A Survey on Detection of LLMs-Generated Content

Anonymous ACL submission

Abstract

The burgeoning capabilities of advanced large language models (LLMs) such as ChatGPT have led to an increase in synthetic content generation with implications across a variety of sectors, including media, cybersecurity, public discourse, and education. As such, the ability to detect LLMs-generated content has become of paramount importance. We aim to provide a detailed overview of existing detection strategies and benchmarks, scrutinizing their differences and identifying key challenges and prospects in the field, advocating for more adaptable and robust models to enhance detection accuracy. We also posit the necessity for a multi-faceted approach to defend against various attacks to counter the rapidly advancing capabilities of LLMs. To the best of our knowledge, this work is the first comprehensive survey on the detection in the era of LLMs. We hope it will provide a broad understanding of the current landscape of LLMs-generated content detection, and we have maintained a website to consistently update the latest research as a guiding reference for researchers and practitioners.

1 Introduction

With the rapid development of powerful AI tools, the risk of LLMs-generated content has raised considerable concerns, such as misinformation spread (Bian et al., 2023; Hanley and Durumeric, 2023; Pan et al., 2023), fake news (Oshikawa et al., 2018; Zellers et al., 2019; Dugan et al., 2022), gender bias (Sun et al., 2019), education (Perkins et al., 2023; Vasilatos et al., 2023), and social harm (Kumar et al., 2023; Yang et al., 2023c).

We also find on the Google search trend, that the concerns about AI-written text have witnessed a significant increase since the release of the latest powerful Large Langue Models (LLMs) such as ChatGPT (Schulman et al., 2022) and GPT-4 (OpenAI, 2023b). Humans are already unable to directly distinguish between LLMs- and human-written text, with the fast advancement of the model size, data scale, and AI-human alignment (Brown et al., 2020; Ouyang et al., 2022). Concurrently, growing interests are shown to detectors, like the commercial tool GPTZero (Tian, 2023), or OpenAI's own detector (OpenAI, 2023a) since humans can be easily fooled by improvements in decoding methods (Ippolito et al., 2019). However, the misuse of detectors also raises protests from students on the unfair judgment on their homework and essays (Herbold et al., 2023; Liu et al., 2023b) and popular detectors perform poorly on code detection (Wang et al., 2023a). Alongside these advancements, there has been a proliferation of detection algorithms aimed at identifying LLMs-generated content. However, there remains a dearth of comprehensive surveys encompassing the latest methodologies, benchmarks, and attacks on LLMs-based detection systems.

Earlier work on text detection dates back to feature engineering (Badaskar et al., 2008). For example, GTLR (Gehrmann et al., 2019a) assumes the generated word comes from the top distribution on small LMs like BERT (Devlin et al., 2019) or GPT-2 (Radford et al., 2019). Recently, there has been an increasing focus on detecting ChatGPT (Weng et al., 2023; Liu et al., 2023b; Desaire et al., 2023), to mitigate ChatGPT misuse or abuse (Sison et al., 2023). In particular, it has recently been called for regulation[1] on powerful AI like ChatGPT usage (Hacker et al., 2023; Wahle et al., 2023).

Therefore, we firmly believe that the timing is ideal for a comprehensive survey on the detection of LLMs-generated content. It would serve to invite further exploration of detection approaches, offer valuable insights into the strengths and weaknesses of previous research, and highlight potential chal-

[1]https://www.nytimes.com/2023/05/16/technology/openai-altman-artificial-intelligence-regulation.html

lenges and opportunities for the research community to address. Our paper is organized as follows: we first briefly describe the problem formulation, including the task definition, metrics, and datasets in Section 2. In Section 3, we classify detection by their working mechanism and scope of application. In section 4, we summarize the three popular detection methods: training-based, zero-shot and watermarking. We also investigate various attacks in Section A.2 since defending against attacks is of increasing importance and point out some challenges in Section A.3. Finally, in Section 5 we provide additional insights into this topic on potential future directions, as well as the conclusion in Section 6.

2 Problem formulation

2.1 Overview

We refer to any textual outputs from LLMs following specific inputs as LLMs-Generated Content. It can be generally classified into natural languages like news, essays, reviews, and reports, or programming languages like codes of Python, C++, and Java. Current research usually aims at the detection of content with moderate length and specific topics. It is meaningless to detect a short sentence describing some facts like *EMNLP started in 1996* or simple coding question *def hello_world(): print('Hello World')*, to be human or AI written.

Formally, consider an LLM denoted as LLM, which generates a candidate text S of length $|S|$ based on an input prompt. Let $f()$ represent a potential detector we aim to use for classification, assigning $f(S)$ to 0 or 1, where 0 and 1 signify human or machine, respectively. The LLM can be classified into unknown (Black-box), fully known (White-box), or partially known (known model name with unknown model parameters) to the detectors. In practice, we are usually given a candidate corpus C comprising both human and LLMs-generated content to test $f()$.

Apart from the standard definition, machine-generated content can undergo additional modifications in practical scenarios, including rephrasing by humans or other AI models. Besides, it is also possible that the candidate text is a mix of human and machine-written text. For example, the first several sentences are written by humans, and the remaining parts by machines, or vice versa. When a text undergoes revisions, the community often perceives it as paraphrasing and treats it as either machine- or human-generated text, depending on the extent of these modifications and the intent behind them. However, it is important to highlight that if a substantial majority of the text is authored by humans, or if humans have extensively revised machine-generated text, it becomes challenging to maintain the assertion that the text is purely machine-generated. Hence, in this survey, we adhere to the traditional definition by considering machine-generated content as text that has not undergone significant modifications, and we consistently classify such text as machine-generated.

2.2 Metrics

Previous studies (Mitchell et al., 2023; Sadasivan et al., 2023) predominantly used the Area Under the Receiver Operating Characteristic (AUROC) score to gauge the effectiveness of detection algorithms. As a binary classification problem, AUROC shows the results under different thresholds, and the F1 score is also helpful. Krishna et al. (2023); Yang et al. (2023b) suggest that AUROC may not consistently provide a precise evaluation, particularly as the AUROC score nears the optimal limit of 1.0 since two detectors with identical AUROC score of 0.99 could exhibit substantial variations in detection quality from a user's perspective. From a practical point of view, ensuring a high True Positive Rate (TPR) is imperative while keeping the False Positive Rate (FPR) to a minimum. As such, current research (Krishna et al., 2023; Yang et al., 2023b) both report TPR scores at a fixed 1% FPR, along with the AUROC. Other work (Sadasivan et al., 2023) also refer to Type I and Type II errors following the binary hypothesis test and even report TPR at 10^{-6} FPR (Fernandez et al., 2023).

2.3 Datasets

In this section, we discuss the common datasets used for this task. The corpus is usually adopted from previous NLP tasks, and reconstructed by prompting LLMs to generate new outputs as candidate machine-generated text. Usually, there are two prompting methods: 1). prompting LLMs with the questions in some question-answering datasets. 2). prompting LLMs with the first 20 to 30 tokens to continue writing in datasets without specific questions. Specifically, several datasets have been compiled and utilized in the field. Some noteworthy datasets include TURINGBENCH (Uchendu et al., 2021), HC3 (Guo et al., 2023), CHEAT (Yu et al., 2023a), Ghostbuster (Verma et al., 2023), OpenGPTText (Chen et al., 2023c), M4 (Wang et al.,

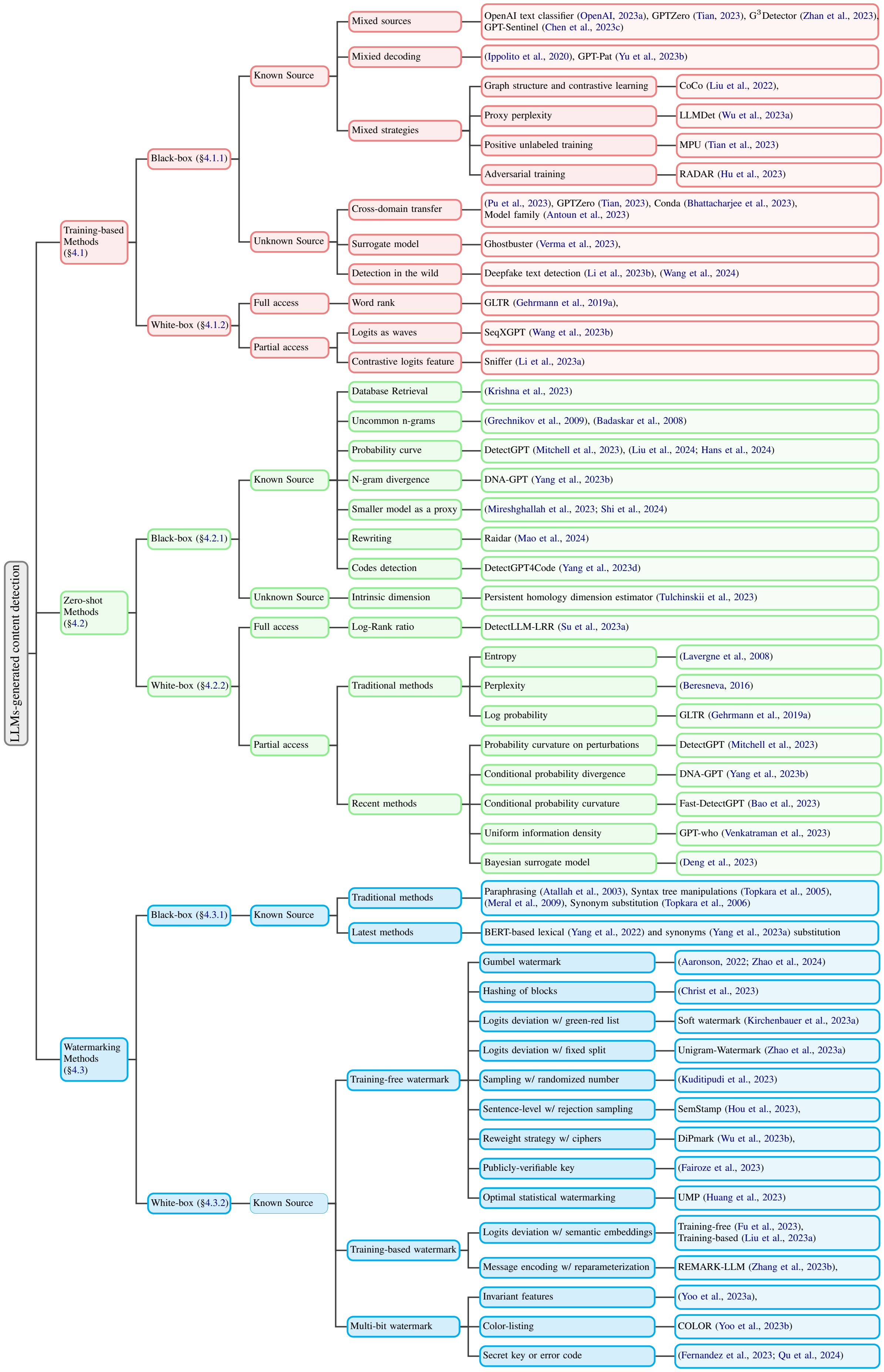

Figure 1: Taxonomy on detection methods. We list the most representative approaches for each category.

2023c), MGTBench (He et al., 2023), and MULTITuDE (Macko et al., 2023) and some other datasets not explicitly built for detection have also been used, such as C4 (Raffel et al., 2019), shareGPT [2], and alpaca (Taori et al., 2023), as summarized in Table 2. For text detection, we only list datasets explicitly built for detection, while some general datasets like C4 (Raffel et al., 2019) or alpaca (Taori et al., 2023) can also be used. For code detection, we only list datasets that have been used in previous code detection work (Lee et al., 2023; Yang et al., 2023d). And other codegeneration corpora can also be adopted. The detailed description is included in Appendix A.7.

Data Contamination. Despite those released standard datasets, we argue that static evaluation benchmarks might not be desirable for this problem with the rapid progress of LLMs trained, tuned, or aligned on large amounts of data across the whole internet. On the one hand, Aaronson (2022) mentioned that some text from Shakespeare or the Bible is often classified as AI-generated because such classic text is frequently used in the training datasets for generative language models. On the other hand, many detectors did not fully disclose their training data, especially commercial tools like GPTZero (Tian, 2023). It is natural to worry that those standard evaluation benchmarks would face a serious test data contamination problem, considering the commercial detectors would consistently improve their products for profits. So, with the rapid evolution of LLMs and detectors, the traditional paradigm of providing standard benchmarks might no longer be suitable for AI-generated text detection. We provide a unique solution to this:

👑 **Utilize the most latest human-written content to reduce data contamination problem by collecting such content from the most updated open-source websites, which themselves explicitly forbid posting AI-written posts.**

3 Detection Scenarios

The findings of previous research, such as (Gehrmann et al., 2019b) and (Dugan et al., 2022), highlight the general difficulty faced by humans in distinguishing between human- and machine-generated text, motivating the development of automatic solutions. The detection process can be classified into black-box or white-box detection based on whether the detector has access to the source model output logits. In black-box detection, there are two

[2]https://sharegpt.com/

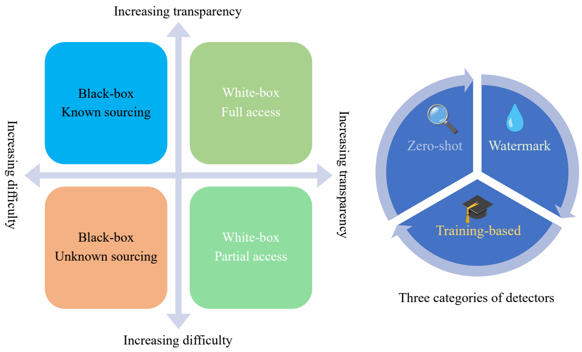

Figure 2: Three categories of detectors and four detection scenarios: as the transparency decreases, the detection difficulty increases.

distinct cases: 1). when the source model name is known, such as `GPT-4`; 2). when the source model name is unknown, and the content might have been generated by models like `GPT-4`, `Bard`, or other undisclosed models. On the other hand, white-box detection also encompasses two cases: 1). the detector only has access to the model's output logits or partial logits, such as the top-5 token log probability in `text-davinci-003`; 2). the detector has access to the entire model weights. Table 2 shows four categories according to application scenarios and three detector methods. Specifically, we can categorize the usage of detecting LLM-generated content into four distinct scenarios based on their application: These categorizations highlight the different levels of information available to the detectors, ranging from limited knowledge to complete access and demonstrate the various scenarios encountered in detecting machine-generated content.

3.1 Black-Box Detection with Unknown Model Source

This scenario closely resembles real-world applications, particularly when users, such as students, utilize off-the-shelf AI services to assist them in writing their essays. In such cases, teachers are often unaware of the specific AI service being employed. Consequently, this situation poses the greatest challenge as very limited information is available to identify instances of deception.

3.2 Black-Box Detection with Known Model Source

In this scenario, we possess knowledge regarding the specific model from which the text originates, yet we lack access to its underlying parameters. This aspect carries considerable significance due to the market domination of major language model providers such as OpenAI and Google. Many users rely heavily on their services, enabling us to make informed assumptions about the model sources.

3.3 White-Box Detection with Full Model Parameters

While access to the most powerful LLMs, such as Anthropic's Claude or OpenAI's ChatGPT, is typically limited, assuming full access to the model parameters is an active research area. This approach is reasonable, considering that researchers often encounter resource constraints, making it challenging to experiment with large-scale models. For instance, watermarking-based methods (Kirchenbauer et al., 2023a) typically require full access to the model parameters. This technique manipulates the next token prediction at each sampling position by modifying the distribution. Although this approach necessitates access to the complete model parameters, it has shown promise and could potentially be adapted for practical use.

3.4 White-Box Detection with Partial Model Information

This corresponds to the scenarios when only the partial model outputs, like the top-5 token logits are provided by `text-davinci-003`. Previous work like DetectGPT (Mitchell et al., 2023) and DNA-GPT (Yang et al., 2023b) both utilize such probability to perform detection.

3.5 Model Sourcing

Furthermore, another aspect related to detection goes beyond distinguishing between human and machine-generated content. This task involves determining which specific model may have generated the content and is referred to as authorship attribution (Uchendu et al., 2020), origin tracing (Li et al., 2023a), or model sourcing (Yang et al., 2023b). We consider this task as a special scenario since it is slightly different from other detection tasks.

4 Detection Methodologies

In this section, we delve into further details about the detection algorithms. Based on their distinguishing characteristics, existing detection methods can be categorized into three classes: 1) Training-based classifiers, which typically fine-tune a pre-trained language model on collected binary data - both human and AI-generated text distributions. 2) Zero-shot detectors leverage the intrinsic properties of typical LLMs, such as probability curves or representation spaces, to perform self-detection. 3) Watermarking involves hiding identifying information within the generated text that can later be used to determine if the text came from a specific language model, rather than detecting AI-generated text in general. We summarize the representative approaches in Figure 1 as classified by the scenarios listed in Section 3.

4.1 Training-based 🔥

The earlier work of training a detection classifier focuses on fake review (Bhagat and Hovy, 2013), fake news (Zellers et al., 2019) or small models (Solaiman et al., 2019; Bakhtin et al., 2019; Uchendu et al., 2020) detection. Subsequently, growing interest in this line of research turns to detecting high-quality text brought by LLMs.

4.1.1 Black-box

The first line of work focuses on black-box detection. *When the model source is known*, some work use the text generated by (1) **mixed sources** and subsequently train a classifier together for detection. For example, OpenAI (OpenAI, 2023a) collects text generated from different model families and trains a robust detector for detection text with more than 1,000 tokens. GPTZero (Tian, 2023) also collects their human-written text spans student-written articles, news articles, and Q&A datasets spanning multiple disciplines from a variety of LLMs. Similarly, G^3Detector (Zhan et al., 2023) claims to be a general GPT-Generated text detector by finetuning RoBERTa-large (Liu et al., 2019) and explores the effect of the use of synthetic data on the training process. GPT-Sentinel (Chen et al., 2023c) trains the RoBERTa and T5 (Raffel et al., 2020) classifiers on their constructed dataset OpenGPTText. (2) **Mixed decoding** is also utilized by incorporating text generated with different decoding parameters to account for the variance. Ippolito et al. (2020) find that, in general, discriminators transfer poorly between decoding strategies, but training on a mix of data can help. GPT-Pat (Yu et al., 2023b) train a siamese network to compute the similarity between the original text and the re-decoded text. Besides, (3) **mixed strategies** involves additional information, such as graph structure and contrastive learning in CoCo (Liu et al., 2022), proxy model perplexity in LLMDet (Wu et al., 2023a), positive unlabeled training in MPU (Tian et al., 2023) and adversarial training in RADAR (Hu et al., 2023).

On the other hand, *when the source model is unknown*, OpenAI text classifier (OpenAI, 2023a) and GPTZero (Tian, 2023) still works by 1 **cross-domain transfer**. Other works like (Pu et al., 2023; Antoun et al., 2023), Conda (Bhattacharjee et al., 2023) also rely on the zero-shot generalization ability of detectors trained on a variety of model families and tested on unseen models. Besides, Ghostbuster (Verma et al., 2023) directly uses outputs from known 2 **surrogate model** as the signal for training a classifier to detect unknown model. Additionally, 3 **detection in the wild** (Li et al., 2023b) contributes a wild testbed by gathering texts from various human writings and deepfake texts generated by different LLMs for detection without knowing their sources.

4.1.2 White-box

The second kind of work lies in the white-box situation when the model's full or partial parameters are accessible. For example, when we have full access to the model, GLTR (Gehrmann et al., 2019a) trains a logistic regression over absolute word ranks in each decoding step. When only partial information like the model output logits are available, SeqXGPT (Wang et al., 2023b) introduce a sentence-level detection challenge by synthesizing a dataset that contains documents that are polished with LLMs and propose to detect it with logits as waves from white-box LLMs. Sniffer (Li et al., 2023a) utilizes the contrastive logits between models as a typical feature for training to perform both detection and origin tracking.

4.2 Zero-Shot

In the zero-shot setting, we do not require extensive training data to train a discriminator. Instead, we can leverage the inherent distinctions between machine-generated and human-written text, making the detector training-free. The key advantage of training-free detection is its adaptability to new data distributions without the need for additional data collection and model tuning. It's worth noting that while watermarking methods can also be considered zero-shot, we treat them as an independent track. Previous work utilizes entropy (Lavergne et al., 2008), average log-probability score (Solaiman et al., 2019), perplexity (Beresneva, 2016), uncommon n-gram frequencies (Grechnikov et al., 2009; Badaskar et al., 2008) obtained from a language model as the judge for determining its origin. However, those simple features fail as LLMs are becoming diverse and high-quality text generators. Similarly, there are also black- and white-box detection, as summarized below.

4.2.1 Black-Box

When the source of the black-box model is known, DNA-GPT (Yang et al., 2023b) achieves superior performance by utilizing N-Gram divergence between the continuation distribution of re-prompted text and the original text. Besides, DetectGPT (Mitchell et al., 2023) also investigates using another surrogate model to replace the source model but achieves unsatisfactory results. In contrast, Mireshghallah et al. (2023) proves that a smaller surrogate model like OPT-125M (Zhang et al., 2022) can serve as a universal black-box text detector, achieving close or even better detection performance than using the source model. Additionally, Krishna et al. (2023) suggests building a database of generated text and detecting the target text by comparing its semantic similarity with all the text stored in the database. Finally, DetectGPT4Code (Yang et al., 2023d) also investigates detecting codes generated by ChatGPT through a proxy small code generation models by conditional probability divergence and achieves significant improvements on code detection tasks.

When the source of the model is unknown, PHD (Tulchinskii et al., 2023) observes that real text exhibits a statistically higher intrinsic dimensionality compared to machine-generated texts across various reliable generators by employing the Persistent Homology Dimension Estimator (PHD) as a means to measure this intrinsic dimensionality, combined with an additional encoder like Roberta to facilitate the estimation process.

4.2.2 White-Box

When the partial access to the model is given, traditional methods use the features such as entropy (Lavergne et al., 2008), average log-probability score (Solaiman et al., 2019) for detection. However, these approaches struggle to detect text from the most recent LLMs. Then, the pioneer work DetectGPT (Mitchell et al., 2023) observes that LLM-generated text tends to occupy negative curvature regions of the model's log probability function and leverages the curvature-based criterion based on random perturbations of the passage. DNA-GPT (Yang et al., 2023b) utilizes the probability difference between the continuous distribution among re-prompted text and original text and achieves state-of-the-art performance. Later, Deng et al. (2023) improves the efficiency of DetectGPT with a Bayesian surrogate model by selecting typical samples based on Bayesian uncertainty and interpolating scores from typical samples to other ones. Furthermore, similar to DNA-GPT (Yang et al., 2023b) on using the conditional probability for discrimination, Fast-DetectGPT (Bao et al., 2023) builds an efficient zero-shot detector by replacing the probability in DetectGPT with conditional probability curvature and witnesses significant efficiency improvements. Additionally, GPT-who (Venkatraman et al., 2023) utilizes Uniform Information Density (UID) based features to model the unique statistical signature of each LLM and human author for accurate authorship attribution.

When the full access to the model is given, Su et al. (2023a) leverages the log-rank information for zero-shot detection through one fast and efficient DetectLLM-LRR (Log-**L**ikelihood **L**og-Rank **r**atio) method, and another more accurate DetectLLM-NPR (**N**ormalized **p**erturbed log **r**ank) method, although slower due to the need for perturbations.

4.3 Watermarking 💧

Text watermarking injects algorithmically detectable patterns into the generated text while ideally preserving the quality and diversity of language model outputs. Although the concept of watermarking is well-established in vision, its application to digital text poses unique challenges due to the text's discrete and semantic-sensitive nature (Kutter et al., 2000). Early works are edit-based methods that modify a pre-existing text. The earliest work can be dated back to Atallah et al. (2001), which designs a scheme for watermarking natural language text by embedding small portions of the watermark bit string in the syntactic structure of the text, followed by paraphrasing (Atallah et al., 2003), syntax tree manipulations (Topkara et al., 2005; Meral et al., 2009) and synonym substitution (Topkara et al., 2006). Besides, text watermarking has also been used for steganography and secret communication (Fang et al., 2017; Ziegler et al., 2019; Abdelnabi and Fritz, 2021), and intellectual property protection (He et al., 2022a,b; Zhao et al., 2022, 2023b), but this is out the scope of this work. In light of growing ethical considerations, text watermarking has been increasingly used to ascertain the origin of textual content and detect AI-generated content (Grinbaum and Adomaitis, 2022). The primary focus of this paper is on the use of text watermarking to detect AI-generated text.

In general, watermarking for text detection can also be classified into white-box and black-box watermarking. Watermarking is designed to determine whether the text is coming from a specific language model rather than universally detecting text generated by any potential model. As such, knowledge of the model source is always required in text watermarking for detection.

4.3.1 Black-Box Watermarking

In black-box setting, such as API-based applications, the proprietary nature of the language models used by LLM providers precludes downstream users from accessing the sampling process for commercial reasons. Alternatively, a user may wish to watermark human-authored text via post-processing. In such cases, black-box watermarking aims to automatically manipulate generated text to embed watermarks readable by third parties. Traditional works designed complex linguistic rules such as paraphrasing (Atallah et al., 2003), syntax tree manipulations (Topkara et al., 2005; Meral et al., 2009) and synonym substitution (Topkara et al., 2006), but lack scalability. Later work turns to pre-trained language models for efficient watermarking. For example, Yang et al. (2022) proposes a natural language watermarking scheme based on context-aware lexical substitution (LS). Specifically, they employ BERT (Devlin et al., 2019) to suggest LS candidates by inferring the semantic relatedness between the candidates and the original sentence. Yang et al. (2023a) first defines a binary encoding function to compute a random binary encoding corresponding to a word. The encodings computed for non-watermarked text conform to a Bernoulli distribution, wherein the probability of a word representing bit-1 is approximately 0.5. To inject a watermark, they alter the distribution by selectively replacing words representing bit-0 with context-based synonyms that represent bit-1. A statistical test is then used to identify the watermark.

4.3.2 White-Box Watermarking

The most popular 1 **training-free** watermark directly manipulates the decoding process when the model is deployed. In the efforts of watermarking GPT outputs, Aaronson (2022) works with OpenAI to first develop a technique for watermarking language models using exponential minimum sampling to sample text from the model, where the inputs to the sampling mechanism are a hash of the previous k consecutive tokens through a pseudo-random number generator. By Gumbel Softmax (Jang et al., 2016) rule, their method is proven to ensure guaranteed quality. Besides, Christ et al. (2023) provides the formal definition and construction of undetectable watermarks. Their cryptographically inspired watermark design proposes watermarking blocks of text from a language model by hashing each block to seed a sampler for the next block. However, there are only theoretical concepts for this method without experimental results. Another pioneering work of training-free watermark (Kirchenbauer et al., 2023a) embeds invisible watermarks in the decoding process by dividing the vocabulary into a "green list" and a "red list" based on the hash of prefix token and subtly increases the probability of choosing from the green list. Then, a third party, equipped with knowledge of the hash function and random number generator, can reproduce the green list for each token and monitor the violation of the green list rule. Subsequently, Zhao et al. (2023a) simplifies the scheme by consistently using a fixed green-red list split, showing that the new watermark persists in guaranteed generation quality and is more robust against text editing. Kuditipudi et al. (2023) create watermarks that are distortion-free by utilizing randomized watermark keys to sample from token probability distribution by inverse transform sampling and exponential minimum sampling. Hou et al. (2023) propose a sentence-level semantic watermark based on locality-sensitive hashing (LSH), which partitions the semantic space of sentences. The advantage of this design is its enhanced robustness against paraphrasing attacks. DiPmark (Wu et al., 2023b) is an unbiased distribution-preserving watermark that preserves the original token distribution during watermarking and is robust to moderate changes of tokens by incorporating a novel reweight strategy, combined with a hash function that assigns unique i.i.d. ciphers based on the context. Drawn on the drawbacks of random green-red list splitting, Fu et al. (2023) uses input sequence to get semantically related tokens for watermarking to improve certain conditional generation tasks.

Despite training-free watermarking, text watermarks can also be injected through pre-inference training or post-inference training: (2) **training-based watermark**. One example of pre-inference training is REMARK-LLM (Zhang et al., 2023b), which injects the watermark by a message encoding module to generate a dense token distribution, following a message decoding module to extract messages from the watermarked textual and reparameterization is used as a bridge to connect the dense distribution with tokens' one-hot encoding. The drawback is that training is required on source data and might not generalize well to unseen text data. On the contrary, post-inference training involves adding a trained module to assist in injecting watermarks during inference. For instance, Liu et al. (2023a) proposes a semantic invariant robust watermark for LLMs, by utilizing another embedding LLM to generate semantic embeddings for all preceding tokens. However, it is not training-free since these semantic embeddings are transformed into the watermark logits through their trained watermark model.

Despite from 0-bit watermark, there is also (3) **multi-bit watermarking**. For example, Yoo et al. (2023a) designs a multi-bit watermarking following a well-known proposition from image watermarking that identifies natural language features invariant to minor corruption and proposes a corruption-resistant infill model. COLOR (Yoo et al., 2023b) subsequently designs another multi-bit watermark by embedding traceable multi-bit information during language model generation while allowing zero-bit detection simultaneously. Fernandez et al. (2023) also consolidates watermarks for LLMs through more robust statistical tests and multi-bit watermarking.

5 Attack, Challenges, Future Outlook

The detection of LLM-generated content is an evolving field. Detection attacks can be found in Appendix A.2 and we also summarize the challenges in Appendix A.3. Additionally, we list some potential avenues for future work (details are included in Appendix A.8): 1). robust and scalable detection techniques; 2). rigorous and standard evaluation; 3). fine-grained detection; 4). user education and awareness; 5). transparency and explainability.

6 Conclusion

We comprehensively survey LLMs-generated content detection over existing task formulation, benchmark datasets, evaluation metrics, and different detection methods to help the research community quickly learn the progress in this field.

Limitations

Despite conducting a comprehensive literature review on AI-generated content detection, we acknowledge the potential for omissions due to incomplete searches.

Ethics Statement

The utilization of AI detection presents significant ethical considerations, particularly when it comes to the detection of plagiarism among students. Misclassifications in this context can give rise to substantial concerns. This survey aims to summarize the current techniques employed in this field comprehensively. However, it is important to note that no flawless detectors have been developed thus far. Consequently, users should exercise caution when interpreting the detection outcomes, and it should be understood that we cannot be held accountable for any inaccuracies or errors that may arise.

References

Scott Aaronson. 2022. My ai safety lecture for ut effective altruism. *Shtetl-Optimized: The blog of Scott Aaronson. Retrieved on January*, 11:2023.

Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *42nd IEEE Symposium on Security and Privacy.*

Anirudh Ajith, Sameer Singh, and Danish Pruthi. 2023. Performance trade-offs of watermarking large language models. *arXiv preprint arXiv:2311.09816.*

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998.*

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

Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*, pages 185–200. Springer.

Mikhail J Atallah, Victor Raskin, Christian F Hempelmann, Mercan Karahan, Radu Sion, Umut Topkara, and Katrina E Triezenberg. 2003. Natural language watermarking and tamperproofing. In *Information Hiding: 5th International Workshop, IH 2002 Noordwijkerhout, The Netherlands, October 7-9, 2002 Revised Papers 5*, pages 196–212. Springer.

Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II.*

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351.*

Guangsheng Bao, 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.*

Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. 2023. Paraphrase detection: Human vs. machine content. *arXiv preprint arXiv:2303.13989.*

Daria Beresneva. 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, Salford, UK, June 22-24, 2016, Proceedings 21*, pages 421–426. Springer.

Rahul Bhagat and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. Conda: Contrastive domain adaptation for ai-generated text detection. *ArXiv*, abs/2309.03992.

Ning Bian, Peilin Liu, Xianpei Han, Hongyu Lin, Yaojie Lu, Ben He, and Le Sun. 2023. A drop of ink may make a million think: The spread of false information in large language models. *arXiv preprint arXiv:2305.04812.*

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Megha Chakraborty, S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Krish Sharma, Niyar R Barman, Chandan Gupta, Shreya Gautam, Tanay Kumar, Vinija Jain, Aman Chadha, Amit P. Sheth, and Amitava Das. 2023a. Counter turing test ct$\hat{2}$: Ai-generated text detection is not as easy as you may think – introducing ai detectability index.

Souradip Chakraborty, Amrit Singh Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2023b. On the possibilities of ai-generated text detection. *arXiv preprint arXiv:2304.04736.*

Liang Chen, Yatao Bian, Yang Deng, Shuaiyi Li, Bingzhe Wu, Peilin Zhao, and Kam-fai Wong. 2023a. X-mark: Towards lossless watermarking through lexical redundancy. *arXiv preprint arXiv:2311.09832*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Shu Chen, Zeqian Ju, Xiangyu Dong, Hongchao Fang, Sicheng Wang, Yue Yang, Jiaqi Zeng, Ruisi Zhang, Ruoyu Zhang, Meng Zhou, et al. 2020. Meddialog: a large-scale medical dialogue dataset. *arXiv preprint arXiv:2004.03329*.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023b. Token prediction as implicit classification to identify llm-generated text. *arXiv preprint arXiv:2311.08723*.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Ramakrishnan. 2023c. Gpt-sentinel: Distinguishing human and chatgpt generated content. *arXiv preprint arXiv:2305.07969*.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable watermarks for language models. *Cryptology ePrint Archive*.

Evan Crothers, Nathalie Japkowicz, and Herna Viktor. 2022. Machine generated text: A comprehensive survey of threat models and detection methods. *arXiv preprint arXiv:2210.07321*.

Wanyun Cui, Linqiu Zhang, Qianle Wang, and Shuyang Cai. 2023. Who said that? benchmarking social media ai detection. *arXiv preprint arXiv:2310.08240*.

Zhijie Deng, Hongcheng Gao, Yibo Miao, and Hao Zhang. 2023. Efficient detection of llm-generated texts with a bayesian surrogate model. *arXiv preprint arXiv:2305.16617*.

Heather Desaire, 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. *arXiv preprint arXiv:2303.16352*.

Jacob Devlin, 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mahdi Dhaini, Wessel Poelman, and Ege Erdogan. 2023. Detecting chatgpt: A survey of the state of detecting chatgpt-generated text. *arXiv preprint arXiv:2309.07689*.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2022. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. *arXiv preprint arXiv:2212.12672*.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. 2023. Publicly detectable watermarking for language models. *arXiv preprint arXiv:2310.18491*.

Angela Fan, 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, Florence, Italy. Association for Computational Linguistics.

Tina Fang, Martin Jaggi, and Katerina Argyraki. 2017. Generating steganographic text with LSTMs. In *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada. Association for Computational Linguistics.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three bricks to consolidate watermarks for large language models. *arXiv preprint arXiv:2308.00113*.

Yu Fu, Deyi Xiong, and Yue Dong. 2023. Watermarking conditional text generation for ai detection: Unveiling challenges and a semantic-aware watermark remedy. *arXiv preprint arXiv:2307.13808*.

Sebastian Gehrmann, SEAS Harvard, Hendrik Strobelt, and Alexander M Rush. 2019a. Gltr: Statistical detection and visualization of generated text. *ACL 2019*, page 111.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019b. 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, Florence, Italy. Association for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2023. Openwebtext corpus, 2019. *URL http://Skylion007. github. io/OpenWebTextCorpus*.

EA Grechnikov, GG Gusev, AA Kustarev, and AM Raigorodsky. 2009. Detection of artificial texts. *RCDL2009 Proceedings. Petrozavodsk*, pages 306–308.

Alexei Grinbaum and Laurynas Adomaitis. 2022. The ethical need for watermarks in machine-generated language. *arXiv preprint arXiv:2209.03118*.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. 2023. On the learnability of watermarks for language models. *arXiv preprint arXiv:2312.04469*.

Biyang Guo, 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 arXiv:2301.07597*.

Zhen Guo and Shangdi Yu. 2023. Authentigpt: Detecting machine-generated text via black-box language models denoising. *arXiv preprint arXiv:2311.07700*.

Philipp Hacker, Andreas Engel, and Marco Mauer. 2023. Regulating chatgpt and other large generative ai models. *arXiv preprint arXiv:2302.02337*.

Hans WA Hanley and Zakir Durumeric. 2023. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. *arXiv preprint arXiv:2305.09820*.

Abhimanyu Hans, 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. *arXiv preprint arXiv:2401.12070*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. 2022a. Protecting intellectual property of language generation apis with lexical watermark. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. 2022b. Cater: Intellectual property protection on text generation apis via conditional watermarks. *Advances in Neural Information Processing Systems*, 35:5431–5445.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. 2021. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*.

Steffen Herbold, Annette Hautli-Janisz, Ute Heuer, Zlata Kikteva, and Alexander Trautsch. 2023. Ai, write an essay for me: A large-scale comparison of human-written versus chatgpt-generated essays. *arXiv preprint arXiv:2304.14276*.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. Semstamp: A semantic watermark with paraphrastic robustness for text generation. *arXiv preprint arXiv:2310.03991*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *arXiv preprint arXiv:2307.03838*.

Baihe Huang, Banghua Zhu, Hanlin Zhu, Jason D Lee, Jiantao Jiao, and Michael I Jordan. 2023. Towards optimal statistical watermarking. *arXiv preprint arXiv:2312.07930*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Daphne Ippolito, 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, Online. Association for Computational Linguistics.

Nafis Irtiza Tripto, Saranya Venkatraman, Dominik Macko, Robert Moro, Ivan Srba, Adaku Uchendu, Thai Le, and Dongwon Lee. 2023. A ship of theseus: Curious cases of paraphrasing in llm-generated texts. *arXiv e-prints*, pages arXiv–2311.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Ganesh Jawahar, 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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kaan Efe Keleş, Ömer Kaan Gürbüz, and Mucahid Kutlu. 2023. I know you did not write that! a sampling based watermarking method for identifying machine generated text. *arXiv preprint arXiv:2311.18054*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.

John Kirchenbauer, 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.

Tomáš Kočiský, 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.

Kalpesh Krishna, 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 arXiv:2303.13408*.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust distortion-free watermarks for language models. *ArXiv*, abs/2307.15593.

Sachin Kumar, Vidhisha Balachandran, Lucille Njoo, Antonios Anastasopoulos, and Yulia Tsvetkov. 2023. Language generation models can cause harm: So what can we do about it? an actionable survey. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3299–3321, Dubrovnik, Croatia. Association for Computational Linguistics.

Tharindu Kumarage, Paras Sheth, Raha Moraffah, Joshua Garland, and Huan Liu. 2023. How reliable are ai-generated-text detectors? an assessment framework using evasive soft prompts. *arXiv preprint arXiv:2310.05095*.

Martin Kutter, S. Andpetitcolas, and Olympia Nikolaeva Roeva. 2000. Information hiding: Techniques for steganography and digital watermarking. In *Artech House Books*.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. *PAN*, 8:27–31.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2023. Who wrote this code? watermarking for code generation. *arXiv preprint arXiv:2305.15060*.

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. 2023a. Origin tracing and detecting of llms. *arXiv preprint arXiv:2304.14072*.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023b. Deepfake text detection in the wild. *arXiv preprint arXiv:2305.13242*.

Yuhang Li, Yihan Wang, Zhouxing Shi, and Cho-Jui Hsieh. 2023c. Improving the generation quality of watermarked large language models via word importance scoring. *arXiv preprint arXiv:2311.09668*.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. 2022. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *arXiv preprint arXiv:2304.02819*.

Stephanie Lin, 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, Dublin, Ireland. Association for Computational Linguistics.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023a. A semantic invariant robust watermark for large language models.

Shengchao Liu, Xiaoming Liu, Yichen Wang, Zehua Cheng, Chengzhengxu Li, Zhaohan Zhang, Yu Lan, and Chao Shen. 2024. Does\textsc {DetectGPT} fully utilize perturbation? selective perturbation on model-based contrastive learning detector would be better. *arXiv preprint arXiv:2402.00263*.

Xiaoming Liu, 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 arXiv:2212.10341*.

Yepeng Liu and Yuheng Bu. 2024. Adaptive text watermark for large language models. *arXiv preprint arXiv:2401.13927*.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023b. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models. *arXiv preprint arXiv:2304.07666*.

Yinhan Liu, 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. *arXiv preprint arXiv:1907.11692*.

Ning Lu, Shengcai Liu, Rui He, and Ke Tang. 2023. Large language models can be guided to evade ai-generated text detection. *arXiv preprint arXiv:2305.10847*.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Samuel Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. Multitude: Large-scale multilingual machine-generated text detection benchmark.

Dominik Macko, Robert Moro, Adaku Uchendu, Ivan Srba, Jason Samuel Lucas, Michiharu Yamashita, Nafis Irtiza Tripto, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2024. Authorship obfuscation in multilingual machine-generated text detection. *arXiv preprint arXiv:2401.07867*.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pages 1941–1942.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative ai detection via rewriting. *arXiv preprint arXiv:2401.12970*.

Hasan Mesut Meral, Bülent Sankur, A Sumru Özsoy, Tunga Güngör, and Emre Sevinç. 2009. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language*, 23(1):107–125.

Fatemehsadat Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2023. Smaller language models are better black-box machine-generated text detectors. *arXiv preprint arXiv:2305.09859*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

OpenAI. 2023a. AI text classifier.

OpenAI. 2023b. Gpt-4 technical report.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the risk of misinformation pollution with large language models. *arXiv preprint arXiv:2305.13661*.

Xinlin Peng, Ying Zhou, Ben He, Le Sun, and Yingfei Sun. 2024. Hidding the ghostwriters: An adversarial evaluation of ai-generated student essay detection. *arXiv preprint arXiv:2402.00412*.

Mike Perkins, Jasper Roe, Darius Postma, James McGaughran, and Don Hickerson. 2023. Game of tones: Faculty detection of gpt-4 generated content in university assessments. *arXiv preprint arXiv:2305.18081*.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. 2023. Mark my words: Analyzing and evaluating language model watermarks. *arXiv preprint arXiv:2312.00273*.

Yury Polyanskiy and Yihong Wu. 2022. Information theory: From coding to learning.

Xiao Pu, Jingyu Zhang, Xiaochuang Han, Yulia Tsvetkov, and Tianxing He. 2023. On the zero-shot generalization of machine-generated text detectors. *arXiv preprint arXiv:2310.05165*.

Wenjie Qu, Dong Yin, Zixin He, Wei Zou, Tianyang Tao, Jinyuan Jia, and Jiaheng Zhang. 2024. Provably robust multi-bit watermarking for ai-generated text via error correction code. *arXiv preprint arXiv:2401.16820*.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Colin Raffel, 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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, 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, Austin, Texas. Association for Computational Linguistics.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

J Schulman, B Zoph, C Kim, J Hilton, J Menick, J Weng, JFC Uribe, L Fedus, L Metz, M Pokorny, et al. 2022. Chatgpt: Optimizing language models for dialogue.

Yuhui Shi, Qiang Sheng, Juan Cao, Hao Mi, Beizhe Hu, and Danding Wang. 2024. Ten words only still help: Improving black-box ai-generated text detection via proxy-guided efficient re-sampling.

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

Alejo Jose G Sison, Marco Tulio Daza, Roberto Gozalo-Brizuela, and Eduardo C Garrido-Merchán. 2023. Chatgpt: More than a weapon of mass deception, ethical challenges and responses from the human-centered artificial intelligence (hcai) perspective. *arXiv preprint arXiv:2304.11215*.

Irene Solaiman, 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 arXiv:1908.09203*.

Rafael Rivera Soto, Kailin Koch, Aleem Khan, Barry Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. *arXiv preprint arXiv:2401.06712*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023a. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text.

Zhenpeng Su, Xing Wu, Wei Zhou, Guangyuan Ma, and Songlin Hu. 2023b. Hc3 plus: A semantic-invariant human chatgpt comparison corpus. *arXiv preprint arXiv:2309.02731*.

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy. Association for Computational Linguistics.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*.

Rohan Taori, 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`.

Edward Tian. 2023. Gptzero: An ai text detector.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. 2023. Multiscale positive-unlabeled detection of ai-generated texts. *arXiv preprint arXiv:2305.18149*.

Mercan Topkara, Cuneyt M Taskiran, and Edward J Delp III. 2005. Natural language watermarking. In *Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 441–452. SPIE.

Umut Topkara, 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 and security*, pages 164–174.

Shangqing Tu, Chunyang Li, Jifan Yu, Xiaozhi Wang, Lei Hou, and Juanzi Li. 2023a. Chatlog: Recording and analyzing chatgpt across time. *arXiv preprint arXiv:2304.14106*.

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. 2023b. Waterbench: Towards holistic evaluation of watermarks for large language models. *arXiv preprint arXiv:2311.07138*.

Eduard Tulchinskii, 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.

Adaku Uchendu, 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, Online. Association for Computational Linguistics.

Adaku Uchendu, 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, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Varab and Natalie Schluter. 2021. MassiveSumm: a very large-scale, very multilingual, news summarisation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10150–10161, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Christoforos Vasilatos, 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 arXiv:2305.18226*.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models.

Jan Philip Wahle, Terry Ruas, Saif M Mohammad, Norman Meuschke, and Bela Gipp. 2023. Ai usage cards: Responsibly reporting ai-generated content. *arXiv preprint arXiv:2303.03886*.

Jian Wang, Shangqing Liu, Xiaofei Xie, and Yi Li. 2023a. Evaluating aigc detectors on code content. *arXiv preprint arXiv:2304.05193*.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023b. Seqxgpt: Sentence-level ai-generated text detection.

Rongsheng Wang, Haoming Chen, Ruizhe Zhou, Han Ma, Yaofei Duan, Yanlan Kang, Songhua Yang, Baoyu Fan, and Tao Tan. 2024. Llm-detector: Improving ai-generated chinese text detection with open-source llm instruction tuning. *arXiv preprint arXiv:2402.01158*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Alham Fikri Aji, and Preslav Nakov. 2023c. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection.

Luoxuan Weng, 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 arXiv:2304.05011*.

Bram Wouters. 2023. Optimizing watermarks for large language models. *arXiv preprint arXiv:2312.17295*.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023a. Llmdet: A large language models detection tool.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2023b. Dipmark: A stealthy, efficient and resilient watermark for large language models.

Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. 2023a. Watermarking text generated by black-box language models. *arXiv preprint arXiv:2305.08883*.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. 2022. Tracing text provenance via context-aware lexical substitution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11613–11621.

Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023b. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 2023c. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*.

Xianjun Yang, Kexun Zhang, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023d. Zero-shot detection of machine-generated codes. *arXiv preprint arXiv:2310.05103*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. 2023a. 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)*, pages 2092–2115, Toronto, Canada. Association for Computational Linguistics.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2023b. Advancing beyond identification: Multi-bit watermark for language models. *arXiv preprint arXiv:2308.00221*.

Peipeng Yu, Jiahan Chen, Xuan Feng, and Zhihua Xia. 2023a. Cheat: A large-scale dataset for detecting chatgpt-written abstracts. *arXiv preprint arXiv:2304.12008*.

Xiao Yu, 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 arXiv:2305.12519*.

Daoguang Zan, Bei Chen, Fengji Zhang, Dianjie Lu, Bingchao Wu, Bei Guan, Yongji Wang, and Jian-Guang Lou. 2022. When neural model meets nl2code: A survey. *arXiv preprint arXiv:2212.09420*.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Haolan Zhan, Xuanli He, Qiongkai Xu, Yuxiang Wu, and Pontus Stenetorp. 2023. G3detector: General gpt-generated text detector. *arXiv preprint arXiv:2305.12680*.

Hanlin Zhang, Benjamin L Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. 2023a. Watermarks in the sand: Impossibility of strong watermarking for generative models. *arXiv preprint arXiv:2311.04378*.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. 2023b. Remark-llm: A robust and efficient watermarking framework for generative large language models. *arXiv preprint arXiv:2310.12362*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. 2023a. Provable robust watermarking for ai-generated text. *ArXiv*, abs/2306.17439.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. 2022. Distillation-resistant watermarking for model protection in NLP. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5044–5055, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. 2024. Permute-and-flip: An optimally robust and watermarkable decoder for llms. *arXiv preprint arXiv:2402.05864*.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023b. Protecting language generation models via invisible watermarking. In *International Conference on Machine Learning*.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

Zachary Ziegler, Yuntian Deng, and Alexander Rush. 2019. Neural linguistic steganography. 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 1210–1215, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Commercial Detection Tool

Despite from academic research, AI text detection also draws considerable attention from commercial companies. Table 1 summarizes the popular commercial detectors. Although the majority of them simultaneously claim to be the most accurate AI detectors on the homepage of their website, it is essential to evaluate their performance based on various factors such as accuracy, speed, robustness, and compatibility with different platforms and frameworks. Regrettably, a dearth of articles exists that explicitly delve into the comparative analysis of the aforementioned properties among popular commercial detectors.

A.2 Detection Attack

Despite the progress of detection work, there are also continuous efforts to evade existing detectors, and we summarize the main streams in this section.

A.2.1 Paraphrasing Attack

Paraphrasing could be performed by human writers or other LLMs, and even by the same source model. Paraphrasing can also undergo several rounds, influenced by a mixture of different models. Current research mostly focuses on the simple paraphrase case where another model rewrites a machine-generated text for one round. For instance, Krishna et al. (2023) trains a T5-11b model for paraphrasing text and discovers that all detectors experience a significant drop in quality when faced with paraphrased text. Additionally, simple paraphrasing attacks involve word substitutions (Shi et al., 2023). Moreover, paraphrasing can also be achieved through translation attacks. However, conducting more in-depth analysis and research on complex paraphrasing techniques in the future is crucial. Becker et al. (2023) systemically examines different classifiers encompassing both classical approaches and Transformer techniques for detecting machine (like T5) or human paraphrased text.

A.2.2 Adversarial Attack

Though the adversarial attack is popular for general NLP tasks (Alzantot et al., 2018), there has been little work specifically addressing adversarial attacks on detectors for LLM-generated content. However, we can consider the following two types of attacks for further investigation and exploration:

Adversarial Examples: Attackers can generate specially crafted inputs by making subtle modifications to the text that fool the AI text detectors while remaining mostly unchanged to human readers (Shi et al., 2023). These modifications can include adding or removing certain words or characters, introducing synonyms, or leveraging linguistic tricks to deceive the detector. Evasion attacks aim to manipulate the AI text detector's behavior by exploiting its vulnerabilities. Attackers can use techniques such as obfuscation, word permutation, or introducing irrelevant or misleading content to evade detection. The goal is to trigger false negatives and avoid being flagged as malicious or inappropriate.

Model Inversion Attacks: Attackers can launch model inversion attacks by exploiting the responses of AI text detectors. They might submit carefully crafted queries and observe the model's responses to gain insights into its internal workings, architecture, or training data, which can be used to create more effective attacks or subvert the system's defenses.

A.2.3 Prompt Attack

Current LLMs are vulnerable to prompts (Zhu et al., 2023), thus, users can utilize smartly designed prompts to evade established detectors. For example, Shi et al. (2023) examines instructional prompt attacks by perturbing the input prompt to encourage LLMs to generate texts that are difficult to detect. Lu et al. (2023) also show that LLMs can be guided to evade AI-generated text detection by a novel substitution-based In-Context example Optimization method (SICO) to automatically generate carefully crafted prompts, enabling ChatGPT to evade six existing detectors by a significant 0.54 AUC drop on average. Nevertheless, limited attention has been devoted to this topic, indicating a notable research gap that merits significant scholarly exploration in the immediate future. Notably, a recent work (Chakraborty et al., 2023a) introduces the Counter Turing Test (CT2), a benchmark consisting of techniques aiming to evaluate the robustness of existing six detection techniques comprehensively. Their empirical findings unequivocally highlight the fragility of almost all the proposed detection methods under scrutiny. Despite the hard prompt attack, Kumarage et al. (2023) first creates an evasive soft prompt tailored to a specific PLM through prompt tuning; and then, they leverage the transferability of soft prompts to transfer the

Product Name	Website	Price	API available
Originality.AI	https://app.originality.ai/api-access	$0.01/100 words	Yes
Quil.org	https://aiwritingcheck.org/	Free website version	No
Sapling	https://sapling.ai/ai-content-detector	1 million chars at $25/month	Yes
OpenAI text classifier	https://openai-openai-detector.hf.space/	Free website version	Yes
Crossplag	https://crossplag.com/ai-content-detector/	Free website version	No
GPTZero	https://gptzero.me/	0.5 million words at $14.99/mo	Yes
ZeroGPT	https://www.zerogpt.com/	Free website version	No
CopyLeaks	https://copyleaks.com/ai-content-detector	25000 words at $10.99/Month	No

Table 1: A summary of popular commercial tools to detect AI-generated text.

learned evasive soft prompt from one PLM to another and find the universal efficacy of the evasion attack.

A.3 Challenges

A.3.1 Theorical Analysis

Inspired by the binary hypothesis test in (Polyanskiy and Wu, 2022), (Sadasivan et al., 2023) claims that machine-generated text will become indistinguishable as the total variance between the distributions of human and machine approaches zero. In contrast, Chakraborty et al. (2023b) demonstrates that it is always possible to distinguish them by curating more data to make the detection of AUROC increase exponentially with the number of training instances. Additionally, DNA-GPT (Yang et al., 2023b) demonstrates the difficulty of obtaining a high TPR while maintaining a low FPR. Nevertheless, a dearth of theoretical examination persists regarding the disparities in intrinsic characteristics between human-written language and LLMs. Scholars could leverage the working mechanisms of GPT models to establish a robust theoretical analysis, shedding light on detectability and fostering the development of additional detection algorithms.

A.3.2 LLM-Generated Code Detection

Previous detectors usually only focus on the text, but LLMs-generated codes also show increasing quality (see a recent survey (Zan et al., 2022)). Among the first, Lee et al. (2023) found that previous watermarking (Kirchenbauer et al., 2023a) for text does not work well in terms of both detectability and generated code quality. It is evidenced that low entropy persists in generated code (Lee et al., 2023), thus, the decoding process is more deterministic. They thus adapt the text watermarks to code generation by only injecting watermarks to tokens with higher entropy than a given threshold and achieve more satisfactory results. Code detection is generally believed to be even harder than text detection due to its shorter length, low entropy, and non-natural language properties. DetectGPT4Code (Yang et al., 2023d) detects codes generated by ChatGPT by using a proxy code model to approximate the logits on the conditional probability curve and achieves the best results over previous detectors.

A.3.3 Model Sourcing

Model sourcing (Yang et al., 2023b), is also known as origin tracking (Li et al., 2023a) or authorship attribution (Uchendu et al., 2020). Unlike the traditional distinction between human and machine-generated texts, it focuses on identifying the specific source model from a pool of models, treating humans as a distinct model category. With the fast advancement of LLMs from different organizations, it is vital to tell which model or organization potentially generates a certain text. This has practical applications, particularly for copyright protection. Consequently, we believe that in the future, it may become the responsibility of organizations releasing powerful LLMs to determine whether a given text is a product of their system. Previous work either (Li et al., 2023a) trains a classifier or utilizes the intrinsic genetic properties (Yang et al., 2023b) to perform model sourcing, but still can not handle more complicated scenarios. GPT-who (Venkatraman et al., 2023) utilizes Uniform Information Density (UID) based features to model the unique statistical signature of each LLM and human author for accurate authorship attribution.

A.3.4 Bias

It has been found that current detectors tend to be biased against non-native speakers (Liang et al., 2023). Also, Yang et al. (2023b) found that previous detection tools often perform poorly on other languages other than English. Besides, current research usually focuses on the detection of text within a certain length, thus showing bias against the shorter text. How to ensure the integrity of detectors under various scenarios without showing bias against certain groups is of central importance.

A.3.5 Generalization

Currently, the most advanced LLMs, like ChatGPT, are getting actively updated, and OpenAI will make a large update every three months. How to effectively adapt existing detectors to the updated LLMs is of great importance. For example, Tu et al. (2023a) records the ChatLog of ChatGPT's response to long-form generation every day in one month, observes performance degradation of the Roberta-based detector, and also finds some stable features to improve the robustness of detection. As LLMs continuously benefit from interacting with different datasets and human feedback, exploring ways to effectively and efficiently detect their generations remains an ongoing research area. Additionally, Kirchenbauer et al. (2023b) investigates the reliability of watermarks for large language models and claims that watermarking is a reliable solution under human paraphrasing and various attacks at the context length of around 1000. Pu et al. (2023) examines the zero-shot generalization of machine-generated text detectors and finds that none of the detectors can generalize to all generators. All those findings reveal the difficulty of reliable generalization to unseen models or data sources of detection.

A.4 News Reports

We summarize several influential news on the false use of AI detectors and concerns brought by AI-generated information.

1. International students are concerned their original writing is being flagged as AI-generated text. link

2. Professor Flunks All His Students After ChatGPT Falsely Claims It Wrote Their Papers. link

3. China reports first arrest over fake news generated by ChatGPT. link

4. Professors have a summer assignment: Prevent ChatGPT chaos in the fall. link

5. AI makes plagiarism harder to detect, argue academics – in paper written by chatbot. link

6. How AI Could Take Over Elections—And Undermine Democracy. link

A.5 Related Survey

In the literature, there are some other surveys on this topic. For example, Jawahar et al. (2020) discusses the detection of small language models. Tang et al. (2023) provides an overview of previous detection methods but does not fully cover the recent progress in the era of LLMs. Very recently, Crothers et al. (2022) surveys threat models and detection methods but also summarizes previous detection methods rather than the latest progress with LLMs. Unlike them, our work aims to fill this gap by providing the first comprehensive survey about detection, attack, and benchmarks, especially focusing on detecting LLMs like ChatGPT. Thus, our survey includes the most advanced approaches.

Dhaini et al. (2023) gives a survey of the state of detecting only ChatGPT-Generated text but ignores various detection methods on other models.

A.6 Additional Latest Work

Very recently, there have been some additional work released very close to our submission, including watermarking methods (Fairoze et al., 2023; Tu et al., 2023b; Chen et al., 2023a; Ajith et al., 2023; Zhang et al., 2023a; Li et al., 2023c; Keleş et al., 2023; Piet et al., 2023; Gu et al., 2023; Huang et al., 2023; Zhao et al., 2024; Qu et al., 2024; Liu and Bu, 2024; Wouters, 2023), training-based methods (Chen et al., 2023b; Guo and Yu, 2023; Wang et al., 2024; Soto et al., 2024), zero-shot methods (Mao et al., 2024; Hans et al., 2024; Shi et al., 2024; Liu et al., 2024), attacks (Irtiza Tripto et al., 2023; Macko et al., 2024; Peng et al., 2024).

A.7 Datasets

- Uchendu et al. (2021) presents the TURINGBENCH benchmark for Turing Test and Authorship Attribution across 19 language models.
- HC3 (Guo et al., 2023) collectes the Human ChatGPT Comparison Corpus (HC3) with both long- and short-level documents from ELI5 (Fan et al., 2019), WikiQA (Yang et al., 2015), Crawled Wikipedia, Medical Dialog (Chen et al., 2020), and FiQA (Maia et al., 2018).
- CHEAT (Yu et al., 2023a) provides 35,304 synthetic academic abstracts, with Generation, Polish, and Mix as prominent representatives.
- Ghostbuster (Verma et al., 2023) provides a detection benchmark that covers student essays, creative fiction, and news at document-level detection and paragraph-level.
- OpenGPTText (Chen et al., 2023c) consists of 29,395 rephrased content generated using ChatGPT, originating from OpenWebText (Gokaslan and Cohen, 2019).

Datasets	Length	Size	Data type	#Language
TuringBench (2021)	100∼400	200K	News articles	1
HC3 (2023)	100∼250	44,425	Reddit, Wikipedia, medicine and finance	2
CHEAT (2023a)	100∼300	35,304	Academical abstracts	1
Ghostbuster (2023)	200∼1200	12,685	Student essays, creative fiction, and news	1
GPT-Sentinel (2023c)	100∼400	29,395	OpenWebText (2023)	1
M4 (2023c)	200-300	122,481	Multi-domains	6
MGTBench (2023)	10∼200	2,817	Question-answering datasets	1
Deepfake (2023b)	∼264	447,674	Multi-domains	1
HC3 Plus (2023b)	100∼250	214,498	Summarization, translation, and paraphrasing	2
MULTITuDE (2023)	150∼400	74,081	MassiveSumm (2021)	11
HumanEval (2021)	∼181	164	Code Exercise	1
APPS (2021)	∼474	5,000	Code Competitions	1
CodeContests (2022)	∼2239	165	Code Competitions	6

Table 2: A summarization of the detection datasets. Length is reported in the number of words for text and characters for codes. #Language represents the number of types of natural languages for text and programming languages for codes.

- M4 (Wang et al., 2023c) is a large-scale benchmark covering multi-generator, multi-domain, and multi-lingual corpus for machine-generated text detection.
- MULTITuDE (Macko et al., 2023) Large-Scale Multilingual Machine-Generated Text Detection Benchmark comprising 74,081 authentic and machine-generated texts in 11 languages generated by 8 multilingual LLMs. They find that the most currently available black-box methods do not work in multilingual settings.
- MGTBench (He et al., 2023) focuses on ChatGPT-generated content on: TruthfulQA (Lin et al., 2022), SQuAD (Rajpurkar et al., 2016) and NarrativeQA (Kočiský et al., 2018).
- SAID(Social media AI Detection) Cui et al. (2023) is curated for real AI-generate text from popular social media platforms like Zhihu and Quora, and conducting detection tasks on actual social media platforms prove to be more challenging compared to traditional simulated AI-text detection.
- HC3 Plus (Su et al., 2023b) is a more extensive and comprehensive dataset that considers more types of tasks, considering tasks such as summarization, translation, and paraphrasing to possess semantic-invariant properties and are more difficult to detect.

We summarize them in Table 2.

A.8 Future Outlooks

Details on the future outlook are as follows.

- *Robust and Scalable Detection Techniques*: Current LLMs are getting constant improvements from big tech companies. Thus, the development of advanced algorithms and detection techniques capable of accurately identifying LLM-generated content in real time is a priority. Future research should focus on improving the accuracy, robustness to attacks, and scalability of detection methods to keep up with the increasing volume and complexity of LLM-generated content.
- *Rigorous and Standard Evaluation*: As discussed in Section 2.3, current evaluation faces data contamination issues; either the LLMs or the detectors might encounter the human data in their training stage. Besides, the evaluation benchmark also varies. The detection results affect the length, prompting methods, and adopted datasets. However, unlike traditional machine learning tasks where one benchmark can be used for a long period, how to avoid any potential data contamination is very critical.
- *Fine-grained Detection*: LLM-generated content can vary in its intentions, ranging from malicious propaganda to unintentional misinformation. Future work should explore approaches that can detect and differentiate between various categories of LLM-generated content, allowing for more tailored interventions and countermeasures.
- *User Education and Awareness*: Educating users about the existence and capabilities of LLMs detectors is essential. For example, in Appendix A.4, we show some reported misuse of AI detectors in education. Future work should focus on raising awareness among users about the reliability of detection tools. This can empower users to make more informed decisions and mitigate the

impact of deceptive or misleading decisions.

- *AI Regulations*: As LLMs become more sophisticated, the ethical implications of their usage in generating deceptive content become increasingly important. Future research should explore ethical frameworks and guidelines for the responsible development and deployment of LLMs while considering the potential consequences and risks associated with their misuse.
- *Transparency and Explainability*: Enhancing the transparency and explainability of LLM-generated content detection algorithms is crucial for building trust and understanding among users. For example, Yang et al. (2023b) uses the non-trivial N-gram overlaps to support the detection results. But currently, most detectors can only give a predictive probability, with no clues about evidence. Future work should focus on developing techniques that can provide explanations or evidence for the classification decisions made by detection systems, enabling users to understand the rationale behind content identification better.