The Fortieth AAAI Conference on Artificial Intelligence (AAAI-26)

Model-Agnostic Sentiment Distribution Stability Analysis for Robust LLM-Generated Texts Detection

Siyuan Li[1,2], **Xi Lin**[1,2*], **Guangyan Li**[3], **Zehao Liu**[1,2], **Aodu Wulianghai**[1],
Li Ding[1,2], **Jun Wu**[1,2], **Jianhua Li**[1,2*]
[1]School of Computer Science, Shanghai Jiao Tong University, Shanghai, China
[2]Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai, China
[3]Institute of Automation, Chinese Academy of Sciences, Beijing, China
{siyuanli, linxi234, liuzehao, melusine.wlhad, liding, junwuhn, lijh888}@sjtu.edu.cn, liguangyan2022@ia.ac.cn

Abstract

The rapid advancement of large language models (LLMs) has resulted in increasingly sophisticated AI-generated content, posing significant challenges in distinguishing LLM-generated text from human-written language. Existing detection methods, primarily based on lexical heuristics or fine-tuned classifiers, often suffer from limited generalizability and are vulnerable to paraphrasing, adversarial perturbations, and cross-domain shifts. In this work, we propose ***`SentiDetect`***, a model-agnostic framework for detecting LLM-generated text by analyzing the divergence in sentiment distribution stability. Our method is motivated by the empirical observation that LLM outputs tend to exhibit emotionally consistent patterns, whereas human-written texts display greater emotional variability. To capture this phenomenon, we define two complementary metrics: sentiment distribution consistency and sentiment distribution preservation, which quantify stability under sentiment-altering and semantic-preserving transformations. We evaluate SentiDetect on five diverse datasets and a range of advanced LLMs, including **`Gemini-1.5-Pro`**, **`Claude-3`**, **`GPT-4-0613`**, and **`LLaMa-3.3`**. Experimental results demonstrate its superiority over state-of-the-art baselines, with over 16% and 11% F1 score improvements on Gemini-1.5-Pro and GPT-4-0613, respectively. Moreover, SentiDetect also shows greater robustness to paraphrasing, adversarial attacks, and text length variations, outperforming existing detectors in challenging scenarios.

Introduction

Advanced large language models (LLMs) have achieved remarkable proficiency in processing and generating natural language across diverse domains, from logical reasoning to creative writing (Yang et al. 2023). As these powerful generative models become increasingly integrated into writing workflows (Sun et al. 2024; Lei, Hsu, and Chen 2025), their outputs now permeate various forms of digital content, including news articles, academic papers, and social platforms (Yuan et al. 2022). However, LLM-generated content still poses numerous security risks and potential misuses, such as phishing and the creation of fake news, which can lead to fraud and factual inaccuracies (Lee et al. 2023; Sadasivan et al. 2023). Furthermore, the growing sophistication of LLM-generated text presents new challenges in distinguishing machine-generated texts from human-written texts (Chen and Wang 2025).

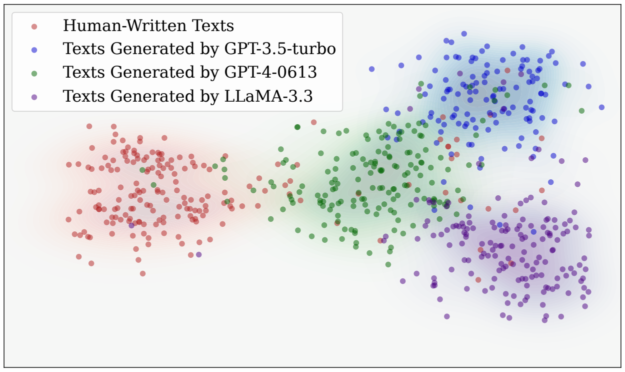


Figure 1: UMAP projection illustration of ***sentiment distribution stability features for LLM-generated texts and human-written*** in the Review dataset. Each point represents an embedded paragraph (≤ 64 tokens) encoded with Sentiment Distribution Consistency features. ***LLM-generated texts exhibit consistent sentiment patterns, forming distinguishable clusters from human-written texts***.

The emerging various challenge has sparked considerable research into developing *automatic LLM-generated text detection approaches*. To address the misuse of LLMs, there are three types of LLM-generated text detectors: watermark (Kirchenbauer et al. 2024), classification (Solaiman et al. 2019), and statistic-based detectors (Tulchinskii et al. 2024). Although watermark-based detectors can identify embedded watermarks in the output text, they are susceptible to paraphrasing (Krishna et al. 2024) and can be bypassed by open-source LLMs (Touvron et al. 2023). Supervised classification is effective within specific domains but struggles with text from new sources or unfamiliar LLMs (Pu et al. 2023). Existing statistic-based detectors analyze statistical features of texts to determine their origins and obtain comparable performance to supervised detectors. However, these methods require access to the source LLM, which is often unfeasible, and current detectors remain vulnerable to paraphrasing attacks (Krishna et al. 2024).

---

[*] Corresponding authors.

In contrast to prior approaches, we investigate a different perspective: ***(RQ) can sentiment patterns embedded in language be leveraged for authorship attribution without supervised training or access to model parameters?*** Motivated by this question, we conduct empirical studies and make a key observation: the stability of emotional expressions in text emerges as a strong and distinctive indicator for attributing authorship. Building on this insight, we propose the hypothesis that LLM-generated content tends to exhibit relatively stable sentiment distributions under specific "low-emotional" transformations, irrespective of textual genre or stylistic variations (Hu et al. 2023; Vijay, Priyanshu, and KhudaBukhsh 2025). This phenomenon likely stems from the nature of LLM training, which involves exposure to vast, uniformly curated corpora (Naveed et al. 2023; Shen et al. 2024). Figure 1 demonstrates how this feature can effectively distinguish LLM-generated text from human-written texts. These subtle yet measurable differences in sentiment fluctuation offer a previously underexplored fingerprint for distinguishing LLM-generated from human-written text.

Building on this insight, we present ***`SentiDetect`***, a novel model-agnostic framework that detects LLM-generated text by analyzing divergence in sentiment distribution stability. Motivated by the observation that LLM-generated content tends to exhibit more emotionally consistent patterns, SentiDetect quantifies this distinction through two complementary unsupervised metrics: sentiment distribution consistency and sentiment distribution preservation. These metrics capture stability under sentiment-altering and semantic-preserving transformations, enabling relatively interpretable detection without access to model internals or fine-tuned data. Comprehensive experiments across five datasets and multiple advanced LLMs demonstrate that SentiDetect significantly outperforms existing baselines in both accuracy and robustness, particularly under paraphrasing, adversarial, and length-variant settings.

The contributions are summarized as follows:

- ***First-of-its-kind study focusing on sentiment analysis for LLM-generated text.*** `SentiDetect` is a model-agnostic framework that identifies LLM-generated text through the divergence in sentiment distribution stability, requiring neither model access nor supervised training.
- ***Strong zero-shot performance on various LLMs and textual domains.*** Extensive evaluations are conducted on `Gemini-1.5-Pro`, `LLaMa-3.3`, `GPT-3.5-Turbo`, and `GPT-4-0613`. Besides, SentiDetect improves F1 score on five datasets, including news, code, essays, papers, and community comments by up to 22.61%.
- ***In-depth analysis of detection robustness.*** We provide an in-depth analysis of the robustness against various adversarial tactics, including paraphrasing, perturbation, and text length across various datasets and LLMs.

Related Works

LLM-Generated Text Detection

Current LLM-generated text detection approaches can be categorized into three paradigms: supervised classification, watermarking techniques, and statistical analysis methods (Chakraborty et al. 2024).

Supervised Training Detectors. Classification-based methods treat the detection as a binary prediction task. Early works like Grover (Zellers et al. 2019) generate articles from titles, and OpenAI employed a fine-tuned RoBERTa model to classify GPT-2 outputs, noting larger models produce better text (Solaiman et al. 2019). Furthermore, adversarial learning can also be introduced to train the robust detector (Hu, Chen, and Ho 2024). However, these supervised methods require extensive labeled datasets and constant updates to keep pace with evolving generative models (Soto et al. 2024; Zhang et al. 2024).

Watermarking Detectors. Watermark-based approaches embed identifiable patterns during text generation. Initial work included adversarial watermarking transformers for binary message encoding (Abdelnabi and Fritz 2021), with recent advances proposing probability distribution perturbations for more robust marking (Kirchenbauer et al. 2023). Despite effective watermarks resisting removal without degrading text quality, these methods struggle with conditional text generation (Fu, Xiong, and Dong 2024) and are vulnerable to paraphrasing (Krishna et al. 2024).

Statistical Analysis Detectors. Statistical-based detectors (Bao et al. 2024) identify abnormal values in text features like entropy and n-gram frequency. OpenAI's detector uses the logarithmic probability of model (Solaiman et al. 2019), while GLTR employs entropy and probability rank. Recently, DetectGPT demonstrated that LLM-generated text often falls in the negative curvature region of the LLM's logarithmic probability function, resulting in a curvature-based classifier (Mitchell et al. 2023). Subsequent studies have developed a faster version of this method (Bao et al. 2024). These methods provide valuable insights into the distinctive features of LLM-generated text without supervised training.

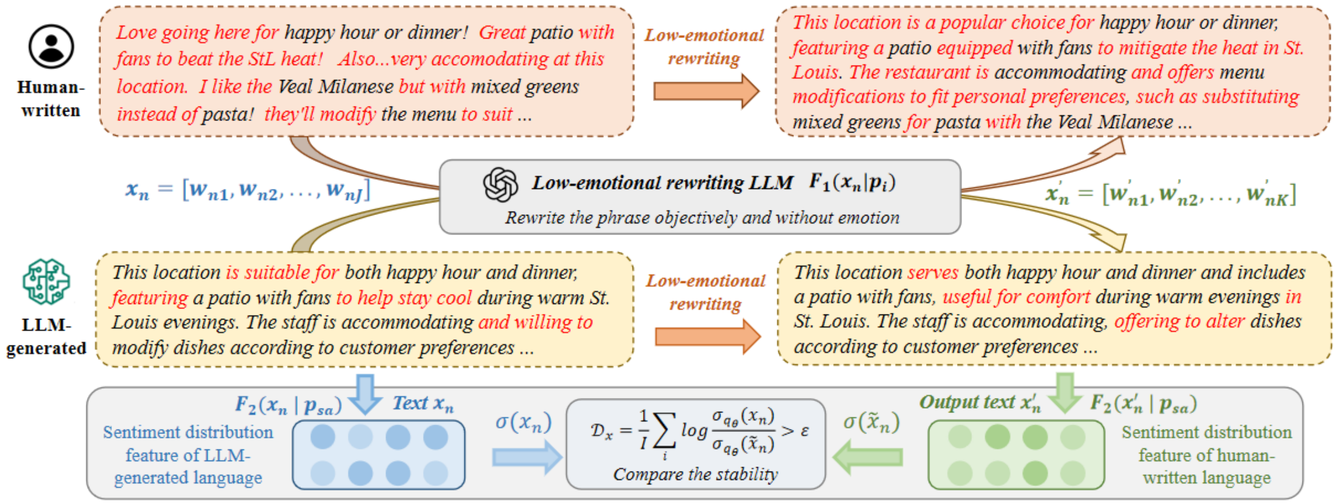
Sentiment Analysis in the Era of LLMs

Recent research has expanded sentiment analysis beyond basic classification to more nuanced examination of textual sentiment patterns, such as aspect-based or multifaceted subjective analysis (Zhang et al. 2022). The understanding ability of LLM has a significant influence on sentiment analysis (Zhong et al. 2023). Notably, researchers have explored innovative methods to leverage generative models for sentiment analysis tasks. Deng et al. developed a semi-supervised framework that utilizes language models to generate sentiment labels for training smaller models (Deng et al. 2023), highlighting the potential of LLMs in analyzing sentiment features. Besides, LLMs consistently outperform smaller models for zero-shot sentiment analysis of diverse text corpora (Zhang et al. 2024). Therefore, for the sentiment stability analysis considered in this work, we designed a pipeline combined with LLMs to implement the idea.

Methodology

LLM-Generated Text Detection Definition

Given the probability measures $\mathcal{P}$ and $\mathcal{Q}_\theta$ on metric space $\mathcal{X}$, suppose we have a set of IID candidate texts $\{x_n\}_{n=1}^N$,



Love going here for happy hour or dinner! Great patio with fans to beat the StL heat! Also...very accomodating at this location. I like the Veal Milanese but with mixed greens instead of pasta! they'll modify the menu to suit ...

This location is a popular choice for happy hour or dinner, featuring a patio equipped with fans to mitigate the heat in St. Louis. The restaurant is accommodating and offers menu modifications to fit personal preferences, such as substituting mixed greens for pasta with the Veal Milanese ...

This location is suitable for both happy hour and dinner, featuring a patio with fans to help stay cool during warm St. Louis evenings. The staff is accommodating and willing to modify dishes according to customer preferences ...

This location serves both happy hour and dinner and includes a patio with fans, useful for comfort during warm evenings in St. Louis. The staff is accommodating, offering to alter dishes according to customer preferences ...

Figure 2: Illustration of the proposed `SentiDetect`. Detecting LLM-generated text by sentiment distribution stability analysis through low-emotional rewriting. See Appendix A for more examples of low-emotional rewriting.

which are either from the human-written distribution $\mathcal{P}$ or LLM-generated distribution $\mathcal{Q}_\theta$. Our goal is to determine whether each x_n originates from the $\mathcal{Q}_\theta$ of source q_θ:

$$\hat{y}_n = \underset{y_n \in \{y_{\mathcal{P}}, y_{\mathcal{Q}_\theta}\}}{\operatorname{argmax}} P(y_n \mid x_n, N, \mathcal{P}, \mathcal{Q}_\theta). \tag{1}$$

Here, $x_n = [w_{n,1}, w_{n,2}, \dots, w_{n,J}]$ denotes the n-th text sequence, $w_{n,j}$ represent the j-th word in x_n. When $N = 1$, the paradigm collapses into a single-instance detection task.

Overview of SentiDetect. An overview of the designed detection pipeline is shown in Figure 2, since LLMs are less effective at directly conducting complex sentiment feature analysis. The pipeline comprises three stages: *(I) Low-emotional rewriting (LER)*: Using zero-shot LLMs to rewrite the text under ”low-emotional” instructions; *(II) Sentiment feature extraction*: Extracting sentiment distribution features from both original and rewritten texts; *(III) Stability divergence analysis*: Measuring the difference in stability of sentiment distributions to identify the input text.

Sentiment Distribution Stability Analysis

While LLM-generated text often appears contextually appropriate, its sentiment distribution patterns exhibit subtle yet measurable differences from human-written text. Our finding reveals this key phenomenon as follows:

> **Divergence in low-emotional stability**: *LLM-generated text* ***maintains consistent sentiment distribution patterns*** *during low-emotional rewriting, while human-written text exhibits significant shifts in sentiment expression patterns.*

Besides, to operationalize this hypothesis, we implement the detection pipeline as illustrated in Figure 2. First, we introduce a low-emotional but semantic-preserving rewriting function $F_1(\cdot \mid p_i)$ to modify the sentiment pattern of the text x_n strategically. To be specific, $F_1(\cdot \mid p_i)$ maps text x_n to $x_n' = [w_{n,1}', w_{n,2}', \dots, w_{n,K}']$. We repeat the above process for each p_i to ensure robust pattern capture.

Subsequently, the sentiment distribution analyzer $F_2(\cdot \mid p_{sa})$ is employed to quantify the sentiment distribution feature of both the original text x_n and its rewritten versions x', as shown in Figure 2. The function $F_2(\cdot)$ maps each input text x_n to a fixed-length vector representation $F_2(x_n \mid p_{sa})$, referred to as the sentiment distribution pattern. To construct this vector, we apply F_2 across semantic-preserving LERs of the original text, denoted as $x_{n,1}', \dots, x_{n,I}'$. This process yields I sentiment distributions, which are then concatenated to form a unified feature embedding. This embedding serves as a stable representation for measuring divergence across different samples. Specifically, the $F_2(\cdot \mid p_{sa})$ in the experiments is implemented using a standard 3-class sentiment classifier, which outputs a probability vector over the classes (negative, neutral, positive). This level of granularity empirically captures patterns of sentiment stability while avoiding the ambiguity introduced by overly fine-grained labels. The sentiment distributions of all the rewritten versions are concatenated, resulting in a fixed-length signal that enables a reliable comparison between the samples.

The sentiment distribution pattern of text x is denoted as $\sigma(x_n) = F_2(x_n \mid p_{sa})$. By comparing the sentiment distribution before and after LER, we can quantify the distribution divergence as $\log(\sigma_{q_\theta}(x_n)) - \log(\sigma_{q_\theta}(x_{n,i}'))$, highlighting differences in sentiment stability between LLM outputs and human-written text. The parameters p_i and p_{sa} represent carefully designed transformation prompts. With the help of the concepts of functions $F_1(\cdot \mid p_i)$ and $F_2(\cdot \mid p_{sa})$, we define the distribution stability divergence in sentiment features as follows:

$$\mathcal{D}(x_n, q_\theta) \triangleq \log \sigma(x_n) - \mathbb{E}_{x_{n,i}' \sim F_1(\cdot|p_i)} \log \sigma(x_{n,i}'). \tag{2}$$

This is referred to as the sentiment distribution stability divergence hypothesis, which describes the characteristic differences between LLM-generated and human-written texts.

LLM-Generated Text Detection via Sentiment Distribution Stability Divergence

Based on the empirical hypothesis, LLM-generated texts and human-written texts can be distinguished by the change of the sentiment distribution feature:

$$\mathcal{D}_x = \frac{1}{I}\sum_{i}\log\Big(\frac{\sigma_{q_\theta}(x)}{\sigma_{q_\theta}(F_1(x \mid p_i))}\Big). \tag{3}$$

The input x will be identified as LLM-generated text when the divergence $\mathcal{D}_x < \varepsilon$, where ε is the decision threshold.

LER Rewriting Instructions. For the implementation of the sentiment distribution stability analysis, we design a set of transformation instructions that guide language models to modify the emotional tone of a text while preserving its semantic content. For example, instructions such as "*Decrease the emotional intensity while preserving semantic content*" or "*Rewrite this paragraph objectively*" serve to elicit controlled variations in sentiment. We formally define each instruction as p_i, drawn from templates like "*Rewrite objectively*" or "*Use a machine-like tone*", and optimize them using AutoPrompt (Ma et al. 2024) to ensure effectiveness and consistency. These transformation instructions are simple in form yet central to our method: they induce minimal changes in LLM-generated text due to the inherent regularity of its sentiment structure, while often triggering more noticeable sentiment shifts in human-written text. This contrast serves as a key signal for distinguishing between LLM-generated and human-written text.

To capture this phenomenon, we implement SentiDetect by leveraging two key properties of sentiment distribution stability: *sentiment distribution consistency (SDC)* and *sentiment distribution preservation (SDP)*.

Sentiment Distribution Consistency. Language models show minimal changes in sentiment distributions when transforming their own generated text, as autoregressive models tend to produce content with stable pattern features. Therefore, we use the low-emotional invariance between output and input to measure the sentiment distribution recognition of LLM towards a given input text x. We use prompt p_i to prompt LLM q_θ to perform LER transformation on text x. If the text x is generated from LLM, *the degree of change from output to input should be small.* We quantify the sentiment distribution consistency through:

$$\begin{aligned}\mathrm{SDC}(x) &= \mathbb{E}_{p_i}\left[\|\log\sigma(x), \log\sigma(x')\|_1\right] \\ &= \frac{1}{I}\sum_{i=1}^{I}(\log\sigma(x) - \log\sigma(F_1(x \mid p_i))).\end{aligned} \tag{4}$$

For the instruction p_i, we present some of them:

- *Please rewrite this more straightforwardly.*
- *Polish this in a machine-like objective tone.*

We employ automated methods to optimize the instruction p_i (Ma et al. 2024).

Sentiment Distribution Preservation. In addition, we give the definition of the other key property, i.e. sentiment distribution preservation. Suppose we perform sentiment-independent and inverse transformations (such as converting people or extensions and abbreviations) on the texts. In that case, it will *output with the same sentiment distribution as the input text in the previous*. We refer to the property that LLM-generated text maintains stable sentiment distributions under semantic-preserving transformations as sentiment distribution preservation. Besides, we elaborate instructions to represent a pair of inverse mapping pairs $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$. $\mathcal{F}(\cdot)$ represents requesting LLM by forward transformation hint words to produce a transformed text output, while $\mathcal{F}^{-1}(\cdot)$ represents its opposite hint word. We can use the following indicators to measure sentiment distribution preservation (Mao et al. 2024):

$$\mathrm{SDP}(x) = \|\log\sigma(x), \log\sigma(\mathcal{F}^{-1}(\mathcal{F}(F_1(x \mid p_i))))\|_1. \tag{5}$$

By mapping twice through instructions with opposite meanings, the LLM-generated text preserves its original sentiment distribution feature. Detailed transformation pair examples ($\mathcal{F}^{-1}(\cdot)$, $\mathcal{F}(\cdot)$) appear in Appendix B. The detailed algorithm for detecting LLM-generated text through sentiment distribution stability analysis appears in Appendix C.

Detection on short texts. At the same time, we observe that this zero-shot detection method requires sufficient text length for reliable analysis. For example, suppose we have two short text samples:

- LLM-generated text: *I feel very happy today because it's sunny. I plan to take a walk in the park and enjoy the beauty of nature.*
- Human-written text: *Feeling extra good today with the sun shining brightly. Planning to stroll in the park and breathe in some fresh air.*

Notably, LLM-generated short texts can closely mimic human-written content, making them harder to distinguish based on limited context. For shorter samples, LLMs can closely approximate human-like sentiment expressions, making detection more challenging. However, this limitation can be mitigated by aggregating multiple short texts to reveal stable distributional patterns, a direction further explored in our experiments on input text length.

Experiments

Experimental Setup

Datasets. We evaluate our method on five datasets, following (Verma et al. 2024; Mao et al. 2024): **News** contains 5,000 real news articles from 50 journalists and corresponding AI-generated versions, produced via a two-stage title-to-article process. **HumanEval** includes 164 programming tasks with signatures, docstrings, and test cases, covering reasoning, algorithms, and math. **Student Essay** comprises high school and university-level papers from the BAWE corpus along with LLM-generated counterparts. **Yelp Review** features 1,000 human-written Yelp reviews and machine-generated versions of similar length. **Paper Abstract** consists of 500 ACL 2023–2024 abstracts, with synthetic versions created from the first 15 words using LLMs, ensuring zero-shot validity by avoiding overlap with pretraining data.

Baselines. We compare our method with several state-of-the-art LLM-generated text detectors: **GPTZero** (Tian 2023) is a leading commercial detector for texts from models

Methods	News	HumanEval	Student Essay	Paper	Yelp Review	Avg
`GPT-4-0613`						
LogRank (GPT-2)	41.50±1.72	56.08±2.14	42.86±1.35	55.48±1.89	58.46±2.03	50.88±1.83
RoBERTa-base (Liu et al. 2019)	42.14±2.25	39.51±0.87	36.40±1.62	39.45±1.03	37.40±0.95	38.98±1.34
RoBERTa-large (Liu et al. 2019)	43.47±1.84	54.94±2.21	45.03±1.57	49.26±1.42	63.27±2.35	51.19±1.88
GPT-Zero (Tian 2023)	52.68±0.97	53.43±1.45	38.60±2.12	71.21±2.24	69.20±2.17	57.02±1.79
DetectGPT (Mitchell et al. 2023)	29.47±0.85	48.53±1.63	48.09±1.54	61.20±1.96	53.16±1.72	48.09±1.54
Ghostbuster (Verma et al. 2024)	46.85±1.33	35.84±0.92	52.77±0.93	62.38±2.12	62.68±2.08	52.10±1.48
RAIDAR (Mao et al. 2024)	54.52±1.88	55.20±1.76	53.16±1.65	75.28±2.41	68.74±2.22	61.38±1.98
Binoculars (Hans et al. 2024)	58.02±0.91	67.48±1.17	51.60±1.04	78.23±1.88	62.76±1.76	63.62±1.35
R-Detect (Song et al. 2025)	56.18±1.63	63.41±1.73	54.52±0.94	76.17±1.72	67.18±1.43	63.49±1.49
`SentiDetect-SDC` (Ours)	**59.03**$_{\uparrow 1.01}$±1.92	78.28$_{\uparrow 10.8}$±2.33	**58.83**$_{\uparrow 4.31}$±1.81	**82.80**$_{\uparrow 4.57}$±1.45	77.02$_{\uparrow 7.82}$±2.28	**71.19**$_{\uparrow 7.57}$±1.96
`SentiDetect-SDP` (Ours)	56.73±1.75	**82.74**$_{\uparrow 15.26}$±2.39	55.61$_{\uparrow 1.09}$±1.68	79.20$_{\uparrow 0.97}$±2.31	**80.16**$_{\uparrow 10.96}$±1.42	70.89$_{\uparrow 7.27}$±1.91
`Gemini-1.5-Pro`						
LogRank (GPT-2)	44.79±3.15	53.71±1.91	45.19±3.24	46.25±3.59	55.78±2.34	48.72±3.06
RoBERTa-base (Liu et al. 2019)	43.52±2.89	49.93±0.68	33.76±2.65	39.45±1.40	44.71±2.50	41.94±2.02
RoBERTa-large (Liu et al. 2019)	43.15±1.22	55.35±2.60	32.83±2.85	47.24±2.66	58.35±3.48	51.33±2.73
GPT-Zero (Tian 2023)	46.24±2.49	46.92±3.51	39.46±1.19	66.92±3.12	62.50±1.83	55.98±2.57
DetectGPT (Mitchell et al. 2023)	41.67±2.98	64.88±1.60	47.55±2.32	66.64±3.52	63.76±0.60	59.41±2.29
Ghostbuster (Verma et al. 2024)	44.31±0.80	40.53±1.51	39.97±2.22	52.72±0.32	48.69±6.13	46.77±2.36
RAIDAR (Mao et al. 2024)	45.84±4.43	71.64±5.09	51.94±5.64	68.72±4.72	69.14±4.83	61.53±5.15
Binoculars (Hans et al. 2024)	52.24±1.63	60.31±1.82	50.63±1.28	70.29±1.10	58.95±1.50	58.48±1.47
R-Detect (Song et al. 2025)	49.08±1.11	57.38±1.69	53.12±1.97	68.95±1.39	67.43±0.97	59.19±1.43
`SentiDetect-SDC` (Ours)	**54.93**$_{\uparrow 2.69}$±1.04	77.34$_{\uparrow 5.70}$±2.91	**61.85**$_{\uparrow 8.73}$±2.50	71.55$_{\uparrow 1.26}$±0.83	**79.19**$_{\uparrow 10.05}$±3.20	**71.65**$_{\uparrow 10.12}$±2.10
`SentiDetect-SDP` (Ours)	50.93$_{\uparrow 4.69}$±2.16	**80.88**$_{\uparrow 9.24}$±2.71	54.29$_{\uparrow 1.17}$±1.69	**72.35**$_{\uparrow 2.06}$±2.07	78.03$_{\uparrow 8.89}$±0.68	70.39$_{\uparrow 8.86}$±2.29

Table 1: Detection F1 scores of the content generated from GPT-4-0613 and Gemini-1.5-Pro for News, HumanEval, Student Essay, Paper, and Yelp Review datasets. The **best** and the second-best scores are in bold and underlined, respectively.

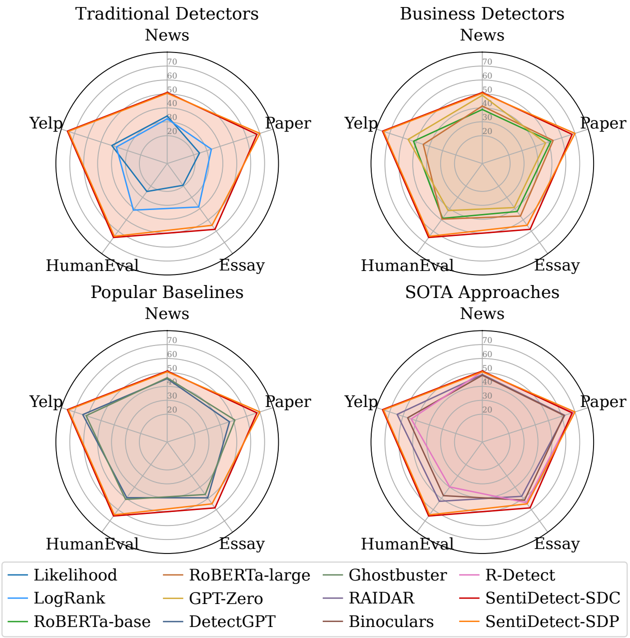


Figure 3: Visualized comparison between SentiDetect with four categories of baselines (traditional detectors, business detectors, popular baselines, and SOTA approaches) on five datasets, evaluated on content from `Claude-3`.

like *ChatGPT*, *GPT-4*, and *LLaMa*; **DetectGPT** (Mitchell et al. 2023) identifies GPT-generated text using probability curvature in a zero-shot setting without supervised training; **Ghostbuster** (Verma et al. 2024) detects texts through directly using a series of language models; **RAIDAR** (Mao et al. 2024) improves accuracy through rewriting-based analysis that captures text modification patterns; **Binoculars** (Hans et al. 2024) is a zero-shot detection method through contrasting predictions from a pair of pre-trained LLMs; **R-Detect** (Song et al. 2025) employs a non-parametric kernel relative test to compare the distributional closeness of a sample to text corpora.

Notably, although combining SDC and SDP into a unified metric has been explored, the performance gain was marginal and reduced interpretability. Thus, we report them separately for clarity. We also verify the semantic fidelity of inverse mapping pairs using BLEURT (Sellam, Das, and Parikh 2020), observing high cross-domain consistency to support reliable SDP analysis. Further detailed experiments are provided in Appendix D.

Main Detection Performance

(I) Commercial LLMs: Robust Detection Across Popular LLMs. We conduct detection experiments at the paragraph level and compare the results with existing baselines. The experimental results on GPT-4-0613 and Gemini-1.5-Pro are shown in Table 1. For processing sentiment distribution information in the input text, we employ sentiment distribution feature analysis using GPT-4-0613 as $F_1(\cdot)$ and $F_2(\cdot)$. Our experiments cover multiple datasets, including News, HumanEval, Student Essays, Paper Abstracts, and Review datasets. SentiDetect shows consistently superior performance on all datasets, proving its universality in detecting different text types. It demonstrates significant advantages in all experimental scenarios, outperforming all baselines on GPT-4-0613 by at least 11.90% and 11.43%, respectively. When tested on Gemini-1.5-Pro, the performance shows no significant decrease compared to GPT-4-0613, confirming our method's effectiveness even with more powerful LLMs. Notably, as illustrated in Appendix A, our analysis of code examples reveals that the ”sentiment” in

Method	News	HumanEval	Student Essay	Paper	Yelp Review	Avg
`LLaMa-3.3`						
LogRank (GPT-2)	58.27±2.50	69.38±3.05	54.37±1.85	68.39±0.66	72.25±1.81	67.07±1.97
RoBERTa-base (Liu et al. 2019)	58.45±0.79	61.30±2.97	40.72±0.91	73.60±3.19	49.91±2.60	60.82±2.09
RoBERTa-large (Liu et al. 2019)	54.06±0.93	72.71±3.71	38.48±1.55	63.62±2.12	64.02±1.30	63.60±1.92
GPT-Zero (Tian 2023)	55.83±2.82	68.49±2.91	51.78±1.64	87.61±2.38	77.56±3.67	72.37±2.68
DetectGPT (Mitchell et al. 2023)	57.42±3.12	76.70±0.78	59.13±2.81	70.38±3.81	72.48±2.77	69.25±2.06
Ghostbuster (Verma et al. 2024)	57.36±3.28	50.02±0.85	50.72±1.43	67.78±3.23	65.42±3.66	60.21±2.49
RAIDAR (Mao et al. 2024)	61.35±3.29	81.06±3.41	62.56±2.63	79.04±1.82	75.06±4.27	71.81±3.08
Binoculars (Hans et al. 2024)	65.80±0.98	67.82±1.31	58.32±1.72	83.64±1.93	58.28±1.29	66.77±1.45
R-Detect (Song et al. 2025)	67.73±1.25	59.38±1.10	62.40±1.51	81.77±1.07	63.27±1.29	66.91±1.24
`SentiDetect-SDC` (Ours)	**70.17**$_{\uparrow 2.44}$±0.87	**95.69**$_{\uparrow 14.63}$±2.05	**72.39**$_{\uparrow 9.83}$±2.02	**89.22**$_{\uparrow 5.58}$±0.94	85.61$_{\uparrow 10.55}$±2.68	**85.17**$_{\uparrow 13.36}$±1.71
`SentiDetect-SDP` (Ours)	68.28$_{\uparrow 0.55}$±0.82	90.01$_{\uparrow 8.95}$±1.84	68.96$_{\uparrow 6.40}$±1.70	87.98$_{\uparrow 4.34}$±3.08	**89.05**$_{\uparrow 13.99}$±3.45	83.84$_{\uparrow 12.03}$±2.18

Table 2: Detection F1 scores of the content generated from LLaMa-3.3 for News, HumanEval, Student Essay, Paper, and Yelp Review datasets. The **best** and the second-best scores are in bold and underlined, respectively.

code primarily reflects in programming style characteristics.

(II) Open-Source LLMs: Detecting Texts from LLaMa-3.3. Table 2 highlights the superior performance of SentiDetect in detecting AI-generated content from open-source LLaMa-3.3. These results demonstrate its ability to capture underlying generation patterns regardless of domain or complexity. This performance is particularly significant given the architectural and training disparities between open-source models and commercial counterparts. Compared with existing methods, SentiDetect delivers an average 13-point F1 improvement, confirming its adaptability to the traits of open-source LLMs.

(III) Performance Across Diverse Text Domains. As shown in Figure 3, SentiDetect (both SDC and SDP variants) consistently outperforms all four categories of baselines across five text domains, achieving the broadest coverage as indicated by the outermost curves in each radar chart. The most significant gains appear in HumanEval and Yelp, where SentiDetect leads by up to 24% in F1 score. This reflects its ability to capture stable sentiment patterns inherent in LLM-generated content, especially in structured code and social reviews. While margins are smaller for News, Paper, and Essay, our method still demonstrates consistent superiority. These results confirm that sentiment distribution stability is a robust and transferable signal, enabling SentiDetect to generalize effectively across diverse content types and outperform both business and state-of-the-art detectors.

Robustness Analysis and Ablation Study

(I) Robustness Against Adversarial Perturbation. We evaluate the detection robustness of various methods under adversarial attack (Ren et al. 2019) using *TextAttack*[1] library to implement it, which simulates practical evasion tactics by applying lexical-level perturbations to deceive classifiers. Table 3 reports the F1 scores post-attack, alongside original scores and relative degradation percentages. Across all five datasets, most baseline detectors, such as GPT-Zero, DetectGPT, and RoBERTa, experience sharp performance drops, often exceeding 70% loss. In contrast, SentiDetect exhibits significantly greater resilience. SentiDetect-SDC achieves the highest average detection score under attack, with a comparatively small drop, while SentiDetect-SDP also attains low degradation on average. These consistent advantages across datasets underscore that sentiment distribution features remain less sensitive to adversarial text transformations than token-level or syntactic signals.

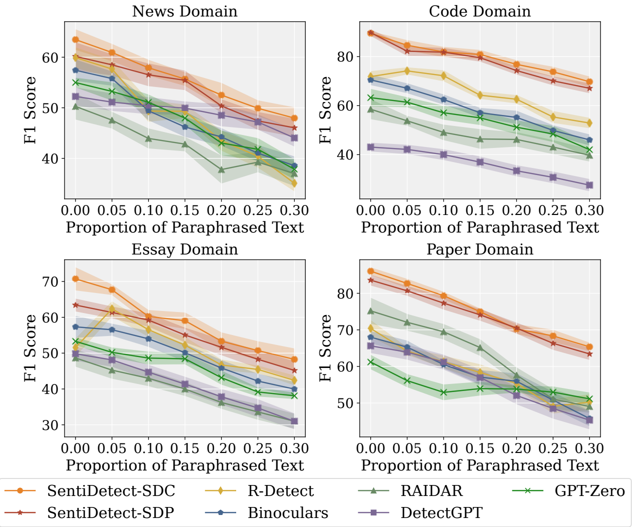


Figure 4: Comparison of paraphrased text on the News, HumanEval, Student Essay, and Paper datasets.

(II) Robustness Against Paraphrasing. To evaluate the robustness of SentiDetect against paraphrasing attacks, we demonstrate the detection performance of detectors on paraphrased text through GPT-3.5-Turbo in Figure 4. We control the proportion of changed paraphrasing texts and test the performance at different paraphrasing ratios. Figure 4 shows the experimental results of our method and various baseline methods on the News, HumanEval, Student Essay, and paper datasets. As the proportion of paraphrased text increases, the performance of all detection methods shows a downward trend, but SentiDetect consistently outperforms all other methods in F1 score and achieves considerable F1 score values on all datasets. Notably, even if 30% of the content is replaced, SentiDetect still maintains its strength.

---

[1] https://github.com/QData/TextAttack

<table>
<tr><th>Method</th><th>News</th><th>HumanEval</th><th>Student Essay</th><th>Paper</th><th>Yelp Review</th></tr>
<tr><td>LogRank</td><td>11.61 | 52.24 | ↓77.8%</td><td>7.04 | 63.25 | ↓88.9%</td><td>12.42 | 53.30 | ↓76.7%</td><td>14.10 | 61.24 | ↓77.0%</td><td>22.49 | 66.54 | ↓66.2%</td></tr>
<tr><td>RoBERTa-base (Liu et al. 2019)</td><td>10.49 | 50.47 | ↓79.2%</td><td>11.10 | 47.72 | ↓76.8%</td><td>9.43 | 43.15 | ↓78.2%</td><td>16.73 | 53.59 | ↓68.8%</td><td>7.85 | 48.87 | ↓83.9%</td></tr>
<tr><td>RoBERTa-large (Liu et al. 2019)</td><td>10.42 | 51.48 | ↓79.8%</td><td>18.02 | 67.92 | ↓73.5%</td><td>12.06 | 45.58 | ↓73.6%</td><td>16.65 | 63.20 | ↓73.7%</td><td>16.77 | 72.70 | ↓76.9%</td></tr>
<tr><td>GPT-Zero (Tian 2023)</td><td>17.42 | 54.99 | ↓68.3%</td><td>14.55 | 58.54 | ↓75.2%</td><td>19.22 | 48.66 | ↓60.5%</td><td>16.56 | 75.21 | ↓78.0%</td><td>15.57 | 78.23 | ↓80.2%</td></tr>
<tr><td>DetectGPT (Mitchell et al. 2023)</td><td>14.20 | 41.62 | ↓65.9%</td><td>18.60 | 70.50 | ↓73.7%</td><td>22.63 | 57.33 | ↓60.5%</td><td>28.47 | 67.98 | ↓58.1%</td><td>19.43 | 68.60 | ↓71.7%</td></tr>
<tr><td>Ghostbuster (Verma et al. 2024)</td><td>15.26 | 53.80 | ↓71.7%</td><td>12.46 | 43.08 | ↓71.1%</td><td>15.48 | 49.83 | ↓68.9%</td><td>14.69 | 65.63 | ↓77.6%</td><td>13.50 | 66.86 | ↓79.8%</td></tr>
<tr><td>RAIDAR (Mao et al. 2024)</td><td>18.03 | 50.28 | ↓64.1%</td><td>24.26 | 71.86 | ↓66.3%</td><td>24.62 | 51.56 | ↓52.2%</td><td>22.26 | 70.40 | ↓68.4%</td><td>24.61 | 65.21 | ↓62.3%</td></tr>
<tr><td>Binoculars (Hans et al. 2024)</td><td>20.37 | 57.42 | ↓64.5%</td><td>20.70 | 67.91 | ↓69.5%</td><td>21.54 | 57.38 | ↓62.5%</td><td>31.33 | 82.39 | ↓62.0%</td><td>25.17 | 60.31 | ↓58.3%</td></tr>
<tr><td>R-Detect (Song et al. 2025)</td><td>18.38 | 59.92 | ↓69.3%</td><td>22.51 | 70.62 | ↓68.13%</td><td>19.25 | 55.71 | ↓65.4%</td><td>27.42 | 73.60 | ↓62.7%</td><td>24.27 | 66.33 | ↓63.4%</td></tr>
<tr><td><code>SentiDetect-SDC</code> (Ours)</td><td>32.22 | 63.48 | ↓49.2%</td><td>48.05 | 89.53 | ↓46.3%</td><td>37.03 | 70.72 | ↓47.6%</td><td>46.05 | 86.54 | ↓46.8%</td><td>45.31 | 86.40 | ↓59.2%</td></tr>
<tr><td><code>SentiDetect-SDP</code> (Ours)</td><td>28.60 | 60.16 | ↓52.5%</td><td>46.79 | 89.79 | ↓47.9%</td><td>32.39 | 63.42 | ↓48.9%</td><td>40.28 | 83.59 | ↓51.8%</td><td>43.53 | 88.29 | ↓50.7%</td></tr>
</table>

Table 3: Detection performance degradation under adversarial perturbations on News, HumanEval, Student Essay, Paper, and Yelp datasets. Each cell shows: perturbed score(↑) | original score(↑) | ↓relative decrease percentage(↓). The **best** and second-best perturbed scores are marked in bold and underlined, respectively.

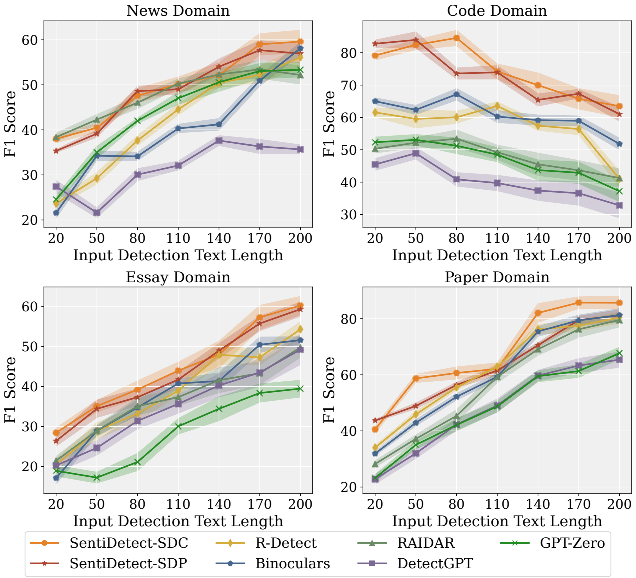


Figure 5: Comparison of varying text length on the News, HumanEval, Student Essay, and Paper datasets.

(III) Effect of Input Text Length. Due to the sensitivity of detection to the text length, we evaluate the performance of our method on GPT-4-0613 under different input lengths, as shown in Figure 5. For the Yelp and HumanEval datasets, the input texts tend to be short, and thus, it may be necessary to concatenate multiple samples together. Each data point represents the average performance for inputs of a specific length, calculated by aggregating samples with a length within ±10 words. We can see that SentiDetect achieves the highest scores across different lengths and is robust to samples from various fields. It is also worth noting that many existing detectors show poor performance in situations with short inputs. However, even if the input is as short as about 20 words, our method can still achieve superior detection scores, demonstrating its robustness to short text detection.

Short Text Detection. Notably, we observe that zero-shot detectors often require sufficient text length for reliable analysis. For shorter text samples, advanced language models can more closely approximate human expression patterns, making the distinction more challenging based on limited content. Aggregating multiple short texts can help capture broader sentiment distribution stability features, thereby improving detection performance. While longer inputs generally enhance detection performance, the results on HumanEval first increase with text length and then decrease, suggesting that excessively long inputs may not always lead to better results and could even impair detection.

	Student Essay				Yelp Review			
	I=3	I=5	I=7	I=9	I=3	I=5	I=7	I=9
SentiDetect-SDC	59.03	59.56	56.62	56.60	77.02	77.58	77.74	77.63
SentiDetect-SDP	56.73	57.01	56.94	56.92	80.16	80.31	80.42	80.44

Table 4: Effect of the number of LER prompts on the Student Essay and Yelp datasets.

(IV) Number of LER Prompts (I). We investigate how the number of LER rewriting (I) influences detection performance using GPT-4-0613 on the Essay and Yelp datasets. As shown in Table 4, both SentiDetect-SDC and SentiDetect-SDP show performance gains as I increases. However, the improvements tend to plateau with larger I, indicating diminishing returns. These findings suggest that moderate rewriting (e.g., I=5) is generally effective, while excessive rewriting may offer limited benefits.

Conclusion

In this paper, we present SentiDetect, a novel model-agnostic framework for detecting LLM-generated text by analyzing sentiment distribution stability, particularly the stability under low-emotional rewriting. Based on empirical observations of sentiment divergence, our method achieves superior detection performance across five diverse domains: news articles, programming code, student essays, academic papers, and community comments, leveraging various advanced LLMs. Experiments demonstrate that SentiDetect outperforms existing state-of-the-art baselines, significantly improving F1 detection scores by over 16% on Gemini-1.5-Pro and over 11% on GPT-4-0613. Furthermore, it also exhibits strong resilience against several adversarial challenges, including paraphrasing attacks, adversarial perturbations, and variable text lengths. This work advances the field by bridging the robustness gap in LLM-generated text detection and offers a scalable, cross-domain solution.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback. We thank Qinghua Mao for the valuable discussion. This work was partially funded by the National Natural Science Foundation of China Program (Grant No. 62572311, 62202302, 62471301, and U21B2019).

References

Abdelnabi, S.; and Fritz, M. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, 121–140. IEEE.

Bao, G.; Zhao, Y.; Teng, Z.; Yang, L.; and Zhang, Y. 2024. Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature. In *The Twelfth International Conference on Learning Representations*.

Chakraborty, S.; Bedi, A.; Zhu, S.; An, B.; Manocha, D.; and Huang, F. 2024. Position: On the possibilities of AI-generated text detection. In *Forty-first International Conference on Machine Learning*.

Chen, C.; and Wang, J.-K. 2025. Online Detection of LLM-Generated Texts via Sequential Hypothesis Testing by Betting. In *Forty-second International Conference on Machine Learning*.

Deng, X.; Bashlovkina, V.; Han, F.; Baumgartner, S.; and Bendersky, M. 2023. Llms to the moon? reddit market sentiment analysis with large language models. In *Companion Proceedings of the ACM Web Conference 2023*, 1014–1019.

Fu, Y.; Xiong, D.; and Dong, Y. 2024. Watermarking Conditional Text Generation for AI Detection: Unveiling Challenges and a Semantic-Aware Watermark Remedy. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI 2024)*.

Hans, A.; Schwarzschild, A.; Cherepanova, V.; Kazemi, H.; Saha, A.; Goldblum, M.; Geiping, J.; and Goldstein, T. 2024. Spotting LLMs With Binoculars: Zero-Shot Detection of Machine-Generated Text. In *International Conference on Machine Learning*, 17519–17537. PMLR.

Hu, L.; Liu, Z.; Zhao, Z.; Hou, L.; Nie, L.; and Li, J. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*.

Hu, X.; Chen, P.-Y.; and Ho, T.-Y. 2024. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, 36.

Kirchenbauer, J.; Geiping, J.; Wen, Y.; Katz, J.; Miers, I.; and Goldstein, T. 2023. A watermark for large language models. In *International Conference on Machine Learning*, 17061–17084. PMLR.

Kirchenbauer, J.; Geiping, J.; Wen, Y.; Shu, M.; Saifullah, K.; Kong, K.; Fernando, K.; Saha, A.; Goldblum, M.; and Goldstein, T. 2024. On the Reliability of Watermarks for Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Krishna, K.; Song, Y.; Karpinska, M.; Wieting, J.; and Iyyer, M. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.

Lee, J.; Le, T.; Chen, J.; and Lee, D. 2023. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, 3637–3647.

Lei, E.; Hsu, H.; and Chen, C.-F. 2025. PaLD: Detection of Text Partially Written by Large Language Models. In *The Thirteenth International Conference on Learning Representations*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ma, H.; Zhang, C.; Bian, Y.; Liu, L.; Zhang, Z.; Zhao, P.; Zhang, S.; Fu, H.; Hu, Q.; and Wu, B. 2024. Fairness-guided few-shot prompting for large language models. *Advances in Neural Information Processing Systems*, 36.

Mao, C.; Vondrick, C.; Wang, H.; and Yang, J. 2024. Detecting Generated Text via Rewriting. In *The Twelfth International Conference on Learning Representations*.

Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C. D.; and Finn, C. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, 24950–24962. PMLR.

Naveed, H.; Khan, A. U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; and Mian, A. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Pu, J.; Sarwar, Z.; Abdullah, S. M.; Rehman, A.; Kim, Y.; Bhattacharya, P.; Javed, M.; and Viswanath, B. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*, 1613–1630. IEEE.

Ren, S.; Deng, Y.; He, K.; and Che, W. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, 1085–1097.

Sadasivan, V. S.; Kumar, A.; Balasubramanian, S.; Wang, W.; and Feizi, S. 2023. Can AI-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Sellam, T.; Das, D.; and Parikh, A. 2020. BLEURT: Learning Robust Metrics for Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7881–7892.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2024. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. arXiv:2308.03825.

Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Krueger, G.; Kim, J. W.; Kreps, S.; et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Song, Y.; Yuan, Z.; Zhang, S.; Fang, Z.; Yu, J.; and Liu, F. 2025. Deep kernel relative test for machine-generated text detection. In *The Thirteenth International Conference on Learning Representations*.

Soto, R. R.; Koch, K.; Khan, A.; Chen, B.; Bishop, M.; and Andrews, N. 2024. Few-Shot Detection of Machine-Generated Text using Style Representations. In *The Twelfth International Conference on Learning Representations*.

Sun, L.; Huang, Y.; Wang, H.; Wu, S.; Zhang, Q.; Gao, C.; Huang, Y.; Lyu, W.; Zhang, Y.; Li, X.; et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Tian, E. 2023. Gptzero: An ai text detector,.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tulchinskii, E.; Kuznetsov, K.; Kushnareva, L.; Cherniavskii, D.; Nikolenko, S.; Burnaev, E.; Barannikov, S.; and Piontkovskaya, I. 2024. Intrinsic dimension estimation for robust detection of ai-generated texts. *Advances in Neural Information Processing Systems*, 36.

Verma, V.; Fleisig, E.; Tomlin, N.; and Klein, D. 2024. Ghostbuster: Detecting Text Ghostwritten by Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 1702–1717.

Vijay, S.; Priyanshu, A.; and KhudaBukhsh, A. R. 2025. When Neutral Summaries Are Not That Neutral: Quantifying Political Neutrality in LLM-Generated News Summaries (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 29514–29516.

Yang, J.; Jin, H.; Tang, R.; Han, X.; Feng, Q.; Jiang, H.; Zhong, S.; Yin, B.; and Hu, X. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*.

Yuan, A.; Coenen, A.; Reif, E.; and Ippolito, D. 2022. Wordcraft: story writing with large language models. In *27th International Conference on Intelligent User Interfaces*, 841–852.

Zellers, R.; Holtzman, A.; Rashkin, H.; Bisk, Y.; Farhadi, A.; Roesner, F.; and Choi, Y. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Zhang, S.; Song, Y.; Yang, J.; Li, Y.; Han, B.; and Tan, M. 2024. Detecting Machine-Generated Texts by Multi-Population Aware Optimization for Maximum Mean Discrepancy. In *The Twelfth International Conference on Learning Representations*.

Zhang, W.; Li, X.; Deng, Y.; Bing, L.; and Lam, W. 2022. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

Zhong, Q.; Ding, L.; Liu, J.; Du, B.; and Tao, D. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*.