A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions

Junchao Wu[1], Shu Yang[1], Runzhe Zhan[1], Yulin Yuan[2*], Lidia Sam Chao[3], and Derek Fai Wong[1*]

[1]University of Macau, NLP[2]CT Lab, Faculty of Science and Technology, Institute of Collaborative Innovation
nlp2ct.junchao@gmail.com, nlp2ct.shuyang@gmail.com, nlp2ct.runzhe@gmail.com, derekfw@um.edu.mo

[2]University of Macau, Department of Chinese Language and Literature, Faculty of Arts and Humanities
Peking University, Department of Chinese Language and Literature, Faculty of Humanities
yulinyuan@um.edu.mo, yuanyl@pku.edu.cn

[3]University of Macau, NLP[2]CT Lab, Faculty of Science and Technology, State Key Laboratory of Internet of Things for Smart City
lidiasc@um.edu.mo

The remarkable ability of large language models (LLMs) to comprehend, interpret, and generate complex language has rapidly integrated LLM-generated text into various aspects of daily life, where users increasingly accept it. However, the growing reliance on LLMs underscores the urgent need for effective detection mechanisms to identify LLM-generated text. Such mechanisms are critical to mitigating misuse and safeguarding domains like artistic expression and social networks from potential negative consequences. LLM-generated text detection, conceptualized as a binary classification task, seeks to determine whether an LLM produced a given text. Recent advances in this field stem from innovations in watermarking techniques, statistics-based detectors, and neural-based detectors. Human-assisted methods also play a crucial role. In this survey, we consolidate recent research breakthroughs in this field, emphasizing the urgent need to strengthen detector research. Additionally, we review existing datasets, highlighting their limitations and developmental requirements. Furthermore, we examine various LLM-generated text detection paradigms, shedding light on challenges like out-of-distribution problems, potential attacks, real-world data issues, and ineffective evaluation frameworks. Finally, we outline intriguing directions for future research in LLM-generated text detection to advance responsible artificial

* Yulin Yuan and Derek Fai Wong are co-coresponding authors.

Action Editor: Miguel Ballesteros. Submission received: 14 March 2024; revised version received: 3 September 2024; accepted for publication: 28 November 2024.

https://doi.org/10.1162/coli_a_00549

intelligence. This survey aims to provide a clear and comprehensive introduction for newcomers while offering seasoned researchers valuable updates in the field.[1]

1. Introduction

The rapid advancement of large language models (LLMs) such as GPT-4 (OpenAI 2023), Claude (Anthropic 2023), and PaLM (Chowdhery et al. 2022) has elevated text generation capabilities to near-human level. These systems generate coherent, sophisticated content, driving a notable surge in artificial intelligence (AI)-produced material. Recent research reveals a 55.4% increase in AI-generated news articles on mainstream websites and a staggering 457% rise in misinformation sites between 1 January, 2022, and 1 May, 2023 (Hanley and Durumeric 2023). Beyond content creation, LLMs are transforming numerous sectors, including education (Susnjak 2022), law (Cui et al. 2023), biology (Piccolo et al. 2023), and medicine (Thirunavukarasu et al. 2023). These applications range from personalized learning to medical diagnostics, underscoring their profound impact across creative and professional domains. However, this rapid integration also raises concerns about accountability, misuse, and fairness, highlighting the urgent need for comprehensive regulatory frameworks to ensure their ethical and transparent deployment.

As LLMs become more sophisticated, distinguishing between human-written and LLM-generated text has become a significant challenge, raising critical societal and ethical concerns. The indistinguishability of AI-generated content enables its misuse in creating deceptive material, such as disinformation, online scams, and social media spam (Pagnoni, Graciarena, and Tsvetkov 2022; Weidinger et al. 2021; Mirsky et al. 2022). Additionally, while LLMs are capable of producing high-quality text, they are not immune to generating unreliable or fabricated information, which risks propagating inaccuracies and eroding trust in digital communication (Ji et al. 2023; Christian 2023). The growing reliance on LLMs for data generation in AI research adds to these challenges. This self-referential practice risks recursive degradation, where LLM-generated content becomes part of new training datasets, potentially reducing the quality and diversity of future models and hindering advancements in both generative AI and detection technologies (Cardenuto et al. 2023; Yu et al. 2023a).

The detection of LLM-generated text, has become an emerging challenge. Current detection technologies, including commercial tools, often need help distingushing between human-written and LLM-generated content (Price and Sakellarios 2023; Walters 2023; Weber-Wulff et al. 2023). These systems frequently misclassify outputs, with a tendency to favor human-written classifications. Human-based detection methods are no better, achieving accuracy rates only slightly above random chance (Uchendu et al. 2021; Dou et al. 2022; Clark et al. 2021; Soni and Wade 2023). In fact, humans often need to perform better relative to automated algorithms in diverse evaluation settings (Ippolito et al. 2020; Soni and Wade 2023). This underscores the urgent need for robust, reliable detection mechanisms to prevent the misuse of LLMs in spreading deceptive content and misinformation. Effective detection systems are key to mitigating these risks and fostering responsible AI governance in the rapidly evolving LLM landscape (Stokel-Walker and Van Noorden 2023; Porsdam Mann et al. 2023; Shevlane et al. 2023).

1 The useful resources are publicly available at: https://github.com/NLP2CT/LLM-generated-Text-Detection.

Efforts to detect AI-generated text predate the widespread adoption of tools like ChatGPT. Early studies primarily focused on identifying machine-generated content, such as detecting deepfake text (Pu et al. 2023a), machine-generated text (Jawahar, Abdul-Mageed, and Lakshmanan 2020), and authorship attribution (Uchendu, Le, and Lee 2023a), often relying on statistical methods or simple classification approaches. However, the introduction of ChatGPT marked a significant shift in both the capabilities of LLMs and the challenges they pose, reigniting interest in LLM-generated text detection. Recent methods have evolved to address more complex scenarios, such as distinguishing LLM-generated text from human-authored content, evaluating the robustness of detection systems, and developing adversarial attacks. While previous surveys provide valuable insights into machine-generated text detection (Crothers, Japkowicz, and Viktor 2023; Tang, Chuang, and Hu 2023), they often need more depth in exploring the diverse and rapidly advancing methodologies in the field. This work seeks to bridge that gap by comprehensively reviewing detection techniques and identifying key challenges for future research (see Section 3.1).

This article presents a detailed review of recent advancements in LLM-generated text detection. Our objective is to highlight the challenges in this domain while exploring potential directions for future research. We begin by introducing the task of detecting LLM-generated text, explaining the mechanisms behind LLM text generation, and outlining key technological advancements. We also discuss the relevance and importance of detecting LLM-generated text in various real-world contexts. This review examines widely used datasets and benchmarks, highlighting their limitations and the need for improved data resources. Additionally, we analyze various detection approaches, including neural-based methods, statistical methods, watermarking techniques, and human-assisted methods. We explore in-depth critical challenges such as out-of-distribution issues, adversarial attacks, real-world data complexities, and the lack of robust evaluation frameworks. Finally, we propose several directions for future research to advance the development of adequate LLM-generated text detectors.

2. Background

2.1 LLM-Generated Text Detection Task

Detecting LLM-generated text presents a significant challenge. Humans generally struggle to distinguish between LLM-generated text and human-written text (Uchendu et al. 2021; Dou et al. 2022; Clark et al. 2021; Soni and Wade 2023), and their capability to distinguish such texts exceeds random classification only slightly. Table 1 offers some examples where LLM-generated text often is extremely close to human-written text and can be difficult to distinguish. When LLMs generate fabricated details, discerning their origins and veracity remains equally challenging.

Recent studies (Guo et al. 2023; Ma, Liu, and Yi 2023; Muñoz-Ortiz, Gómez-Rodríguez, and Vilares 2023; Giorgi et al. 2023; Seals and Shalin 2023) have highlighted significant disparities between human-written and LLM-generated text, such as ChatGPT. The differences between LLM-generated text and human-written text are not merely within the scope of individual word choice (Seals and Shalin 2023), but also manifest in stylistic dimensions, such as syntactical simplicity, use of passive voice, and narrativity. Notably, LLM-generated text often exhibits qualities of enhanced organization, logical structure, formality, and objectivity in comparison with human-written text. Additionally, LLMs frequently produce extensive and comprehensive responses,

Table 1
Examples of human-written text and LLM-generated text. Text generated by LLMs during normal operation and instances in which they fabricate facts often exhibit no intuitively discernible differences. When LLMs either abstain from providing an answer or craft neutral responses, certain indicators, such as the explicit statement "I am an AI language model," may facilitate human adjudication, but such examples are less.

Type	Question	Human-written	LLMs-generated
Normal	Explain what is NLP?	Natural language processing (NLP) is an interdisciplinary subfield of linguistics, computer science, and artificial intelligence …	Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics that focuses on …
Refusal	How is today special?	Today's Special is a Canadian children's television show produced by Clive VanderBurgh at TVOntario from 1981 to 1987.	I'm sorry, but I am an AI language model and do not have access to current dates or events. Is there anything else I can help you with …
Fabricated	Explain what is NLP based on one publication in the recent literature.	In "Natural language processing: state of the art, current trends and challenges", NLP is summarized as a discipline that uses various algorithms, tools and methods to …	NLP is a multidisciplinary field at the intersection of computer science, linguistics, and ai, as described in a recent peer-reviewed publication titled "Natural Language Processing: A Comprehensive Overview and Recent Advances" (2023) …

characterized by a lower prevalence of bias and harmful content. Nevertheless, they occasionally introduce nonsensical or fabricated details. Linguistically, LLM-generated text tends to be about twice the length of human-written text but exhibits a more limited vocabulary. LLMs exhibit a higher frequency of noun, verb, determiner, adjective, auxiliary, coordinating conjunction, and particle word categories compared with humans, and less adverb and punctuation, incorporating more deterministic, conjunctive, and auxiliary structures in their syntax. Additionally, LLM-generated text often conveys less emotional intensity and exhibits clearer presentation than human writing. This phenomenon is possibly influenced by inherent positive bias in LLMs (Giorgi, Ungar, and Schwartz 2021; Markowitz, Hancock, and Bailenson 2023; Mitrović, Andreoletti, and Ayoub 2023). While statistical variations exist across datasets, the consistent divergence in language features and human visual perception strongly indicates detectable differences between LLM-generated and human-written text. Chakraborty et al. (2023b) have further substantiated the view by reporting on the detectability of text generated by LLMs, including the high-performance models such as GPT-3.5-Turbo and GPT-4 (Helm, Priebe, and Yang 2023). Additionally, Chakraborty et al. (2023a) introduced an AI Detectability Index to further rank models according to their detectability.

This survey begins by defining key concepts relevant to the field, including human-written text, LLM-generated text, and the LLM-generated text detection task.

Human-written Text. Is characterized as the text crafted by individuals to express thoughts, emotions, and viewpoints. This encompasses articles, poems, and reviews, among others, typically reflecting personal knowledge, cultural milieu, and emotional disposition, spanning the entirety of the human experience.

LLM-Generated Text. Is defined as cohesive, grammatically coherent, and pertinent content generated by LLMs. These models are trained extensively on NLP techniques,

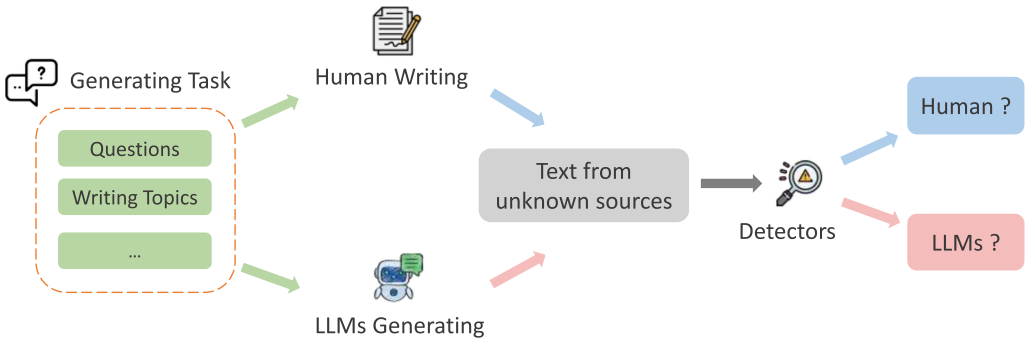

Figure 1
Overview of LLM-generated text detection task. This task is a binary classification task that detects whether the provided text is generated by LLMs or written by humans.

utilizing large datasets and machine learning methodologies. The quality and fidelity of the generated text typically depend on the scale of the model and the diversity of training data.

It is noteworthy that a standardized definition for computer-assisted writing is still absent. Gao et al. (2024) categorize this as a distinct type termed "AI-revised Human-Written Text," which is further discussed in Section 8.3.

LLM-Generated Text Detection Task. Is conceptualized as a binary classification task, aiming to ascertain if a given text is generated by an LLM, as illustrated in Figure 1. The formal representation of this task is given by Equation (1).

$$D(x) = \begin{cases} 1 & \text{if } x \text{ generated by LLMs} \\ 0 & \text{if } x \text{ written by human} \end{cases} \tag{1}$$

where $D(x)$ represents the detector, and x is the text to be detected.

2.2 LLMs Text Generation and Confusion Sources

2.2.1 Generation Mechanisms of LLMs. The mechanism of text generation of LLMs operates by sequentially predicting subsequent tokens, constructing text one word at a time. During generation, LLMs predict the next token based on both the input sequence and previously generated tokens. Assume that the input sequence has a length of N, denoted as $X_N = \{x_1, x_2, \ldots, x_N\}$, and the total number of time steps is T, the current time step is t, and the sequence up to time step $t-1$ is $Y_{t-1} = \{y_1, y_2, \ldots y_{t-1}\}$. At this point, the output next word y_t can be expressed as Equation (2).

$$y_t \sim P(y_t|Y_{t-1}, X_T) = softmax(w_o \cdot h_t) \tag{2}$$

Here, h_t is the hidden state of the model at time step t, w_o is the output matrix, and the softmax function is used to obtain the probability distribution of the vocabulary. The token y_t is sampled from the probability distribution $P(y_t|Y_{t-1}, X_T)$. The final output

sequence can be described as Equation (3) and the joint probability function for the final output sequence can be modeled and represented as Equation (4).

$$Y_T = \{y_1, y_2, \ldots, y_T\} \tag{3}$$

$$P(Y_T \mid X_N) = \prod_{t=1}^{T} P(y_t \mid y_1, y_2, \ldots, y_{t-1}, X_N) \tag{4}$$

The quality of generated text is fundamentally linked to the decoding strategy used during text generation. As models generate text sequentially, the method of selecting the next token from the probability distribution over the vocabulary plays a pivotal role in shaping the output. This involves sampling y_t from the probability distribution over the vocabulary. The predominant decoding techniques include greedy search, beam search (Lowerre and Reddy 1976), top-K sampling (Fan, Lewis, and Dauphin 2018), and top-P sampling (Holtzman et al. 2020).

Greedy search selects the token with the highest probability at each step (Sutskever, Vinyals, and Le 2014), offering simplicity and speed, but it often leads to local optima and lacks diversity, making it ineffective in managing uncertainty. Beam search attempts to mitigate these limitations by considering multiple candidates simultaneously, which improves text quality but can result in repetitive fragments and struggles with open-ended tasks due to its difficulty in handling uncertainty (He et al. 2023a). In contrast, top-K sampling enhances diversity by restricting choices to the K most probable tokens, thus introducing variability and managing uncertainty, although it can sometimes result in incoherence (Holtzman et al. 2020; Basu et al. 2021). Top-P sampling, or nucleus sampling, further refines this approach by selecting from the smallest set of tokens whose cumulative probability surpasses a threshold P, effectively balancing coherence and diversity. Its effectiveness, however, heavily depends on the model's prediction quality, with diversity being influenced by the parameter P (Holtzman et al. 2020).

These strategies provide various trade-offs between speed, diversity, coherence, and flexibility in text generation, helping elucidate the specific characteristics of the text produced by LLMs.

2.2.2 Sources of LLMs' Strong Generation Capabilities. Notably, beyond a certain scale, models exhibit abilities that defy prediction by conventional scaling laws. These phenomena, absent in smaller models but emergent in larger ones, are collectively referred to as the "Emergent Abilities" of LLMs.

In-Context Learning (ICL). The origins of ICL capabilities remain a topic of ongoing debate (Dai et al. 2023). However, this capability introduces a paradigm where model parameters remain unchanged, and only the design of the prompt is modified to elicit desired outputs from LLMs. This concept was first introduced in GPT-3 (Brown et al. 2020). Brown et al. (2020) argued that the presence of ICL is fundamental for the swift adaptability of LLMs across a diverse set of tasks. With only a few examples, LLMs can adeptly handle downstream tasks, eliminating the need for the earlier BERT-based approach that depended on pretraining followed by task-specific fine-tuning (Raffel et al. 2020).

Alignment of Human Preference. Although LLMs can be guided to generate content using carefully designed prompts, the resulting text might lack control, potentially leading to

the creation of misleading or biased content (Zhang et al. 2023b). The primary limitation of these models lies in predicting subsequent words to form coherent sentences based on vast corpora, rather than ensuring that the content generated is both beneficial and innocuous to humans. To address these concerns, OpenAI introduced the Reinforcement Learning from Human Feedback (RLHF) approach, as detailed in Ouyang et al. (2022) and Lambert et al. (2022). This approach begins by fine-tuning LLMs using data from user-directed quizzes and subsequently evaluating the model's outputs with human assessors. Simultaneously, a reward function is established, and the LLM is further refined using the Proximal Policy Optimization (PPO) algorithm (Schulman et al. 2017). The end result is a model that aligns with human values, understands human instructions, and genuinely assists users.

Complex Reasoning Capabilities. Although the ICL and alignment capabilities of LLMs enable meaningful interactions and assistance, their effectiveness diminishes when tasked with logical reasoning and increased complexity. Wei et al. (2022) observed that encouraging LLMs to produce more intermediate steps through a Chain of Thought can enhance their effectiveness. Tree of Thoughts (Yao et al. 2023) and Graph of Thoughts (Besta et al. 2023) are extensions of this methodology. Both strategies augment LLM performance on intricate tasks by amplifying the computational effort required for the model to deduce an answer.

2.3 Why Do We Need to Detect Text Generated by LLMs?

As LLMs undergo iterative refinements and reinforcement learning through human feedback, their outputs become increasingly harmonized with human values and preferences. This alignment facilitates the broader acceptance and integration of LLM-generated text into everyday life. The emergence of various AI tools has played a significant role in fostering intuitive human–AI interactions and democratizing access to the advanced capabilities of previously esoteric models. From interactive web-based assistants like ChatGPT,[2] to search engines enhanced with LLM technology like the contemporary version of Bing,[3] to specialized tools like Copilot,[4] and Scispace[5] that assist professionals in code generation and scientific research, LLMs have subtly woven into the digital fabric of our lives, propagating their content across diverse platforms.

It is important to acknowledge that for the majority of users, LLMs and their applications are still considered black-box AI systems. For individual users, this often serves as a benign efficiency boost, sidestepping laborious retrieval, and summarization. However, within specific contexts and against the backdrop of the broader digital landscape, it becomes crucial to discern, filter, or even exclude LLM-generated text. It is crucial to note that not all scenarios necessitate the detection of LLM-generated content. Unnecessary detection can lead to inefficiencies and increased development costs. Generally, detecting LLM-generated text might be superfluous when:

- The utilization of LLMs poses minimal risk, especially when they handle routine, replicable tasks.

2 https://chat.openai.com/.
3 https://www.bing.com/.
4 https://github.com/features/copilot/.
5 https://typeset.io/.

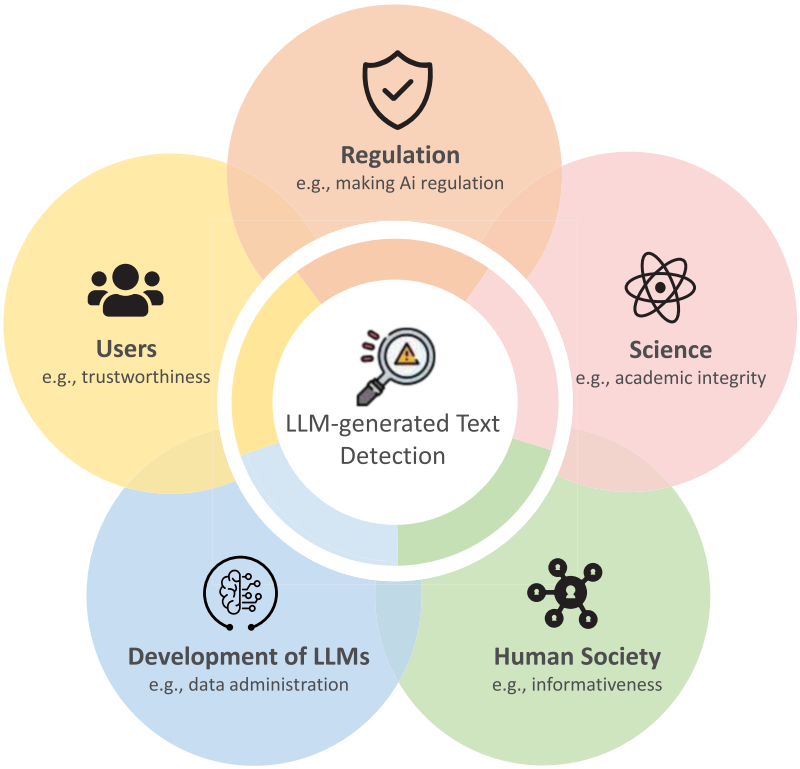

Figure 2
The most critical reasons why LLM-generated text detection is needed urgently. We discuss it from five perspectives: Regulation, Users, Developments, Science, and Human Society.

- The dissemination of LLM-generated text is confined to predictable, limited domains, like closed information circles with few participants.

Drawing upon the literature reviewed in this study, the rationale behind detecting LLM-generated text can be elucidated from multiple perspectives, as illustrated in Figure 2. The delineated perspectives are, in part, informed by the insights presented in Gade et al. (2020) and Saeed and Omlin (2023). Gade et al. (2020) suggested the need for explainable AI for social, legal, scientific, enterprise, end-user applications, particularly in high-stakes domains. Saeed and Omlin (2023) emphasized aspects like fairness, accountability, and transparency of trustworthy AI. While they focus on the general AI applications, we have adapted and extended these concepts to detecting AI-generated text.

While these perspectives provided in previous works may not be exhaustive and some facets may intersect or further delineate as LLMs and AI systems mature, we posit that these points underscore the paramount reasons for the necessity of detecting text generated by LLMs.

Regulation. As AI tools, often characterized as black boxes, the inclusion of LLM-generated text in creative endeavors raises significant legal issues. A pressing concern is the eligibility of LLM-generated texts for intellectual property rights protection, a

subject still mired in debate (Epstein et al. 2023; Wikipedia 2023), although the *EU AI Act*[6] has begun to continuously improve to regulate the use of AI. The main challenges arise from issues such as ownership of the training data used by the AI in generating output and how to determine how much human involvement is enough to make the work theirs. The prerequisite for copyright protection for AI supervision and AI-generated content is that human creativity in the materials used to train AI systems can be distinguished, so as to further promote the implementation of more complete legal supervision.

Users. LLM-generated text, refined through various alignment methods, is progressively aligning with human preferences. This content permeates numerous user-accessible platforms, including blogs and Questions & Answers (Q&A) forums. However, excessive reliance on such content can undermine user trust in AI systems and, by extension, digital content as a whole. In this context, the role of LLM-generated text detection becomes crucial as a gatekeeper to regulate the prevalence of LLM-generated text online.

Developments. With the evolving prowess of LLMs, Li et al. (2023b) suggested that LLMs can self-assess and even benchmark their own performances. Due to its excellent text generation performance, LLMs are also used to construct many training data sets through preset instructions (Taori et al. 2023). However, if LLMs heavily rely on web-sourced data for training, and a significant portion of this data originates from LLM outputs, it could hinder their long-term progress (Alemohammad et al. 2023; Tang, Chuang, and Hu 2024; Shumailov et al. 2024).

Science. The relentless march of human progress owes much to the spirit of scientific exploration and discovery. However, the increasing presence of LLM-generated text in academic writing (Májovskỳ et al. 2023) and the use of LLM-originated designs in research endeavors raise concerns about potentially diluting human ingenuity and exploratory drive. At the same time, it could also undermine the ability of higher education to validate student knowledge and comprehension, and diminish the academic reputation of specific higher education institutions (Ibrahim et al. 2023). Although current methodologies may have limitations, further enhancements in detection capabilities will strengthen academic integrity and preserve human independent thinking in scientific research.

Human Society. From a societal perspective, analyzing the implications of LLM-generated text reveals that these models essentially mimic specific textual patterns while predicting subsequent tokens. If used improperly, these models have the potential to diminish linguistic diversity and contribute to the formation of information silos within societal discourse. In the long run, detecting and filtering LLM-generated text is crucial for preserving the richness and diversity of human communication, both linguistically and informatively.

3. Related Work and Our Investigation

3.1 Related Work

The comprehensive review article by Beresneva (2016) represents the first extensive survey of methods for detecting computer-generated text. At that time, the detection

6 https://artificialintelligenceact.eu/the-act/.

process was relatively simple, mainly aimed at identifying machine-translated text through simple statistical methods. The advent of autoregressive models significantly heightened the complexity of text detection tasks. Jawahar, Abdul-Mageed, and Lakshmanan (2020) provide a detailed survey on the detection of machine-generated text. This work laid a solid foundation for the field, with a particular focus on detection methods tailored to the state-of-the-art (SOTA) generative models like GPT-2. The detection methods prevalent at the time were classified into four main categories: classifiers trained from scratch, zero-shot classifiers, fine-tuning of neural LMs, and human–machine collaboration. These methods have since been regarded as relatively traditional in current research contexts.

The subsequent release of ChatGPT sparked a surge of interest in LLMs, and signified a major shift in research directions. In response to the rapid challenges posed by LLM-generated text, the NLP community has intensified efforts to develop robust detection mechanisms and investigate the dynamics of evasive techniques used by such models. Recent surveys by Crothers, Japkowicz, and Viktor (2023) and Dhaini, Poelman, and Erdogan (2023) have offered fresh insights into the detection of LLM-generated text. Notably, Crothers, Japkowicz, and Viktor (2023) performed a comprehensive analysis of threat models proposed by contemporary NLG systems, which covered speech-to-text and end-to-end models. However, this review's coverage of detection techniques was quite similar to that of Jawahar, Abdul-Mageed, and Lakshmanan (2020) and did not encompass many cutting-edge works, including current popular zero-shot techniques, for example, DetectGPT (Mitchell et al. 2023) and Fast-DetectGPT (Bao et al. 2023), as well as advanced neural-based methods like CoCo (Liu et al. 2022) and OUTFOX (Koike, Kaneko, and Okazaki 2023b). Similarly, Dhaini, Poelman, and Erdogan (2023) also lagged in capturing many innovative works, with the primary focus still on Encoder-based classification methods. Another survey by Tang, Chuang, and Hu (2023) categorized detection methods into black-box and white-box approaches and highlighted emerging technologies such as watermarking. However, it, too, was slightly outdated in terms of dataset and detector discussed, including only five datasets and not covering currently popular zero-shot techniques and some advanced neural-based methods, which could benefit from a more comprehensive analysis and critical evaluation. Ghosal et al. (2023) focused primarily on current attacks and defenses in LLM-generated text detection, discussing the state of LLM-generated text detection tasks, including watermarks and some SOTA zero-shot detectors. However, by concentrating on more specific aspects, this review missed some broader review angles, such as a detailed examination of the motivations for detecting LLM-generated text, data resources, and and the history of various evaluation methods that could enrich the discourse. Liu et al. (2023c) provided a comprehensive review of watermarking techniques, primarily focusing on watermarking itself rather than the detection of texts generated by LLMs. While not all watermarking methods covered in the review are applicable to LLM-generated text detection, the classification criteria for watermarking methods proposed in the paper can provide valuable insights and serve as an important reference for organizing and investigating watermarking techniques.

In this article, we strive to provide a more comprehensive and insightful review of the latest research on LLM-generated text detection, enriched with comprehensive discussion. We highlight the strengths of our review in comparison to others:

- Systematic and Comprehensive Review: Our survey offers an extensive exploration of LLM-generated text detection, covering the task's

description and underlying motivation, various benchmarks and datasets, the latest detection and attack methods, evaluation frameworks, the most pressing challenges faced today, potential future directions, and a critical examination of each aspect.

- In-depth Analysis of Detection Mechanisms: We provide a detailed overview of detection strategies, from traditional approaches to the latest research, and systematically evaluate their effectiveness, strengths, and weaknesses in the current environment of LLMs.
- More Pragmatic Insights. Our discussion delves into research questions with practical implications, such as how model size affects detection capabilities, the challenges of identifying text that is not purely generated by LLMs, and the lack of effective evaluation frameworks.

In summary, we firmly believe that this review is more systematic and comprehensive than existing works. More importantly, our critical discussion not only provides guidance to new researchers but also imparts valuable insights into established works within the field.

3.2 Systematic Investigation and Implementation

Our survey utilized the System for Literature Review (SLR) as delineated by Kitchenham and Charters (2007), a methodological framework designed for evaluating the extent and quality of extant evidence pertaining to a specified research question or topic. Offering a more expansive and accurate insight compared with conventional literature reviews, this approach has been prominently utilized in numerous scholarly surveys, as evidenced by Murtaza et al. (2020) and Saeed and Omlin (2023). The research questions guiding our SLR were as follows:

What are the prevailing methods for detecting LLM-generated text, and *what* are the main challenges associated with these methods?

Upon delineating the research problems, our study utilized search terms directly related to the research issue, specifically: "LLM-generated text detection." "machine-generated text detection," "AI-written text detection," "authorship attribution," and "deepfake text detection." These terms were strategically combined using the Boolean operator OR to formulate the following search string: ("LLM-generated text detection" OR "machine-generated text detection" OR "AI-written text detection" OR "authorship attribution" OR "deepfake text detection"). Subsequently, using this search string, we engaged in a preliminary search through pertinent and authoritative electronic dissertation databases and search engines. Our investigation mainly focused on scholarly articles that were publicly accessible prior to November 2023. Table 2 outlines the sources used and provides an overview of our results.

Subsequently, we established the ensuing criteria to scrutinize the amassed articles:

- The article should be a review focusing on the methods and challenges pertinent to LLM-generated (machine-generated/AI-written) text detection.

Table 2
Overview of the diverse databases and search engines utilized in our research, along with the incorporated search schemes and the consequent results obtained. Google Scholar predominates as the search engine yielding the maximum number of retrievable documents. Upon meticulous examination, it is observed that a substantial portion of the documents originate from ArXiv, primarily shared by researchers.

Databases	Search Engine	Search Scheme	Retrieved
Google Scholar	https://scholar.google.com/	Full Text	210
ArXiv	https://arxiv.org/	Full Text	N/A[a]
Scopus	https://www.scopus.com/	TITLE-ABS-KEY: (Title, Abstract, Author Keywords, Indexed Keywords)	133
Web of Science	https://www.webofscience.com/	Topic: (Searches Title, Abstract, Author Keywords, Keywords Plus)	92
IEEE Xplore	https://ieeexplore.ieee.org/	Full Text	49
Springer Link	https://link.springer.com/	Full Text	N/A[a]
ACL Anthology	https://aclanthology.org/	Full Text	N/A[a]
ACM Digital Library	https://dl.acm.org/	Title	N/A[b]

[a] Search engines cannot use all keywords in a single search string. Therefore the retrieved results are inaccurate and there may be duplicate results of thesis queries.
[b] The search engine retrieved an inaccurate number of papers that were weakly related to our topic.

- The article should propose a methodology specifically designed for the detection of LLM-generated (machine-generated/AI-written) text.
- The article should delineate challenges and prospective directions for future research in the domain of text generation for LLMs.
- The article should articulate the necessity and applications of LLM-generated text detection.

If any one of the aforementioned four criteria was met, the respective work was considered valuable for our study. Following a process of de-duplication and manual screening, we identified 83 pertinent pieces of literature. The distribution by year is illustrated in Figure 3. Notably, the majority of relevant research on LLM-generated text detection was published in the year 2023 (as shown in Figure 3), underscoring the vibrant development within this field and highlighting the significance of our study.

In the subsequent sections, we provide a comprehensive analysis, starting with Section 4, where we delve into the datasets and benchmarks relevant to detection tasks. This section highlights datasets that can be extended to detection applications and discusses the inherent challenges they present. Building on this foundation, Section 5 examines various detection methods. We explore technologies ranging from watermarking and statistics-based approaches to neural-based detectors and human-assisted methods, providing insights into their effectiveness and limitations. Section 7 focuses on evaluation metrics include accuracy, precision, recall, false positive rate, true negative rate, false negative rate, F_1 score, and the area under the receiver operating

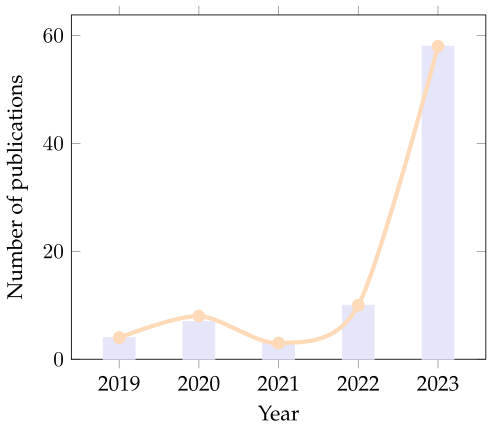

Figure 3
The distribution by year of the last 5 years of literature obtained from the screening is plotted. The number of published articles obtain significant attention in 2023.

characteristic curve (AUROC), linking these measures to the methods discussed in earlier sections. In Section 8, we discuss the challenges faced by detection methods. This includes out-of-distribution issues, potential attacks, real-world data problems, the impact of model size on detector performance, and the current lack of a robust evaluation framework, all of which underscore the need for further research. Finally, Section 9 outlines potential avenues for future research. We propose developing more robust detectors against attacks, enhancing zero-shot capabilities, optimizing performance in low-resource settings, detecting not purely LLM-generated text, constructing detectors amidst data ambiguity, creating effective real-world evaluation frameworks, and improving misinformation discrimination capabilities.

Through this structured analysis, we aim to provide a clear and cohesive understanding of the current landscape and future directions in detection technology.

4. Data

High-quality datasets are essential for advancing research in the LLM-generated text detection task. These datasets enable researchers to swiftly develop and calibrate efficient detectors and establish standardized metrics for evaluating the efficacy of their methodologies. However, procuring such high-quality labeled data often demands substantial financial, material, and human resources. Presently, the development of datasets focused on detecting LLM-generated text is in its nascent stages, hindered by issues such as limited data volume and sample complexity, both crucial for crafting robust detectors. This section introduces the most widely used datasets for training LLM-generated text detectors. Additionally, we highlight datasets from unrelated domains or tasks that, though not initially designed for detection tasks, can be repurposed for various detection scenarios, which is a prevailing strategy in many contemporary detection studies. We subsequently introduce benchmarks for verifying the effectiveness of LLM-generated text detectors, which are carefully designed to evaluate the performance of the detector from different perspectives. Lastly, we evaluate these training datasets and benchmarks, identifying current shortcomings and challenges in dataset construction for LLM-generated text detection, aiming to inform the design of future data resources.

Table 3
Summary of detection datasets for LLM-generated text detection.

Corpus	Use	Human	LLMs	LLMs Type	Language	Attack	Domain
HC3 (Guo et al. 2023)	train	~80k	~43k	ChatGPT	English, Chinese	–	Web Text, QA, Social Media
CHEAT (Yu et al. 2023a)	train	~15k	~35k	ChatGPT	English	Paraphrase	Scientific Writing
HC3 Plus (Su et al. 2023b)	train valid test	~95k ~10k ~38k		GPT-3.5-Turbo	English, Chinese	Paraphrase	News Writing, Social Media
OpenLLMText (Chen et al. 2023a)	train, valid, test	~52k ~8k ~8k	~209k ~33k ~33k	ChatGPT, PaLM, LLaMA, GPT2-XL	English	–	Web Text
GROVER Dataset (Zellers et al. 2019b)	train	~24k		Grover-Mega	English	–	News Writing
TweepFake (Fagni et al. 2021)	train	~12k	~12k	GPT-2, RNN, Markov, LSTM, CharRNN	English	–	Social Media
GPT-2 Output Dataset[7]	train test	~250k ~5k	~2000k ~40k	GPT-2 (small, medium, large, xl)	English	–	Web Text
ArguGPT (Liu et al. 2023d)	train valid test	~6k 700 700		GPT2-Xl, Text-Babbage-001, Text-Curie-001, Text-Davinci-001, Text-Davinci-002, Text-Davinci-003, GPT-3.5-Turbo	English	–	Scientific writing
DeepfakeTextDetect (Li et al. 2023c)	train valid test	~236k ~56k ~56k		GPT (Text-Davinci-002, Text-Davinci-003, GPT-Turbo-3.5), LLaMA (6B, 13B, 30B, 65B), GLM-130B, FLAN-T5 (small, base, large, xl, xxl), OPT(125M, 350M, 1.3B, 2.7B, 6.7B, 13B, 30B, iml1.3B, iml-30B), T0 (3B, 11B), BLOOM-7B1, GPT-J-6B, GPT-NeoX-20B)	English	Paraphrase	Social Media, News Writing, QA, Story Generation, Comprehension and Reasoning, Scientific writing

4.1 Training

4.1.1 Detection Datasets. Massive and high-quality datasets can assist researchers in rapidly training their detectors. Table 3 provides a comprehensive organization and comparison. Given that different studies focus on various practical issues, our aim is to facilitate researchers in conveniently selecting high-quality datasets that meet their specific needs through our comprehensive review work.

HC3. The Human ChatGPT Comparison Corpus (HC3) (Guo et al. 2023) stands as one of the initial open-source efforts to compare ChatGPT-generated text with human-written text. It involves collecting both human and ChatGPT responses to identical questions. Due to its pioneering contributions in this field, the HC3 corpus has been utilized in numerous subsequent studies as a valuable resource. The corpus offers datasets in both English and Chinese. Specifically, HC3-en comprises 58k human responses and 26k ChatGPT responses, derived from 24k questions, sourced from the ELI5, WikiQA, Crawled Wikipedia, Medical Dialog, and FiQA datasets. Meanwhile, HC3-zh encompasses a broader spectrum of domains, featuring 22k human answers and 17k

ChatGPT responses. The data within HC3-zh spans seven sources: WebTextQA, BaikeQA, Crawled BaiduBaike, NLPCC-DBQA, Medical Dialog, Baidu AI Studio, and LegalQA datasets. However, it is pertinent to note some limitations of the HC3 dataset, such as the lack of diversity in prompts used for data creation.

CHEAT. The CHEAT dataset (Yu et al. 2023a) is specifically designed to detect spurious academic content generated by ChatGPT. It includes human-written academic abstracts sourced from IEEE Xplore, with an average abstract length of 163.9 words and a vocabulary size of 130k words. Following the ChatGPT generation process, the dataset contains 15k human-written abstracts and 35k ChatGPT-generated summaries. To better simulate real-world applications, the outputs were guided by ChatGPT for further refinement and amalgamation. The "polishing" process aims to simulate scenarios where users refine LLM-generated text to bypass plagiarism detection, while "blending" represents cases where users combine manually drafted content with ChatGPT-generated text to evade detection. However, a limitation of the CHEAT dataset is its focus on narrow academic disciplines, overlooking cross-domain challenges, which stems from constraints related to its primary data source.

HC3 Plus. HC3 Plus (Su et al. 2023b) built on the original HC3 dataset, introducing an augmented section named *HC3-SI*. This new section specifically targets tasks requiring semantic invariance, such as summarization, translation, and paraphrasing, thus extending the scope of HC3. To compile the human-written text corpus for HC3-SI, data was curated from several sources, including the CNN/DailyMail dataset, Xsum, LCSTS, the CLUE benchmark, and datasets from the Workshop on Machine Translation (WMT). Simultaneously, the LLM-generated texts were generated using GPT-3.5-Turbo. The expanded English dataset now includes a training set of 95k samples, a validation set of 10k samples, and a test set of 38k samples. The Chinese dataset, in comparison, contains 42k training samples, 4k for validation, and 22k for testing. Despite these expansions, HC3-SI still mirrors HC3's approach to data construction, which is somewhat monolithic and lacks diversity, particularly in the variety of LLMs and the use of complex and varied prompts for generating data.

OpenLLMText. The OpenLLMText dataset (Chen et al. 2023a) incorporates four types of LLMs: GPT-3.5, PaLM, LLaMA-7B, and GPT2-1B (also known as GPT-2 Extra Large). The samples from GPT2-1B are sourced from the GPT-2 Output dataset, which OpenAI has made publicly available. Text generation from GPT-3.5 and PaLM was generated using the prompt "Rephrase the following paragraph by paragraph: [Human_Sample]," while LLaMA-7B generated text by completing the first 75 tokens of human samples. The dataset comprises a total of 344k samples, including 68k written by humans. It is divided into training, validation, and test sets at 76%, 12%, and 12%, respectively. Notably, this dataset features LLMs like PaLM, which are commonly used in everyday applications. However, it does not fully capture the nuances of cross-domain and multilingual text, which limits its usefulness for related research.

TweepFake Dataset. TweepFake (Fagni et al. 2021) is a foundational dataset designed for the analysis of fake tweets on Twitter, derived from both genuine and counterfeit accounts. It encompasses a total of 25k tweets, with an equal distribution between human-written and machine-generated samples. The machine-generated tweets were crafted using various techniques, including GPT-2, RNN, Markov, LSTM, and Char-RNN. Although TweepFake remains a valuable dataset resource of choice for many

scholars, those working with LLMs should critically assess its relevance and rigor in light of evolving technological capabilities.

GPT2-Output Dataset. The GPT2-Output Dataset,[7] introduced by OpenAI, is based on 250k documents sourced from the WebText test set for its human-written text. Regarding the LLM-generated text, the dataset includes 250k randomly generated samples using a temperature setting of 1 without truncation and an additional 250k samples produced with Top-K 40 truncation. This dataset was conceived to further research into the detectability of the GPT-2 model. However, a notable limitation lies in the insufficient complexity of the dataset, marked by the uniformity of both the generative models and data distribution.

GROVER Dataset. The GROVER Dataset, presented by Zellers et al. (2019b), is styled after news articles. Its human-written text is sourced from RealNews, a comprehensive corpus of news articles derived from Common Crawl. The LLM-generated text is produced by Grover-Mega, a transformer-based news generator with 1.5 billion parameters. A limitation of this dataset, particularly in the current LLM landscape, is the uniformity and singularity of both its generative model and data distribution.

ArguGPT Dataset. The ArguGPT Dataset (Liu et al. 2023d) is specifically designed for detecting LLM-generated text in various academic contexts such as classroom exercises, TOEFL, and GRE writing tasks. It comprises 4k argumentative essays, generated by seven distinct GPT models. Its primary aim is to tackle the unique challenges associated with teaching English as a second language.

DeepfakeTextDetect Dataset. Attention is also drawn to the DeepfakeTextDetect Dataset (Li et al. 2023c), a robust platform tailored for deepfake text detection. The dataset combines human-written text from ten diverse datasets, encompassing genres like news articles, stories, scientific writing, and more. The dataset comprises texts generated by 27 prominent LLMs, sourced from entities such as OpenAI, LLaMA, and EleutherAI. Furthermore, the dataset introduces an augmented challenge with the inclusion of text produced by GPT-4 and paraphrased text.

4.1.2 Potential Datasets. Creating datasets from scratch that include both human-written and LLM-generated text can be highly resource-intensive. As a result, researchers often use existing datasets to represent human-written text and generate new text using LLMs for training detectors. We refer to such datasets as "potential datasets." These datasets can be categorized into various writing domains such as question answering, scientific writing, creative writing, social media, and web text, aligning closely with real-world use cases and taking into account the potential harm and higher likelihood of misuse of LLM-generated text (Mitchell et al. 2023). Table 4 provides an organized classification of commonly used datasets in current LLM-generated text detection research. Moreover, with the advancement of sophisticated LLMs, it has become increasingly challenging to ensure that new human-written datasets are free from LLM-generated content. Consequently, older human-written datasets may play a crucial role in developing future defenses against LLM-generated text.

7 https://github.com/openai/gpt-2-output-dataset.

Table 4
Summary of other potential datasets that can be easily extended to LLM-generated text detection tasks.

Corpus	Size	Source	Language	Domain
XSum (Narayan, Cohen, and Lapata 2018)	42k	BBC	English	News Writing
SQuAD (Rajpurkar et al. 2016)	98.2k	Wiki	English	Question Answering
WritingPrompts (Fan, Lewis, and Dauphin 2018)	302k	Reddit WRITINGPROMPTS	English	Creative Writing
Wiki40B (Guo et al. 2020)	17.7m	Wiki	40+ Languages	Web Text
PubMedQA (Jin et al. 2019)	211k	PubMed	English	Question Answering
Children's Book Corpus (Hill et al. 2016)	687k	Books	English	Question Answering
Avax Tweets Dataset (Muric, Wu, and Ferrara 2021)	137m	Twitter	English	Social Media
Climate Change Dataset (Littman and Wrubel 2019)	4m	Twitter	English	Social Media
Yelp Dataset (Asghar 2016)	700k	Yelp	English	Social Media
ELI5 (Fan et al. 2019)	556k	Reddit	English	Question Answering
ROCStories (Mostafazadeh et al. 2016)	50k	Crowdsourcing	English	Creative Writing
HellaSwag (Zellers et al. 2019a)	70k	ActivityNet Captions, Wikihow	English	Question Answering
SciGen (Moosavi et al. 2021)	52k	arXiv	English	Scientific Writing, Question Answering
WebText (Radford et al. 2019)	45m	Web	English	Web Text
TruthfulQA (Lin, Hilton, and Evans 2022)	817	authors writtEnglish	English	Question Answering
NarrativeQA (Kočiský et al. 2018)	1.4k	Gutenberg3, web	English	Question Answering
TOEFL11 (Blanchard et al. 2013)	12k	TOEFL test	11 Languages	Scientific Writing
Peer Reviews (Kang et al. 2018)	14.5k	NIPS 2013–2017, CoNLL 2016, ACL 2017 ICLR 2017, arXiv 2007–2017	English	Scientific Writing

There are two main approaches to the construction of LLM generated text. One involves using prompts to directly instruct the model to write or answer questions. For example, in news writing, you might prompt the model with, "Please write an article for BBC News with the following headline: <headline>." Examples of such prompts can be found in the work by Li et al. (2023c), which provides many specified prompts. The other method involves providing the LLM with an opening sentence and guiding it to continue the narrative. For instance, in news writing, you might instruct the model with, "Please write an article starting exactly with: <prefix>." This approach helps align LLM-generated text more closely with human writing. For more details, please refer to the prompt settings and examples in Bao et al. (2023).

4.2 Evaluation Benchmarks

Benchmarks with higher quality can help researchers verify whether their detectors are rapidly feasible and effective. We sort out and compare the benchmarks that are currently popular or have potential, as shown in Table 5. On the one hand, we hope to help researchers better understand their differences to choose suitable benchmarks for their experiments. On the other hand, we hope to draw researchers' attention to the latest benchmarks, which have been fully designed to verify the latest issues for the task, with great potential.

TuringBench. The TuringBench dataset (Uchendu et al. 2021) is an initiative designed to explore the challenges of the "Turing test" in the context of neural text generation techniques. It comprises human-written content derived from 10k news articles, predominantly from reputable sources such as CNN. For the purpose of this dataset, only articles

Table 5
Summary of benchmarks for LLM-generated text detection.

Corpus	Use	Human	LLMs	LLMs Type	Language	Attack	Domain
TuringBench (Uchendu et al. 2021)	train	~8k	~159k	GPT-1, GPT-2, GPT-3, GROVER, CTRL, XLM, XLNET, FAIR, TRANSFORMER_XL, PPLM	English	–	News Writing
MGTBench (He et al. 2023b)	train test	~2.4k ~0.6k	~14.4k ~3.6k	ChatGPT, ChatGPT-turbo, ChatGLM, Dolly, GPT4All, StableLM	English	Adversarial	Scientific Writing, Story Generation, News Writing
GPABenchmark (Liu et al. 2023e)	test	~150k	~450k	GPT-3.5	English	Paraphrase	Scientific Writing
Scientific-articles Benchmark (Mosca et al. 2023)	test	~16k	~13k	SCIgen, GPT-2, GPT-3, ChatGPT, Galactica	English	–	Scientific Writing
MULTITuDE (Macko et al. 2023)	train test	~4k ~3k	~40k ~26k	Alpaca-lora, GPT-3.5-Turbo, GPT-4, LLaMA, OPT, OPT-IML-Max, Text-Davinci-003, Vicuna	Arabic, Catalan, Chinese, Czech, Dutch, English, German, Portuguese, Russian, Spanish, Ukrainian	–	Scientific Writing, News Writing, Social Media
HANSEN (Tripto et al. 2023)	test	–	~21k	ChatGPT, PaLM2, Vicuna13B	English	–	Spoken Text
M4 (Wang et al. 2023b)	train valid test	~35k ~3.5k ~3.5k	~112k ~3.5k ~3.5k	GPT-4, ChatGPT, GPT-3.5, Cohere, Dolly-v2, BLOOMz 176B	English, Chinese, Russian, Urdu, Indonesian, Bulgarian, Arabic	–	Web Text, Scientific Writing, News Writing, Social Media, QA
DetectRL (Wu et al. 2024b)	train test	~100k	~134k	GPT-3.5-turbo, Claude-instant, Palm-2-bison, Llama-2-70b	English	Prompt, Paraphrase, Adversarial	Scientific Writing, News Writing, Story Generation, Social Media

ranging between 200 to 400 words were selected. LLM-generated text within this dataset is produced by 19 distinct text generation models, including GPT-1, GPT-2 variants (small, medium, large, xl, and PyTorch), GPT-3, different versions of GROVER (base, large, and mega), CTRL, XLM, XLNET variants (base and large), FAIR for both WMT19 and WMT20, Transformer-XL, and both PLM variants (distil and GPT-2). Each model contributed 8k samples, categorized by label type. Notably, TuringBench emerged as one of the pioneering benchmark environments for the detection of LLM-generated text. However, given the rapid advancements in LLM technologies, the samples within TuringBench are now less suited for training and validating contemporary detector performances. As such, timely updates incorporating the latest generation models and their resultant texts are imperative.

MGTBench. Introduced by He et al. (2023b), MGTBench stands as the inaugural benchmark framework for machine-generated text (MGT) detection. It boasts a modular architecture, encompassing an input module, a detection module, and an evaluation module. The dataset draws upon several of the foremost LLMs, including ChatGPT, ChatGLM, Dolly, ChatGPT-turbo, GPT4All, and StableLM, for text generation. Additionally, it incorporates over ten widely recognized detection algorithms, demonstrating significant potential.

GPABenchmark. The GPABenchmark (Liu et al. 2023e) is a comprehensive dataset encompassing 600k samples. These samples encompass four categories: human-written, GPT-written, GPT-completed, and GPT-polished abstracts from a broad spectrum of academic disciplines, such as computer science, physics, and the humanities and social sciences. This dataset meticulously captures the critical scenarios reflecting both the utilization and potential misapplication of LLMs in academic composition. Consequently, it delineates three specific tasks: generation of text based on a provided title, completion of a partial draft, and refinement of an existing draft. Within the domain of academic writing detection, GPABenchmark stands as a robust benchmark, attributed to its voluminous data and its holistic approach to scenario representation.

Scientific-articles Benchmark. The Scientific-articles Benchmark (Mosca et al. 2023) comprises 16k human-written articles alongside 13k LLM-generated samples. The human-written articles are sourced from the ArXiv dataset available on Kaggle. In contrast, the machine-generated samples, which include abstracts, introductions, and conclusions, are produced by SCIgen, GPT-2, GPT-3, ChatGPT, and Galactica using the titles of the respective scientific articles as prompts. A notable limitation of this dataset is its omission of various adversarial attack types.

MULTITuDE. This is a benchmark for detecting machine-generated text in multiple languages. This dataset consists of 74k machine-generated texts and 7k human-written texts across 11 languages (Macko et al. 2023), including Arabic, Catalan, Chinese, Czech, Dutch, English, German, Portuguese, Russian, Spanish, and Ukrainian. The machine-generated texts are produced by eight generative models, including Alpaca-Lora, GPT-3.5-turbo, GPT-4, LLaMA, OPT, OPT-IML-Max, Text-Davinci-003, and Vicuna. In an era of rapidly increasing numbers of multilingual LLMs, MULTITuDE serves as an effective benchmark for assessing the detection capabilities of LLM-generated text detectors in various languages.

HANSEN. The Human and AI Spoken Text Benchmark (HANSEN) (Tripto et al. 2023) is the largest benchmark for spoken text, encompassing the organization of 17 speech datasets and records, as well as 23k novel AI-generated spoken texts. The AI-generated spoken texts in HANSEN were created by ChatGPT, PaLM2, and Vicuna-13B. Due to the stylistic differences between spoken and written language, detectors may require a more nuanced understanding of spoken text. HANSEN can effectively assess the progress in research aimed at developing such nuanced detectors.

M4. M4 (Wang et al. 2023b) represents a comprehensive benchmark corpus for the detection of text generated by LLMs. It spans a variety of generators, domains, and languages. Compiled from diverse sources, including wiki pages from various regions, news outlets, and academic portals, the dataset reflects common scenarios where LLMs are utilized in daily applications. The LLM-generated texts in M4 are created using

cutting-edge generative models such as ChatGPT, LLaMa, BLOOMz, FlanT5, and Dolly. Notably, the dataset captures cross-lingual subtleties, featuring content in more than ten languages. While the dataset effectively addresses challenges across diverse domains, languages, and models, it could be further enhanced by incorporating a wider array of adversarial scenarios to broaden its applicability.

DetectRL. DetectRL (Wu et al. 2024b) is a benchmark explicitly designed to evaluate the effectiveness of LLM-generated text detectors in real-world application scenarios. It consists of four evaluation tasks: In-domain Robustness, Generalization, Varying Text Length, and Real-World Human Writing Assessment. This benchmark covers writing domains that are particularly susceptible to misuse, including academic writing, news writing, creative writing, and social media. It supports popular LLMs such as GPT-3.5-turbo, Claude-instant, Palm-2-bison, and Llama-2-70b. Unlike prior studies, DetectRL employs heuristic rules to generate adversarial LLM-generated texts, simulating realistic scenarios such as various prompt usages, human revisions (e.g., word substitutions), and typographical errors. The benchmark contains a total of 100k human-written samples and 134k LLM-generated samples, including both original and attacked examples.

5. Advances in Automated Detection Research

This section provides an overview of different detector designs and detection algorithms, including watermarking technology, statistics-based detectors, and neural-based detectors. The discussion is systematically structured and is organized by their underlying principles (see Figure 4), with a detailed analysis of each method's contributions, comparisons to other approaches, examination of the datasets used, and examples to illustrate their effectiveness.

5.1 Watermarking Technology

Initially developed in the field of computer vision, watermarking techniques have played a critical role in detecting AI-generated images and safeguarding intellectual property in the visual arts. With the rise of LLMs, watermarking technology has been adapted to identify text generated by these models. These techniques not only protect models from unauthorized access, such as sequence distillation, but also mitigate risks related to the replication and misuse of LLM-generated text.

It is important to note that watermarking technology differs significantly from statistics-based or neural-based detection approaches. It is not solely designed for the broad identification of text generated by LLMs. Instead, it serves as a regulatory framework tailored to specific models, requiring access to the deployment of the model for implementation. Thus, while watermarking is a specialized detection method with effective capabilities, it also has the potential to complement statistics-based or neural-based detection techniques (Mitchell et al. 2023).

5.1.1 Data-Driven Watermarking. Data-driven methods enable the verification of data ownership or the tracking of illegal copying or misuse by embedding specific patterns or tags within the training datasets of LLMs. These methods typically rely on backdoor insertion, where a small number of watermarked samples are added to the dataset, enabling the model to implicitly learn a secret function set by the defender. When a specific trigger is activated, the backdoor watermark is triggered, which is usually implemented in a black-box setting (Gu et al. 2022). This mechanism protects the model

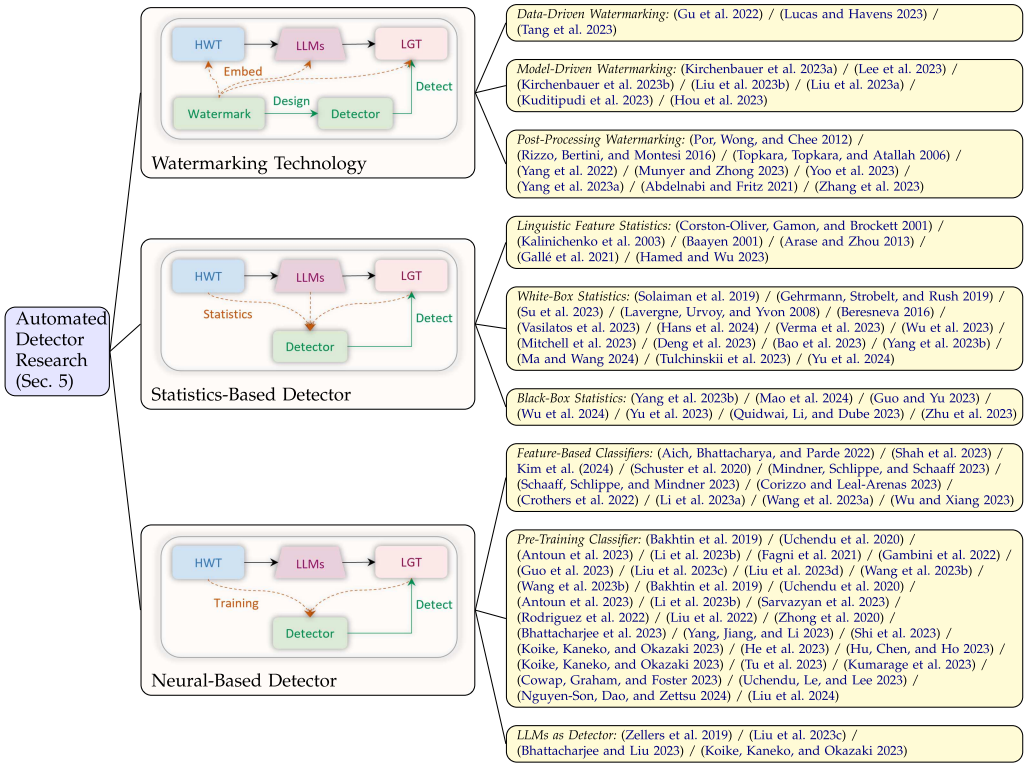

Figure 4
Classification of LLM-generated text detectors with corresponding diagrams and paper lists. We categorize the detectors into watermarking technology, statistics-based detectors, neural-based detectors, and human-assisted methods. In the diagrams, HWT represents Human-Written Text and LGT represents LLM-Generated Text. We use the brown lines to highlight the source of the detector's detection capability, and the green lines to describe the detection process.

from unauthorized fine-tuning or use beyond the terms of the license by embedding a backdoor during the foundational and multi-task learning framework phases of model training, specified by the owner's input. Experimental results show that the watermark embedded using this method can be robustly extracted with a high success rate and will not be erased by subsequent fine-tuning.

However, subsequent studies identified vulnerabilities in this technology, showing that it can be relatively easily compromised. Lucas and Havens (2023) detailed an attack method on this watermarking strategy by analyzing the content generated by autoregressive models to precisely identify the trigger words or phrases of the backdoor watermark. Their research indicates that triggers composed of randomly combined common words are easier to detect than those made up of unique and rare markers. Additionally, the research mentions that access to the model's weights is the only prerequisite for detecting the backdoor watermark. Recently, Tang et al. (2023) introduced a clean-label backdoor watermarking framework that uses subtle adversarial perturbations to mark and trigger samples. Unlike previous methods that require adding arbitrary and mislabeled data to the training set, this approach minimizes the impact on the original task performance. Results indicate that incorporating just 1% of watermarked samples can embed a traceable watermark feature while remaining visually imperceptible.

It is important to note that data-driven methods were initially designed to protect the copyright of datasets and hence generally lack substantial payload capacity and generalizability. Moreover, applying such techniques in the field of LLM-generated text detection requires significant resource investment, including the embedding of watermarks in large datasets and retraining LLMs.

5.1.2 Model-Driven Watermarking. Model-driven methods embed watermarks directly into the LLMs by manipulating the logits output distribution or token sampling during the inference process. As a result, the LLMs generate responses that carry the embedded watermark, enabling effective regulation.

Logits-Based Methods. Kirchenbauer et al. (2023a) were the first to design a logits-based watermarking framework for LLMs, known as WLLM. This approach minimizes impact on text quality and does not require access to the LLM's API or parameters. It involves selecting a random set of "green" tokens before generating words and subtly promoting their use during sampling. The watermark is then identified through statistical analysis of "red" and "green" tokens in the text. Experiments demonstrate that WLLM has a low false-positive rate, and the watermark degrades gracefully when under attack, showcasing strong reliability. Building on WLLM, Lee et al. (2023) proposed a method called Selective Watermarking via Entropy Thresholding (SWEET) for code generation. Unlike WLLM, SWEET enhances "green" tokens only at positions with high token distribution entropy during generation, ensuring the watermark remains both stealthy and intact. Results show that SWEET outperforms watermark baselines WLLM (Kirchenbauer et al. 2023a) and EXP-EDIT (Kuditipudi et al. 2024) and zero-shot methods including LogRank (Solaiman et al. 2019) and DetectGPT (Mitchell et al. 2023). SWEET improves detection capability (by over 10% AUROC) and demonstrates robust performance against real-world challenges, including paraphrasing attacks.

Despite the impressive performance of WLLM (Kirchenbauer et al. 2023a) and SWEET (Lee et al. 2023), Krishna et al. (2023) have shown that the robustness of these methods in paraphrasing LLM-generated text still requires improvement. A key issue is that the token's watermark logit is influenced by a specific number of preceding tokens—too few to reduce security, while too many to compromise robustness against attacks. To address this challenge, Liu et al. (2023b) proposed a Semantic Invariant Robust Watermark (SIR) for LLMs. This approach generates semantic embeddings for all previous tokens and uses them to determine the watermark logic, offering greater robustness against synonym substitution and text paraphrasing compared with WLLM (Kirchenbauer et al. 2023a) and EXP-EDIT (Kuditipudi et al. 2024). Recently, a multilingual version, X-SIR (He et al. 2024), was developed to tackle the inconsistency issues faced by logits-based text watermarking when translating text into different languages.

Moreover, current watermark detection algorithms require a secret key during generation, which can introduce security vulnerabilities and the risk of forgery in public detection processes. To address these issues, Liu et al. (2023a) introduced an Unforgeable Publicly Verifiable (UPV) watermarking algorithm. This method utilizes two different neural networks for watermark generation and detection to avoid using the same key in both stages. Compared to WLLM (Kirchenbauer et al. 2023a), it demonstrates a lower false positive rate and stronger resistance to paraphrasing attacks.

In evaluating the robustness of logits-based text watermarking, Kirchenbauer et al. (2023b) assessed the resilience of watermarked text against various attacks, including manual rewriting, rewriting using non-watermarked LLMs, and integration into large handwritten document corpora. The findings revealed that watermarks could be

detected after an average of just 800 tokens, even though real-world attacks do weaken the watermark's effectiveness. This suggests that watermarking might be the most reliable method compared with other LLM-generated text detectors.

Token Sampling-Based Methods. During the normal model inference process, token sampling is determined by the sampling strategy and is often random, which helps guide the LLMs to produce more unpredictable text. Token sampling-based methods achieve watermarking by influencing the token sampling process, either by setting random seeds or specific patterns for token sampling. Kuditipudi et al. (2023) introduced a sequence of random numbers as a secret watermark key to intervene in and guide token sampling, which is then embedded into the LLMs to generate watermarked text. This approach is the first distortion-free watermarking strategy. Experiments on OPT-1.3B, LLaMA-7B, and Alpaca-7B demonstrate robustness against various paraphrasing attacks (e.g., editing and cropping), even when about 40–50% of the tokens are modified.

SemStamp (Hou et al. 2023) addresses the vulnerability of existing watermarking algorithms to paraphrasing attacks due to their token-level design. It is a robust sentence-level semantic watermarking algorithm based on Locality-Sensitive Hashing (LSH). The algorithm starts by encoding candidate sentences generated by the LLM and uses LSH hashing to partition the semantic embedding space into watermarked and non-watermarked regions. It then continuously performs sentence-level rejection sampling until a sentence falls into the watermarked region of the semantic embedding space. Experimental results show that this approach is more resilient against common and effective paraphrasing attacks, such as bigram paraphrase attacks, compared with WLLM (Kirchenbauer et al. 2023a), while maintaining superior text generation quality.

In general, model-driven watermarking is a plug-and-play method that does not require any changes to the model's parameters and has minimal impact on text quality, making it a reliable and practical watermarking approach. However, there is still significant opportunity for improvement in its robustness, and its specific usability needs to be further explored through additional experiments and practical applications.

5.1.3 Post-Processing Watermarking. Post-processing watermarking refers to a technique that involves embedding a watermark by processing the text after it has been generated by an LLM. This method typically functions as a separate module that works in a pipeline with the output of the generative model.

Character-Embedded Methods. Early post-processing watermarking techniques relied on the insertion or substitution of special Unicode characters in text, which was known at the time as the open space method (Bender et al. 1996) or Steganography (Provos and Honeyman 2003). These characters, which are difficult to detect with the naked eye, carried unique encoding information. Subsequent research utilized sentence spacing, word spacing, paragraph spacing, and end-of-line spacing to further reduce the visibility of embedded messages (Chotikakamthorn 1998; Por, Ang, and Delina 2008).

Por, Wong, and Chee (2012) presented a straightforward character-embedded method based on space character operations. They inserted selected Unicode space characters between sentences, words, lines, and paragraphs to embed watermarks in text, offering strong imperceptibility.

Rizzo, Bertini, and Montesi (2016) introduced Easymark, a technique that cleverly took advantage of the fact that Unicode has many visually identical or similar code points. Unlike earlier methods (Por, Wong, and Chee 2012), Easymark was not restricted

by specific file formats and could be effectively applied to short texts. Specifically, Easymark embedded watermarks by replacing the regular space character (U+0020) with another blank code point (e.g., U+2004), using Unicode's variant selectors, substituting substrings, or using spaces and homoglyphs of slightly different lengths, while ensuring the text's appearance remained nearly unchanged. Results on a real dataset of 1.8 million New York Times articles showed that watermarks embedded with Easymark could be reliably detected.

Synonym Substitution-Based Methods. In response to the vulnerability of character-level methods to targeted attacks, some research has shifted towards embedding watermarks at the word level, primarily through synonym substitution.

Early watermark embedding strategies involved the continuous replacement of words with synonyms until the text carried the intended watermark content, as demonstrated in T-tex (Winstein 1998). To address the limitations of these initial methods, Topkara, Topkara, and Atallah (2006) developed a more quantifiable and resilient watermarking technique utilizing Wordnet (Fellbaum 1998). This approach was not entirely reliant on the token insertion process. Specifically, it involved further modifying the document to maximize ambiguity and deliberately operating close to the distortion threshold after the watermark content was embedded. Xiang et al. (2018) presented another approach that effectively combined arithmetic coding with synonym substitution for robust lossless recovery. In this technique, synonyms capable of carrying a meaningful payload were quantified into binary sequences and compressed using adaptive binary arithmetic coding before being embedded in the text. However, these techniques often did not adequately address the impact of synonym substitution on the overall meaning of sentences.

Building upon these foundations, Yang et al. (2022), Munyer and Zhong (2023), and Yoo et al. (2023) have utilized pre-trained or additionally fine-tuned neural models to execute word substitution and detection tasks more effectively, thereby better preserving the semantic relevance of the original sentences. A notable development in this area is presented by Yang et al. (2022), who introduced a context-aware lexical substitution (LS) scheme for natural language watermarking. This method utilizes BERT (Devlin et al. 2019) to assess the semantic relevance between candidate words and the original sentence, recommending LS candidates that ensure superior semantic relevance compared to earlier methods (Topkara, Topkara, and Atallah 2006; Xiang et al. 2018), with average relevance exceeding 98% across six datasets.

Given the prevalence of black-box models, Yang et al. (2023a) have developed a watermarking framework tailored for black-box LMs, enabling third parties to independently embed watermarks into generated texts. This framework involves defining a binary encoding function to randomly encode words into binaries, selectively replacing words denoting binary "0" with contextually relevant synonyms denoting binary "1" to embed the watermark. Experimental results on the Chinese and English datasets from HC3 (Guo et al. 2023) have demonstrated that this method maintains robustness against various attacks including retranslation, text polishing, word deletion, and synonym substitution, without sacrificing the original semantic integrity.

Sequence-to-Sequence Methods. Recent research has explored end-to-end watermark encryption techniques aimed at enhancing flexibility and minimizing artifacts introduced by watermarks. For instance, Abdelnabi and Fritz (2021) proposed Adversarial Watermark Transformer (AWT), the first end-to-end framework to automate the learning of word replacements and their contents for watermark embedding. This method

combines end-to-end and adversarial training, enabling the injection of binary messages into specific input texts at the encoding level. The result is an output text that is virtually imperceptible, with minimal impact on the semantic accuracy and integrity of the original input. In comparison to Synonym Substitution-Based Methods (Topkara, Topkara, and Atallah 2006), AWT offers a superior balance between effectiveness, confidentiality, and robustness.

Zhang et al. (2023a) introduced a Robust and Efficient Watermarking Framework for Generative LLMs (REMARK-LLM), which includes three components: (i) a message encoding module that embeds binary signatures into texts generated by LLMs; (ii) a reparametrization module that converts the dense distribution of message encoding into a sparse distribution for generating watermarked text tokens; and (iii) a decoding module dedicated to extracting signatures. Experiments demonstrated that REMARK-LLM can embed significantly more signature bits (more than double) into the same text while maintaining its semantic integrity. Moreover, compared to AWT (Abdelnabi and Fritz 2021), it shows enhanced resilience against a variety of watermark removal and detection attacks.

Compared to model-driven watermarking, post-processing watermarking may depend more heavily on specific rules, making it more vulnerable to sophisticated attacks that exploit visible clues. For example, systematic patterns introduced through methods like character embedding or synonym substitution can be identified and targeted for removal or disruption, rendering the watermarks ineffective. Despite this risk, post-processing watermarking has significant potential for various applications. Many existing watermarking techniques typically necessitate training within white-box models, making them unsuitable for black-box LLMs settings. For instance, embedding watermarks in GPT-4 is nearly impossible given its proprietary and closed-source nature. Nevertheless, post-processing watermarking provides a solution for adding watermarks to text generated by black-box LLMs, enabling third parties to embed watermarks independently.

5.2 Statistics-Based Methods

Unlike watermarking methods, which embed identifiable patterns within text and require access to the LLM's deployment, statistics-based methods focus on detecting LLM-generated text by analyzing inherent text features. These methods do not rely on additional training through supervised signals; instead, they leverage statistical data to uncover unique patterns and regularities in the generated text. By computing thresholds or analyzing distributional characteristics, statistics-based methods can proficiently identify LLM-generated text without requiring supervised training or specialized access to the model. This independence makes statistics-based approaches more broadly applicable and less dependent on the specific deployment or modification of LLMs. As a result, they provide a complementary or alternative detection strategy to watermarking. In this section, we classify these methods into three categories: linguistic feature statistics, white-box statistics, and black-box statistics, and discuss each in detail.

5.2.1 Linguistic Feature Statistics. The inception of statistics-based detection research can be traced back to the pioneering work of Corston-Oliver, Gamon, and Brockett (2001). In this foundational study, the authors utilized linguistic features, such as the branching properties observed in grammatical analyses of text, function word density, and constituent length, to determine whether a given text was generated by a machine

translation model. These features served as key indicators in distinguishing machine-generated text from human-generated text.

Another early method, dedicated to achieving similar detection goals, uses frequency statistics. For instance, Kalinichenko et al. (2003) and Baayen (2001) utilized frequency statistics associated with the occurrence of word pairs and the distribution characteristics of words within texts. This mechanism was used to determine whether texts were autonomously generated by a generative system. Building on this foundation, Arase and Zhou (2013) developed a detection technique that could identify the "phrase salad" phenomenon in sentences—right and fluent phrases that are sequenced unnaturally. This technique, based on the statistics of fluency feature, grammaticality feature, and gappy-phrase feature, effectively detected low-quality machine-translated sentences from large-scale web texts, achieving an accuracy of 95.8% for sentences and 80.6% for noisy web texts.

Recent studies on LLM-generated text detection have proposed methodologies based on count-based language model features statistics. Gallé et al. (2021) proposed a method of using repeated high-order n-grams to detect LLM-generated documents. This approach is predicated on the observation that certain n-grams appear with unusual frequency within LLM-generated text. Similarly, Hamed and Wu (2023) developed a detection system based on the statistical similarity of bigram counts. Their research revealed that texts produced by ChatGPT accounted for only 23% of all academic bigram content, highlighting substantial variations in terminology between human authors and LLM-generated content. These findings suggest that ChatGPT may possess restricted academic aptitude from a human viewpoint. Their algorithm accurately detected 98 of 100 academic papers authored by LLMs, thus proving the efficacy of their feature engineering strategy in differentiating between texts created by humans and those produced by LLMs.

However, our empirical observations reveal a conspicuous limitation in the application of linguistic feature statistics: their effectiveness heavily depends on access to extensive corpus statistics and diverse types of LLMs.

5.2.2 White-box Statistics. White-box methods for detecting LLM-generated text require direct access to the source model for implementation. The existing white-box detection techniques primarily use zero-shot approaches, which involves obtaining the model's logits output and calculating specific metrics. These metrics are then compared against predetermined thresholds obtained through statistical methods to identify LLM-generated text.

Logits-Based Methods. Logits are the raw outputs produced by LLMs during text generation, specifically from the model's final linear layer before the softmax function. These outputs indicate the model's confidence levels associated with generating each potential subsequent word. The Log-likelihood (Solaiman et al. 2019), a metric derived directly from the logits, measures the average token-wise log probability for each token within the given text. This metric helps determine the likelihood that the text was generated by an LLM and is widely recognized as one of the most popular baseline metrics for LLM-generated text detection.

Similarly, Rank (Solaiman et al. 2019) is another normal baseline computed from logits. The Rank metric calculates the ranking of each word in a sample within the model's output probability distribution. This ranking is determined by comparing the logit score of the word against the logit scores of all other possible words. If the average rank of each word in the sample is high, it suggests that the sample is likely generated by

LLMs. To refine this process, the Log-rank method applies a logarithmic function to each token's rank. A notable application of Log-rank is the GLTR tool (Gehrmann, Strobelt, and Rush 2019), which is designed as a visual forensic tool to facilitate comparative analysis. This tool uses different colors to represent tokens based on their sampling frequency levels, highlighting the proportion of words that a model tends to use in the analyzed text. With the help of GLTR, the human detection rate of fake texts increased from 54% to 72%. Su et al. (2023a) introduced Log-likelihood Ratio Ranking (LRR), which combines Log-likelihood (Solaiman et al. 2019) and Log-rank (Gehrmann, Strobelt, and Rush 2019) by taking the ratio of these two metrics. This approach provides a more comprehensive evaluation by effectively integrating Log-likelihood assessments and Log-rank analysis. Experiments conducted across three datasets and seven language models demonstrated that this method outperforms the Log-likelihood (Solaiman et al. 2019), Rank (Solaiman et al. 2019), and Log-rank (Gehrmann, Strobelt, and Rush 2019) methods by approximately 3.9 and 1.75 AUROC points, respectively.

Entropy represents another early zero-shot method used for evaluating LLM-generated text. It measures the uncertainty or amount of information in a text or model output, typically calculated through the probability distribution of words. High entropy suggests that the content of the sample text is unclear or highly diversified, meaning that many words have a similar probability of being chosen. In such cases, the sample is likely to have been generated by an LLM. Lavergne, Urvoy, and Yvon (2008) used the Kullback-Leibler (KL) divergence to assign scores to n-grams, taking into account the semantic relationships between their initial and final words. This approach identifies n-grams with significant dependencies between the initial and terminal words, thus aiding in the detection of spurious content and enhancing the overall performance of the detection process. However, in recent works by Mitchell et al. (2023) and Bao et al. (2023), methods based on entropy have performed poorly, achieving only an average AUROC of nearly 50%.

The perplexity method based on traditional n-gram language models evaluates the predictive capability of LMs (Beresneva 2016). Recent studies, such as HowkGPT (Vasilatos et al. 2023), use this approach to differentiate between texts written by students and those produced by ChatGPT. This is achieved by calculating and comparing the perplexity scores of each text. Findings indicate that ChatGPT-generated texts exhibit lower perplexity, in contrast to the more evenly distributed scores observed in student responses. This comparative analysis establishes thresholds to accurately identify the origins of submitted assignments. Wu et al. (2023) introduced LLMDet, a tool designed to quantify and categorize the perplexity scores of various models by calculating the probability of the next token for selected n-grams. This approach leverages the text's intrinsic self-watermarking properties (evidenced by surrogate perplexity). Compared to other perplexity-based methods like HowkGPT (Vasilatos et al. 2023), LLMDet allows for effective tracing of text origins and facilitates more fine-grained detection. With a classification accuracy of up to 98.54%, this tool also boasts greater computational efficiency than finely-tuned RoBERTa classifiers (Liu et al. 2019). However, Hans et al. (2024) found that perplexity alone is insufficient as a detection feature, as the text generated by LLMs may exhibit high perplexity scores depending on the specified prompt. This variability renders simple perplexity-based detectors ineffective. Therefore, it is recommended to use the ratio of perplexity measurement to cross-perplexity to address this issue. This approach can detect over 90% of samples generated by ChatGPT (and other LLMs) with a false positive rate of 0.01%. The method, referred to as "Binoculars," describes the extent to which the next-token prediction of one model surprises another model.

In addition, some white-box based derived features are beneficial for LLM-generated text detection. Ghostbuster (Verma et al. 2023) feeds LLM-generated texts into a series of weaker LLMs (from unigram models to unadjusted GPT-3 davinci) to obtain token probabilities, and then conducts a structured search on the combinations of these model outputs to train a linear classifier for distinguishing LLM-generated texts. This detector achieves an average F_1 score of 99.0, which is an increase of 41.6 F_1 score over previous methods such as GPTZero[8] and DetectGPT (Mitchell et al. 2023). Uniform Information Density (UID) (Jain et al. 2018; Wei, Meister, and Cotterell 2021; Venkatraman, He, and Reitter 2023) proves to be another influential feature. By analyzing the token probabilities within samples, Venkatraman, Uchendu, and Lee (2023) have been able to extract UID-based features and then trains a logistic regression classifier to fit the UID characteristics of texts generated by different LLMs. Experimental results on datasets such as TuringBench (Uchendu et al. 2021), GPABenchmark (Liu et al. 2023e), ArguGPT (Liu et al. 2023d), and MAGE (Li et al. 2023c) demonstrate that the method significantly outperforms, by over 20%, traditional supervised and statistical methods like the OpenAI detector (Radford et al. 2019) and DetectGPT (Mitchell et al. 2023) across multiple domains.

Perturbed-Based Methods. A notable study by Mitchell et al. (2023) presents a method to identify text produced by LLMs, based on structural patterns within LLM probability functions. Specifically, it has been observed that LLM-sampled texts tend to be located in regions exhibiting negative curvature in the model's log probability function. This technique merely requires the use of a pre-trained mask-filling model (e.g., small T5) to generate semantically similar text perturbations. DetectGPT demonstrates greater discriminative power than traditional zero-shot approaches, including Log-likelihood (Solaiman et al. 2019), Rank (Solaiman et al. 2019), Log-rank (Solaiman et al. 2019), and Entropy Gehrmann, Strobelt, and Rush (2019). It significantly enhances the detection rate of fake news articles generated by the GPT-NeoX-20B, improving the AUROC from the strongest zero-shot baseline of 0.81 to 0.95 with DetectGPT. Another contemporary work, NPR (Su et al. 2023a), shares a conceptual framework with DetectGPT (Mitchell et al. 2023). NPR utilizes normalized perturbation log-rank to detect text generated by LLMs. Compared to DetectGPT, NPR is less sensitive to the type and quantity of perturbations, achieving improved performance with an approximate 2% increase in AUROC.

While innovative and sometimes more effective than supervised methods, DetectGPT has limitations, including potential performance drops if rewrites don't adequately represent the space of meaningful alternatives, and high computational demands, as it requires perturbing and scoring numerous texts. To address these challenges, Deng et al. (2023) proposed a method using a Bayesian surrogate model to score a small set of representative samples. By extrapolating the scores from these samples to others, the method enhances query efficiency and reduces computational overhead while preserving performance. Extensive empirical studies on datasets such as GPT-2, LLaMA2, and Vicuna have demonstrated its efficiency over DetectGPT (Mitchell et al. 2023), especially in detecting texts generated by the LLaMA. This method achieved superior results using just 2–3 queries compared to DetectGPT's 200 queries. Another significant advancement is reported by Bao et al. (2023), who introduced an approach based on the hypothesis that token-level conditional probability curvature is a more fundamental metric

8 https://gptzero.me/.

of LLM-generated text. This approach replaces the perturbation steps in DetectGPT (Mitchell et al. 2023) with more efficient sampling steps. Evaluations across a variety of datasets, source models, and testing conditions have shown that Fast-DetectGPT not only significantly enhances detection accuracy by approximately 75% over DetectGPT in both white-box and black-box settings under consistent experimental conditions but also boosts detection speed by 340 times. Another approach stems from the white-box configuration in DNA-GPT (Yang et al. 2023b), which also utilizes probability curves but diverges from DetectGPT (Mitchell et al. 2023). Rather than using a perturbation framework, this method leverages LLMs like ChatGPT to repeatedly extend truncated texts. By calculating the probability divergence, it analyzes the differences between the original and the extended texts. This technique has shown superior performance in distinguishing between human-written and GPT-generated texts on four English and one German dataset, achieving nearly 100% detection accuracy and outperforming the OpenAI classifier (Radford et al. 2019).

To enhance white-box LLM-generated text detection, Ma and Wang (2024) proposed a method called TOCSIN. The key innovation is Token Cohesiveness, a feature that quantifies the semantic tightness of a text by randomly deleting 1.5% of its tokens and calculating the average semantic difference between the original and altered versions. Due to the causal self-attention mechanism in LLMs, LLM-generated text tends to exhibit higher cohesiveness, while human-written text is more flexible and less cohesive. TOCSIN uses a dual-channel framework: One channel calculates token cohesiveness, and the other uses an existing zero-shot detector to generate predictions. By combining the two scores, TOCSIN achieves more accurate classification. Experiments show that TOCSIN significantly improves the performance of four mainstream detectors (Likelihood, LogRank, LRR, Fast-DetectGPT), with AUROC gains of 0.59% to 10.97% in white-box scenarios and 0.38% to 3.77% in black-box scenarios, while also enhancing the reliability of cross-model detection.

Intrinsic Features-Based Methods. Tulchinskii et al. (2023) proposed a method for constructing a detector using the intrinsic dimension of the manifold, based on the assumption that humans and LLMs exhibit consistent capabilities in their respective textual domains. Specifically, a contextual embedding is extracted for each token, converting the text into a high-dimensional point cloud. The intrinsic dimension of this point cloud is estimated using the persistent homology dimension (PHD) method, which involves constructing a minimum spanning tree, analyzing the "lifetime" of topological features, and fitting a regression model. This intrinsic dimension is then used as the sole feature in a logistic regression classifier to detect whether the text is LLM-generated. Observations show that the average intrinsic dimension of natural fluent text in various alphabetic languages is about 9 and about 7 for Chinese, while the average intrinsic dimension of LLM-generated text in each language is generally about 1.5 units lower. The proposed detector shows consistent accuracy across text domains, generator, and different levels of human author skills. It significantly outperforms detectors including DetectGPT (Mitchell et al. 2023) and OpenAI Detector (Radford et al. 2019) in scenarios involving multi-domain challenges, model shifts, and adversarial attacks. However, its reliability degrades when applied to suboptimal or high-temperature generators. Text Fluoroscopy (Yu et al. 2024) classifies texts by extracting intrinsic features from the intermediate layers of language models. Unlike methods that rely solely on features from the first or last layer of the model, this approach calculates the distributional differences (KL divergence) between each intermediate layer and the first and last layers, selecting the intermediate layer with the largest distributional difference as the

intrinsic feature layer. This enables more effective identification of differences between human-generated and LLM-generated texts. Experimental results demonstrate that Text Fluoroscopy outperforms existing methods across various datasets and generative models (e.g., ChatGPT, GPT-4, Claude3) with an average improvement of 7.36%, and it exhibits greater robustness against paraphrasing and back-translation attacks.

5.2.3 Black-box Statistics. In contrast to white-box statistical methods, black-box statistical approaches utilize models that calculate scores for specific text features without accessing the logits of either the source or surrogate models. One method, utilized in DNA-GPT (Yang et al. 2023b), uses n-gram probability divergence similarity. It extends the writing of a truncated text under review using an LLM, and evaluates the similarity of n-gram probability divergence between the continuation and the original text. This process helps distinguish between human-written texts and those generated by LLMs. Experiments conducted on models such as GPT-3.5-turbo and GPT-4, as well as open-source models like GPT-NeoX-20B and LLaMa-13B, have shown superior performance compared to the OpenAI detector (Radford et al. 2019), with a minimal drop from 99.09 AUROC to 98.48 AUROC when facing modification attacks.

Another approach calculates similarity scores between original texts and their rewritten or revised versions to identify LLM-generated text, as demonstrated by Mao et al. (2024) and Zhu et al. (2023). This method is based on the observation that texts rewritten and edited by an LLM tend to undergo fewer modifications than human-written texts, as LLMs favor their own generative logic and statistical patterns, making them less likely to initiate changes. Thus, texts that show higher similarity between their original and modified versions are more likely to be LLM-generated. Experiments across various domains, including news writing, creative writing, academic writing, code generation, and social media, have significantly enhanced the efficacy of current AI detectors, showing better generalization capabilities than white-box statistical methods like DetectGPT (Mitchell et al. 2023) and Ghostbuster (Verma et al. 2023). GECScore (Wu et al. 2024a) is a simple yet effective black-box zero-shot method, designed based on the observation that, from the perspective of LLMs, human-written texts often contain more grammatical errors than texts generated by LLMs. The method achieves an AUROC of approximately 98.62% and demonstrates greater reliability in real-world scenarios compared with methods relying on simple LLM revision preferences (Zhu et al. 2023) and the powerful zero-shot method Fast-DetectGPT (Bao et al. 2023), showcasing outstanding generalization capabilities. AuthentiGPT (Guo and Yu 2023) utilizes a similar black-box approach with a zero-shot denoising technique to identify LLM-generated texts. It involves using a black-box LLM to remove noise artificially added to the input texts, then comparing the denoised text semantically to the original. This method has achieved an AUROC score of 91.8% in specific academic writing domain, surpassing commercial detectors such as GPTZero[9] and Originality.AI,[10] and baseline methods using GPT-3.5 and GPT-4 as detectors.

Yu et al. (2023b) introduced a novel detection mechanism that leverages the similarity between original texts and their regenerated versions. This method considers the generation process as a coupling of the generative model's prompt features and intrinsic characteristics. It reconstructs prompts corresponding to the candidate text using an auxiliary LLM, then regenerates the text from these prompts, aligning the candidate

9 https://gptzero.me/.
10 https://originality.ai/.

and regenerated texts with the prompts accordingly. This alignment's similarity is then used as the primary detection feature, enabling the detector to focus on the intrinsic properties of the generative model rather than the prompts. Compared to supervised methods like the RoBERTa classifier (Liu et al. 2019) and RADAR (Hu, Chen, and Ho 2023), and statistical methods such as DNA-GPT (Yang et al. 2023b), DetectGPT (Mitchell et al. 2023), and Fast-DetectGPT (Bao et al. 2023), DPIC (Yu et al. 2023b) achieves an average improvement of 6.76% and 2.91% over the best baseline in detecting texts generated by GPT-4 and Claude3 across various domains.

Additionally, acknowledging the absence of quantifiable metrics at the sentence level, Quidwai, Li, and Dube (2023) proposed a more comprehensive comparison technique to provide more accurate and interpretable evaluations. This method involves analyzing sets of sentences within LLM-generated texts and their paraphrases, differentiating them from human-written texts through cosine similarity measurements, achieving an accuracy rate as high as 94%, which outperforms supervised methods like the RoBERTa classifier (Liu et al. 2019).

However, the approaches of black-box statistics involves challenges, including the substantial overhead of accessing the LLM and long response times.

5.3 Neural-Based Methods

5.3.1 Feature-Based Classifiers.

Linguistic Feature-Based Classifiers. When comparing texts generated by LLMs with those written by humans, the differences in numerous linguistic features provide a solid foundation for feature-based classifiers to effectively distinguish between them. The workflow of such classifiers typically starts with the extraction of key statistical language features, followed by the application of machine learning techniques to train a classification model. This approach has been widely used in the identification of fake news. For instance, in a recent study, Aich, Bhattacharya, and Parde (2022) achieved impressive accuracy of 97% on fake news detection by extracting 21 textual features and using a k-nearest neighbor classifier. Drawing inspiration from the tasks of detecting fake news and LLM-generated texts, the linguistic features of texts can be extensively categorized into stylistic features, complexity features, semantic features, psychological features, and knowledge-based features. These features are primarily obtained through statistical methods.

Stylistic features primarily focus on the frequency of words that specifically highlight the stylistic elements of the text, including the frequency of capitalized words, proper nouns, verbs, past tense words, stopwords, technical words, quotes, and punctuation (Horne and Adali 2017). Complexity Features are extracted to represent the complexity of the text, such as the type-token ratio (TTR) and textual lexical diversity (MTLD) (McCarthy 2005). Semantic Features include Advanced Semantic (AdSem), Lexico Semantic (LxSem), and statistics of semantic dependency tags, among other semantic-level features. These can be extracted using tools like LingFeat (Lee, Jang, and Lee 2021). Psychological Features are generally related to sentiment analysis and can be derived based on tools like SentiWordNet (Baccianella, Esuli, and Sebastiani 2010) to calculate sentiment scores, or extracted using sentiment classifiers. Information Features include named entities (NE), opinions (OP), and entity relation extraction (RE), and can be extracted using tools such as UIE (Lu et al. 2022) and CogIE (Jin et al. 2021).

Shah et al. (2023) constructed a classifier based on stylistic features such as syllable count, word length, sentence structure, frequency of function word usage, and

punctuation ratio. This classifier achieved an accuracy of 93%, which effectively demonstrates the significance of stylistic features for LLM-generated text detection. Other work integrated text modeling with a variety of linguistic features through data fusion techniques (Corizzo and Leal-Arenas 2023), which included different types of punctuation marks, the use of the Oxford comma, paragraph structures, average sentence length, the repetitiveness of high-frequency words, and sentiment scores, subsequently training a deep neural network for detection tasks. On datasets in both English and Spanish, this methodology reached F_1 score of 98.36% and 98.29%, respectively, surpassing the performance of both single neural network-based methods (e.g., BERT + SVM) and feature-based methods (e.g., Emotional Semantics + SVM). Furthering this line of inquiry, Mindner, Schlippe, and Schaaff (2023) explored a range of both traditional and novel features for detecting LLM-generated text, adopting a multidimensional approach to bolster the discriminative power of their classifiers. This approach included perplexity-based, semantic, list lookup, document, error-based, readability, AI-feedback, and text vector features. By training the detector using XGBoost, random forest, and neural network-based methods, they achieved an F_1 score of 98% for basic LLM-generated text detection, and 78.9% for basic LLM-rewritten text. The optimized detector outperformed GPTZero[11] by an impressive 183.8% in F_1 score, demonstrating its exceptional detection capabilities.

While these features provide valuable insights into various aspects of the text, research has shown that incorporating deeper structural features can further improve performance. For example, Kim et al. (2024) proposed an approach based on Rhetorical Structure Theory (RST) and recursive hypergraph analysis, focusing on extracting hierarchical discourse topics. Compared with methods directly fine-tuning encoder-based classifiers, models incorporating discourse topic features achieve significant performance improvements on multiple datasets. On the HC3 dataset, the classification model leveraging discourse topic features achieved an F_1 score of 92.4%, which is about 4% higher than the baseline without discourse features, showing higher robustness, especially in long texts and cross-domain tasks.

Although classifiers based on linguistic features have their advantages in distinguishing between human-written and LLM-generated texts, their shortcomings cannot be overlooked. The results from Schaaff, Schlippe, and Mindner (2023) indicate that such classifiers have poor robustness against ambiguous semantics and often underperform neural network features. Moreover, classifiers based on stylistic features may be capable of differentiating between texts written by humans and those generated by LLMs, but their ability to detect LLM-generated misinformation is limited. This limitation is highlighted in Schuster et al. (2020), which shows that language models tend to produce stylistically consistent texts. However, Crothers et al. (2022) suggest that statistical features can offer additional adversarial robustness and can be utilized in constructing integrated detection models.

White-box Features-Based Classifiers. In addition to linguistic features, classifiers based on model features have recently garnered considerable attention from researchers. These classifiers are not only capable of detecting texts generated by LLMs but can also be used for text origin tracing. Sniffer (Li et al. 2023a) was the first to focus on tracing the origins of LLM outputs, using contrastive features between models along with token-level perplexity that aligns with model contrasts. These features evaluate the percentage

11 https://gptzero.me/.

of words that show lower perplexity when comparing one model, θ_i, with another, θ_j. A classifier trained on these features reached an accuracy of 86.0%. SeqXGPT (Wang et al. 2023a) represents a further advancement in the field of text origin tracing, extending the granularity of detection to the sentence level. It utilizes a context network designed around the log probabilities in a white-box LLM, which combines a CNN and a two-layer transformer to encode text and detect LLM-generated content through a sequence tagging task. Experiments demonstrate that SeqXGPT achieves superior results over RoBERTa (Liu et al. 2019), DetectGPT (Mitchell et al. 2023), and Sniffer (Li et al. 2023a) in both sentence-level and document-level LLM-generated text detection.

However, a common limitation of these methods is their reliance on accessing the source models' logits. This requirement may limit their effectiveness when applied to other powerful, closed-source models where logits are inaccessible.

5.3.2 Pre-training Classifiers.

In-domain Fine-tuning is All You Need. This subsection explores methods involving the fine-tuning of encoder-based classifiers to distinguish between texts generated by LLMs and those written by humans. This approach requires paired samples to facilitate supervised training processes. According to Qiu et al. (2020), pre-trained LMs have demonstrated exceptional capabilities in natural language understanding, which is crucial for enhancing various NLP tasks, particularly text categorization. Prominent pre-trained models, such as BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), and XLNet (Yang et al. 2019), have outperformed traditional statistical machine learning and deep learning counterparts when applied to the text classification tasks within GLUE (Wang et al. 2019). There is an extensive body of prior work (Bakhtin et al. 2019; Uchendu et al. 2020; Antoun et al. 2023; Li et al. 2023c) that has meticulously examined the capabilities of fine-tuned LMs in detecting LLM-generated text. Notably, studies conducted in 2019 have acknowledged fine-tuned LMs, with RoBERTa (Liu et al. 2019) being especially prominent, as being among the most formidable detectors of LLM-generated text. In the following discourse, we will introduce recent scholarly contributions in this vein, providing an updated review and summary of the methods deployed.

Fine-tuning RoBERTa provides a robust baseline for detecting text generated by LLMs. Fagni et al. (2021) observed that fine-tuning RoBERTa led to optimal classification outcomes in various encoding configurations (Gambini et al. 2022), with the subsequent OpenAI detector (Radford et al. 2019) also adopting a RoBERTa fine-tuning approach. Recent work (Guo et al. 2023; Liu et al. 2023d, 2023e; Chen et al. 2023b; Wang et al. 2023c) further corroborated the superior performance of fine-tuned members of the BERT family, such as RoBERTa, in identifying LLM-generated text. Moreover, these fine-tuned models yielded a 95% accuracy rate within their respective domains, outperforming zero-shot methods like DetectGPT (Mitchell et al. 2023) and watermarking methods like WLLM (Kirchenbauer et al. 2023a). Additionally, exhibiting a modicum of resilience to various attack techniques within in-domain settings. Nevertheless, like their counterparts, these encoder-based fine-tuning approaches lack robustness (Bakhtin et al. 2019; Uchendu et al. 2020; Antoun et al. 2023; Li et al. 2023c), as they tend to overfit to their training data or the source model's training distribution, resulting in a decline in performance when faced with unseen domain or data. Additionally, fine-tuning LM classifiers is limited in facing data generated by different models (Sarvazyan et al. 2023). Despite this, detectors based on RoBERTa exhibit significant potential for robustness, requiring as few as a few hundred labels to fine-tune and deliver impressive results (Rodriguez et al. 2022). mBERT (Devlin et al. 2019) has demonstrated consistently robust

performance in document-level LLM-generated text classification and various model attribution settings, maintaining optimal performance particularly in English and Spanish tasks. In contrast, encoder models like XLM-RoBERTa (Conneau et al. 2020) and TinyBERT (Jiao et al. 2020) have shown significant performance disparities in the same document-level tasks and model attribution setups, suggesting that these two tasks may require different capabilities from the models. Additionally, SimLLM (Nguyen-Son, Dao, and Zettsu 2024) improved on the zero-shot work of Mao et al. (2024) and Zhu et al. (2023) and combines it with a fine-tuning approach. The method used a candidate LLM to generate proofread versions of input sentences, compares them with the original input to assess similarity, and organizes the input and its proofread versions based on the similarity score. The RoBERTa model is then fine-tuned to determine the source of the connected sequence and distinguish between human-written content and model-generated content. Results show that this method outperforms DetectGPT (Mitchell et al. 2023) and Revise-Detect (Zhu et al. 2023) in both performance and generalization.

Contrastive Learning. Data scarcity has propelled the application of contrastive learning (Yan et al. 2021; Gao, Yao, and Chen 2021; Chen et al. 2022) to Encoder-based classifiers, with the core of this approach being self-supervised learning. This strategy minimizes the distance between the anchor and positive samples while maximizing the distance to negative samples through spatial transformations. Liu et al. (2022) proposed an enhanced contrastive loss that allocates increased weight to challenging negative samples, thus optimizing model utility and sensitivity, which improves performance in resource-scarce environments. This approach effectively integrates linguistic features and sentence structure, presenting text as coherence graphs to encapsulate inherent entity consistency. Research demonstrates that leveraging information fact structures can significantly enhance the effectiveness of Encoder-based detectors. Experiments on generators such as GROVER, GPT-2, and GPT-3.5 yielded results that surpassed those of the encoder-based classifiers including RoBERTa (Conneau et al. 2020) and XLNet (Yang et al. 2019) classifier, the mainstream contrastive learning method DualCL (Chen et al. 2022), and the zero-shot approach DetectGPT (Mitchell et al. 2023). Similar findings were also highlighted in the work by Zhong et al. (2020). Another contrastive learning application for LLM-generation detection, the Contrastive Domain Adaptation framework (ConDA), was introduced by Bhattacharjee et al. (2023). This framework merges standard domain adaptation techniques with the representational capabilities of contrastive learning, substantially boosting the model's defenses against unknown models. Compared to DetectGPT, ConDA shows an average performance improvement of 31.7%, with only a 0.8% discrepancy from fully supervised RoBERTa detectors. Building on these advancements, PECOLA (Liu et al. 2024) introduced a multi-pairwise contrastive learning strategy combined with selective perturbation to improve noise robustness and token-level sensitivity. By leveraging token importance weights and bridging zero-shot and fine-tuned approaches, PECOLA achieves better generalization and performance. Experiments demonstrate that PECOLA outperforms DetectGPT by 3.84% and fine-tuned RoBERTa classifier by 1.62%.

Adversarial Learning Methods. In light of the vulnerability of detectors to different attacks and robustness issues, there has been significant academic interest in utilizing adversarial learning as a countermeasure. Adversarial learning approaches are mainly linked with the fine-tuning of LMs. A notable recent study by Koike, Kaneko, and Okazaki (2023b) demonstrates that adversarial training can be conducted without fine-tuning the model, using context as a guide to freeze the model parameters. To more clearly describe

this method, we categorize such research into sample-enhancement-based adversarial training and two-player games.

A prominent approach within sample enhancement based adversarial training centers on deploying adversarial attacks predicated on sample augmentation, with the overarching aim of crafting deceptive inputs to thereby enhance the model's competency in addressing a broader array of scenarios that bear deception potential. Specifically, this method emphasizes the importance of sample augmentation and achieves it by injecting predetermined adversarial attacks. This augmentation process is integral to fortifying the detector's robustness by furnishing it with an expanded pool of adversarial samples. Subsection 8.2 of the article outlines various potential attack mechanisms, including paraphrase attacks, adversarial attacks, and prompt attacks. Yang, Jiang, and Li (2023), Shi et al. (2023), and He et al. (2023b) conducted the adversarial data augmentation process on LLM-generated text, the findings of which indicated that models trained on augmented data exhibited commendable robustness against potential attacks.

The methods of Two-Player Games, fundamentally aligned with the principles underpinning Generative Adversarial Networks (Goodfellow et al. 2020) and Break-It-Fix-It strategies (Yasunaga and Liang 2021), typically involve the configuration of an attack model alongside a detection model, with the iterative confrontation between the two culminating in enhanced detection capabilities. Hu, Chen, and Ho (2023) introduced a framework, RADAR, envisaged for the concurrent training of robust detectors through adversarial learning. This framework facilitates interaction between a paraphrasing model, responsible for generating realistic content that evades detection, and a detector whose goal is to enhance its capability to identify text produced by LLMs. The RADAR framework incrementally refines the paraphrase model, drawing on feedback garnered from the detector and employing PPO (Schulman et al. 2017), outperforming zero-shot detection methods including DetectGPT (Mitchell et al. 2023) and OpenAI detector across eight different LLMs and four datasets. Despite its commendable performance in countering paraphrase attacks, the study by Hu, Chen, and Ho (2023) did not provide a comprehensive analysis of RADAR's defense mechanism against other attack modalities. In a parallel vein, Koike, Kaneko, and Okazaki (2023b) proposed a training methodology for detectors predicated on a continual interaction between an attacker and a detector. Distinct from RADAR, OUTFOX allocates greater emphasis on the likelihood of detectors using ICL (Dong et al. 2023) for attacker identification. Specifically, the attacker in the OUTFOX framework utilizes predicted labels from the detector as ICL exemplars to generate text that poses detection challenges. Conversely, the detector uses the content generated adversarially as ICL exemplars to enhance its detection capabilities against formidable attackers. This reciprocal consideration of each other's outputs fosters improved robustness in detectors for text generated by LLMs. Empirical evidence shows that OUTFOX outperforms previous statistical methods (e.g., Log-likelihood [Solaiman et al. 2019] and DetectGPT [Mitchell et al. 2023]) and RoBERTa-based methods (Conneau et al. 2020), achieving up to 96.9 F_1 score and maintaining good performance against attacks utilizing TF-IDF and DIPPER (Krishna et al. 2023).

Features-Enhanced Approaches. In addition to enhancements in training methodology, Tu et al. (2023) demonstrated that the extraction of linguistic features can effectively improve the robustness of a RoBERTa-based detector, with benefits observed in various related models. Cowap, Graham, and Foster (2023) developed an emotion-aware detector by fine-tuning a Pre-trained Language Model (PLM) for sentiment analysis, thereby enhancing the potential of emotion as a signal for identifying synthetic text.

They achieved this by further fine-tuning BERT specifically for sentiment classification, resulting in a detection performance F_1 score improvement of up to 9.03%. Uchendu, Le, and Lee (2023b) used RoBERTa to capture contextual representations, such as semantic and syntactic linguistic features, and integrated Topological Data Analysis to analyze the shape and structure of data, which includes linguistic structure. This approach surpassed the performance of RoBERTa alone on the SynSciPass and M4 datasets. The framework J-Guard (Kumarage et al. 2023a) guides existing supervised LLM-generated text detectors in detecting LLM-generated news by extracting Journalism Features, which help the detector recognize LLM-generated fake news text. The framework demonstrates strong performance and robustness, achieving over 96% AUROC in all tests within TuringBench and 93.4% AUROC on ChatGPT. Even when faced with adversarial attacks, the average performance degradation is as low as 7%.

5.3.3 LLMs as Detectors.

Questionable Reliability of Using LLMs. Several studies have examined the feasibility of utilizing LLMs as detectors to distinguish between text generated by either themselves or other LLMs. This approach was first introduced by Zellers et al. (2019b), who noted that the text generation model Grover produced disinformation that was remarkably deceptive due to its inherently controllable nature. Subsequent analyses by Zellers et al. (2019b), involving various architectural models like GPT-2 (Radford et al. 2019) and BERT (Devlin et al. 2019), revealed that Grover's most effective countermeasure was itself, achieving an accuracy rate of 92%. In contrast, other detector types experienced a decline in accuracy to approximately 70% as Grover's size increased. A recent reevaluation conducted by Bhattacharjee and Liu (2023) on more recent LLMs like ChatGPT and GPT-4 yielded that neither could reliably identify text generated by various LLMs. During the observations, it was noted that ChatGPT and GPT-4 exhibited contrasting tendencies. ChatGPT tended to classify text generated by LLMs as if it were written by humans, with a misclassification probability of about 50%; whereas GPT-4 leaned towards labeling human-written text as if it were generated by LLMs, and about 95% of human-written texts are misclassified as LLM-generated texts. ArguGPT (Liu et al. 2023d) further attested to the lackluster performance of GPT-4-Turbo in detecting text generated by LLMs, with accuracy rates languishing below 50% across zero-shot, one-shot, and two-shot settings. These findings collectively demonstrate the diminishing reliability of utilizing LLMs for direct self-generated text detection, particularly when compared to statistical and neural network methods. This is particularly evident in light of the increasing complexity of LLMs.

ICL: A Powerful Technique for LLM-Based Detection. While using LLMs to detect texts generated by other LLMs often raises reliability concerns, recent empirical research underscores the effectiveness of Instructional Contextual Learning (ICL) in improving detection capabilities. ICL involves an advanced form of cue engineering, which integrates examples directly into the prompts provided to the model, thereby enabling LLMs to learn new tasks more effectively. This approach allows existing LLMs to adeptly manage a variety of tasks without requiring further fine-tuning. Koike, Kaneko, and Okazaki (2023b) introduced a framework that enhances the robustness of detectors for LLM-generated text by facilitating a reciprocal consideration of outputs between detectors and attackers. In this setup, attackers use the prediction labels from the detector as contextual learning examples to craft more elusive texts adversarially. Simultaneously, detectors use these adversarially generated texts as learning examples

to improve their ability to identify content from sophisticated attackers. Experimental results demonstrate that the ICL strategy surpasses traditional zero-shot methods and detectors based on RoBERTa, achieving an F_1 score of up to 96.9 and enhancing the detection performance of texts generated by attackers by as much as 41.3 F_1 score.

6. Human Detection and Enhancements

This section explores human-assisted methods for detecting text generated by LLMs. These methods leverage human prior knowledge and analytical skills, offering notable interpretability and credibility in the detection process.

6.1 Intuitive Indicators

Numerous studies have delved into the disparities between human and machine classification capabilities. Human classification primarily depends on visual observation to discern features indicative of text generation by LLMs. Uchendu et al. (2023) noted that a lack of coherence and consistency in LLM-generated text serves as a strong indicator of falsified content. Texts produced by LLMs often exhibit semantic inconsistencies and logical errors. Additionally, Dugan et al. (2023) identified that the human discernment of LLM-generated text varies across different domains. For instance, LLMs tend to generate more "generic" text in the news domain, whereas, in story domains, the text might be more "irrelevant." Ma et al. (2023) noted that evaluators of academic writing typically emphasize style. Summaries generated by LLMs frequently lack detail, particularly in describing the research motivation and methodology, which hampers the provision of fresh insights. In contrast, LLM-generated papers exhibit fewer grammatical and other types of errors and demonstrate a broader variety of expression (Yan et al. 2023; Liao et al. 2023). However, these papers often use general terms instead of effectively tailored information relevant to the specific problem context. In human-written texts, such as scientific papers, authors are prone to composing lengthy paragraphs and using ambiguous language (Desaire et al. 2023), often incorporating terms like "but," "however," and "although." Dugan et al. (2023) also noted that relying solely on grammatical errors as a detection strategy is unreliable. In addition, LLMs frequently commit factual and common-sense reasoning errors, which, while often overlooked by neural network-based detectors, are easily noticed by humans (Jawahar, Abdul-Mageed, and Lakshmanan 2020).

6.2 Imperceptible Features

Ippolito et al. (2020) suggested that text perceived as high quality by humans tends to be more easily recognizable by detectors. This observation implies that some features, imperceptible to humans, can be efficiently captured by detection algorithms. While humans are adept at identifying errors in many LLM-generated texts, unseen features also significantly influence their decision-making. In contrast, statistical thresholds commonly employed in zero-shot detector research to distinguish LLM-generated text can be manipulated. However, humans typically possess the ability to detect such manipulations through various metrics; GLTR (Gehrmann, Strobelt, and Rush 2019) pioneered this approach, serving as a visual forensic tool to assist human vetting processes, while also providing rich interpretations easily understandable by non-experts (Clark et al. 2021).

6.3 Enhancing Human Detection Capabilities

Ippolito et al. (2020) indicated that human evaluators might not be as proficient as detection algorithms in recognizing LLM-generated text across various settings. However, exposing evaluators to examples before evaluation enhances their detection capabilities, especially with longer samples. The platform RoFT (Dugan et al. 2020) allows users to engage with LLM-generated text, shedding light on human perception of such text. While revealing true boundaries post-annotation did not lead to an immediate improvement in annotator accuracy, it is worth noting that with proper incentives and motivations, annotators can indeed improve their performance over time (Dugan et al. 2023). The SCARECROW framework (Dou et al. 2022) facilitates the annotation and review of LLM-generated text, outlining ten error types to guide users. The result from SCARECROW reports manual annotation outperformed detection models on half of the error types, highlighting potential in developing efficient annotation systems despite the associated human resource overhead.

6.4 Mixed Detection: Understanding and Explanation

Weng et al. (2023) introduced a prototype amalgamating human expertise and machine intelligence for visual analysis, based on the belief that human judgment is the benchmark. Initially, experts label text based on their prior knowledge, elucidating the distinctions between human and LLM-generated text. Subsequently, machine-learning models are trained and iteratively refined based on labeled data. Finally, the most intuitive detector is selected through visual statistical analysis to fulfill the detection purpose. This analytical approach not only bolsters experts' trust in decision-making models but also fosters learning from the models' behavior, improving the identification of LLM-generated samples.

7. Evaluation

7.1 Evaluation Metrics

Evaluation metrics are indispensable for the assessment of model performance within any NLP task, including LLM-generated text detection. This section discusses conventionally utilized metrics in these tasks. Common metrics include *Accuracy*, *Precision*, *Recall*, F_1 *score*, and *AUROC* (Dalianis 2018). *Accuracy* provides an overall measure of success in correctly identifying both human-written and LLM-generated texts, making it a straightforward metric for assessing the general effectiveness of a detection model. *Precision* and *Recall* in LLM-generated text detection are used to assess the precision of identifying LLM-generated text and the ability to capture all relevant instances of such content. The F_1 score constitutes a harmonic mean of Precision and Recall, integrating the considerations of false positives and false negatives (Sokolova, Japkowicz, and Szpakowicz 2006). *AUROC*, derived from receiver operating characteristic curves, evaluates model performance across varying thresholds and is particularly valuable for assessing the model's robustness and behavior under different levels of detection sensitivity.

8. Important Issues of LLM-Generated Text Detection

This section discusses the primary issues and limitations of contemporary SOTA techniques for detecting text generated by LLMs. It is important to note that no technique

has been deemed infallible. The issues highlighted here may pertain to one or multiple classes of detectors.

8.1 Out-of-Distribution Challenges

Out-of-distribution challenges significantly impede the efficacy of current techniques dedicated to the detection of LLM-generated text. This section elucidates the constraints of these detectors to variations in domains and languages.

Multi-domain. The dilemma of multi-domain application is a ubiquitous challenge inherent to numerous NLP tasks. Studies conducted by Antoun et al. (2023) and Li et al. (2023c) underscored considerable limitations in the performance of sophisticated detectors, including but not limited to DetectGPT (Mitchell et al. 2023), GLTR (Gehrmann, Strobelt, and Rush 2019), and fine-tuned RoBERTa models, when applied to a new domain. These detectors exhibit substantial performance degradation when confronted with out-of-distribution data prevalent in real-world scenarios, with the efficacy of some classifiers marginally surpassing that of random classification. This disparity between high reported performance and actual reliability underlines the need for critical evaluation and enhancement of existing methods.

Multilingual. The issue of multilingual application introduces a set of complex challenges that hinder the global applicability of existing detector research. Predominantly, contemporary detectors designed for LLM-generated text primarily target monolingual applications, often neglecting to evaluate and optimize performance across multiple languages. Wang et al. (2023b) and Chaka (2023) have noted significant gaps in control and consistency when detecting multilingual LLM-generated text, despite some language migration capabilities. These multilingual challenges are pivotal for improving the usability and fairness of LLM-generated text detectors. Additionally, the study by Liang et al. (2023a) showed a discernible decline in the performance of SOTA detectors when processing texts authored by non-native English speakers. While effective prompt strategies can mitigate this bias, they also inadvertently allow generated text to bypass detection. Consequently, there is a risk that detectors might inadvertently penalize writers who exhibit non-standard linguistic styles or use limited expressions, thereby introducing issues of discrimination within the detection process.

Cross-LLMs. Another significant out-of-distribution issue in the LLM-generated text detection task is the cross-LLMs challenge. Current white-box detection approaches primarily rely on accessing the source model and comparing features such as Log-likelihood. As a result, white-box methods may underperform when encountering text generated by unknown LLMs. For example, DetectGPT (Mitchell et al. 2023) highlights the vulnerability of white-box methods when faced with unknown models, especially powerful ones like GPT-3.5-Turbo. However, the recent findings from Fast-DetectGPT (Bao et al. 2023) show that statistical comparisons with surrogate models can significantly mitigate this issue. Additionally, identifying the type of the generative model before applying white-box methods could be beneficial. In this regard, the methodologies of Siniff (Li et al. 2023a), SeqXGPT (Wang et al. 2023a), and LLMDet (Wu et al. 2023) may offer useful insights. On the other hand, methods based on neural classifiers, especially those fine-tuned classifiers susceptible to overfitting training data, may struggle to recognize types of LLMs not seen during training. This limitation is evident for newly emerging LLMs, where detectors may fail to effectively identify generated texts

(Pagnoni, Graciarena, and Tsvetkov 2022). For instance, the OpenAI detector[12] (trained on texts generated by GPT-2) struggles to discern texts generated by GPT-3.5-Turbo and GPT-4, achieving an AUROC of only 74.74%, while it performs nearly perfectly on GPT-2 generated texts (Bao et al. 2023). Findings by Sarvazyan et al. (2023) demonstrate that supervised LLM-generated text detectors generalize well across model scales but face challenges in generalizing across different model families. Enhancing the cross-LLM robustness of neural classifiers is thus essential for the practical deployment of detectors. Nonetheless, classifiers fine-tuned on RoBERTa still possess strong transfer capabilities, and with additional fine-tuning on just a few hundred samples, detectors can effectively generalize to texts generated by other models. Therefore, incorporating LLM-generated text from various sources into the training data could substantially improve the cross-LLMs' robustness of detectors in real-world applications, even with a small sample size.

8.2 Potential Attacks

Potential attacks, defined as deliberate manipulations of text or the generative models used to produce it, aim to evade or confuse detection systems. These attacks significantly contribute to the ongoing unreliability of current LLM-generated text detectors. In this section, we discuss these attacks to encourage researchers to focus on developing more comprehensive defensive measures.

Paraphrase Attacks. Paraphrasing attacks are one of the most effective strategies, capable of undermining detectors using watermarking technology, fine-tuned supervised detectors, and zero-shot detectors (Sadasivan et al. 2023; Orenstrakh et al. 2023). The underlying principle involves applying a lightweight paraphrase model on LLMs' outputs and changing the distribution of lexical and syntactic features of the text by paraphrasing, thereby confusing the detector. Sadasivan et al. (2023) reported on Parrot (Damodaran 2021), a T5-based paraphrase model, and DIPPER (Krishna et al. 2023), an 11B paraphrasing model that allows for tuning paraphrase diversity and the degree of content reordering that attacks the overall superiority of existing detection methods. While retrieval-based defenses have shown promise against paraphrasing attacks (Krishna et al. 2023), these defenses require ongoing maintenance by language model API providers and remain vulnerable to recursive paraphrasing attacks (Sadasivan et al. 2023).

Adversarial Attacks. Normal LLM-generated texts are highly identifiable, yet adversarial perturbations, such as substitution, can effectively reduce the accuracy of detectors (Peng et al. 2024). We summarize attacks that process on textual features as adversarial attacks, including cutoff (cropping a portion of the feature or input) (Shen et al. 2020), shuffle (randomly disrupting the word order of the input) (Lee et al. 2020), mutation (character and word mutation) (Liang, Guerrero, and Alsmadi 2023), word swapping (substituting other suitable words given the context) (Shi and Huang 2020; Ren et al. 2019; Crothers et al. 2022), and misspelling (Gao et al. 2018). There are also adversarial attack frameworks such as TextAttack (Morris et al. 2020), which systematically construct attacks using four components: an objective function, a set of constraints, a transformation, and a search method. Shi et al. (2023) and He et al. (2023b) reported on

12 `openai-community/roberta-large-openai-detector`.

the effectiveness of the permutation approach on attack detectors. Specifically, Shi et al. (2023) replaced words with synonyms based on context, which forms an effective attack on the fine-tuned classifier, watermarking (Kirchenbauer et al. 2023a), and DetectGPT (Mitchell et al. 2023), reducing detector performance by more than 18%, 10%, and 25%, respectively. He et al. (2023b) used probability-weighted word saliency (Ren et al. 2019) to generate adversarial examples, which further maintains semantic similarity.

Stiff and Johansson (2022) utilized the DeepWordBug (Gao et al. 2018) adversarial attack algorithm to introduce character-level perturbations to generated texts, including adjacent character swaps, character substitutions, deletions, and insertions, which resulted in more than a halving of the performance of the OpenAI large detector.[13] Wolff (2020) presented two types of black-box attacks against these detectors: random substitutions of characters with visually similar homoglyphs and the intentional misspelling of words. These attacks drastically reduced the recall rate of popular neural text detectors from 97.44% to 0.26% and 22.68%, respectively. Moreover, Bhat and Parthasarathy (2020) showed that detectors are more sensitive to syntactic perturbations, including breaking longer sentences, removing definite articles, using semantic-preserving rule conversions (such as changing "that's" to "that is"), and reformatting paragraphs of machine-generated text.

Although existing detection methods are highly sensitive to adversarial attacks, different types of detectors exhibit varying degrees of resilience to such attacks. Antoun et al. (2023) reported that supervised approaches are effective defensive measures against these attacks: Training on adversarial samples can significantly improve a detector's ability to recognize texts that have been manipulated by such attacks. Additionally, Kulkarni et al. (2023) explored the impact of semantic perturbations on the Grover detector, finding that synonym substitution, fake-fake replacement, insertion instead of substitution, and changes in the position of substitution had no effect on Grover's detection capabilities. However, adversarial embedding techniques can effectively deceive Grover into classifying false articles as genuine. The attack degrades the performance of the fine-tuning classifier significantly, even though the distributional features of the attack can be learned by the fine-tuning classifier to form a strong defense.

Prompt Attacks. Prompt attacks pose a significant challenge for current LLM-generated text detection techniques. The quality of LLM-generated text is associated with the complexity of the prompts used to instruct LLMs to generate text. As the model and corpus size increase, LLMs emerge with excellent ICL capabilities for more complex text generation capabilities. Numerous efficient prompting methods have been developed, including few-shot prompt (Brown et al. 2020), combining prompt (Zhao et al. 2021), Chain of Thought (CoT) (Wei et al. 2022), and zero-shot CoT (Kojima et al. 2022), etc., which significantly enhance the quality and capabilities of LLMs. Existing research on LLM-generated text detectors primarily utilize datasets created with simple direct prompts. For instance, the study by Guo et al. (2023) demonstrates that detectors might struggle to identify text generated with complex prompts. Liu et al. (2023e) reported a noticeable decrease in the detection ability of a detector using a fine-tuned language model when faced with varied prompts, which indicates that the use of different prompts results in large differences in the detection performance of existing detectors (Koike, Kaneko, and Okazaki 2023a).

13 `openai-community/roberta-large-openai-detector`.

The Substitution-based Contextual Example Optimization method, proposed by Lu et al. (2023), uses sophisticated prompts to bypass the defenses of current detection systems. This leads to a substantial reduction in the area under the curve, averaging a decrease of 0.54, and achieves a higher success rate with better text quality compared to paraphrase attacks. It is worth mentioning that both paraphrase attacks and adversarial attacks mentioned above could be executed through careful prompt design (Shi et al. 2023; Koike, Kaneko, and Okazaki 2023b). With ongoing research in prompt engineering, the risk posed by prompt attacks is expected to escalate further. This underscores the need for developing more robust detection methods that can effectively counteract such evolving threats.

Training Threat Models. Further training of language models has been preliminarily proven to effectively attack existing detectors. Nicks et al. (2023) used the "humanity" scores of various open source and commercial detectors as a reward function for reinforcement learning, enabling fine-tuning of language models to confound existing detectors. Without significantly altering the model, further fine-tuning of Llama-2-7B can reduce the AUROC of the OpenAI RoBERTa-Large detector from 0.84 AUROC to 0.62 AUROC in a short training period. A similar idea is demonstrated in Schneider et al. (2023): Using reinforcement learning to refine generative models can successfully circumvent BERT-based classifiers with detection accuracy as low as 0.15 AUROC, even when using linguistic features as a reward function. Kumarage et al. (2023b) propose a universal evasion framework named EScaPe to guide PLMs in generating "human-like text" that may mislead detectors. Through evasive soft prompt learning and transfer, the performance of DetectGPT and OpenAI Detector can be effectively reduced by up to 40% AUROC. Additionally, the results from Henrique, Kucharavy, and Guerraoui (2023) reveal another potential vulnerability of detectors. If a generative model can access the human-written text used to train the detector and use them for fine-tuning, it is impossible to use detector for text detection on this generative model. This indicates that LLMs trained on a more human-written corpus will be more robust against existing detectors, and training against a specific detector can provide the LLMs with a sharp spear to breach its defenses.

8.3 Real-World Data Issues

Detection for Not Purely LLM-Generated Text. In practice, there are many texts that are not purely generated by LLMs, and they may even contain a mix of human-written text. Specifically, this can be categorized as either data-mixed text or human-edited text. Data-mixed text refers to the sentence or paragraph level mixture of human-written text and LLM-generated text. For instance, in a document, some sentences may be generated by LLMs, while others are written by humans. In such cases, identifying the category of the document becomes challenging. Data-mixed text necessitates more fine-grained detection methods, such as sentence-level detection, to effectively address this challenge. However, current LLM-generated text detectors struggle to perform effectively with short texts. Recent research, such as that by Wang et al. (2023a), indicates that sentence-level detection appears to be feasible. Encouragingly, studies have started addressing this issue. Zeng et al. (2023) proposed a two-step method to effectively identify a mix of human-written and LLM-generated text. This method first uses contrastive learning to distinguish between content generated by LLMs and human-written content. It then calculates the similarity between adjacent prototypes, assuming that a boundary exists between the least similar adjacent prototypes.

Another issue that has not been fully discussed is the human-edited text. For example, after applying LLM to generate a text, humans often edit and modify certain words or passages. The detection of such text poses a significant challenge and is an issue we must confront, as it is prevalent in real-world applications. Therefore, there is an urgent need to organize relevant datasets and define tasks to address this issue. One potential approach for tackling this problem is informed by experimental results from paraphrasing and adversarial perturbation attacks. These methods effectively simulate how individuals might use LLMs to refine text or make word substitutions. Current mainstream detectors, however, tend to degrade in performance when dealing with paraphrased text (Wolff 2020), although certain black-box detectors display relatively good robustness. Another potential solution could involve breaking down the detection task to the word level, but as of now, there is no research directly addressing this.

Issues of LLM-Assisted Writing. When discussing the use of LLMs in writing assistance and their impact on human writing, it is essential to consider meticulously how such texts are annotated and managed. It is particularly important to distinguish between texts that are entirely generated by LLMs and those that are co-created by humans with LLM assistance. Texts that are fully generated by an LLM should be marked as "LLM-generated" to enhance transparency and meet regulatory requirements. Moreover, minor edits by humans, such as revising individual words or slightly modifying segments, should not change the designation of the text since the core content remains LLM-generated and subject to regulation due to possible quality inconsistencies that require strict oversight (Wang et al. 2024).

In scenarios where LLMs provide grammar checking, polishing, and editing suggestions during the creative process, the text should not be labeled as entirely LLM-generated, because the primary substantial contribution is from humans. Such texts should not be under strict regulation. It has been suggested that these texts could be described as "AI-revised Human-Written Text" by Gao et al. (2024), a label that accurately reflects the collaborative nature of human and computer in the creative process and respects the human contribution to creativity. Regulation of LLM-assisted human creations should be more lenient to avoid suppressing creative freedom, particularly in the realms of literature and academia. Despite the challenges of regulation and labeling, LLMs have positively influenced human writing by enhancing writing efficiency, improving language quality, and fostering creativity (Kasneci et al. 2023). Appropriate labeling and moderate regulation can harness the benefits of LLMs while mitigating potential risks, ensuring both the quality of content and creative freedom. This balanced approach will optimize the positive impacts of LLMs while safeguarding the interests of both users and creators.

Data Ambiguity. Data ambiguity remains a challenge in the field of LLM-generated text detection, which closely ties to the inherent mechanisms of the detection technology itself. The pervasive deployment of LLMs across various domains exacerbates this issue, rendering it increasingly challenging to discern whether training data comprises human-written or LLM-generated text. Utilizing LLM-generated text as training data under the misapprehension that it is human-written inadvertently instigates a detrimental cycle. Within this cycle, detectors, consequently trained, demonstrate diminished efficacy in distinguishing between human-written and LLM-generated text, thereby undermining the foundational premises of detector research. It is imperative to acknowledge that this quandary poses a significant, pervasive threat to all facets of detection research. However, to our knowledge, no existing studies formally address

this concern. Alemohammad et al. (2023) further highlighted an additional potential risk, suggesting that data ambiguity might lead to the recycling of LLM-generated content in the training processes of subsequent models. This scenario could adversely impact the text generation quality of these emergent LLMs, thereby destabilizing the research landscape dedicated to the detection of LLM-generated text.

8.4 Impact of Model Size on Detectors

Many researchers are concerned about the impact of the model size on detectors, which can be analyzed from two perspectives: one is the size of the generative model, and the other is the size of the supervised classifiers. The size of the generative model is closely related to the quality of the generated text. Generally, texts generated by smaller-sized models are easier to recognize, while those generated by larger models pose a greater challenge for detection. Another concern is how the texts generated by models of different sizes affect the detectors when used as training samples. Pu et al. (2023b) report that detectors trained with data generated by medium-sized LLMs can generalize to larger versions without any samples, while training samples generated by overly large or small models may reduce the generalization ability of the detectors. Antoun, Sagot, and Seddah (2023) further explores the apparent negative correlation between classifier effectiveness and the size of the generative model. Their findings indicate that text generated by larger LLMs is more difficult to detect, especially when the classifier is trained on data generated by smaller LLMs. Aligning the distribution of the generative models for the training and test sets can improve the performance of the detectors. From the perspective of the size of the supervised classifiers, the detection capability of the detectors is directly proportional to the size of the fine-tuned LMs (Guo et al. 2023). However, recent findings suggest that while larger detectors perform better on test sets with the same distribution as the training set, their generalization ability is somewhat diminished.

8.5 Lack of Effective Evaluation Framework

Comprehensiveness of Evaluation Frameworks. To gain users' trust, a reliable detector must undergo a multifaceted assessment. The current benchmarks are somewhat limited, providing only superficial challenges and thereby not facilitating a holistic evaluation of detectors. We highlight five crucial dimensions that are essential for the development of more robust benchmarks for LLM-generated text detection tasks. These dimensions include the incorporation of multiple types of attacks, diverse domains, varied tasks, a spectrum of models, and the inclusion of multiple languages.

Multiple types of attack are instrumental in ascertaining the efficacy of detection methodologies. In practical environments, LLM-generated text detectors often encounter texts that are generated using a wide range of attack mechanisms, which differ from texts generated through simple prompts. For instance, the *prompt attack* elucidated in Section 8.2 impels the generative model to produce superior-quality text, leveraging intricate and sophisticated prompts. Integrating such texts into existing datasets is essential, as echoed in the limitations outlined by Guo et al. (2023).

Multi-domain and multi-task configurations are crucial in assessing a detector's performance across diverse domains and LLM applications. These dimensions bear significant implications for a detector's robustness, usability, and credibility. In academic contexts, for instance, an effective detector should perform consistently across all disciplines. In everyday scenarios, it should adeptly identify LLM-generated text spanning

academic compositions, news articles, and Q&A sessions. It is important to highlight that many existing studies have explicitly considered this in their experimental setups (Wang et al. 2023b). We encourage the construction and promotion of more high-quality, multi-domain, and task-specific LLM-generated datasets for future researchers to adopt. The ongoing research momentum in LLMs has ushered in formidable counterparts like LLaMa (Touvron et al. 2023), PaLM (Chowdhery et al. 2022), and Claude-2,[14] rivaling ChatGPT's prowess. As the spotlight remains on ChatGPT, it is essential to concurrently address potential risks emanating from other emerging LLMs.

Multilingual considerations demand increased attention. Specifically, existing benchmarks are mainly developed for English datasets, with only a few like MULTITuDE (Macko et al. 2023) and M4 (Wang et al. 2023b) covering multiple languages, including Arabic, Catalan, Chinese, Czech, Dutch, English, German, Portuguese, Russian, Spanish, Indonesian, Bulgarian, and Ukrainian. However, there is still a significant lack of datasets for other language resources, especially low-resource languages. Therefore, we strongly encourage researchers to spearhead the creation of multilingual datasets to facilitate the evaluation of text detectors generated by LLMs across different languages. We strongly encourage researchers to spearhead the creation of multilingual datasets to facilitate the evaluation of text detectors generated by LLMs across different languages. The utilization of pre-trained models may uncover instances where certain detectors struggle with underrepresented languages, while LLMs could exhibit more noticeable inconsistencies. This dimension presents a rich avenue for exploration and discourse.

Request for Objective and Fair Benchmark. A prevalent issue in LLM-generated text detection research is the discrepancy between claimed detector performance and practical results. While many studies report impressive and robust detector capabilities, these methods often underperform on test sets created by other researchers. This variance arises from using different strategies to construct their test sets including the parameters used to generate the test set, the computational environment, text distribution, and text processing strategies, including truncation, which can all influence the effectiveness of detectors. Due to these factors' complex nature, the reproducibility of evaluation results is often compromised, even when researchers adhere to identical dataset production protocols. As discussed in Section 4, the limitations of existing benchmarks necessitate the creation of high-quality and comprehensive evaluation frameworks. We strongly encourage future research to adopt these frameworks to ensure consistency in testing standards. Additionally, we urge researchers focusing on specific challenges to openly share their test sets, emphasizing the adaptability of current evaluation frameworks to incorporate diverse datasets. Establishing objective and fair benchmarks is vital to advancing LLM-generated text detection research and moving beyond isolated, siloed efforts.

Temporal of Current Benchmark. It is evident that certain contemporary studies persistently rely on seminal but somewhat outdated benchmark datasets, which had significantly shaped prior GPT-generated text and fake news detection endeavors. However, these datasets predominantly originate from backward LLMs, implying that validated methodologies might not invariably align with current real-world dynamics. We emphasize the significance of utilizing datasets formulated with advanced and powerful

14 https://www.anthropic.com/index/claude-2.

LLMs, while also urging benchmark dataset developers to regularly update their contributions to reflect the rapid evolution of the field.

9. Future Research Directions

This section explores potential avenues for future research to develop more efficient and practically developed detectors for LLM-generated text.

9.1 Building Robust Detectors with Attacks

The attack methods discussed in Section 8.2 encompass Paraphrase Attacks (Sadasivan et al. 2023), Adversarial Attacks (He et al. 2023b), and Prompt Attacks (Lu et al. 2023). These methods underscore the primary challenges impeding the utility of current detectors. While recent research, such as that of Yang, Jiang, and Li (2023), addresses robustness against specific attacks, it often neglects potential threats posed by other attack forms. Consequently, it is imperative to develop and validate diverse attack types, thereby gaining insights into vulnerabilities inherent to LLM-generated text detectors. Additionally, we further advocate for the establishment of comprehensive benchmarks to assess existing detection strategies. Although some studies (He et al. 2023b; Wang et al. 2023b) purport to provide such benchmarks, the scope and diversity of the validated attacks remain limited.

9.2 Enhancing the Efficacy of Zero-Shot Detectors

Zero-shot methods are recognized for their notable stability as detectors (Deng et al. 2023). These approaches offer enhanced controllability and interpretability for users (Mitrović, Andreoletti, and Ayoub 2023). Recent research (Giorgi et al. 2023; Liao et al. 2023) has highlighted distinct disparities between LLM-generated text and human-written text, revealing a tangible and discernible gap between the two. This revelation has invigorated research in the domain of LLM-generated text detection. We advocate for a proliferation of these studies that delve into the nuanced distinctions between LLM-generated texts and human-written text, spanning from low-dimensional to high-dimensional features. Unearthing metrics that more accurately distinguish the two can bolster the evolution of automatic detectors and furnish more compelling justifications for decision-making processes. We have observed that the latest emerging black-box zero-shot methods (Yang et al. 2023b; Mao et al. 2024; Zhu et al. 2023; Quidwai, Li, and Dube 2023; Guo and Yu 2023) demonstrate enhanced stability and application potential compared to white-box based zero-shot methods by extracting discriminative metrics that are independent of white-box models. These methods do not rely on an understanding of the model's internal workings, thereby offering broader applicability across various models and environments.

9.3 Optimizing Detectors for Low-Resource Environments

Many contemporary detection techniques tend to overlook the challenges faced by resource-constrained settings, often neglecting the need for resources in developing the detector. The relative efficacy of various detectors across different data volume settings remains inadequately explored. Concurrently, determining the minimal resource prerequisites for different detection methods to yield satisfactory results is imperative. Beyond examining the model's adaptability across distinct domains (Rodriguez et al.

2022) and languages (Wang et al. 2023b), we advocate for investigating the defensive adaptability against varied attack strategies. Such exploration can guide users in selecting the most beneficial approach to establish a dependable detector under resource constraints.

9.4 Detection for Not Purely LLM-Generated Text

As highlighted in Section 8.3, a significant challenge encountered in real-world scenarios is the detection of text that is not purely produced by LLMs. We examined this issue by separately discussing texts that are a mixture of data sources and those that have been edited by humans, and review the latest related work and propose potential solutions, which are still pending verification. We emphasize that organizing relevant datasets and defining tasks to address this issue is an urgent need at present, because fundamentally, this type of text may be the most commonly encountered in detector applications.

9.5 Constructing Detectors Amidst Data Ambiguity

Verifying the authenticity of the training data poses a significant challenge. When aggregating textual data from sources such as blogs and web comments, there is a potential risk of inadvertently including a substantial amount of LLM-generated text. This incorporation can fundamentally compromise the integrity of detector research, perpetuating a detrimental feedback loop. We urge forthcoming detection studies to prioritize the authenticity assessment of real-world data, anticipating this as a pressing challenge in the future.

9.6 Developing an Effective Evaluation Framework Aligned with Real-World Settings

In Section 8.5, we discussed the objective differences between evaluation environments and real-world settings, which limit the effectiveness of existing detectors when applied in practice. On one hand, biases in the construction of test sets can be found in many works because they often favor the detectors built by their creators. On the contrary, current benchmarks frequently reflect idealized scenarios far removed from real-world applications. We call on researchers to develop a fair and effective evaluation framework closely linked to the practical needs of LLM-generated detection tasks; for instance, considering the necessity of the application domain, the black-box nature of LLM-generated texts, and the various attacks and post-editing strategies that texts may encounter. We believe such an evaluation framework will promote the research and development of detectors that are more practical and aligned with real-world scenarios.

9.7 Constructing Detectors with Misinformation Discrimination Capabilities

Contemporary detection methodologies have largely overlooked the capacity to identify misinformation. Existing detectors primarily emphasize the distribution of features within the text generated by LLMs but often overlooked their potential for factual verification. A proficient detector should possess the capability to discern the veracity or falsity of factual claims presented in text. In the initial stages of generative modeling's emergence, when it had yet to pose significant societal challenges, the emphasis was on assessing the truth or falsity of the content in LLM-generated text, with less regard for its

source (Schuster et al. 2020). Constructing detectors with misinformation discrimination capabilities can aid in more accurately attributing the source of text, rather than relying solely on distributional features, and subsequently contribute to mitigating the proliferation of misinformation. Recent studies (Gao et al. 2023; Chern et al. 2023) highlight the potential of LLMs to detect factual content in texts. We recommend bolstering such endeavors through integration with external knowledge bases (Asai et al. 2023) or search engines (Liang et al. 2023b), which could significantly enhance their ability to verify claims and improve reliability in practical applications.

10. Conclusion

With the rapid advancements and application of LLMs, the presence of LLM-generated text in our daily lives has transitioned from expectation to pervasive reality. LLM-generated text detectors play a pivotal role in distinguishing between human-written and LLM-generated text, serving as a crucial defense against the misuse of LLMs for generating deceptive news, engaging in scams, or exacerbating issues such as educational inequality. This survey provides a comprehensive overview of the task of LLM-generated text detection, examines the underlying mechanisms enhancing LLM capabilities, and highlights the increasing need for robust detection methodologies. We also list popular or promising datasets that point out the challenges and requirements associated with existing detectors. In addition, we shed light on the critical limitations of contemporary detectors, including issues related to out-of-distribution data, potential attacks, real-world data issues, and the lack of an effective evaluation framework, to direct researchers' attention to the focal points of the field, thereby sparking innovative ideas and approaches. Finally, we propose potential future research directions that are poised to guide the development of more powerful and effective detection systems, ensuring their alignment with real-world applications.

Acknowledgments

This work was supported in part by the Major Program of the State Commission of Science Technology of China (grant no. 2020AAA0106701), the Science and Technology Development Fund of Macau SAR (grant no. 0007/2024/AKP), the Science and Technology Development Fund of Macau SAR (grant no. FDCT/0070/2022/AMJ, China Strategic Scientific and Technological Innovation Cooperation Project grant no. 2022YFE0204900), the Science and Technology Development Fund of Macau SAR (grant no. FDCT/060/2022/AFJ, National Natural Science Foundation of China grant no. 62261160648), the UM and UMDF (grant nos. MYRG-GRG2023-00006-FST-UMDF, MYRG-GRG2024-00165-FST-UMDF), and the National Natural Science Foundation of China (grant no. 62266013). This work was performed in part at SICC which is supported by SKL-IOTSC, and HPCC supported by ICTO of the University of Macau. We would like to thank the anonymous reviewers for their insightful comments.

References

Abdelnabi, Sahar and Mario Fritz. 2021. Adversarial Watermarking Transformer: Towards tracing text provenance with data hiding. In *42nd IEEE Symposium on Security and Privacy, SP 2021*, pages 121–140. https://doi.org/10.1109/SP40001.2021.00083

Aich, Ankit, Souvik Bhattacharya, and Natalie Parde. 2022. Demystifying neural fake news via linguistic feature-based interpretation. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*, pages 6586–6599.

Alemohammad, Sina, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. 2023. Self-consuming generative models go MAD. *CoRR*,

abs/2307.01850. https://doi.org/10.52591/lxai202312101

Anthropic. 2023. Model card and evaluations for Claude models. https://paperswithcode.com/paper/model-card-and-evaluations-for-claude-models

Antoun, Wissam, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah. 2023. Towards a robust detection of language model generated text: Is ChatGPT that easy to detect? *CoRR*, abs/2306.05871. https://doi.org/10.48550/arXiv.2306.05871

Antoun, Wissam, Benoît Sagot, and Djamé Seddah. 2023. From text to source: Results in detecting large language model-generated content. *CoRR*, abs/2309.13322. https://doi.org/10.48550/ARXIV.2309.13322

Arase, Yuki and Ming Zhou. 2013. Machine translation detection from monolingual web-text. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1597–1607.

Asai, Akari, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46. https://doi.org/10.18653/v1/2023.acl-tutorials.6

Asghar, Nabiha. 2016. Yelp dataset challenge: Review rating prediction. *ArXiv preprint*, abs/1605.05362.

Baayen, R. Harald. 2001. *Word Frequency Distributions*, volume 18. Springer Science & Business Media. https://doi.org/10.1007/978-94-010-0844-0

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010*.

Bakhtin, Anton, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? Learning to discriminate machine from human generated text. *CoRR*, abs/1906.03351.

Bao, Guangsheng, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*, abs/2310.05130.

Basu, Sourya, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *9th International Conference on Learning Representations, ICLR 2021*, OpenReview.net.

Bender, Walter, Daniel Gruhl, Norishige Morimoto, and Anthony Lu. 1996. Techniques for data hiding. *IBM Systems Journal*, 35(3/4):313–336. https://doi.org/10.1147/SJ.353.0313

Beresneva, Daria. 2016. Computer-generated text detection using machine learning: A systematic review. In *Natural Language Processing and Information Systems: 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016*, pages 421–426. https://doi.org/10.1007/978-3-319-41754-7_43

Besta, Maciej, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *ArXiv preprint*, abs/2308.09687.

Bhat, Meghana Moorthy and Srinivasan Parthasarathy. 2020. How effectively can machines defend against machine-generated fake news? An empirical study. In *Proceedings of the First Workshop on Insights from Negative Results in NLP, Insights 2020, Online, November 19, 2020*, pages 48–53. https://doi.org/10.18653/v1/2020.insights-1.7

Bhattacharjee, Amrita, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for ai-generated text detection. *CoRR*, abs/2309.03992. https://doi.org/10.48550/arXiv.2309.03992

Bhattacharjee, Amrita and Huan Liu. 2023. Fighting fire with fire: Can ChatGPT detect AI-generated text? *ArXiv preprint*, abs/2308.01284.

Blanchard, Daniel, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A corpus of non-native English. *ETS Research Report Series*, 2013(2):i–15. https://doi.org/10.1002/j.2333-8504.2013.tb02331.x

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on*

Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual.
Cardenuto, João Phillipe, Jing Yang, Rafael Padilha, Renjie Wan, Daniel Moreira, Haoliang Li, Shiqi Wang, Fernanda A. Andaló, Sébastien Marcel, and Anderson Rocha. 2023. The age of synthetic realities: Challenges and opportunities. *CoRR*, abs/2306.11503. https://doi.org/10.48550/arXiv.2306.11503, https://doi.org/10.1561/116.00000138
Chaka, Chaka. 2023. Detecting AI content in responses generated by ChatGPT, YouChat, and Chatsonic: The case of five AI content detection tools. *Journal of Applied Learning and Teaching*, 6(2). https://doi.org/10.37074/jalt.2023.6.2.12
Chakraborty, Megha, S. M. Towhidul Islam Tonmoy, S. M. Mehedi Zaman, Shreya Gautam, Tanay Kumar, Krish Sharma, Niyar R. Barman, Chandan Gupta, Vinija Jain, Aman Chadha, Amit P. Sheth, and Amitava Das. 2023a. Counter Turing Test (CT2): AI-generated text detection is not as easy as you may think - Introducing AI Detectability Index (ADI). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 2206–2239. https://doi.org/10.18653/v1/2023.emnlp-main.136
Chakraborty, Souradip, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023b. On the possibilities of AI-generated text detection. *CoRR*, abs/2304.04736. https://doi.org/10.48550/ARXIV.2304.04736
Chen, Qianben, Richong Zhang, Yaowei Zheng, and Yongyi Mao. 2022. Dual contrastive learning: Text classification via label-aware data augmentation. *ArXiv preprint*, abs/2201.08702.
Chen, Yutian, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023a. Token prediction as implicit classification to identify LLM-generated text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 13112–13120. https://doi.org/10.18653/v1/2023.emnlp-main.810
Chen, Yutian, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Ramakrishnan. 2023b. GPT-sentinel: Distinguishing human and ChatGPT generated content. *ArXiv preprint*, abs/2305.07969.
Chern, I., Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. FacTool: Factuality detection in generative AI–A tool augmented framework for multi-task and multi-domain scenarios. *ArXiv preprint*, abs/2307.13528.
Chotikakamthorn, Nopporn. 1998. Electronic document data hiding technique using inter-character space. In *IEEE. APCCAS 1998. 1998 IEEE Asia-Pacific Conference on Circuits and Systems. Microelectronics and Integrating Systems. Proceedings (Cat. No. 98EX242)*, pages 419–422. https://doi.org/10.1109/APCCAS.1998.743799
Chowdhery, Aakanksha, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling language modeling with pathways. *ArXiv preprint*, abs/2204.02311.
Christian, Jon. 2023. CNET secretly used AI on articles that didn't disclose that fact, staff say. *Futurism, January*. https://futurism.com/cnet-ai-articles-label
Clark, Elizabeth, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296. https://doi.org/10.18653/v1/2021.acl-long.565
Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747
Corizzo, Roberto and Sebastian Leal-Arenas. 2023. A deep fusion model for human vs. Machine-generated essay classification. In *International Joint Conference on Neural Networks, IJCNN 2023*, pages 1–10. https://doi.org/10.1109/IJCNN54540.2023.10191322
Corston-Oliver, Simon, Michael Gamon, and Chris Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proceedings of the*

39th Annual Meeting of the Association for Computational Linguistics, pages 148–155. https://doi.org/10.3115/1073012.1073032

Cowap, Alan, Yvette Graham, and Jennifer Foster. 2023. Do stochastic parrots have feelings too? Improving neural detection of synthetic text via emotion recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9928–9946. https://doi.org/10.18653/v1/2023.findings-emnlp.665

Crothers, Evan, Nathalie Japkowicz, and Herna L. Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002. https://doi.org/10.1109/ACCESS.2023.3294090

Crothers, Evan, Nathalie Japkowicz, Herna L. Viktor, and Paula Branco. 2022. Adversarial robustness of neural-statistical features in detection of generative transformers. In *International Joint Conference on Neural Networks, IJCNN 2022*, pages 1–8. https://doi.org/10.1109/IJCNN55064.2022.9892269

Cui, Jiaxi, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. ChatLaw: Open-source legal large language model with integrated external knowledge bases. *ArXiv preprint*, abs/2306.16092.

Dai, Damai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? Language models implicitly perform gradient descent as meta-optimizers. pages In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*. https://doi.org/10.18653/v1/2023.findings-acl.247

Dalianis, Hercules. 2018. Evaluation metrics and evaluation. *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer, pages 45–53. https://doi.org/10.1007/978-3-319-78503-5_6

Damodaran, Prithiviraj. 2021. Parrot: Paraphrase generation for NLU. https://github.com/PrithivirajDamodaran/Parrot_Paraphraser

Deng, Zhijie, Hongcheng Gao, Yibo Miao, and Hao Zhang. 2023. Efficient detection of LLM-generated texts with a Bayesian surrogate model. *ArXiv preprint*, abs/2305.16617.

Desaire, Heather, Aleesa E. Chua, Madeline Isom, Romana Jarosova, and David Hua. 2023. ChatGPT or academic scientist? Distinguishing authorship with over 99% accuracy using off-the-shelf machine learning tools. *CoRR*, abs/2303.16352. https://doi.org/10.1016/j.xcrp.2023.101426, PubMed: 37426542

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. https://doi.org/10.18653/v1/n19-1423

Dhaini, Mahdi, Wessel Poelman, and Ege Erdogan. 2023. Detecting ChatGPT: A survey of the state of detecting ChatGPT-generated text. *CoRR*, abs/2309.07689. https://doi.org/10.48550/ARXIV.2309.07689

Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2023. A survey for in-context learning. *ArXiv preprint*, abs/2301.00234.

Dou, Yao, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A. Smith, and Yejin Choi. 2022. Is GPT-3 text indistinguishable from human text? Scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7250–7274. https://doi.org/10.18653/v1/2022.acl-long.501

Dugan, Liam, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. RoFT: A tool for evaluating human detection of machine-generated text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 189–196. https://doi.org/10.18653/v1/2020.emnlp-demos.25

Dugan, Liam, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023*, pages 12763–12771. https://doi.org/10.1609/aaai.v37i11.26501

Epstein, Ziv, Aaron Hertzmann; Investigators of Human Creativity; Memo Akten, Hany Farid, Jessica Fjeld, Morgan R. Frank, Matthew Groh, Laura Herman, Neil Leach, et al. 2023. Art and the science of generative AI. *Science*, 380(6650):1110–1111. https://doi.org/10.1126/science.adh4451, PubMed: 37319193

Fagni, Tiziano, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. TweepFake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415. https://doi.org/10.1371/journal.pone.0251415 PubMed: 33984021

Fan, Angela, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567. https://doi.org/10.18653/v1/P19-1346

Fan, Angela, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898. https://doi.org/10.18653/v1/P18-1082

Fellbaum, Christiane. 1998. *WordNet: An electronic lexical database*. MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

Gade, Krishna, Sahin Geyik, Krishnaram Kenthapadi, Varun Mithal, and Ankur Taly. 2020. Explainable AI in industry: Practical challenges and lessons learned. In *Companion Proceedings of the Web Conference 2020*, pages 303–304. https://doi.org/10.1145/3366424.3383110

Gallé, Matthias, Jos Rozen, Germán Kruszewski, and Hady Elsahar. 2021. Unsupervised and distributional detection of machine-generated text. *CoRR*, abs/2111.02878.

Gambini, Margherita, Tiziano Fagni, Fabrizio Falchi, and Maurizio Tesconi. 2022. On pushing DeepFake tweet detection capabilities to the limits. In *Proceedings of the 14th ACM Web Science Conference 2022*, pages 154–163. https://doi.org/10.1145/3501247.3531560

Gao, Chujie, Dongping Chen, Qihui Zhang, Yue Huang, Yao Wan, and Lichao Sun. 2024. LLM-as-a-coauthor: The challenges of detecting LLM-human mixcase. *CoRR*, abs/2401.05952. https://doi.org/10.48550/ARXIV.2401.05952

Gao, Ji, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. https://doi.org/10.1109/SPW.2018.00016

Gao, Luyu, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508. https://doi.org/10.18653/v1/2023.acl-long.910

Gao, Tianyu, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910. https://doi.org/10.18653/v1/2021.emnlp-main.552

Gehrmann, Sebastian, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116. https://doi.org/10.18653/v1/P19-3019

Ghosal, Soumya Suvra, Souradip Chakraborty, Jonas Geiping, Furong Huang, Dinesh Manocha, and Amrit Bedi. 2023. A survey on the possibilities & impossibilities of AI-generated text detection. *Transactions on Machine Learning Research*.

Giorgi, Salvatore, David M. Markowitz, Nikita Soni, Vasudha Varadarajan, Siddharth Mangalik, and H. Andrew Schwartz. 2023. "I slept like a baby": Using human traits to characterize deceptive ChatGPT and human text. In *Proceedings of the IACT - The 1st International Workshop on Implicit Author Characterization from Texts for Search and Retrieval held in conjunction with the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023)*, volume 3477 of *CEUR Workshop Proceedings*, pages 23–37.

Giorgi, Salvatore, Lyle Ungar, and H. Andrew Schwartz. 2021. Characterizing social spambots by their human traits. In *Findings of the Association for Computational*

Linguistics: ACL-IJCNLP 2021, pages 5148–5158. https://doi.org/10.18653/v1/2021.findings-acl.457

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144. https://doi.org/10.1145/3422622

Gu, Chenxi, Chengsong Huang, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. Watermarking pre-trained language models with backdooring. *ArXiv preprint*, abs/2210.07543.

Guo, Biyang, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. *ArXiv preprint*, abs/2301.07597.

Guo, Mandy, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40B: Multilingual language model dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2440–2452.

Guo, Zhen and Shangdi Yu. 2023. AuthentiGPT: Detecting machine-generated text via black-box language models denoising. *CoRR*, abs/2311.07700. https://doi.org/10.48550/ARXIV.2311.07700

Hamed, Ahmed Abdeen and Xindong Wu. 2023. Improving detection of ChatGPT-generated fake science using real publication text: Introducing xFakeBibs a supervised-learning network algorithm. *CoRR*, abs/2308.11767. https://doi.org/10.48550/ARXIV.2308.11767, https://doi.org/10.21203/rs.3.rs-2851222/v1

Hanley, Hans W. A. and Zakir Durumeric. 2023. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. *CoRR*, abs/2305.09820. https://doi.org/10.48550/ARXIV.2305.09820

Hans, Abhimanyu, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. In *Forty-first International Conference on Machine Learning, ICML 2024*, OpenReview.net.

He, Jianfei, Shichao Sun, Xiaohua Jia, and Wenjie Li. 2023a. Empirical analysis of beam search curse and search errors with model errors in neural machine translation. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 91–101.

He, Xinlei, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023b. MGTBench: Benchmarking machine-generated text detection. *ArXiv preprint*, abs/2303.14822.

He, Zhiwei, Binglin Zhou, Hongkun Hao, Aiwei Liu, Xing Wang, Zhaopeng Tu, Zhuosheng Zhang, and Rui Wang. 2024. Can watermarks survive translation? On the cross-lingual consistency of text watermark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 4115–4129. https://doi.org/10.18653/v1/2024.acl-long.226

Helm, Hayden S., Carey E. Priebe, and Weiwei Yang. 2023. A statistical Turing test for generative models. *CoRR*, abs/2309.08913. https://doi.org/10.48550/ARXIV.2309.08913

Henrique, Da Silva Gameiro, Andrei Kucharavy, and Rachid Guerraoui. 2023. Stochastic parrots looking for stochastic parrots: LLMs are easy to fine-tune and hard to detect with other LLMs. *CoRR*, abs/2304.08968. https://doi.org/10.48550/ARXIV.2304.08968

Hill, Felix, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks principle: Reading children's books with explicit memory representations. In *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*.

Holtzman, Ari, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020*, OpenReview.net.

Horne, Benjamin and Sibel Adali. 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 759–766. https://doi.org/10.1609/icwsm.v11i1.14976

Hou, Abe Bohan, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Vanzz Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. SemStamp: A

semantic watermark with paraphrastic robustness for text generation. *CoRR*, abs/2310.03991. https://doi.org/10.48550/ARXIV.2310.03991

Hu, Xiaomeng, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: Robust AI-text detection via adversarial learning. *ArXiv preprint*, abs/2307.03838.

Ibrahim, Hazem, Fengyuan Liu, Rohail Asim, Balaraju Battu, Sidahmed Benabderrahmane, Bashar Alhafni, Wifag Adnan, Tuka Alhanai, Bedoor K. AlShebli, Riyadh Baghdadi, et al. 2023. Perception, performance, and detectability of conversational artificial intelligence across 32 university courses. *CoRR*, abs/2305.13934. https://doi.org/10.48550/ARXIV.2305.13934, https://doi.org/10.1038/s41598-023-38964-3, PubMed: 37620342

Ippolito, Daphne, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822. https://doi.org/10.18653/v1/2020.acl-main.164

Jain, Ayush, Vishal Singh, Sidharth Ranjan, Rajakrishnan Rajkumar, and Sumeet Agarwal. 2018. Uniform information density effects on syntactic choice in Hindi. In *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, pages 38–48.

Jawahar, Ganesh, Muhammad Abdul-Mageed, and Laks Lakshmanan, V. S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309. International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.208

Ji, Ziwei, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38. https://doi.org/10.1145/3571730

Jiao, Xiaoqi, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

Jin, Qiao, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577. https://doi.org/10.18653/v1/D19-1259

Jin, Zhuoran, Yubo Chen, Dianbo Sui, Chenhao Wang, Zhipeng Xue, and Jun Zhao. 2021. CogIE: An information extraction toolkit for bridging texts and CogNet. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL 2021 - System Demonstrations*, pages 92–98. https://doi.org/10.18653/v1/2021.acl-demo.11

Kalinichenko, Leonid A., Vladimir V. Korenkov, Vladislav P. Shirikov, Alexey N. Sissakian, and Oleg V. Sunturenko. 2003. Digital libraries: Advanced methods and technologies, digital collections. *D-Lib Magazine*, 9(1):1082–9873.

Kang, Dongyeop, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1647–1661. https://doi.org/10.18653/v1/N18-1149

Kasneci, Enkelejda, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274. https://doi.org/10.1016/j.lindif.2023.102274

Kim, Zae Myung, Kwang Hee Lee, Preston Zhu, Vipul Raheja, and Dongyeop Kang. 2024. Threads of subtlety: Detecting machine-generated texts through discourse motifs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), ACL 2024, pages 5449–5474. https://doi.org/10.18653/v1/2024.acl-long.298

Kirchenbauer, John, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084.

Kirchenbauer, John, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023b. On the reliability of watermarks for large language models. *CoRR*, abs/2306.04634. https://doi.org/10.48550/arXiv.2306.04634

Kitchenham, Barbara and Stuart Charters. 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical Report, EBSE Technical Report EBSE-2007-01. pages 1–57. https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf

Kočiský, Tomáš, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328. https://doi.org/10.1162/tacl_a_00023

Koike, Ryuto, Masahiro Kaneko, and Naoaki Okazaki. 2023a. How you prompt matters! Even task-oriented constraints in instructions affect LLM-generated text detection. *CoRR*, abs/2311.08369. https://doi.org/10.48550/ARXIV.2311.08369, https://doi.org/10.18653/v1/2024.findings-emnlp.841

Koike, Ryuto, Masahiro Kaneko, and Naoaki Okazaki. 2023b. OUTFOX: LLM-generated essay detection through in-context learning with adversarially generated examples. *ArXiv preprint*, abs/2307.11729.

Kojima, Takeshi, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

Krishna, Kalpesh, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *ArXiv preprint*, abs/2303.13408.

Kuditipudi, Rohith, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust distortion-free watermarks for language models. *CoRR*, abs/2307.15593. https://doi.org/10.48550/ARXIV.2307.15593

Kuditipudi, Rohith, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2024. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*.

Kulkarni, Pranav, Ziqing Ji, Yan Xu, Marko Neskovic, and Kevin Nolan. 2023. Exploring semantic perturbations on Grover. *CoRR*, abs/2302.00509. https://doi.org/10.48550/ARXIV.2302.00509

Kumarage, Tharindu, Amrita Bhattacharjee, Djordje Padejski, Kristy Roschke, Dan Gillmor, Scott W. Ruston, Huan Liu, and Joshua Garland. 2023a. J-Guard: Journalism guided adversarially robust detection of AI-generated news. *CoRR*, abs/2309.03164. https://doi.org/10.48550/ARXIV.2309.03164

Kumarage, Tharindu, Paras Sheth, Raha Moraffah, Joshua Garland, and Huan Liu. 2023b. How reliable are AI-generated-text detectors? An assessment framework using evasive soft prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1337–1349. https://doi.org/10.18653/v1/2023.findings-emnlp.94

Lambert, Nathan, Louis Castricato, Leandro von Werra, and Alex Havrilla. 2022. Illustrating reinforcement learning from human feedback (RLHF). *HuggingFace Blog*. Https://huggingface.co/blog/rlhf

Lavergne, Thomas, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse-Volume 377*, pages 27–31.

Lee, Bruce W., Yoo Sung Jang, and Jason Hyung-Jong Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event*, pages 10669–10686. https://doi.org/10.18653/v1/2021.emnlp-main.834

Lee, Haejun, Drew A. Hudson, Kangwook Lee, and Christopher D. Manning. 2020. SLM: Learning a discourse language representation with sentence unshuffling. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pages 1551–1562. https://doi.org/10.18653/v1/2020.emnlp-main.120

Lee, Taehyun, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2023. Who wrote this code? Watermarking for code generation. *CoRR*, abs/2305.15060. https://doi.org/10.48550/arXiv.2305.15060

Li, Linyang, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. 2023a. Origin tracing and detecting of LLMs. *CoRR*, abs/2304.14072. https://doi.org/10.48550/ARXIV.2304.14072

Li, Xian, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2023b. Self-alignment with instruction backtranslation. *ArXiv preprint*, abs/2308.06259.

Li, Yafu, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023c. Deepfake text detection in the wild. *CoRR*, abs/2305.13242. https://doi.org/10.48550/arXiv.2305.13242

Liang, Gongbo, Jesus Guerrero, and Izzat Alsmadi. 2023. Mutation-based adversarial attacks on neural text detectors. *ArXiv preprint*, abs/2302.05794.

Liang, Weixin, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023a. GPT detectors are biased against non-native English writers. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*. https://doi.org/10.1016/j.patter.2023.100779, PubMed: 37521038

Liang, Yaobo, Chenfei Wu, Ting Song, Wenshan Wu, Yan Xia, Yu Liu, Yang Ou, Shuai Lu, Lei Ji, Shaoguang Mao, et al. 2023b. TaskMatrix.AI: Completing tasks by connecting foundation models with millions of APIs. *ArXiv preprint*, abs/2303.16434. https://doi.org/10.34133/icomputing.0063

Liao, Wenxiong, Zhengliang Liu, Haixing Dai, Shaochen Xu, Zihao Wu, Yiyang Zhang, Xiaoke Huang, Dajiang Zhu, Hongmin Cai, Tianming Liu, and Xiang Li. 2023. Differentiate ChatGPT-generated and human-written medical texts. *CoRR*, abs/2304.11567. https://doi.org/10.48550/ARXIV.2304.11567, https://doi.org/10.2196/48904, PubMed: 38153785

Lin, Stephanie, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252. https://doi.org/10.18653/v1/2022.acl-long.229

Littman, Justin and Laura Wrubel. 2019. Climate Change Tweets Ids. https://doi.org/10.7910/DVN/5QCCUU

Liu, Aiwei, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S. Yu. 2023a. A private watermark for large language models. *ArXiv preprint*, abs/2307.16230.

Liu, Aiwei, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023b. A semantic invariant robust watermark for large language models. *CoRR*, abs/2310.06356. https://doi.org/10.48550/ARXIV.2310.06356

Liu, Aiwei, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2023c. A survey of text watermarking in the era of large language models. *CoRR*, abs/2312.07913. https://doi.org/10.48550/ARXIV.2312.07913

Liu, Shengchao, Xiaoming Liu, Yichen Wang, Zehua Cheng, Chengzhengxu Li, Zhaohan Zhang, Yu Lan, and Chao Shen. 2024. Does DetectGPT fully utilize perturbation? Bridging selective perturbation to fine-tuned contrastive learning detector would be better. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 1874–1889. https://doi.org/10.18653/v1/2024.acl-long.103

Liu, Xiaoming, Zhaohan Zhang, Yichen Wang, Yu Lan, and Chao Shen. 2022. CoCo: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *ArXiv preprint*, abs/2212.10341.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Liu, Yikang, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023d. ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models. *ArXiv preprint*, abs/2304.07666.

Liu, Zeyan, Zijun Yao, Fengjun Li, and Bo Luo. 2023e. Check me if you can: Detecting ChatGPT-generated academic writing using CheckGPT. *ArXiv preprint*, abs/2306.05524.

Lowerre, Bruce P. and B. Raj Reddy. 1976. Harpy, a connected speech recognition system. *The Journal of the Acoustical Society of America*, 59(S1):S97–S97. https://doi.org/10.1121/1.2003013

Lu, Ning, Shengcai Liu, Rui He, and Ke Tang. 2023. Large language models can be guided to evade AI-generated text detection. *ArXiv preprint*, abs/2305.10847.

Lu, Yaojie, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*, pages 5755–5772. https://doi.org/10.18653/v1/2022.acl-long.395

Lucas, Evan and Timothy Havens. 2023. GPTs don't keep secrets: Searching for backdoor watermark triggers in autoregressive language models. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 242–248. https://doi.org/10.18653/v1/2023.trustnlp-1.21

Ma, Shixuan and Quan Wang. 2024. Zero-shot detection of LLM-generated text using token cohesiveness. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pages 17538–17553. https://doi.org/10.18653/v1/2024.emnlp-main.971

Ma, Yongqiang, Jiawei Liu, and Fan Yi. 2023. Is this abstract generated by AI? A research for the gap between AI-generated scientific text and human-written scientific text. *ArXiv preprint*, abs/2301.10416.

Ma, Yongqiang, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, and Xiaozhong Liu. 2023. AI vs. human–differentiation analysis of scientific content generation. *arXiv*, https://arxiv.org/abs/2301.10416

Macko, Dominik, Róbert Móro, Adaku Uchendu, Jason Samuel Lucas, Michiharu Yamashita, Matús Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Mária Bieliková. 2023. Multitude: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 9960–9987. https://doi.org/10.18653/v1/2023.emnlp-main.616

Májovskỳ, Martin, Martin Černỳ, Matěj Kasal, Martin Komarc, and David Netuka. 2023. Artificial intelligence can generate fraudulent but authentic-looking scientific medical articles: Pandora's box has been opened. *Journal of Medical Internet Research*, 25:e46924. https://doi.org/10.2196/46924, PubMed: 37256685

Mao, Chengzhi, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: geneRative AI Detection viA Rewriting. *CoRR*, abs/2401.12970. https://doi.org/10.48550/ARXIV.2401.12970

Markowitz, David M., Jeffrey Hancock, and Jeremy Bailenson. 2023. Linguistic markers of inherent AI deception and intentional human deception: Evidence from hotel reviews. *PsyArXiv preprint*. https://doi.org/10.31234/osf.io/mnyz8

McCarthy, Philip M. 2005. *An Assessment of the Range and Usefulness of Lexical Diversity Measures and the Potential of the Measure of Textual, Lexical Diversity (MTLD)*. Ph.D. thesis, The University of Memphis.

Mindner, Lorenz, Tim Schlippe, and Kristina Schaaff. 2023. Classification of human- and AI-generated texts: Investigating features for ChatGPT. *CoRR*, abs/2308.05341. https://doi.org/10.48550/ARXIV.2308.05341, https://doi.org/10.1007/978-981-99-7947-9_12

Mirsky, Yisroel, Ambra Demontis, Jaidip Kotak, Ram Shankar, Deng Gelei, Liu Yang, Xiangyu Zhang, Maura Pintor, Wenke Lee, Yuval Elovici, et al. 2022. The threat of offensive AI to organizations. *Computers & Security*. page 103006. https://doi.org/10.1016/j.cose.2022.103006

Mitchell, Eric, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962.

Mitrović, Sandra, Davide Andreoletti, and Omran Ayoub. 2023. ChatGPT or human? Detect and explain. Explaining decisions of machine learning model for detecting short ChatGPT-generated text. *ArXiv preprint*, abs/2301.13852.

Moosavi, Nafise Sadat, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. SciGen: A dataset for reasoning-aware text generation from scientific tables. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Morris, John X., Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos*, pages 119–126, https://doi.org/10.18653/v1/2020.emnlp-demos.16

Mosca, Edoardo, Mohamed Hesham Ibrahim Abdalla, Paolo Basso, Margherita Musumeci, and Georg Groh. 2023. Distinguishing fact from fiction: A benchmark dataset for identifying machine-generated scientific papers in the LLM era. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 190–207. https://doi.org/10.18653/v1/2023.trustnlp-1.17

Mostafazadeh, Nasrin, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849. https://doi.org/10.18653/v1/N16-1098

Muñoz-Ortiz, Alberto, Carlos Gómez-Rodríguez, and David Vilares. 2023. Contrasting linguistic patterns in human and LLM-generated text. *ArXiv preprint*, abs/2308.09067. https://doi.org/10.21203/rs.3.rs-4077382/v1

Munyer, Travis J. E. and Xin Zhong. 2023. DeepTextMark: Deep learning based text watermarking for detection of large language model generated text. *CoRR*, abs/2305.05773. https://doi.org/10.48550/ARXIV.2305.05773

Muric, G., Y. Wu, and E. Ferrara. 2021. Covid-19 vaccine hesitancy on social media: Building a public Twitter dataset of anti-vaccine content, vaccine misinformation and conspiracies. 2021; 1–10. *ArXiv preprint*, abs/2105.05134. https://doi.org/10.2196/30642, PubMed: 34653016

Murtaza, Ghulam, Liyana Shuib, Ainuddin Wahid Abdul Wahab, Ghulam Mujtaba, Ghulam Mujtaba, Henry Friday Nweke, Mohammed Ali Al-garadi, Fariha Zulfiqar, Ghulam Raza, and Nor Aniza Azmi. 2020. Deep learning-based breast cancer classification through medical imaging modalities: State of the art and research challenges. *Artificial Intelligence Review*, 53:1655–1720. https://doi.org/10.1007/s10462-019-09716-5

Narayan, Shashi, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807. https://doi.org/10.18653/v1/D18-1206

Nguyen-Son, Hoang-Quoc, Minh-Son Dao, and Koji Zettsu. 2024. SimLLM: Detecting sentences generated by large language models using similarity between the generation and its re-generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pages 22340–22352. https://doi.org/10.18653/v1/2024.emnlp-main.1246

Nicks, Charlotte, Eric Mitchell, Rafael Rafailov, Archit Sharma, Christopher D. Manning, Chelsea Finn, and Stefano Ermon. 2023. Language model detectors are easily optimized against. In *The Twelfth International Conference on Learning Representations*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774. https://doi.org/10.48550/arXiv.2303.08774

Orenstrakh, Michael Sheinman, Oscar Karnalim, Carlos Anibal Suarez, and Michael Liut. 2023. Detecting LLM-generated text in computing education: A comparative study for ChatGPT cases. *ArXiv preprint*, abs/2307.07411.

Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback, 2022. *ArXiv preprint*, abs/2203.02155.

Pagnoni, Artidoro, Martin Graciarena, and Yulia Tsvetkov. 2022. Threat scenarios and best practices to detect neural fake news. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1233–1249.

Peng, Xinlin, Ying Zhou, Ben He, Le Sun, and Yingfei Sun. 2024. Hidding the ghostwriters: An adversarial evaluation of AI-generated student essay detection. *CoRR*, abs/2402.00412. https://doi.org/10.48550/ARXIV.2402.00412

Piccolo, Stephen R., Paul Denny, Andrew Luxton-Reilly, Samuel Payne, and Perry G. Ridge. 2023. Many bioinformatics programming tasks can be automated with ChatGPT. *ArXiv preprint*, abs/2303.13528.

Por, L. Y., T. F. Ang, and B. Delina. 2008. WhiteSteg: A new scheme in information hiding using text steganography. *WSEAS Transactions on Computers*, 7(6):735–745.

Por, Lip Yee, KokSheik Wong, and Kok Onn Chee. 2012. UniSpaChi: A text-based data hiding method using Unicode space characters. *Journal of Systems and Software*, 85(5):1075–1082. https://doi.org/10.1016/J.JSS.2011.12.023

Porsdam Mann, Sebastian, Brian D Earp, Sven Nyholm, John Danaher, Nikolaj Møller, Hilary Bowman-Smart, Joshua Hatherley, Julian Koplin, Monika Plozza, Daniel Rodger, et al. 2023. Generative AI entails a credit–blame asymmetry. *ArXiv preprint*, abs/2305.15324. https://doi.org/10.1038/s42256-023-00653-1

Price, Gregory and Marc D. Sakellarios. 2023. The effectiveness of free software for detecting AI-generated writing. *International Journal of Teaching, Learning and Education*, 2(6). https://doi.org/10.22161/ijtle.2.6.4

Provos, Niels and Peter Honeyman. 2003. Hide and seek: An introduction to steganography. *IEEE Security & Privacy*, 1(3):32–44. https://doi.org/10.1109/MSECP.2003.1203220

Pu, Jiameng, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023a. Deepfake text detection: Limitations and opportunities. In *44th IEEE Symposium on Security and Privacy, SP 2023*, pages 1613–1630. https://doi.org/10.1109/SP46215.2023.10179387

Pu, Xiao, Jingyu Zhang, Xiaochuang Han, Yulia Tsvetkov, and Tianxing He. 2023b. On the zero-shot generalization of machine-generated text detectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4799–4808. https://doi.org/10.18653/v1/2023.findings-emnlp.318

Qiu, Xipeng, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897. https://doi.org/10.1007/s11431-020-1647-3

Quidwai, Mujahid Ali, Chunhui Li, and Parijat Dube. 2023. Beyond black box AI generated plagiarism detection: From sentence to document level. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2023*, pages 727–735. https://doi.org/10.18653/v1/2023.bea-1.58

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. https://doi.org/10.18653/v1/D16-1264

Ren, Shuhuai, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097. https://doi.org/10.18653/v1/P19-1103

Rizzo, Stefano Giovanni, Flavio Bertini, and Danilo Montesi. 2016. Content-preserving text watermarking through Unicode homoglyph substitution. In *Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS 2016*, pages 97–104. https://doi.org/10.1145/2938503.2938510

Rodriguez, Juan, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233. https://doi.org/10.18653/v1/2022.naacl-main.88

Sadasivan, Vinu Sankar, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can AI-generated text be reliably detected? *ArXiv preprint*, abs/2303.11156.

Saeed, Waddah and Christian Omlin. 2023. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273. https://doi.org/10.1016/j.knosys.2023.110273

Sarvazyan, Areg Mikael, José Ángel González, Paolo Rosso, and Marc Franco-Salvador. 2023. Supervised machine-generated text detectors: Family and scale matters. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 121–132. https://doi.org/10.1007/978-3-031-42448-9_11

Schaaff, Kristina, Tim Schlippe, and Lorenz Mindner. 2023. Classification of human- and AI-generated texts for English, French, German, and Spanish. In *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 1–10.

Schneider, Sinclair, Florian Steuber, Joao A. G. Schneider, and Gabi Dreo Rodosek. 2023. How well can machine-generated texts be identified and can language models be trained to avoid identification? *CoRR*, abs/2310.16992. https://doi.org/10.48550/ARXIV.2310.16992, https://doi.org/10.24251/HICSS.2023.328

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv preprint*, abs/1707.06347.

Schuster, Tal, Roei Schuster, Darsh J. Shah, and Regina Barzilay. 2020. The limitations of stylometry for detecting machine-generated fake news. *Computational Linguistics*, 46(2):499–510. https://doi.org/10.1162/coli_a_00380

Seals, S. M. and Valerie L. Shalin. 2023. Long-form analogies generated by ChatGPT lack human-like psycholinguistic properties. *CoRR*, abs/2306.04537. https://doi.org/10.48550/ARXIV.2306.04537

Shah, Aditya, Prateek Ranka, Urmi Dedhia, Shruti Prasad, Siddhi Muni, and Kiran Bhowmick. 2023. Detecting and unmasking AI-generated texts through explainable artificial intelligence using stylistic features. *International Journal of Advanced Computer Science and Applications*, 14(10):110. https://doi.org/10.14569/IJACSA.2023.01410110

Shen, Dinghan, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *ArXiv preprint*, abs/2009.13818.

Shevlane, Toby, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, Lewis Ho, Divya Siddarth, Shahar Avin, Will Hawkins, Been Kim, Iason Gabriel, Vijay Bolina, Jack Clark, Yoshua Bengio, Paul F. Christiano, and Allan Dafoe. 2023. Model evaluation for extreme risks. *CoRR*, abs/2305.15324. https://doi.org/10.48550/arXiv.2305.15324

Shi, Zhouxing and Minlie Huang. 2020. Robustness to modification with shared words in paraphrase identification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 164–171. https://doi.org/10.18653/v1/2020.findings-emnlp.16

Shi, Zhouxing, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2023. Red teaming language model detectors with language models. *ArXiv preprint*, abs/2305.19713.

Shumailov, Ilia, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759. https://doi.org/10.1038/s41586-024-07566-y PubMed: 39048682

Sokolova, Marina, Nathalie Japkowicz, and Stan Szpakowicz. 2006. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In *Australian Conference on Artificial Intelligence*. https://doi.org/10.1007/11941439_114

Solaiman, Irene, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *ArXiv preprint*, abs/1908.09203.

Soni, Mayank and Vincent Wade. 2023. Comparing abstractive summaries generated by ChatGPT to real summaries through blinded reviewers and text classification algorithms. *CoRR*,

abs/2303.17650. https://doi.org/10.48550/ARXIV.2303.17650

Stiff, Harald and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *International Journal of Data Science and Analytics*, 13(4):363–383. https://doi.org/10.1007/S41060-021-00299-5

Stokel-Walker, Chris and Richard Van Noorden. 2023. What ChatGPT and generative AI mean for science. *Nature*, 614(7947):214–216. https://doi.org/10.1038/d41586-023-00340-6, PubMed: 36747115

Su, Jinyan, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023a. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. *CoRR*, abs/2306.05540. https://doi.org/10.48550/arXiv.2306.05540, https://doi.org/10.18653/v1/2023.findings-emnlp.827

Su, Zhenpeng, Xing Wu, Wei Zhou, Guangyuan Ma, and Songlin Hu. 2023b. HC3 Plus: A semantic-invariant human ChatGPT comparison corpus. *CoRR*, abs/2309.02731. https://doi.org/10.48550/ARXIV.2309.02731

Susnjak, Teo. 2022. ChatGPT: The end of online exam integrity? *ArXiv preprint*, abs/2212.09292.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112.

Tang, Ruixiang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting LLM-generated texts. *CoRR*, abs/2303.07205. https://doi.org/10.48550/arXiv.2303.07205

Tang, Ruixiang, Yu-Neng Chuang, and Xia Hu. 2024. The science of detecting LLM-generated text. *Communications of the ACM*, 67(4):50–59. https://doi.org/10.1145/3624725

Tang, Ruixiang, Qizhang Feng, Ninghao Liu, Fan Yang, and Xia Hu. 2023. Did you train on my dataset? Towards public dataset protection with clean-label backdoor watermarking. *CoRR*, abs/2303.11470. https://doi.org/10.48550/ARXIV.2303.11470, https://doi.org/10.1145/3606274.3606279

Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMa model. https://github.com/tatsu-lab/stanford_alpaca

Thirunavukarasu, Arun James, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature Medicine*. 29(8):1930–1940. https://doi.org/10.1038/s41591-023-02448-8, PubMed: 37460753

Topkara, Umut, Mercan Topkara, and Mikhail J. Atallah. 2006. The hiding virtues of ambiguity: Quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th Workshop on Multimedia & Security, MM&Sec 2006*, pages 164–174. https://doi.org/10.1145/1161366.1161397

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMa: Open and efficient foundation language models. *CoRR*, abs/2302.13971. https://doi.org/10.48550/arXiv.2302.13971

Tripto, Nafis Irtiza, Adaku Uchendu, Thai Le, Mattia Setzu, Fosca Giannotti, and Dongwon Lee. 2023. HANSEN: Human and AI spoken text benchmark for authorship analysis. *CoRR*, abs/2310.16746. https://doi.org/10.48550/ARXIV.2310.16746, https://doi.org/10.18653/v1/2023.findings-emnlp.916

Tu, Shangqing, Chunyang Li, Jifan Yu, Xiaozhi Wang, Lei Hou, and Juanzi Li. 2023. ChatLog: Recording and analyzing ChatGPT across time. *CoRR*, abs/2304.14106. https://doi.org/10.48550/ARXIV.2304.14106

Tulchinskii, Eduard, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Serguei Barannikov, Irina Piontkovskaya, Sergey Nikolenko, and Evgeny Burnaev. 2023. Intrinsic dimension estimation for robust detection of AI-generated texts. *ArXiv preprint*, abs/2306.04723.

Uchendu, Adaku, Thai Le, and Dongwon Lee. 2023a. Attribution and obfuscation of neural text authorship: A data mining perspective. *SIGKDD Explorations Newsletter*, 25(1):1–18. https://doi.org/10.1145/3606274.3606276

Uchendu, Adaku, Thai Le, and Dongwon Lee. 2023b. TOPROBERTA: Topology-aware authorship attribution of deepfake texts. *CoRR*, abs/2309.12934. https://doi.org/10.48550/ARXIV.2309.12934, https://doi.org/10.3233/FAIA240647

Uchendu, Adaku, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395. https://doi.org/10.18653/v1/2020.emnlp-main.673

Uchendu, Adaku, Jooyoung Lee, Hua Shen, and Thai Le. 2023. Does human collaboration enhance the accuracy of identifying LLM-generated deepfake texts? *ArXiv preprint*, abs/2304.01002. https://doi.org/10.1609/hcomp.v11i1.27557

Uchendu, Adaku, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016. https://doi.org/10.18653/v1/2021.findings-emnlp.172

Vasilatos, Christoforos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. 2023. HowkGPT: Investigating the detection of ChatGPT-generated university student homework through context-aware perplexity analysis. *ArXiv preprint*, abs/2305.18226.

Venkatraman, Saranya, He He, and David Reitter. 2023. How do decoding algorithms distribute information in dialogue responses? In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 923–932. https://doi.org/10.18653/v1/2023.findings-eacl.70

Venkatraman, Saranya, Adaku Uchendu, and Dongwon Lee. 2023. GPT-who: An information density-based machine-generated text detector. *CoRR*, abs/2310.06202. https://doi.org/10.48550/ARXIV.2310.06202

Verma, Vivek, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *CoRR*, abs/2305.15047. https://doi.org/10.48550/ARXIV.2305.15047

Walters, William H. 2023. The effectiveness of software designed to detect AI-generated writing: A comparison of 16 AI text detectors. *Open Information Science*, 7(1):20220158. https://doi.org/10.1515/opis-2022-0158

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*, OpenReview.net. https://doi.org/10.18653/v1/W18-5446

Wang, Pengyu, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023a. SeqXGPT: Sentence-level AI-generated text detection. *CoRR*, abs/2310.08903. https://doi.org/10.48550/ARXIV.2310.08903, https://doi.org/10.18653/v1/2023.emnlp-main.73

Wang, Yichen, Shangbin Feng, Abe Bohan Hou, Xiao Pu, Chao Shen, Xiaoming Liu, Yulia Tsvetkov, and Tianxing He. 2024. Stumbling blocks: Stress testing the robustness of machine-generated text detectors under attacks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 2894–2925. https://doi.org/10.18653/v1/2024.acl-long.160

Wang, Yuxia, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, et al. 2023b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *ArXiv preprint*, abs/2305.14902.

Wang, Zecong, Jiaxi Cheng, Chen Cui, and Chenhao Yu. 2023c. Implementing BERT and fine-tuned RobertA to detect AI generated news by ChatGPT. *CoRR*, abs/2306.07401. https://doi.org/10.48550/ARXIV.2306.07401

Weber-Wulff, Debora, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltýnek, Jean Guerrero-Dib, Olumide Popoola, Petr Šigut, and Lorna Waddington. 2023. Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*, 19(1):26. https://doi.org/10.1007/s40979-023-00146-z

Wei, Jason, Clara Meister, and Ryan Cotterell. 2021. A cognitive regularizer for language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 5191–5202. https://doi.org/10.18653/v1/2021.acl-long.404

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Weidinger, Laura, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models (2021). *ArXiv preprint*, abs/2112.04359.

Weng, Luoxuan, Minfeng Zhu, Kam Kwai Wong, Shi Liu, Jiashun Sun, Hang Zhu, Dongming Han, and Wei Chen. 2023. Towards an understanding and explanation for mixed-initiative artificial scientific text detection. *ArXiv preprint*, abs/2304.05011.

Wikipedia. 2023. Large language models and copyright. https://en.wikipedia.org/wiki/Wikipedia:Large_language_models_and_copyright

Winstein, Keith. 1998. Lexical steganography through adaptive modulation of the word choice hash. Unpublished. http://www.imsa.edu/~keithw/tlex

Wolff, Max. 2020. Attacking neural text detectors. *CoRR*, abs/2002.11768.

Wu, Junchao, Runzhe Zhan, Derek F. Wong, Shu Yang, Xuebo Liu, Lidia S. Chao, and Min Zhang. 2024a. Who wrote this? The key to zero-shot LLM-generated text detection is GECScore. *CoRR*, abs/2405.04286. https://doi.org/10.48550/ARXIV.2405.04286

Wu, Junchao, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. 2024b. DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. *CoRR*, abs/2410.23746. https://doi.org/10.48550/ARXIV.2410.23746

Wu, Kangxi, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. LLMDet: A third party large language models generated text detection tool. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2113–2133. https://doi.org/10.18653/v1/2023.findings-emnlp.139

Xiang, Lingyun, Yan Li, Wei Hao, Peng Yang, and Xiaobo Shen. 2018. Reversible natural language watermarking using synonym substitution and arithmetic coding. *Computers, Materials & Continua*, 55(3):541–559.

Yan, Duanli, Michael Fauss, Jiangang Hao, and Wenju Cui. 2023. Detection of AI-generated essays in writing assessment. *Psychological Testing and Assessment Modeling*, 65(2):125–144.

Yan, Yuanmeng, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075. https://doi.org/10.18653/v1/2021.acl-long.393

Yang, Lingyi, Feng Jiang, and Haizhou Li. 2023. Is ChatGPT involved in texts? Measure the Polish ratio to detect ChatGPT-generated text. *ArXiv preprint*, abs/2307.11380. https://doi.org/10.1561/116.00000250

Yang, Xi, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. 2023a. Watermarking text generated by black-box language models. *CoRR*, abs/2305.08883. https://doi.org/10.48550/ARXIV.2305.08883

Yang, Xianjun, Wei Cheng, Linda R. Petzold, William Yang Wang, and Haifeng Chen. 2023b. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. *CoRR*, abs/2305.17359. https://doi.org/10.48550/ARXIV.2305.17359

Yang, Xi, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. 2022. Tracing text provenance via context-aware lexical substitution. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event*, pages 11613–11621. https://doi.org/10.1609/aaai.v36i10.21415

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 5754–5764.

Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models, May 2023. *ArXiv preprint*, abs/2305.10601.

Yasunaga, Michihiro and Percy Liang. 2021. Break-it-fix-it: Unsupervised learning for program repair. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 11941–11952.

Yoo, KiYoon, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. 2023. Robust multi-bit natural language watermarking through invariant features. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 2092–2115. https://doi.org/10.18653/v1/2023.acl-long.117

Yu, Peipeng, Jiahan Chen, Xuan Feng, and Zhihua Xia. 2023a. CHEAT: A large-scale dataset for detecting ChatGPT-written abstracts. *CoRR*, abs/2304.12008. https://doi.org/10.48550/arXiv.2304.12008

Yu, Xiao, Kejiang Chen, Qi Yang, Weiming Zhang, and Nenghai Yu. 2024. Text fluoroscopy: Detecting LLM-generated text through intrinsic features. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pages 15838–15846. https://doi.org/10.18653/v1/2024.emnlp-main.885

Yu, Xiao, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Weiming Zhang, and Nenghai Yu. 2023b. GPT paternity test: GPT generated text detection with GPT genetic inheritance. *ArXiv preprint*, abs/2305.12519.

Zellers, Rowan, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019a. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800. https://doi.org/10.18653/v1/P19-1472

Zellers, Rowan, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019b. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 9051–9062.

Zeng, Zijie, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guanliang Chen. 2023. Towards automatic boundary detection for human–AI hybrid essay in education. *arXiv preprint arXiv:* 2307.12267.

Zhang, Ruisi, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. 2023a. REMARK-LLM: A robust and efficient watermarking framework for generative large language models. *CoRR*, abs/2310.12362. https://doi.org/10.48550/ARXIV.2310.12362

Zhang, Yue, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023b. Siren's song in the AI ocean: A survey on hallucination in large language models. *ArXiv preprint*, abs/2309.01219.

Zhang, Yi-Fan, Zhang Zhang, Liang Wang, Tieniu Tan, and Rong Jin. 2023c. Assaying on the robustness of zero-shot machine-generated text detectors. *CoRR*, abs/2312.12918. https://doi.org/10.48550/ARXIV.2312.12918

Zhao, Zihao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706.

Zhong, Wanjun, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2461–2470. https://doi.org/10.18653/v1/2020.emnlp-main.193

Zhu, Biru, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023. Beat LLMs at their own game: Zero-shot LLM-generated text detection via querying ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 7470–7483. https://doi.org/10.18653/v1/2023.emnlp-main.463

Togather-ai → all free instruct models
Hyperbolic → base models.

~~AI~~ Survey on AI detection

LLM-generated text detection → Binary classification task

- neural
- statistics
- Watermarking

Summarizing current progress

1. Introduction

1. Rise of LLM capabilities → a) near-human text generation 2022-23
   b) AI generated news articles ↑55%.
   c) Misinform" sites → ↑457%
2. Broad Integration across sectors → enabling personalized learning & medical diagnostics but misuse risks undermining trust.
3. Detection challange → a) Current automated tools often misclassify
   b) Human evaluators ~~to~~ struggle with accuracy slightly above random chance.
4. Societal & Ethical Concerns → misinfo, Plagiarism
5. Need for Robust Detection → high stake domains (healthcare, legal systems)

Background

2.1 → LLM generated Text Detection Task

* Binary Classification → a) Human written $D(x) = 0$
LLM " " " = 1

a) Human evaluators accuracy ≈ 50 %.

c) LLM text mimics human patterns (logical structure, formality)

a) Fabricated content complicates origin tracing (false citations, eg → if you ask to define NLP using some paper, it fabricates the papers to give the response)

LLMs → higher freq of noun, verb, determiner, less adverbs & is less emotional → to exhibit clearer presentations.

2.2 Generation Mechanism of LLMs → sequentially one word at a time based on input & already generated sequence

$X_n = \{x_1, x_2 \dots x_n\}$ → Input sequence of tokens

$X_{t-1} = \{y_1, y_2 \dots y_{t-1}\}$ → already generated tokens

$y_t \sim P(y_t | X_{t-1}, X_T) = \text{softmax}(W_o \cdot h_t)$

W_o → output matrix; h_t → hidden state

The token y_t is sampled from the probability distribution

Joint Probability function for the final output sequence

$Y_T = \{y_1 \dots y_T\}$

$P(Y_T | X_N) = \prod_{t=1}^{T} P(y_t | y_1, y_2 \sim y_{t-1}, X_N)$

product of probability of each output given previous text

Decoding Strategy → Method for selecting tokens from the probability distribution critically impacts text quality.

Strategies	Mechanism	Pros/cons
a) Greedy →	token with highest probability	Fast / repetitive outputs
b) Beam Search	maintain k-candidate sequences → top k paths	quality, diversity / repetitive fragments.
c) Top-k sampling	Samples from k most probable tokens (randomly)	randomness / incoherence if k too large
d) Top-p	Smallest set of tokens that surpass cumulative probability of p	coherent, diverse / effectiveness depends on p

∴ ∑ sum ≥ p

↓

hence here we choose from top k but k varies as we are considering sum.

2.2.2 Source of LLM's strong generation capabilities

In-Context learning → LLMs adapts to tasks just by giving few examples in the prompt, so we can do thing by prompt tuning, some which we earlier used to do task-specific fine-tuning (BERT)

RLHF → Reinforcement learning from Human Feedback

COT → Chain of thought, tree of thought.

2.3 Necessity of Detection

RLHF + refinements → (LLM tend to humans)

Not all scenarios need detection → LLMs handling routine replicable tasks (no risk)

Regulation (making ai regul^{n})
IP rights, copyright in training

LLM Text detection

- Users: overreliance on LLMs for generating content (blogs) reduce trust on digital
- Academic Integrity (Science): dilute explorations, undermine assessment (cheating), reduce independent thinking
- Human Society (They have specific textual patterns which can homogenize the richness & diversity of human communic^{n})
- Development of LLMs: LLM generated data used for training future models may degrade model quality

3. Earlier work

- statistical methods
- ChatGpt 2 → SOTA evoked many NLP peeps into detection.
- Classifiers trained from scratch, zero-shot classifier, fine tuning neural LMs.

Why this ??

- more depth & detailed overviews
- System for Literature review (SLR)

Sudden rise of papers in this field in 2023

Research questions
(What are prevailing methods, main challenges ?)

Methods & Challenges of LLM Detec^{n}.
Proposal of specific methodology
Scope for future

4. Data

High Quality of Datasets → swift calibration of data stars
→ Standard metrics

Problems
a) for AI detection → New so not much.
b) limited data volume & complexity

4.1 Datasets

a) HC3 (Human ChatGpt Corpus Comparison)
↓
Compares GPT with human answers on identical questions.
lacks diversity.

b) CHEAT → detecting fake academic content
Human written academic abstracts & their ChatGPT summaries
Polishing → Human refined LLM outputs.
Blending → LLM + human
only Acads

c) HC3 Plus → HC3 ⊕ HC3-SI (req. for tasks such as semantic invariance summaries, paraphrasing)
GPT 3.5 turbo
lacks complex prompts, diversity

d) Open LLM text → GPT 3.5, PaLM, Llama-7B, GPT 2/B ⊕ Human written samples

* different prompts but not fully capture cross domain & multilingual text.

e) Deepfake Text Detect Dataset → (Human Text) × 10 datasets
+
(27 LLMs) + Paraphrased + others.

*** * The increasing difficulty in ensuring new human written datasets are free from LLM
→ older Human written datasets become crucial.

4.2 Benchmarks → verify the detectors.

(i) CyPA Benchmark : Focus on academic writing (Robust)
(Human written) + (CyPT written) + (CyPT completed)
+ (CyPT polished) : Abstracts across domains.

(ii) HANSEN (Human and AI spoken Text Benchmark)
largest benchmark for spoken Text of humans, & AI generated spoken text.
(Written & Spoken have style differences)

(iii) M4 → Variety of generators, domains, languages.
common real world applic^ns, (CyPT, Llama)
10 languages (cross lingual)

(iv) DetectRL → Real word applic^n scenarios. Tasks assessing robustness, generaliz^n, varying text length, real world writing.
covers domains of potential misuse, like human revision, minor edits, paraphrased, varying length.

5. Advances in Detection

5.1 Watermarking Technology

* Adapted from C.V. to detect AI gen. images.

Why watermarking ?
- → Protect from unauthorised access (Sequence distill^{n})
- → replic^{n} & misuse

Embedding identifiable patterns within the text during generation process.

a) Logits Based Method

selection: Before text generation, a set of "green tokens" which are randomly selected from models vocab. other tokens → red

purpose ÷ embed a watermark

Mechanism → during token sampling stage (before prob. calc.) the logit scores from "green" tokens are promoted. so more green token in output than general.

Detection → statistically analyse the text, significant higher frequency of token from the "green" set than unwatermarked text,

(WLLM) & (SWEET)
↓ for code
*Watermarking LLM
watermarks at position with high token distribu^{n} entropy.

SIR → Semantic Invariant Robust Watermark.

(Use semantic embedding of previous token to determine watermark logic, offering greater robustness against synonym substitution.

b) Token Sampling Based Method

↳ influence sampling process, which is normally random, to make text less predictable.

* Setting random seeds or specific patterns for token sampling.

* Sequence of random numbers as a secret watermark key to guide token sampling at sentence level → distortion free watermarking. * Highly robust

* SemStamp → ; Addresses vulnerability against paraphrasing attacks by operating at sentence level.

Locality Sensitive Hashing

a) Encode Candidate Sentences

b) Use LSH to partition semantic embedding space in regions → watermarked / non-watermarked.

c) Sentence level rejection sampling - model keeps generating & evaluating sentences until a sentence falls into designated watermarked region.

Better & more resilient against paraphrasing attacks with better quality.

c) Character / Symbol substitution based methods

* Watermarking by substituting characters or symbols
* UniSpaChi → insert selected unicode characters b/w sentences, words, lines & paragraphs.
* Easymark → Unicode has many visually identical or similar code points for substitution, unlike earlier methods only using space characters

d) Synonym substitution Methods

* At word level → synonym substitution.
* T-lex → continuously replaced words with synonyms until the text carried intended watermark.
* Wordnet → select synonyms deliberately operating just below distortion (meaning change) threshold after embedding the watermark.
* Earlier synonyms which can carry meaningful payload were turned into binary sequences before embedding to text.

e) Sequence to Sequence Methods

• Adversarial Watermark Transformer

End-to-end framework that automates the learning of word-replacements and their context for watermark embedding.

Training to inject binary messages into texts as encoding level.

Virtually imperceptible with minimal impact on semantic.

AWT → test

5.3 → Neural Based Method

watermarking methods → require access to model's deployment &

neural → analyse inherent text features.

learn patterns & characteristics that distinguish LLM generated text from human-written text.

Feature Based ← Pre-training

(a) Feature Based

* Exploit differences in linguistic features b/w human & LLM text.

* Extract key statistical language features & then M.L. to classify. (like fake-news detc^n)

• what type of feature? → 93% accuracy

(i) Stylistic features → Focus on frequency of word that highlight stylistic elements, frequency of Capitalised words, nouns, punctuation etc.

(ii) Complexity Features → Texts complexity, type-token ratio (TTR) & Measure of Textual lexical diversity (MTLD)

type-token = unique words / no. of tokens

MTLD → TTR over a window

High MTLD → more rich & diverse

• Semantic Features → Advanced Semantic (AdSem), Lexico-Semantic (LxSem)

and statistics of semantic dependancy tags, extracted using tools like LingFeat.

Agent, theme, predicate argument structure

(Hyperonym/hyponym, Synonym, Polysemy)

roles/themes (Et linguistics)

• Psychological Features → Sentiment Aalysis.

• Inform" Features → named entity, opinions etc.
(NE) (OP)

Incorporating Deeper structural features, such as those based on Rhetorical structure theory (RST) improves performance.

b) Pre-training Classifier

fine tuning of encoder-based classifier.

Pre-trained Language Models → exceptional NLP capabilities like categoriz"

BERT, RoBERTa, XLNet → outperformed traditional methods.

↓

very good for detec"

approaches based on Adversarial Training.

- Sample Enhancement based Adversarial Training:
  This involves using adversarial attacks/training which is a strategy to harden detectors (or discriminators) against malicious or "deceptive" inputs by deliberately teaching them to recognize and resist crafted perturbations.

  # Highly robust models against potential attacks.

- Two Player Games → GANs → An attack model & a detection model with iterative confrontation enhancing detection capabilities.

  * RADAR → Robust detector through adversarial training; with one paraphrasing model (generating content to evade detec^{n}) and a detector (improving its identific^{n} capabilities)

  RADAR refines the paraphrase model using the feedback from detector outperforming.

  * OutFox → Attacker ⟷ Detector
  (continual interac^{n})

  more emphasis of in-context learning for attacker identification. The detectors output act as ICL example to generate challenging text, which further fes detector robustness.

  >> Roberta, DetectGPT

LLMs as detectors

Feasibility and Reliability of using LLMs themselves to detect text generated by themselves or LLMs.

For newer models, GPTzero itself was the best detector.

But ChatGPT → can't identify LLM text reliably.
↓
acc = 0.5

GPT-4 → 85% of human text as AI.

5.2 Statistics Based Method → no extra training
→ Broad application.

a) White Box statistics

Require access to LLM's logits.

Logits → unnormalized log probability of the models output vocab.
= values just before softmax.

- Logits Based Methods: Compares log probability distribution of generated text against Human text.

* Likelihood (or log-likelihood)

logits = $[z_1 \; - \; - \; - \; z_n]$

$p_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$ ← Softmax to get probability / likelihood

$\log(p_i)$ or log likelihood $= z_i - \log(\Sigma e^{z_i})$

* Rank → Calculates ranking of each word based on logit score.
High average rank → text is LLM generated

* Log-Rank → log(Rank)
GLTR tool → using colours to represent tokens based on sampling frequency.

* log likelihood ratio ranking → log(p_i) / rank

• Perplexity Based Methods → how surprised the model is by the data it sees.

$$PP(p) = \exp\left(-\frac{1}{N}\sum \log p(w_i \mid w_{<i})\right)$$

* HowkGPT → used perplexity scores to differentiate text; ChatGPT text → lower perplexity.

* LLMDet → Categorises perplexity scores by calculating probability of next token for selected n-grams.
better than Roberta
We can also try to use ratio of perplexity to cross-perplexity, measuring how one model surprises another.

• Perturbation Based Method

DetectGPT → perturbes the text & then measures log-likelihood score → low score → LLM

a) Black-Box Method → without logits score.

DNA-GPT → use n-gram probability divergence. Extends a text using LLM & compares new n-gram divergence. Robust

* DPIC → similarity b/w original & regenerated versions.

6. Intuitive Detectors

(i) • Humans "Evalue" → look at signs like lack of coherence & consistency, logical errors.

• Domain dependence → "generic" text in news context but in story → "irrelevant"

• Academic writing → may lack detail. less grammatical errors, concise.

(ii) Imperceptible Features

Some features not easily noticed by humans, can be efficiently captured by detector algorithms,

Statistical Thresholds & Visual Forensic Tools

Hybrid approaches → human + AI detector

How ?? → Iterative training → human experts keep explaining the diff. b/w human & LLM text

7. Evaluation Metrics

Confusion matrix

	Predicte (+ve)	Pred (-ve)
Actual +ve	TP	FN
Actual -ve	FP	TN

Accuracy — $\frac{TP+TN}{TP+FP+TN+FN}$

Precision — $\frac{TP}{TP+FP}$

Recall — $\frac{TP}{TP+FN}$

F1-Score — $\frac{2 \times Precision \times Recall}{Precision + recall}$

~~AUROC~~ — ?

AUROC — ?