detection accuracy [15].

Dou et al. [14] performed experiments envisioning the utilization of LLMs by students. In their study, they demonstrated a decrease in DetectGPT's detection accuracy when prompts were employed.

Hans et al. [8] pointed out the difficulty in reproducing likelihoods depending on the presence or absence of prompts, using unique prompts like "Write about a capybara astronomer." In response to the capybara problem, they proposed Binoculars.

We assume performing benign tasks such as summarization. Therefore, unlike malicious prompt attacks, there is no need to deliberately choose prompts that would lower accuracy using the detector when constructing prompts, nor is there a requirement to collect pairs of data for in-context learning.

On the other hand, Dou et al. [14] experimentally demonstrated unintended decreases in detection accuracy. However, they did not delve into why the accuracy decreases or make references to other likelihood-based zero-shot detectors. Additionally, Hans et al. [8] did not provide specific verification regarding the impact of a detector knowing or not knowing the prompt on detection accuracy. Therefore, the resilience of Binoculars to changes in likelihood due to prompts has not been adequately assessed. The supervised learning based approach [15] is excluded from our experiments in this context.

In this study, we demonstrate that even in ordinary tasks such as summarization, the presence or absence of prompts unintentionally leads to a decrease in accuracy when using likelihood-based zero-shot detectors.

3 Preliminary

3.1 Language model

A model that captures the probability of generating words or sentences is referred to as a language

故意设计具有恶意意图的提示，以故意降低检测准确性。相比之下，第二类研究涉及使用无恶意意图的良性提示的任务。

2.1 恶意提示

首先，我们深入研究那些专门关注恶意提示的故意创建的研究。

在[19]中，Koike等人提出了OUTFOX，利用上下文学习，结合问题陈述P、人类生成的文本H和AI生成的文本A。通过构建诸如“pi ∈ P →hi ∈ H是人类的正确标签，以及pi ∈ P →ai ∈ A是AI的正确标签”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创作的内容一致。这种方法使得检测人工生成内容变得具有挑战性。

Shi等人通过使用指令提示对OpenAI的检测器进行了攻击，确认了检测准确性的下降。指令提示涉及添加一个参考文本Xref和一个具有降低检测准确性特征的指令文本Xins，从而削弱对原始输入X的检测准确性。

在[20]中，Lu等人提出了SICO，这是一种通过在提示中指示模型模仿人类撰写文本的写作风格，并更新指令内容以降低检测准确性的方法。

Kumarage 等人提出了一种名为软提示的攻击，该攻击使用强化学习生成一个向量，以诱导检测器的错误分类。这个软提示向量随后作为输入用于 DetectGPT 和 RoBERTa-base 检测器 [12]，显示出检测准确率的下降 [21]。

2.2 良性提示

我们在这里审查涉及温和提示的任务案例。

刘等人使用基于监督学习的Check-GPT模型进行了实验。他们的研究结果表明，在使用不同提示时，尽管所有结果均超过90%，但检测准确率的实验性演示显示约有7%的下降[15]。

Dou等人[14]进行了实验，设想学生使用大型语言模型（LLMs）。在他们的研究中，他们展示了在使用提示时，DetectGPT的检测准确性下降。

汉斯等人[8]指出了在重现依赖于提示的存在或缺失的可能性时的困难，例如使用“写一篇关于水豚天文学家的文章”这样的独特提示。针对水豚问题，他们提出了双筒望远镜。

我们假设执行无害的任务，例如摘要。因此，与恶意提示攻击不同，在构建提示时不需要故意选择会降低检测器准确性的提示，也不需要收集用于上下文学习的数据对。

另一方面，Dou等人[14] 实验性地证明了检测准确性意外下降。然而，他们并没有深入探讨准确性下降的原因，也没有提及其他基于可能性的零-shot 检测器。此外，Hans 等人[8] 并未对检测器是否知道提示对检测准确性的影响提供具体验证。因此，双目镜对由于提示而导致的可能性变化的韧性尚未得到充分评估。在这个背景下，基于监督学习的方法[15] 被排除在我们的实验之外。

在这项研究中，我们证明即使在诸如摘要这样的普通任务中，提示的存在或缺失无意中导致使用基于似然的零-shot检测器时准确性下降。

3 初步

3.1 语言模型

生成单词或句子的概率的模型被称为语言模型。

model. Let V represent the vocabulary. The language model for a word sequence of length n, denoted as $x_1, x_2, \ldots, x_n$ where $x_i \in V$, is defined by the following (1).

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

3.2 Existing zero-shot detectors

We provide a brief introduction to existing zero-shot detectors, summarized in Table 1. Here, P_{T_θ} refers to the language model utilized for detection. The vocabulary V is composed of C tokens. The input text S is composed of N tokens, represented as $S = \{S_1, S_2, ..., S_N\}$, and the token sequence from S_1 to S_{i-1} is denoted as $S_{<i}$.

3.2.1 Log-Likelihood

The log-likelihood is a method that utilizes the likelihood of tokens composing a text for detection. The formula is presented in (2). The log-likelihood is the average of the log-likelihoods of tokens constituting a given text.

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

3.2.2 Entropy

Entropy is a method that utilizes the entropy of the vocabulary for detection. The formula is shown in (3). Entropy is calculated using the likelihood of the vocabulary, taking the average across each context.

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

3.2.3 Rank

Rank is a method that utilizes the order of likelihood magnitude of tokens in the vocabulary when sorted. The formula is presented in (4). Rank is the average position of tokens constituting a given text. The function *sort* is a function that sorts the given array in descending order, and *index* is a function that, given an array and an element as input, returns the index of the element within the given array.

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

3.2.4 DetectGPT

The language model aims to maximize likelihood during text generation, whereas humans create text independently of likelihood. DetectGPT focuses on this phenomenon and posits a hypothesis that by rewriting certain words, the likelihood of the text decreases for AI-generated content and can either increase or decrease for human-generated content [5].

The overview of DetectGPT is presented in Figure 1. The replacement process is achieved by utilizing a mask modelP_M, such as T5 [24], on some of the words contained in the given text S. This operation is repeated for a total of k iterations, and the average log-likelihood of the obtained k replacement texts is then computed. (5) represents the score, calculating the difference between the log-likelihood of the original text and the average log-likelihood of the acquired replacement texts. It is permissible to standardize by dividing by the standard deviation of the log-likelihood of the replacement texts. If the score is above the threshold ε, it is deemed to be AI-generated text.

$$\text{DetectGPT} = \frac{\log P_{T_\theta}(S) - \tilde{m}}{\tilde{\sigma_S}} \quad (5)$$

where

$$\tilde{m} = \frac{1}{k} \sum_{i=1}^{k} \log P_{T_\theta}(\tilde{S_i})$$

$$\tilde{\sigma_S} = \frac{1}{k-1} \sum_{i=1}^{k} (\log P_{T_\theta}(\tilde{S_i}) - \tilde{u})^2$$

and $\tilde{S_i} \sim P_M(S_i)$ represent the mean, sample variance, and a sample from $P_M(S_i)$, respectively.

模型。让 V 代表词汇。长度为 n 的词序列的语言模型，记作 x1, x2, . . . , xn，其中 xi ∈ V，由以下公式定义 (1)。

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

3.2 现有的零样本检测器

我们提供了对现有零样本检测器的简要介绍，汇总在表1中。这里，PTθ指的是用于检测的语言模型。词汇表V由C个标记组成。输入文本S由N个标记组成，表示为S = {S1, S2, ..., SN}，从S1到Si−1的标记序列表示为S

3.2.1 对数似然

对数似然是一种利用构成文本的标记的似然性进行检测的方法。公式在（2）中给出。对数似然是构成给定文本的标记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

熵是一种利用词汇熵进行检测的方法。公式如（3）所示。熵是通过计算词汇的可能性，并在每个上下文中取平均值来得出的。

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

3.2.3 排名

排名是一种利用词汇中令牌按可能性大小排序的顺序的方法。公式在（4）中给出。排名是构成给定文本的令牌的平均位置。函数 sort 是一个将给定数组按降序排序的函数，而 index 是一个函数，给定一个数组和一个元素作为输入，返回该元素在给定数组中的索引。

$$\text{rank} = \frac{-1}{N-1} \sum_{i=2}^{N} \text{索引（排序（日志 PT}\theta\text{（Si|S} \quad (4)$$

3.2.4 检测GPT

语言模型在文本生成过程中旨在最大化似然性，而人类则独立于似然性创造文本。DetectGPT关注这一现象，并提出一个假设，即通过重写某些词，AI生成内容的文本似然性会降低，而人类生成内容的文本似然性则可能增加或减少。

DetectGPT的概述如图1所示。替换过程是通过利用掩码模型PM，例如T5 [24]，对给定文本S中的某些单词进行实现的。此操作重复进行k次迭代，然后计算获得的k个替换文本的平均对数似然值。公式(5)表示得分，计算原始文本的对数似然值与获得的替换文本的平均对数似然值之间的差异。可以通过除以替换文本的对数似然值的标准差进行标准化。如果得分高于阈值ε，则被视为AI生成的文本。

$$\text{DetectGPT} = \frac{\log P_{T_\theta}(S) - \tilde{m}}{\tilde{\sigma_S}} \quad (5)$$

where

$$\tilde{m} = \frac{1}{k} \sum_{i=1}^{k} \log P_{T_\theta}(\tilde{S_i})$$

$$\tilde{\sigma_S} = \frac{1}{k-1} \sum_{i=1}^{k} (\log P_{T_\theta}(\tilde{S_i}) - \tilde{u})^2$$

并且 ˜Si ~PM(Si) 分别表示均值、样本方差和来自 PM(Si) 的样本。

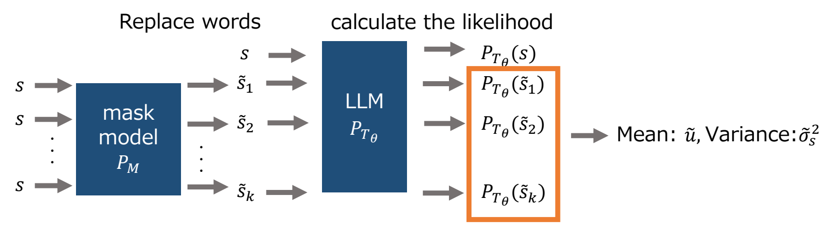

Figure 1: DetectGPT Overview

3.2.5 FastDetectGPT

In [6], Bao et al. highlighted challenges in DetectGPT's use of different models for substitution and score calculation, as well as the cost-related aspect of requiring model access for each substitution iteration. In response, FastDetectGPT is a modified detector that reduces access to the model, addressing the cost issue while enabling substitutions. Although the methodology involves setting hypotheses similar to DetectGPT, there is no fundamental change. It still operates on the assumption that "AI-generated text is likely to be around the maximum likelihood, whereas human-generated text is not."

We present the overall architecture of FastDetectGPT in Figure 2. In FastDetectGPT, the substitution process is replaced with an alternative method that does not rely on a mask model. Similar to the detection model, it utilizes an autoregressive model, and P_{T_θ} and P_{U_θ} can be the same. The substitution for the i-th word involves randomly extracting a word from the next-word list, considering the context up to the $(i-1)$-th word in the input text, and replacing the word with the chosen one. In other words, performing this substitution N times results in the substituted text $\tilde{S}$, and by conducting sampling during word selection, the replacement process generates k substitution texts in a single access.

The subsequent score calculation is omitted as it follows the same procedure as DetectGPT.

3.2.6 LLR & NPR

LLR (**L**ikelihood **L**og-**R**ank **r**atio) and NPR (**N**ormalized **p**erturbed log **r**ank) are classical log-rank enhancement techniques proposed by Su et al. [7]. Both methods have simple configurations. LLR literally takes the ratio of log-likelihood to log-rank, as expressed in (6). Here, r_θ represents the rank when using P_{T_θ}.

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

On the other hand, NPR, like DetectGPT, performs the substitution of words in the text k times. It takes the ratio of the average log-rank of the obtained substituted texts to the log-rank of the original text. This is defined in (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 Binoculars

Hans et al. proposed Binoculars, a detection method utilizing two closely related language models, Falcon-7b [26] and Falcon-7b-instruct, by employing a metric called cross-perplexity [8]. The overall framework is illustrated in Figure 3.

Let the first model be denoted as M_1 (such as Falcon-7b), and the second model as M_2 (like Falcon-7b-instruct). In this case, using M_1, we calculate the log perplexity as shown in (8).

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

Next, using M_1 and M_2, we calculate the cross-perplexity, as shown in (9). Here, the symbol $\cdot$ represents the dot product.

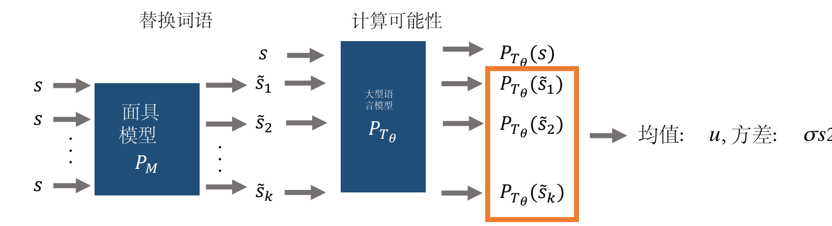

图1：DetectGPT 概述

3.2.5 快速检测GPT

在[6]中，Bao等人强调了Detect-GPT在替换和评分计算中使用不同模型的挑战，以及每次替换迭代都需要模型访问的成本相关问题。作为回应，FastDetectGPT是一种修改过的检测器，减少了对模型的访问，解决了成本问题，同时允许进行替换。尽管该方法涉及设定与DetectGPT类似的假设，但并没有根本性的变化。它仍然基于"AI生成的文本可能接近最大似然，而人类生成的文本则不是"的假设进行操作。

我们在图2中展示了FastDetect-GPT的整体架构。在FastDetect-GPT中，替换过程被一种不依赖于掩码模型的替代方法所取代。与检测模型类似，它利用自回归模型，PT0和PU0可以是相同的。对第i个单词的替换涉及从下一个单词列表中随机提取一个单词，考虑到输入文本中第(i−1)个单词之前的上下文，并用所选单词替换该单词。换句话说，进行N次这种替换会产生替换文本~S，并且通过在单词选择过程中进行采样，替换过程可以在一次访问中生成k个替换文本。

后续的分数计算被省略，因为它遵循与DetectGPT相同的程序。

3.2.6 LLR 和 NPR

LLR（似然对数秩比）和NPR（标准化扰动对数秩）是经典的对数-Su等人提出的排名增强技术[7]。这两种方法的配置都很简单。LLR字面上取对数似然与对数排名的比率，如(6)所示。这里，rθ表示使用PTθ时的排名。

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

另一方面，NPR与DetectGPT一样，在文本中进行k次单词替换。它计算获得的替换文本的平均对数排名与原始文本的对数排名的比率。这在（7）中定义。

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

3.2.7 双筒望远镜

汉斯等人提出了双目镜，这是一种利用两个密切相关的语言模型（Falcon-7b和Falcon-7b-instruct）的检测方法，采用了一种称为交叉困惑度的度量。整体框架如图3所示。

将第一个模型表示为 M1（例如 Falcon-7b），将第二个模型表示为 M2（如 Falcon-7b-instruct）。在这种情况下，使用 M1，我们计算如 (8) 所示的对数困惑度。

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

接下来，我们使用 M1 和 M2 计算交叉困惑度，如 (9) 所示。这里，符号 · 代表点积。

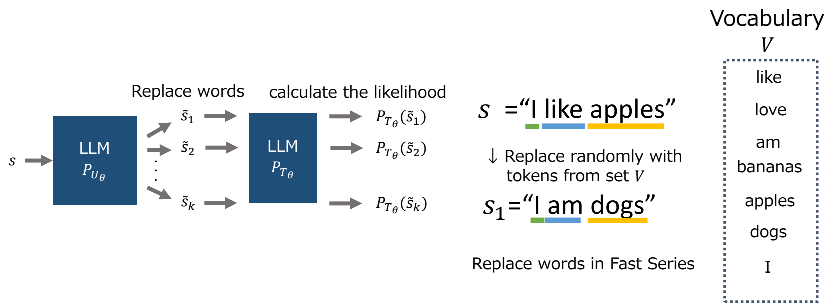

Figure 2: FastDetectGPT and Sampling Overview

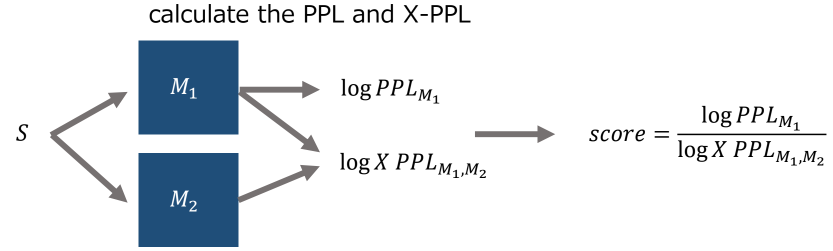

Figure 3: Binoculars Overview

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

The score in Binoculars is determined by (10).

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

4 Proposal

In this study, we propose a detection flow to investigate the impact of prompts on likelihood.

Before presenting the experimental setup, we introduce an additional detection method.

4.1 FastNPR

Word replacements in NPR are performed using a masked model. In this research, aiming for cost reduction, we also employ FastNPR, a method that replaces word replacements with sampling, akin to FastDetectGPT.

4.2 Detection methods

We explain the detection methodology. For the purpose of the explanation, let x represent the text to be detected, and if x is an AI-generated text, let p denote the prompt used for its generation. Detection can be categorized into two patterns: Black-box detection and White-box detection. An overview is presented in Figure 4.

Black-box detection occurs when the detector is unaware of prompt information, essentially mirroring existing detection methods. In this scenario, only the content of x is provided to the detector.

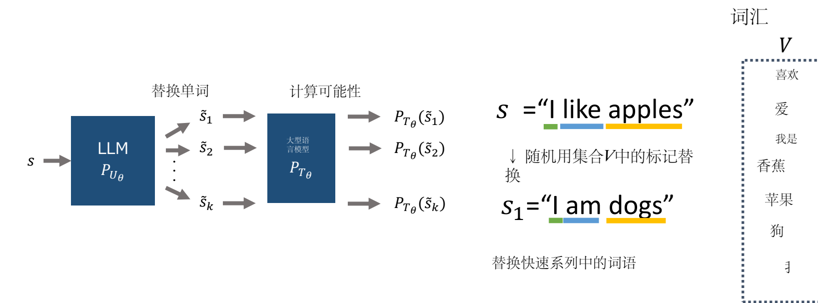

图2：FastDetectGPT和采样概述

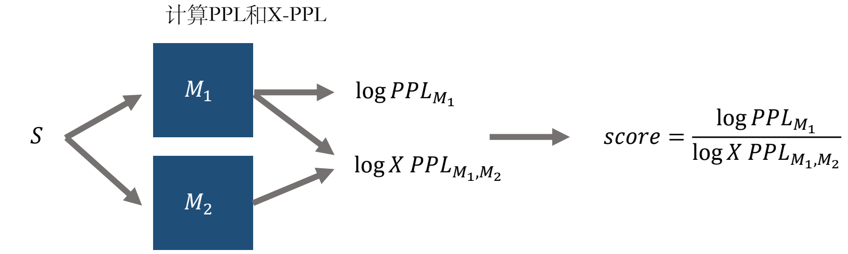

图3：望远镜概述

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

在双筒望远镜中的得分由（10）决定。

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

4 提案

在这项研究中，我们提出了一种检测流程，以调查提示对可能性的影响。

在介绍实验设置之前，我们介绍了一种额外的检测方法。

4.1 快速NPR

在NPR中，单词替换是通过掩码模型执行的。在这项研究中，为了降低成本，我们还采用了FastNPR，这是一种使用采样替代单词替换的方法，类似于FastDetectGPT。

4.2 检测方法

我们解释检测方法。为了说明，设 x 代表待检测的文本，如果 x 是 AI 生成的文本，则 p 表示用于生成它的提示。检测可以分为两种模式：黑箱检测和白箱检测。概述见图 4。

黑箱检测发生在检测器无法获取提示信息时，本质上反映了现有的检测方法。在这种情况下，检测器仅获得 x 的内容。

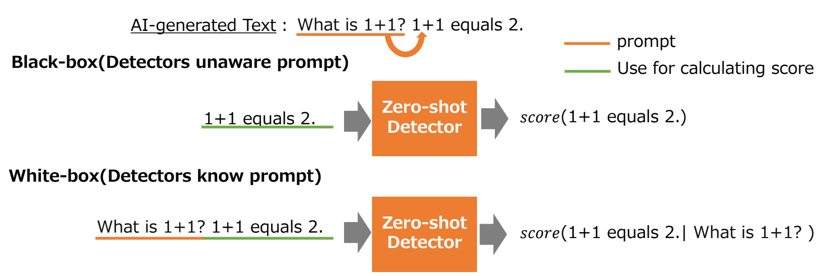

Figure 4: Proposed Detection Methods Overview

White-box detection, on the other hand, involves the detector having knowledge of prompt information. For human-generated text, only x is input. In the case of AI-generated text, the input consists of $p+x$. It is important to note that, in White-box detection, the prompt is used solely for likelihood calculation and is not included in the score computation.

5 Experiment

5.1 Configuration

To begin, we utilize the GPT2-XL [23] as the detection model, excluding Binoculars. Due to GPU constraints, Binoculars employs the pre-trained and instruct-tuned Phi1.5 [27] instead of Falcon.

For DetectGPT and NPR, we generate five replacement sentences for 10% of the entire text, while the Fast series generates 10,000 replacement sentences. T5-Large [24] is used for word replacement in DetectGPT and NPR, while the Fast series employs the GPT2-XL, the same detection model.

Also, we use the XSum dataset [28]. For human-generated text, we extract 200 samples from the XSum dataset, and for AI-generated text, we employ the Llama2 7B Chat model [25], generating up to 200 tokens. The prompt used is "Would you summarize the following sentences, please? text".

5.2 Result

As evident from the results in Table 2, white-box detection exhibits higher accuracy, while black-box detection shows lower accuracy. As anticipated, modifying likelihood through prompts leads to a decrease in the detection accuracy of likelihood-based detectors. Notably, there is a consistent decrease of 0.1 or more across all methods, highlighting a significant observation.

Table 2: Detection of Generated Summaries: Discrepancies Between Cases with and Without Prompts

Method	Black-box	White-box
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

Binoculars and the Fast series detectors demonstrate robustness compared to other methods. In particular, the Fast series detector maintains the same scoring calculation as conventional methods, suggesting robustness factors in the sampling process. For further verification, we conduct additional experiments.

In this experiment, we investigate the differences in detection accuracy when varying the replacement ratio, indicating the extent to which tokens in the text are replaced, and the sample size, representing the number of replacement sentences. DetectGPT and

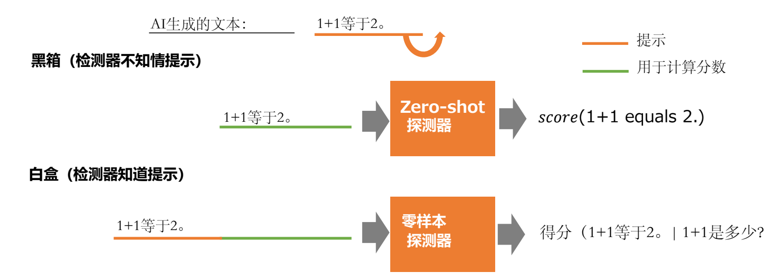

图4：建议的检测方法概述

白盒检测则涉及检测器对提示信息的了解。对于人类生成的文本，仅输入 x。在 AI 生成的文本中，输入由 p + x 组成。重要的是要注意，在白盒检测中，提示仅用于概率计算，而不包括在评分计算中。

5 实验

5.1 配置

首先，我们使用GPT2-XL [23]作为检测模型，排除双目视觉。由于GPU的限制，双目视觉使用预训练和指令调优的Phi1.5 [27]，而不是Falcon。

对于DetectGPT和NPR，我们为整个文本的10%生成五个替换句子，而Fast系列生成10,000个替换句子。T5-Large [24]用于DetectGPT和NPR中的单词替换，而Fast系列则使用GPT2-XL，作为相同的检测模型。

此外，我们使用了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AI生成的文本，我们使用Llama2 7B Chat模型[25]，生成最多200个标记。使用的提示是"请您总结以下句子吗？文本"。

5.2 结果

如表2所示，白盒检测表现出更高的准确性，而黑盒检测则...

表2：生成摘要的检测：有提示与无提示案例之间的差异

方法	黑箱	白盒
检测GPT	0.453	1.000
快速检测GPT	0.819	0.958
LRR	0.532	0.995
NPR (美国国家公共电台)	0.560	0.934
快速NPR	0.768	0.993
熵	0.330	0.978
对数似然	0.474	0.998
排名	0.432	0.977
对数秩检验	0.485	0.999
双筒望远镜	0.877	0.999

检测显示出较低的准确性。正如预期的那样，通过提示修改似然性导致基于似然的检测器的检测准确性下降。值得注意的是，所有方法的准确性均一致下降0.1或更多，突显了一个重要的观察结果。

双筒望远镜和Fast系列探测器相比其他方法展示了更强的鲁棒性。特别是，Fast系列探测器保持与传统方法相同的评分计算，这表明在采样过程中存在鲁棒性因素。为了进一步验证，我们进行额外的实验。

在这个实验中，我们研究了在改变替换比例（指文本中被替换的标记的程度）和样本大小（代表替换句子的数量）时检测准确性的差异。DetectGPT 和

NPR require the use of a masked language model to replace plausible tokens, making replacement not always feasible, especially for higher replacement percentages. Therefore, we primarily vary the replacement ratio in the Fast series to conduct the investigation.

The results for DetectGPT are presented in Table 3, and the results for NPR are shown in Table 4. From these results, it is evident that increasing the replacement ratio and sample size helps mitigate the decrease in detection accuracy. This observation is similar to Chakraborty et al.'s assertion that increasing the sample size can enable detection if the distribution slightly differs [29].

However, in our validation, the improvement in accuracy plateaus at around 10 samples, reaching a maximum AUC of approximately 0.8, which is not considered high. Particularly in recent years, there is a trend toward practical applications, emphasizing high true positive rates at low false positive rates, suggesting that at least an AUC in the late 0.9s would be necessary [31, 8]. Furthermore, the lack of improvement in detection accuracy with DetectGPT and NPR may be attributed to the limited number of substitutable tokens.

Table 3: Effect of Substitution Rate(SR) and Sample Size(SS) Variation on AUC(DetectGPT)

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

Table 4: Effect of Substitution Rate(SR) and Sample Size(SS) Variation on AUC(NPR)

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 Discussions

6.1 Hypotheses for zero-shot detectors

While our investigation has focused solely on prompts, similar phenomena could potentially be observed with other elements. For instance, variations in Temperature or Penalty Repetition between the generation and detection stages might introduce differences in the selected tokens, making detection challenging based on likelihood. Generalizing these observations, we hypothesize that any act that fails to replicate the likelihood during language generation could undermine the detection accuracy of Zero-shot detectors relying on likelihood from next-word prediction.

6.2 Common tasks

While our investigation has focused on summary text generation, there are several other potential tasks to consider, such as paraphrase generation, story generation, and translation text generation. It is plausible that detection accuracy could also decrease in these common tasks. Since these tasks may be utilized without malicious intent, it is crucial to conduct similar evaluations for them.

NPR要求使用掩码语言模型来替换合理的标记，这使得替换并不总是可行，尤其是在较高的替换百分比下。因此，我们主要在Fast系列中改变替换比例以进行研究。

DetectGPT的结果呈现在表3中，NPR的结果显示在表4中。从这些结果来看，增加替换比例和样本大小有助于减轻检测准确度的下降。这一观察与Chakraborty等人的论断相似，即如果分布略有不同，增加样本大小可以实现检测[29]。

然而，在我们的验证中，准确率的提升在大约10个样本时达到了平稳状态，最大AUC约为0.8，这并不算高。特别是在近年来，实际应用的趋势强调在低假阳性率下的高真正阳性率，这表明至少需要在0.9的后期达到AUC [31, 8]。此外，DetectGPT和NPR在检测准确性方面缺乏改进可能归因于可替代令牌的数量有限。

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

方法	SR	SS	曲线下面积
快速检测GPT	10%	5	0.640
快速检测GPT	20%	5	0.697
快速检测GPT	100%	5	0.779
快速检测GPT	10%	10	0.704
快速检测GPT	20%	10	0.739
快速检测GPT	100%	10	0.821
快速检测GPT	100%	10000	0.819
检测GPT	10%	5	0.453
检测GPT	20%	5	0.522
检测GPT	30%	5	0.490
检测GPT	10%	10	0.446
检测GPT	30%	10	0.446

表4：替代率（SR）和样本大小（SS）变化对AUC（NPR）的影响

方法	SR	SS	曲线下面积
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
国家公共广播电台	10%	5	0.560
NPR	20%	5	0.590
美国国家公共电台	30%	5	0.577
美国国家公共电台	10%	10	0.589
美国国家公共广播电台	30%	10	0.588

6 次讨论

6.1 零样本检测器的假设

虽然我们的调查仅专注于提示，但类似现象可能在其他元素中 例如，变化-

在生成和检测阶段的温度或惩罚重复的变化可能会导致所选标记之间的差异，从而使基于可能性的检测变得具有挑战性。概括这些观察结果，我们假设任何未能在语言生成过程中复制可能性的行为都可能削弱依赖于下一个单词预测的可能性的零-shot检测器的检测准确性。

6.2 常见任务

虽然我们的研究集中在摘要文本生成上，但还有其他几个潜在的任务需要考虑，例如释义生成、故事生成和翻译文本生成。在这些常见任务中，检测准确性也可能下降。由于这些任务可能在没有恶意意图的情况下被使用，因此对它们进行类似的评估至关重要。

6.3 Relevance to paraphrase attacks

Paraphrase generation, as briefly discussed in the previous section, assumes a single act. However, currently known paraphrase attacks [30, 31, 18, 13] involve generating paraphrases for each sentence and combining the results. While paraphrase attacks using masked language models may have a slightly different structure, as they utilize both preceding and succeeding contexts for word replacement, it can be argued that reproducing likelihood during detection becomes challenging. Therefore, paraphrase attacks can be viewed as more complex versions of the tasks verified in this study.

6.4 Text length

In the current experiment, the generated texts were fixed at 200 tokens. The length of tokens may impact the ease of reproducing likelihood. Therefore, it would be beneficial to conduct further verification with longer texts. Tasks such as narrative generation, where the length of the text is not a concern, may be suitable for such investigations.

6.5 Number of parameters

In this study, each detection method utilized a language model of approximately 1 billion parameters. It would be of interest to investigate whether increased robustness can be observed when experimenting with larger language models. Conversely, there are experimental studies that have demonstrated the ability of smaller language models to achieve a higher likelihood for AI-generated texts across a broader range of language models [32]. Considering these findings, conducting experiments with smaller language models and verifying if there are differences in robustness could also provide valuable insights.

6.6 Relationship with supervised learning detectors

Even when using supervised learning, it has been noted that generated text from prompt-based tasks may exhibit decreased detection accuracy [15]. However, there is a possibility that these models could be more robust compared to zero-shot detectors. For instance, RADAR [13] achieved an AUC of 0.939 in the task used in this experiment. In comparison, the RoBERTa-large detector [12] had an AUC of 0.767. This suggests that robust detectors against paraphrase attacks might demonstrate similarly robust results in other tasks.

6.7 Relationship with watermarking

Watermarking techniques utilize statistical methods for verification [16]. Since these methods are based on likelihood during both generation and verification, a failure to reproduce likelihood during the verification stage may lead to a decrease in accuracy. On the other hand, robust watermarking techniques against paraphrase attacks have emerged [17]. These methods may exhibit robustness against prompts as well.

6.8 Towards resilient zero-shot detectors

Currently, many methods perform likelihood-based detection. Combining these approaches with other methods may lead to more robust detection. One such approach is Intrinsic Dimension [11]. Intrinsic Dimension refers to the minimum dimension needed to represent a given text. Tulchinskii et al. propose a detector based on Persistent Homology to estimate the Intrinsic Dimension and use it as a score. However, this method requires a constant length of text and was not applicable in our experiment. It would be interesting to explore the application of this method in experiments involving longer texts.

Approaches utilizing representations obtained with masked language models, including Intrinsic Dimension, calculate likelihood in a different way from the detectors used in our experiment, which are based on autoregressive language models. Combining these elements could lead to the development of a more robust zero-shot detector.

6.3 与释义攻击的相关性

如前一节简要讨论的，释义生成假设为单一行为。然而，目前已知的释义攻击涉及为每个句子生成释义并组合结果。虽然使用掩码语言模型的释义攻击可能具有稍微不同的结构，因为它们利用前后文进行词替换，但可以认为在检测过程中重现可能性变得具有挑战性。因此，释义攻击可以被视为本研究中验证任务的更复杂版本。

6.4 文本长度

在当前实验中，生成的文本固定为200个标记。标记的长度可能会影响重现可能性的难易程度。因此，进行更长文本的进一步验证将是有益的。叙事生成等任务，文本长度不是问题，可能适合进行此类研究。

6.5 参数数量

在这项研究中，每种检测方法都利用了大约10亿参数的语言模型。研究更大语言模型时是否能观察到增强的鲁棒性将是一个有趣的课题。相反，有实验研究表明，较小的语言模型能够在更广泛的语言模型中实现对AI生成文本的更高可能性[32]。考虑到这些发现，进行较小语言模型的实验并验证鲁棒性是否存在差异也可能提供有价值的见解。

6.6 与监督学习检测器的关系

即使在使用监督学习时，也注意到基于提示的任务生成的文本可能表现出较低的检测准确性[15]。然而，这些模型可能比零样本检测器更具鲁棒性。例如，RADAR [13] 在本实验中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，针对同义句攻击的鲁棒检测器在其他任务中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可能性，因此在验证阶段未能重现可能性可能会导致准确性下降。另一方面，针对释义攻击的强健水印技术已经出现[17]。这些方法也可能对提示表现出强健性。

6.8 朝着具有弹性的零样本检测器迈进

目前，许多方法采用基于似然的检测。将这些方法与其他方法结合可能会导致更强大的检测。其中一种方法是内在维度 [11]。内在维度是指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调的检测器，用于估计内在维度并将其作为评分。然而，这种方法需要文本长度恒定，在我们的实验中不适用。探索该方法在涉及更长文本的实验中的应用将是很有趣的。

利用掩码语言模型获得的表示方法，包括内在维度，以不同于我们实验中使用的基于自回归语言模型的探测器的方式计算可能性。结合这些元素可能会导致更强大的零-shot 探测器的发展。

Acknowledgement

This research was supported in part by JSPS international scientific exchanges between Japan and India, Bilateral Program DTS-JSP, grant number JPJSBP120227718, and the Kayamori Foundation of Informational Science Advancement.

References

[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., Strobelt, 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 of Machine-Generated Text. 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.

确认

本研究部分得到了日本学术振兴会（JSPS）在日本与印度之间的国际科学交流、双边项目DTS-JSP（资助编号JPJSBP120227718）以及香取信息科学促进基金会的支持。

参考文献

[1] OpenAI. (2023). GPT-4 技术报告，arXiv 电子印刷品。

[2] 微软。微软Copilot，检索于十月 2023年31月，来自https://adoption.microsoft.com/ja-jp/copilot/。

团队，G.，Anil，R.，Borgeaud，S.，Wu，Y.， Alayrac, J. B., Yu, J., ... & Ahn, J. (2023). Gem-ini：一个高能力多模态模型的家族。arXiv预印本 arXiv:2312.11805。

[4] Gehrmann, S., Strobelt, H., & Rush, A. (2019). GLTR：生成文本的统计检测与可视化。在M. R. Costajuss`a和E. Alfonseca（编辑），《计算语言学协会第57届年会会议录：系统演示》（第111－116页）。计算语言学协会。

[5] 米切尔, E., 李, Y., 哈扎茨基, A., 曼宁, C. D. 和 Finn, C. (2023). DetectGPT: 使用概率曲率进行零样本机器生成文本检测. 在第40届国际机器学习会议（ICML’23）论文集 (第202卷, 页24950－24962). JMLR.org

包, G., 赵, Y., 滕, Z., 杨, L., & 张, Y. (2023). Fast-DetectGPT：通过条件概率曲率高效零样本检测机器生成文本。arXiv 预印本 arXiv:2310.05130。

[7] 苏, J., 朱涛, 检测LLM 利用 日志 零样本检测的排名信息 机器生成文本。arXiv预印本arXiv:2306.05540。

汉斯，A.，施瓦茨希尔德，A.，切列潘诺娃，V.， Kazemi, H., Saha, A., Goldblum, M., ... & Gold- stein, T. (2024). 用双筒望远镜识别大型语言模型：零样本检测机器生成文本。arXiv 预印本 arXiv:2401.12070。

刘, S., 刘, X., 王, Y., 程, Z., 李, C., 张, Z., ... & 沈, C. (2024). DoesDetectGPT是否充分利用扰动？基于模型的对比学习检测器的选择性扰动会更好。arXiv预印本 arXiv:2402.00263。

[10] 萨斯，K.，巴哈姆，S.，凯伊，E. S.，& 斯塔利，E. W. (2024). 爆发还是不爆发：生成和量化不可能的文本。arXiv 预印本 arXiv:2401.15476。

图尔钦斯基，E.，库兹涅佐夫，K.，库什纳列娃，L.，切尔尼亚夫斯基，D.，巴拉尼科夫，S.，皮翁特科夫斯卡娅，I.，... & 布尔纳耶夫，E.（2023）。用于稳健检测AI生成文本的内在维度估计。arXiv预印本arXiv:2306.04723。

[12] 索莱曼, I., 布伦代奇, M., 克拉克, J., 阿斯凯尔, A., 赫伯特-沃斯, A., 吴, J., ... & 王, J. (2019). 发布策略与语言模型的社会影响. arXiv 预印本 arXiv:1908.09203.

[13] 胡， X. 陈， P. Y. & 你好， T. Y.（2023）。RADAR：通过对抗学习实现稳健的AI文本检测。arXiv预印本arXiv:2307.03838。

[14] 斗, Z., 郭, Y., 常, C. C., 阮, H. H. 和 Echizen, I. (2024). 提升学术写作中大型语言模型合成文本检测器的鲁棒性：综合分析。arXiv 预印本 arXiv:2401.08046。

[15] 刘, Z., 姚, Z., 李, F., & 罗, B. (2023). 如果你能的话，检查我：使用CheckGPT检测ChatGPT生成的学术写作。arXiv预印本 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 unified text-to-text transformer. The Journal of Machine Learning Research, 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 Blackbox Machine-Generated Text Detectors. arXiv preprint arXiv:2305.09859.

[16] Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I. & Goldstein, T. (2023). 大型语言模型的水印。第40届国际机器学习会议论文集，机器学习研究论文集 202:17061-17084。

任, J., 许, H., 刘, Y., 崔, Y., 王, S., Yin, D., & Tang, J. (2023). 一种基于语义的强健水印，用于防止大型语言模型的改写。arXiv 预印本 arXiv:2311.08721。

[18] 施, Z., 王, Y., 尹, F., 陈, X., 常, K. W.，和谢，C. J.（2023）。使用语言模型进行红队语言模型检测器。arXiv预印本arXiv:2305.19713。

[19] 小池, R., 兼子, M., & 岡崎, N. (2023). Outfox：通过对抗生成示例的上下文学习进行LLM生成的论文检测。arXiv预印本arXiv:2307.11729。

[20] 陆, N., 刘, S., 何, R., & 唐, K. (2023). 大型语言模型可以被引导以规避AI生成文本检测。arXiv预印本 arXiv:2305.10847。

[21] Kumarage, T., Sheth, P., Moraffah, R., Gar-Land, J.，& Liu, H.（2023）。AI生成文本检测器的可靠性如何？使用规避性软提示的评估框架。arXiv预印本 arXiv:2310.05095。

[22] OpenAI. (2023). 新的AI分类器用于指示 AI撰写的文本，检索于2023年11月30日。

[23] 拉德福德, A., 吴, J., 奇尔德, R., 卢安, D., 阿莫代，D.，& 苏茨克弗，I.（2019）。语言模型是无监督的多任务学习者。2023年10月31日检索自 https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_lear

[24] Raffel, C., Shazeer, N., Roberts, A., Lee, K., 纳朗，S.，马特纳，M.，... & 刘，P. J.（2020）。探索迁移学习的极限与一个统一文本到文本的变换器。《机器学习研究杂志》，21(1)，5485-5551。

[25] 图夫龙，H.，马丁，L.，斯通，K.，阿尔-bert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2：开放基础和微调聊天模型。arXiv 预印本 arXiv:2307.09288。

[26] 阿尔马兹鲁伊，E.，阿洛贝德利，H.，阿尔沙姆西，A.，Cappelli, A., Cojocaru, R., Debbah, M., ... & Penedo, G. (2023). 猎鹰系列开放语言模型。arXiv 预印本 arXiv:2311.16867。

[27] 李, Y., 布贝克, S., 埃尔丹, R., 德尔 · 乔尔诺, A., Gunasekar, S.，& Lee, Y. T.（2023）。教科书就是你所需要的一切 ii：phi-1.5 技术报告。arXiv 预印本 arXiv:2309.05463。

[28] Narayan, S., Cohen, S. B., & Lapata, M. (2018). 不要给我细节，只给我总结！主题感知卷积神经网络用于极端摘要。在E. Riloff, D. Chiang, J. Hockenmaier和J. Tsujii（编辑），2018年自然语言处理实证方法会议论文集（第1797-1807页）。计算语言学协会

[29] 查克拉博提，S.，贝迪，A. S.，朱，S.，安，B.，Manocha, D.，& Huang, F.（2023）。关于人工智能生成文本检测的可能性。arXiv预印本 arXiv:2304.04736。

[30] Sadasivan, V. S., Kumar, A., Balasubrama-年, S., 王, W., & 费兹, S. (2023). 人工智能生成的文本能否被可靠检测？arXiv 预印本 arXiv:2303.11156.

[31] 克里希纳，K.，宋，Y.，卡尔平斯卡，M.，维廷，J. 和 Iyyer, M. (2023). 释义可以避开 AI 生成文本的检测器，但检索是一种有效的防御手段。arXiv 预印本 arXiv:2303.13408。

[32] Mireshghallah, F., Mattern, J., Gao, S., Shokri, R.，& Berg-Kirkpatrick, T.（2023）。较小的语言模型是更好的黑箱机器生成文本检测器。arXiv 预印本 arXiv:2305.09859。