ASJET

Al-Salam Journal for Engineering and Technology
Journal Homepage:
http://journal.alsalam.edu.iq/index.php/ajest
e-ISSN: 2790-4822 p-ISSN: 2958-0862

A Systematic Review of AI-Generated Text Detection: Approaches, Tools, and Datasets

Ahmed A. Alethary[1]* and Ahmed H. Aliwy[1]

[1]Department of Computer Science, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq .

*Corresponding Author :Ahmed A. Alethary

DOI: https://doi.org/10.55145/ajest.2026.05.01.012
Received December 2025; Accepted January 2026; Available online Febuary 2026

ABSTRACT: The rapid evolution of Large Language Models (LLMs) has enabled the generation of text that is increasingly indistinguishable from human writing. While this advancement benefits various sectors, it raises significant concerns regarding academic integrity, security, and the spread of misinformation. This paper presents a comprehensive systematic review of AI-Generated Text Detection (AIGTD) techniques, evaluating their current efficacy and limitations. We categorized and analyzed various detection methodologies, including statistical and stylometric approaches, transformer-based models, watermarking strategies, and hybrid frameworks. In addition, the analysis is covered 16 prominent datasets, such as HC3 and M4 for size, diversity, and limitations and domain bias, along with tools such as GPTZero, Originality.ai, and DetectGPT, which were compared on language support, usability, and detection principles. Our findings reveal that detection accuracy averages 80-99% on in-domain benchmarks but drops to 60-75% against adversarial attacks or cross-domain texts. Datasets often lack multilingual coverage and real-world diversity. Tools show high computational costs and biases toward English, with limited Arabic support. Hybrid methods outperforms singles but face scalability issues. Although the field has progressed, developing robust, unbiased, and computationally efficient systems is essential. This review concludes by proposing future research directions to enhance the reliability of detection systems in an era of advancing AI.

Keywords: AIGTD, Transformer Models, LLMs, Detection Methods, Dataset

1. INTRODUCTION

In recent years, generative AI models have grown tremendously, particularly LLMs like GPT [1], Llama [2], and T5 [3]. These systems can produce text that, in some cases, appears almost equally close to something that a human would write. Due to this remarkable skill, LLMs are being applied in various fields such as essay writing, automated conversations, coding help, and producing academic and educational content [4,5].

Despite their advantages, these models raise several serious concerns. Some of the biggest issues include security risks [6], fraudulent product reviews [7], trustworthiness of content [8], and academic integrity [9]. As these models continue to improve, it has become quite difficult and sometimes nearly impossible [10] for the average person to determine whether a human or a machine wrote a text. In response to this challenge, researchers are working tirelessly to develop superior text detection systems and algorithms.

In this regard, transformer-based models are not only powerful for language generation, but they also serve as the backbone for many detection approaches [11]. The power of these models comes from their ability to account for context, capture small details, and extract deep patterns [12]. Many modern detection tools rely on techniques, including Bidirectional Encoder Representations Transformer (BERT)[13], Robustly Optimized BERT Approach (RoBERTa)[14], Distilled version of BERT (DistilBERT)[15], and GPT, either through direct training (Fine-Tuning), using them as a probability generator (Perplexity-based), or extracting contextual representations.

Prior reviews on AI-generated text detection (AIGTD) provide broad overviews but lack a unified evaluation framework, comprehensive multilingual coverage (especially Arabic), structured taxonomies for method comparison, and standardized benchmarks for accuracy, fairness, and explainability [16,17].

*Corresponding author: *ahmeda.alethari@uokufa.edu.iq*

This study aims to conduct an in-depth and systematic analysis of AIGTD. It analyzes the architecture of the models, classifies the methods, discusses current tools and applications, reviews the datasets used, and addresses current and future challenges in this field related to these questions:

1. What are the main techniques used for detecting AI-generated text?
2. How are these methods evaluated, and what results have been reported so far?
3. What obstacles do current systems face, and what future directions are suggested in the literature?

Despite extensive efforts in this field, several challenges remain for detection tools, including: Reduced performance when dealing with new or improved generation models that detection tools have not been previously trained on [18]. Some detection systems can still be fooled rather easily. For instance, if the text is a little rephrased or edited. Performance in many other non-English languages also remains weak, notably Arabic, due to the limited size of the datasets and lack of other processing tools. The absence of a common and reliable evaluation standard makes it difficult to measure the accuracy and fairness of various detectors.

This work is different from the many prior reviews that provided a broad overview of AI text generation detection and offers instead a structured comparison with a more useful taxonomy. The techniques described in this article are divided into four main categories: statistical techniques, watermark-based approaches, transformer-based neural methods, and hybrid models. In addition, the review assembles all relevant tools and data into one location to point out the research gaps, which were not addressed in previous studies, and highlight new research directions.

The organization of this paper is as follows: Section 2 explains the technical background behind language models. Section 3 covers related work in the field. Section 4 describes the review methodology followed in this study. Section 5 presents the different detection techniques and explains how each category works. Section 6 describes the major datasets used for AIGTD and analyzes their features. Section 7 reviews existing detection tools, Section 8 analyzes and the discussion finally section 9 summarizes the key challenges still facing the field, and provides future research paths to help improve detection systems and enhance their usability across different languages and contexts.

.

2. THEORETICAL BACKGROUND

2.1 TRANSFORMER ARCHITECTURE FUNDAMENTALS

The architecture of the transformer, as introduced in the paper [19] "Attention Is All You Need" represents a paradigm shift from sequential processing artefacts to model artefacts based on attention that operate in parallel. The main innovation is the self-attention mechanism. The self-attention mechanism helps the model identify the significance of other parts as relevant while processing each part of the input. Figure 1 illustrates the architecture of the transformer

1- Self-Attention Mechanism: The self-attention mechanism calculates the attention weights for each position within a sequence of all other positions. Given an input sequence, the mechanism generates three matrices: query (Q), key (K), and value (V). The attention output is computed as:

$$\text{Attention } (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

where dk refers to the dimension of the key vector. This feature allows the model to effectively understand dependencies and relationships that are far apart.

2- Multi-head attention performs multiple self-attention functions in parallel, with each acting on a unique linear projection of the original input:

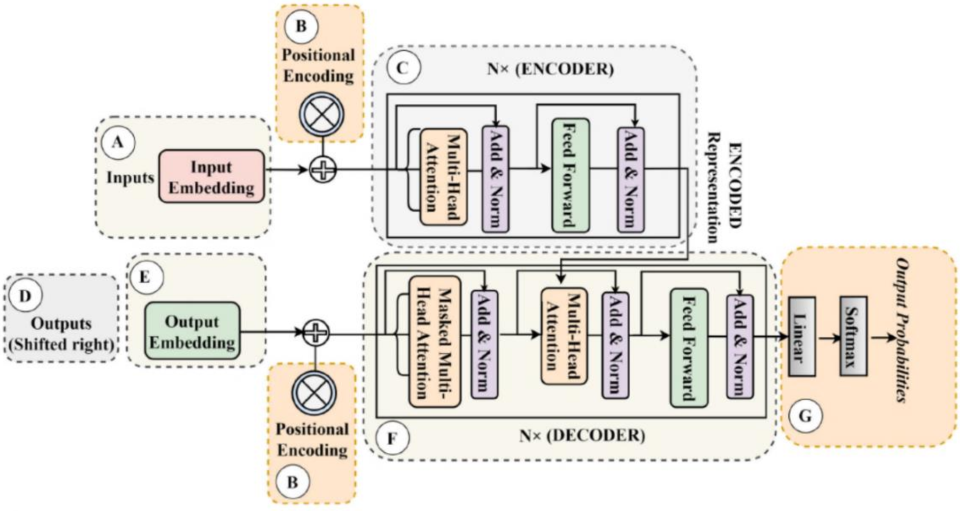

FIGURE 1 The Transformer model architecture [20]

$$\text{MultiHead}(\text{Q},\text{K},\text{V}) = \text{Concat}(head_1, \ldots, head_2)W^O \qquad (2)$$

where each is computed as

$$head_i = Attention(QW_i^Q, KW_i^k, VW_i^V) \qquad (3)$$

BERT is built entirely using the Transformer encoder architecture. It consists of a stack of N = 12 identical layers for the base model (N = 24 for BERT-large). Each layer comprises two primary sublayers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network. Residual connections are applied around each sublayer, followed by layer normalization. The output of each sublayer is computed as LayerNorm(x + Sublayer(x)), where Sublayer(x) represents the operation executed by the sublayer itself. Figure 2 illustrates the overall BERT encoder architecture, highlighting its stacked layers of self-attention and feed-forward networks.

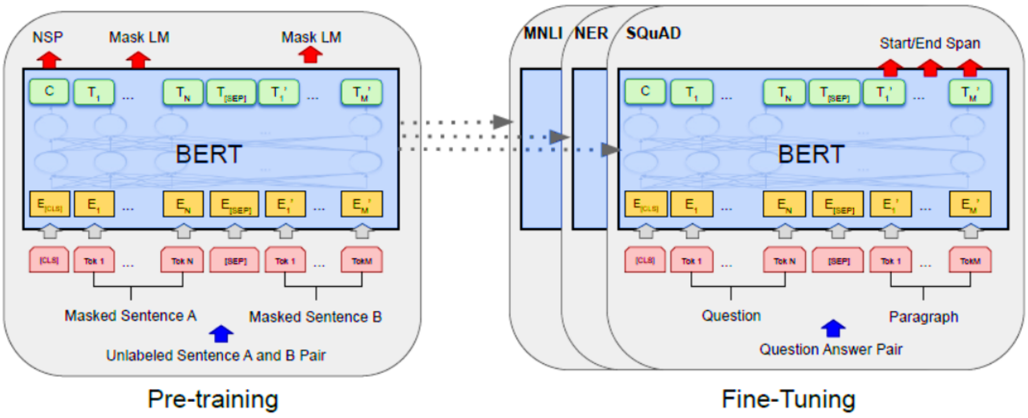

FIGURE 2 Overall pre-training and fine-tuning procedures for BERT

RoBERTa uses the same transformer encoder architecture as BERT, but differs in training. It removes the Next Sentence Prediction (NSP) objective, is trained on much larger datasets, applies dynamic masking during training, and uses larger batch sizes with longer training times. These changes make RoBERTa more robust than BERT.

DistilBERT provides a lightweight alternative to BERT while retaining most of its performance. It has the same transformer encoder architecture as BERT, but with a reduced size. It removes the token-type embeddings and reduces the number of encoder layers by half (six layers instead of 12 in BERT-base). DistilBERT is trained using a knowledge distillation process, where it learns to mimic BERT's behavior, making it 40% smaller and 60% faster while retaining approximately 97% of BERT's performance.

ELECTRA [21] (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) It is a pre-training method for language models that trains a discriminator to detect replaced tokens rather than masking them, as in BERT, making it more efficient than BERT.

GPT (Generative Pre-trained Transformer) uses only the decoder part of the Transformer architecture, unlike BERT, which uses only the encoder. GPT is designed for unidirectional (left-to-right) language modeling, in which each token can only attend to previous tokens. This makes GPT suitable for text-generation tasks. In contrast, BERT uses bidirectional self-attention, making it better suited for understanding tasks such as classification.

GPT is trained with a causal language modeling (CLM) objective, predicting the next token, whereas BERT is trained with masked language modeling (MLM), predicting masked tokens in context.

.

2.2 THREATS POSED BY LLMS

Even though LLMs can generate human-like text, they pose a number of risks and ethical concerns. LLMs cause several issues that affect society, and many people may find it difficult to accept this. Some of the biggest threats are: (1) Misinformation and Disinformation: Powerful language models such as GPT-4, GPT-5, etc., can produce misinformation and disinformation that resembles genuine information. It makes it more likely that misinformation can cross platforms. It becomes more serious on the social media channels because information goes around so easily without checking [22,23]. (2) Academic Dishonesty: Using LLMs to write essays, do assignments, or produce scientific text poses a major risk to academic integrity [24]. (3) Malicious Use: Because LLMs are accessible to everyone, cybercriminals are no longer blocked by technical limitations from developing malware, spam, phishing emails, and social engineering content [25,26]. (4) Loss of Human Creativity: Relying too much on AI-generated content could slowly weaken human creativity in writing, journalism, and creative work in general. A lot of people talk about this problem, but honestly, it still lacks a strong academic investigation so far [27]. Despite the fact that this issue has received a lot of attention, academic literature has not fully examined it. (5) Bias and Discrimination: Training data biases can be replicated and even amplified by LLMs, producing outputs that reinforce racial, cultural, or gender stereotypes. (6) Privacy Risks: Concerns regarding data leakage and privacy attacks arise because LLMs trained on massive datasets may unintentionally memorize and output sensitive personal data [28,29].

2.3 VARIATIONS BETWEEN AI-GENERATED AND HUMAN WRITTEN TEXT

The field of research concerned with finding a way to become able to separate AI-generated text and human-created text is a significant one. Numerous notable differences in linguistic, stylistic, and structural properties remain. These signals are exploited by the detection methods. This is not one signal but various signals. (1) Perplexity and Predictability The perplexity of the AI-generated text is lower than that of the text generated by the actual writer. AI models write a word at a time using prediction, which results in a very predictable word flow. Hence, the predictability of AI-generated text. In contrast, human-written texts exhibit higher perplexity. Human thought process is less predictable, due to which we get plenty of sentence variation in sentence structure, vocabulary, and rhythm; It is known as ‘burstiness’ [30]. This unpredictable quality is a central feature of real human writing. (2) Stylistic and Linguistic Patterns AI-generated text can feel “too clean.” It tends to have relatively good grammar, steady patterns of sentences, and fewer odd expressions. Writing by humans, in contrast, mixes tones and sentence lengths, and may often include a small mistake or two and some wording that is not too formal. People make more varied use of verbs and connectors, while the AI tends to repeat safe, common wording [31]. (3) Content and Depth AI may efficiently summarize and recombine information, but it cannot generate an original thought or deep reasoning. It may also confidently create false information, which is called a ‘hallucination’. Human writers normally utilize real experience, personal knowledge, and evidence; an argument based on these is grounded and specific [32]. (4) Emotional Depth and Personalization. Finally, an important difference between humans and AI is the presence of emotions and personality. AI can generate text that appears emotional, but they don't actually have feelings, nor do they possess a voice. The writer brings together hard data and does not express lived experiences. Nonetheless, human writing has personal stories, opinions, and attachments that make the text relatable and real. A human reader may readily notice this personalization. Automated detectors have difficulty measuring such personalization, which is a key aspect [33]. Table 1 summarizes the key linguistic and statistical differences between human-written and AI-generated texts, highlighting features commonly exploited by AIGTD systems.

Table 1 Comparison of Linguistic and Statistical Features between Human-Written and AI-Generated Text

Feature Category	Human-Written Text	AI-Generated Text	Relevance to Detection
Perplexity	Higher and more variable	Lower and more uniform	AI text tends to follow high-probability token paths
Burstiness	High variability in sentence length and structure	More uniform sentence patterns	Low burstiness is a strong AI indicator
Lexical Diversity	High (rich vocabulary, rare words)	Moderate, favors frequent tokens	Stylometric discrimination
Repetition	Irregular repetition	Subtle n-gram or semantic repetition	Useful for n-gram and entropy-based methods
Sentence Length	Highly variable	Moderately consistent	AI prefers balanced sentence lengths
Syntactic Complexity	Inconsistent, may include errors	Grammatically consistent and polished	Over-regularity signals AI generation
Error Patterns	Typos, informal grammar, and idiosyncratic mistakes	Rare surface errors, consistent grammar	Error absence can indicate AI
Discourse Coherence	Natural but sometimes fragmented	Globally coherent but locally generic	Coherence over-optimization
Stylistic Consistency	Style varies within text	Style remains highly consistent	Low stylistic variance is suspicious
Hedging Language	Mixed use	Frequent use of safe phrases	Risk-avoidance behavior of LLMs
Token Probability Distribution	Heavy-tailed	Skewed toward high-probability tokens	Basis of likelihood-rank detectors
Semantic Redundancy	Low	Moderate–High	Over-explaining concepts
Use of Idioms & Slang	Natural and context-specific	Limited or overly neutral	Weakness in informal language
Creativity & Deviations	Includes unexpected jumps	Constrained creativity	Limited exploration in generation

Editing Traces	Inconsistent revisions	Smooth and uniform	Post-editing detection signal

3. RELATED WORKS

With the rapid advancement of LLMs, detecting AI-generated text has become an increasingly significant and challenging area of research. Because of this, several review studies have tried to summarize and analyze the different techniques used in this field and how they are evolving.[34] The earliest survey focused mostly on simple and direct techniques using basic statistical patterns since the text was usually easier to distinguish, but once autoregressive language models appeared, the whole task became much more complicated, and traditional detection approaches were no longer enough to handle the new level of fluency and coherence in generated text.[35] This study offered one of the early overviews of how to detect machine-generated text and served as a base reference for later work in the area. The authors grouped detection techniques into four main directions: traditional trained classifiers, zero-shot classification methods, fine-tuned language models, and approaches that combine both humans and machines in the decision process.[36] This work reviews the datasets created for identifying text written by ChatGPT and looks at the different detection techniques used with them. It also examines research studies that compare the writing styles of humans and ChatGPT. It examines the main language patterns that make it easy to distinguish between them. [37] The paper discusses several serious issues, including poor performance on out-of-distribution text, sensitivity to adversarial manipulation, difficulty making methods applicable in the real-world setting, lack of a strong evaluation standard. The authors also outline future research directions that could enhance the security and accountability of the detection systems.

As shown in Table 2, previous survey studies on AI-generated text detection differ significantly in terms of scope, methodology, and limitations.

Table 2 Comparative Analysis of Key Survey Studies on AI-Generated Text Detection

Study (Ref)	**Year**	**Taxonomy of Methods**	**Tool Evaluation**	**Dataset Inventory**	**Methodology**	**Limitations**
Beresneva [34]	2016	Traditional (Statistical & ML-based)	No	Partial	Systematic Review (Short version)	Outdated; focuses on Markov chains and simple spam; no coverage of LLMs.
Jawahar et al [35]	2020	Neural-centric (Zero-shot, Fine-tuning)	No	Yes	Critical Survey & Error Analysis	Pre-dates ChatGPT; strictly English-centric; lacks modern watermarking analysis.
Dhaini et al. [36]	2023	Model-Specific (ChatGPT traits)	Yes	Yes	Qualitative & Comparative Survey	Too narrow; restricted only to ChatGPT; ignores other major LLMs (Claude, Llama).
Wu et al [37]	2025	Multi-layered (Neural, Watermark)	Yes	Yes	Taxonomy-driven Review	Does not address computational/prohibitive costs; lacks focus on low-resource languages (Arabic).
This Study	2025	PRISMA-compliant systematic review with unified taxonomy	Yes	Extensive	comparative analysis of detection paradigms, dataset mapping, and structured identification of research gaps across languages and models.	No model-level evaluation; focuses on approach-level comparison and research gap analysis.

4. SURVEY METHODOLOGY

This systematic literature review adhered to the PRISMA 2020 guidelines to ensure transparent and reproducible reporting of studies on the detection of AI-generated text. This review targeted peer-reviewed journal papers, conference papers, and relevant preprints published from January 2020 to December 2025. Searches were conducted across seven major databases: Google Scholar, IEEE Xplore, SpringerLink, ACM Digital Library, arXiv, Scopus, and Web of Science.

The final search was completed on December 15, 2025. No language restrictions were imposed, though English-language publications were prioritized for accessibility.

Search strings employed Boolean operators (AND/OR) to combine relevant terms, such as: ("AI-generated text" OR "LLM-generated text" OR "GPT-generated" OR "synthetic text") AND ("detection" OR "classifier" OR "identification") AND ("machine learning" OR "deep learning" OR "BERT" OR "statistical"). Complete search strings are detailed in Appendix A for full replicability.

The inclusion criteria encompassed original research using statistical models, machine learning, deep learning, or hybrid approaches for detecting AI-generated text (e.g., from LLMs such as GPT or BERT), limited to 2020-2025 publications. Exclusion criteria eliminated reviews, non-academic sources, commercial promotions, pre-2020 works, and irrelevant studies not focused on AI text detection.

The screening process unfolded in four stages: (1) duplicate removal (n=100 from the initial 290 records), (2) title/abstract screening by researchers, (3) full-text eligibility assessment, and (4) final resolution of disagreements through discussion (inter-rater reliability kappa=0.82). At the full-text stage, exclusions included irrelevant methods (n=45), non-empirical works (n=35), and duplicates (n=20), yielding 90 included studies.

Quality assessment was applied using the Mixed Methods Appraisal Tool (MMAT) v.2018 across six criteria, including methodology appropriateness and outcome reliability. Studies scoring ≥70% were retained as high quality (75/90 qualified); risk-of-bias ratings were low (n=62), moderate (n=20), and high (n=8), with low-quality studies excluded.

Data extraction was conducted using a standardized extraction form to collect information on authorship, publication year, research objectives, detection methods, datasets (type, size, and language), evaluation metrics (accuracy, precision, recall, and F1-score), reported challenges, and future research directions.

The updated PRISMA flowchart Figure 3 illustrates: 290 records identified, 100 duplicates removed, 190 screened (100 excluded), 90 full-texts assessed (all eligible post-quality check), resulting in 90 included studies.

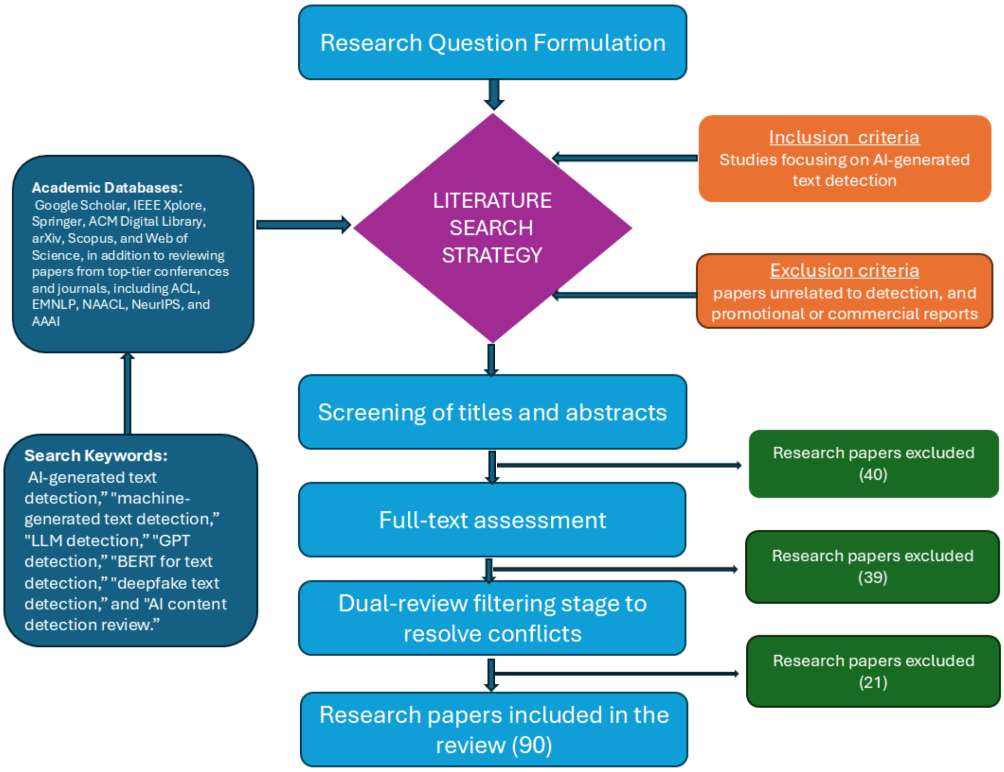
Research Question Formulation
Inclusion criteria
Studies focusing on AI-generated text detection
Academic Databases:
Google Scholar, IEEE Xplore, Springer, ACM Digital Library, arXiv, Scopus, and Web of Science, in addition to reviewing papers from top-tier conferences and journals, including ACL, EMNLP, NAACL, NeurIPS, and AAAI
LITERATURE SEARCH STRATEGY
Exclusion criteria
papers unrelated to detection, and promotional or commercial reports
Screening of titles and abstracts
Research papers excluded (40)
Search Keywords:
AI-generated text detection," "machine-generated text detection," "LLM detection," "GPT detection," "BERT for text detection," "deepfake text detection," and "AI content detection review."
Full-text assessment
Research papers excluded (39)
Dual-review filtering stage to resolve conflicts
Research papers excluded (21)
Research papers included in the review (90)

FIGURE 3 **Study selection process of the systematic review**

5. TAXONOMY OF METHODS FOR AIGTD

After reviewing 90 studies in detail, we suggest a taxonomy for AIGTD that groups the main techniques into four categories: statistical and stylometric approaches, transformer-based models, watermarking methods, and hybrid strategies that combine more than one method.

5.1 STATISTICAL AND STYLOMETRIC METHODS

Statistical and stylometric methodologies are among the earliest and most commonly used techniques for AIGTD. The main idea behind them is that even though modern LLMs write in a way that looks very human, their output still carries small statistical and stylistic "signatures." By measuring these patterns, it becomes possible to determine whether a piece of text was written by a human or produced by a model. By quantifying and analyzing these characteristics, the detectors can differentiate between content authored by humans and that produced by machines.

One common statistical approach is perplexity-based detection, which evaluates the ability of a language model to predict the next word in a sentence. Generally, AI-generated content has lower perplexity scores because it tends to follow highly probable sequences of words. In other words, the writing can appear a bit too predictable compared to human text, which usually has more variation and unexpected phrasing.LLMs often produce text with notably lower perplexity compared to human writing [38]. The other method used in the GPTZero Tool is known as burstiness. This refers to variations in the length and structure of sentences. Typically, human writing has more burstiness, with short sentences alongside longer ones. AI tends to make texts rhythmically and stylistically more similar compared to human writing, with varying levels of difficulty [39]. GLTR (Giant Language model Test Room) [40]is a statistical tool that uses statistical tests to check if the text is a model-generated or human-written text. The following tests were performed: (1) the probability of a word given its preceding context, (2) the absolute rank of a word in the predicted distribution, and (3) the entropy of the predicted distribution. The language patterns are exploited by these methods to identify the generation of artifacts. [41] The paper proposed DetectGPT, a new technique for detecting machine-generated text, based on the observation that LLMs produce text that occupies regions of negative curvature. It uses only log probabilities computed by another generic pre-trained language model (e.g., T5). [42] A technique that relies on the detection of reusable high-order n-grams to identify LLM-generated documents has been proposed. This method works on the premise that certain n-grams occur abundantly within the output of large LLMS, resulting in a frequency distribution that is substantially different from typical human text. [43] The paper presents two novel approaches: DetectLLM-LRR, which employs log-likelihood information for efficiency, and DetectLLM-NPR, which uses the normalized perturbed log rank for improved accuracy.

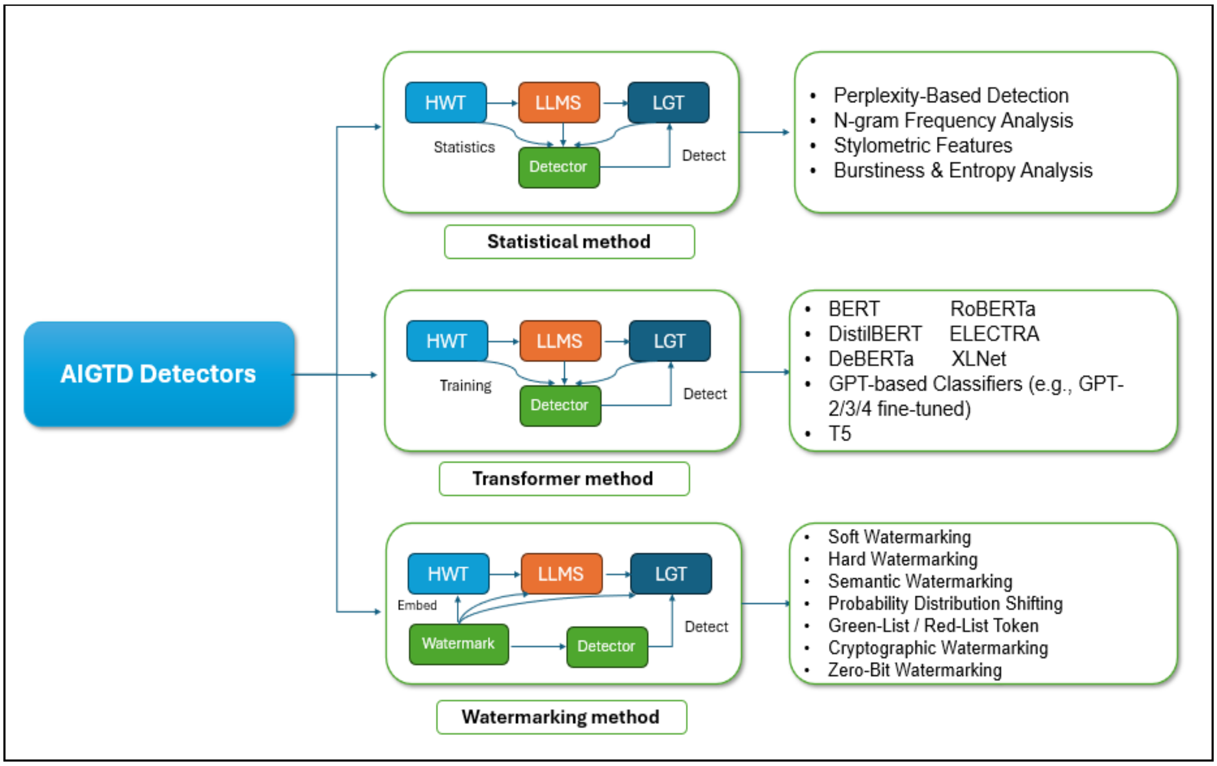

FIGURE 4 Three key research directions in AIGTD

Stylometric techniques measure the numerical characteristics of writing styles, such as vocabulary richness and punctuation usage, creating another option for detecting simulated text by AI and humans alike. [44] The article describes an algorithm that uses three types of stylometric features: phraseology, punctuation, and linguistic diversity to improve the detection of AI-generated tweets in Twitter timelines. [45] The authors suggest a feature-based detection method that uses a variety of intrinsic differences such as named entities, coreference chains, and part-of-speech distributions. [46] StyloAI employs thirty-one stylometric features and was subsequently trained using a Random Forest Classifier. Moreover, the model achieved an accuracy of 81% on the AuTextification dataset and 98% on the Education dataset. The analysis provides specific features, such as lexical diversity metrics, that can help improve the detection. Using this strategy would also help us observe the differences between AI-generated and human texts from diverse samples. Table 3 highlights the main differences between statistical and stylometric techniques in terms of the extracted features and detection assumptions.

5.2 TRANFORMER_BASED METHODS

This method involves adapting previously trained transformer models, such as BERT, RoBERTa, and DistilBERT, to human and AI-generated text datasets. These models learn to differentiate between the two based on their patterns and features. Through this supervised learning, the model learns to differentiate between the two classes by extracting more subtle and complex linguistic features and stylistic differences that are typically present in the AI output. This method works very well if sufficient annotated data are available; consequently, the model generalizes across the domain. [47, 48] The studies show ways to detect AI text using BERT to pick up patterns of language that show generation by machines. These techniques are effective, as shown by experiments and evaluation, and BERT can effectively identify AI-generated texts with high accuracy. [49] The paper proposed a ranking classifier called RoBERTa-Ranker. This classifier is a fine-tuned version of RoBERTa. It was trained as a baseline on a dataset that contains a diverse range of human texts and texts produced by various LLMs. [50] in this study, several transformer architectures like BERT, RoBERTa, DistilBERT were fine-tuned on the Enhanced RODICA dataset. According to the study, the RoBERTa-large model achieved higher accuracy and F-scores of about 83% for monolingual classification. In multilingual scenarios, the DistilBERT-multilingual-cased model excelled, achieving accuracy and F-scores of around 72%, demonstrating its effectiveness in handling diverse language inputs. [51] DistilBERT is utilized in this research for detecting AI-generated text by classifying essays as either human-written or AI-generated. The model, a streamlined version of BERT, was fine-tuned on a curated dataset that included diverse text types to improve detection accuracy.

Table 3 Summary of Statistical and Stylometric Techniques Used for Detecting AI-Generated Text

Ref..	Year	AIGTD Approach	Used Method	Dataset	Language	Accuracy
[38]	2025	Statistical /Perplexity-based	Context-aware perplexity with adaptive thresholds	University homework	English	≈ 74% - 100% (category-dependent)
[39]	2021	Statistical / Distributional	Repeated higher-order N-grams (super-maximal repeats) with a self-training ensemble	(human) + GPT-2 generated text	English	80%
[40]	2019	Statistical /Stylometric	Token probability, rank (top-k), entropy analysis with human-assisted visualization	GPT-2 generated texts, NYT articles, and scientific abstracts	English	Up to 72.3% (human-assisted detection accuracy)
[41]	2023	Statistical /Zero-shot	Probability curvature (perturbation discrepancy)	XSum, SQuAD, WritingPrompts, PubMedQA, WMT16	English (also German tested)	AUROC ≈ 0.92-0.99 (dataset & model-dependent)
[43]	2023	Statistical /Zero-shot	Log-Rank-based statistics (LRR, NPR)	XSum, SQuAD, Writing Prompts	English	AUROC ≈ 0.86 - 0.97 (dataset & method-dependent)
[44]	2023	Statistical /Stylometric (Hybrid)	Stylometric features fused with RoBERTa embeddings + change-point detection (StyloCPA)	TweepFake + In-house Twitter timelines	English	80.7% - 99.2% (timeline-length dependent)
[46]	2024	Statistical /Stylometric	31 stylometric features + Random Forest classifier	AuTexTification + Education datasets	English	81% - 98% (dataset-dependent)
[52]	2024	Statistical /Zero-shot	Conditional probability curvature (sampling-based, no perturbation)	XSum, SQuAD, Writing Prompts, PubMedQA, WMT16	English + German	AUROC ≈ 0.99 (white-box), ≈ 0.97 (black-box)

[53]	2024	Statistical /Zero-shot	Likelihood-based zero-shot detectors with black-boxvs. white-box(prompt-aware) evaluation	XSum summaries (Human vs. LLaMA-2 generated)	English	Accuracy (AUC): ≈ 0.33-0.47 (black-box), up to ≈ 1.00 (prompt-aware)

[54] The researchers used pre-trained transformer-based models, specifically ELECTRA for English and AraELECTRA for Arabic, after fine-tuning on academic essays.[55] The paper presents an ensemble model designed to detect text generated by AI techniques, integrating BERT and DeBERTa. [56] The study presents an ensemble approach that combines RoBERTa-base, OpenAI detector, and BERT-base-cased for the English language. It uses RemBERT, XLM-RoBERTa-base, and BERT-base for multilingual detection. [57] It utilizes multiple transformer models as a stack ensemble: ALBERT, ELECTRA, RoBERTa, and XLNet. Performance Evaluation of transformers by Detection Scenario: A comprehensive literature survey [58-60]. The models differ in their ability to withstand paraphrasing attacks. DeBERTa is the most robust, with a 15-20% drop in performance, followed by RoBERTa with 20-25%. DistilBERT is the least robust model and shows a drop of 30-35% in performance score during the robustness analysis for paraphrasing attacks. For round-trip translation attacks, English to Spanish demonstrated a drop of 10-15% in accuracy, followed by English to Chinese, which demonstrated a score drop of 25-35%, and lastly, English to Arabic, which saw a drop of 35-45% in accuracy. A comparison of transformer-based studies for AI-generated text detection is presented in Table 4.

5.3 WATERMARKING_BASED METHODS

Watermarking-based methods aim to embed a hidden, machine-readable signal into AI-generated text during the generation process to enable the verification of its origin. Watermarking techniques can be categorized into three main types: Post-hoc, Data-driven, and Model-driven watermarking approaches. In the Post-hoc method, watermarks are inserted after the text has already been generated, and no modifications are made to the model or training data. Edits the generated text to embed patterns (e.g., synonym swaps, punctuation patterns, spacing, or controlled phrase insertion) [61] introduce a post-hoc watermarking method by black-box large language models. In embedding, semantically and syntactically fundamental words that are robust to minor modifications are identified as anchor points. Using a paraphrase-based lexical substitution model, selected words are replaced with synonyms that preserve the meaning of the sentence but encode binary watermark bits. Detection uses the same position selection criteria and a one-proportion z-test to determine watermark presence. [62] The paper presents a watermark insertion method that utilizes Word2Vec and Sentence encoding to embed watermarks into the text, ensuring that the process is automatic and maintains blindness,

Table 4 Comparative Analysis of Transformer-Based Approaches in AI-Generated Text Detection

Ref..	Year	AIGTD Approach	Used Method	Dataset	Language	Accuracy
[49]	2024	Transformer-based	RoBERTa-Ranker with domain-aware fine-tuning	LLMCheck (news, tweets, essays, QA)	English	≈ 85.8% - 97.3% (F1, in-domain & cross-domain)
[50]	2025	Transformer-based	Fine-tuned RoBERTa / BERT / DistilBERT	M4, AICrowd, ERH	EN, RO, HU (Multilingual)	83.0% (RoBERTa-large, EN)
[51]	2024	Transformer-based	DistilBERT fine-tuning with contextual embeddings	Kaggle essays (500k AI & human texts)	English	98%
[54]	2025	Transformer-based / Hybrid	ELECTRA & AraELECTRA fine-tuning with stylometric features	GenAI Detection Task 2 (academic essays)	English & Arabic	F1 = 98.5% (EN), 98.4% (AR); up to 99.7% (EN, ELECTRA-Large)
[56]	2025	Transformer-based (Ensemble)	Inverse perplexity-weighted soft voting (RoBERTa, BERT, XLM-R, RemBERT)	GenAI Detection Task 1 (COLING 2025)	English + Multilingual	Macro F1 = 0.7458 (EN), 0.7513 (Multilingual)
[57]	2023	Transformer-based(Ensemble)	Stacking ensemble of ALBERT, ELECTRA, RoBERTa, XLNet with Logistic Regression meta-learner	ALTA 2023 Shared Task (18k train / 2k val / 2k test)	English (legal domain)	96.94% (internal test), 95.55% (official test)

[63]	2024	Transformer-based	BERT fine-tuning for text classification	Private dataset (708 AI / 670 human texts)	English	97.71%
[64]	2025	Transformer-based	Instruction fine-tuning (GPT-4o-mini, BERT, LLaMA-3)	Defactify (human vs. multi-LLM texts)	English	95.47% (Task-A, GPT-4o-mini F1)
[65]	2021	Transformer-based	Fine-tuned RoBERTa	TweepFake (25,572 tweets; human vs. bot)	English (Twitter)	89.6%

meaning the watermarking does not alter the original text's readability significantly. A data-driven method embeds detectable patterns directly into the model's outputs by modifying its training data, so the watermark becomes a natural part of the text style it learns to generate. [66] presents a multi-task learning framework for watermarking pre-trained language models (PLMs) by embedding backdoors triggered by specific inputs defined by the owners. [67] a watermark creation procedure that modulates logits modifiable through dynamic keys. This means modifying the output values for tokens in the language model so that they generate the desired watermark features. [68] It proposes SemStamp, a semantic watermarking algorithm that uses locality-sensitive hashing (LSH) to partition the semantic space of sentences. This technique takes candidate sentences from a language model, encodes and hashes them, and then uses sentence-level rejection sampling to ensure that the sampled sentences belong to the watermarked partitions selected in semantic embedding space. [69] proposes an attack method to identify trigger words or phrases in autoregressive language models that have backdoor watermarks inserted. This method involves analyzing open-ended generations from these models to detect the presence of watermarks. In model-driven watermarking, the watermark is embedded through models that involve fine-tuning, parameter editing, or altering the generation, such as logit modification and token biasing. The model does not require special datasets after training because the watermark signal is part of its natural output. Watermarking-based approaches for AI-generated text detection are listed in Table 5.

5.4 HYBRID METHODS

Hybrid methods aim to integrate multiple complementary strategies to achieve higher detection accuracy and robustness than single-strategy methods. These methods integrate different functionalities of transformers, such as fine-tuning, classification, embedding extraction, zero-shot inference, and few-shot prompting, into a single framework instead of using a single detection paradigm. Methods have been developed to enhance the versatility and reliability of detection systems. In real-world scenarios such as adversarial editing, cross-domain texts, and multilingual inputs, relying on a single detection can result in lower performance and/or false negatives .[70] COCO proposes a hybrid detection framework that incorporates linguistic coherence modeling with contrastive learning to enhance the detection of machine-generated text in low-resource settings. [71] The hybrid model of the research is a BERT-trained CNN and BiLSTM for detecting AI-generated content. [72] The authors used of pre-trained word embeddings fed into a Bi-LSTM network to capture bidirectional contextual dependencies. Subsequently, an attention mechanism focuses on the most informative tokens. The functions that have been attended to are solely responsible for the liquifying or classification of AI and human text. A BERT+CNN+BiLSTM approach that integrates the best aspects of each model component. BERT generates contextual embeddings, CNN recognizes local meanings, and BILSTM learns dependencies across longer sequences. This combination has shown better results than a single model. [73] recorded a 4% improvement in F1-score over base BERT on HC3 data. However, the increased complexity results in higher computational costs, raising questions regarding the scalability of real-time detection. As a viable solution, hybrid models indicate recent compromises in accuracy and efficiency. Table 6 indicates that hybrid approaches are more generic and robust, leveraging multiple detection strategies.

Table 5 Overview of Watermarking Techniques for AI-Generated Text Detection

Ref..	Year	AIGTD Approach	Used Method	Dataset	Language	Accuracy
[61]	2025	Watermarking-based (Post-hoc, Black-box)	Robust lexical-substitution watermarking + one-proportion z-test detection	HC3	English	96%
[62]	2024	Watermarking-based (Deep Learning)	Semantic synonym substitution + BERT-based detector	Dolly (train), C4 (test)	English	76.30%
[66]	2023	Watermarking-based	Backdoor watermarking at embedding layer (WLM, rare/common word triggers, multi-task learning)	IMDB, SST-2, MNLI, SNLI, PAWS	English	93%

[67]	2023	Watermarking-based	Green-list watermarking with statistical z-test detection (soft & hard watermark)	C4-RealNews	English	0.99 %
[68]	2023	Watermarking-based (Semantic)	Sentence-level semantic watermarking using LSH + contrastive SBERT	C4-RealNews	English	AUROC ≈ 0.998 (no attack) 0.974 under paraphrase attacks
[69]	2023	Watermarking / Attack Analysis	Frequency & TF-IDF analysis to discover backdoor watermark triggers in autoregressive LMs	DialogSum (summarization task)	English	Backdoor success rate ≥ 75% (with ≥10% poisoning); False trigger rate very low

Table 6 Hybrid Methods for AI-Generated Text Detection

No	Year	AIGTD Approach	Used Method	Dataset	Language	Accuracy
[70]	2023	Neural / Contrastive	Coherence graph + RoBERTa + contrastive learning (COCO)	GROVER, GPT-2, GPT-3.5	English	0.699 (low-resource), up to 0.997 (full data)
[71]	2025	Neural / Feature-based	POS + CNN + Bi-LSTM + Attention	DAGPap22 / Liyanage	English	85.00% -88.00%
[72]	2024	Hybrid (Statistical + Transformer)	TF-IDF + ML ensemble + DeBERTa-v3-large (stacked ensemble)	Mixed human & AI texts (Pile, SlimPajama, filtered corpora)	English	ROC-AUC = 0.975
[73]	2025	Hybrid (Transformer + Linguistic)	BERT embeddings combined with linguistic/stylometric features + ML classifier	Mixed human vs. LLM-generated texts (multiple domains)	English	98%
[74]	2024	Neural / Contrastive (Hybrid)	Multi-level contrastive learning + KNN retrieval (DeTeCtive)	Deepfake, M4, TuringBench	Multilingual	AvgRec = 98.44%, F1 = 98.38% (M4-mono); AvgRec = 93.42%, F1 = 93.05% (M4-multi)

6. DATASETS FOR AIGTD

Databases are important for the research and advancement of LLMs text detection. These datasets can help design effective detection mechanisms and standardize metrics for evaluating the performance of various approaches. Currently, they are working on datasets for detecting LLM generation. Although it is still in its infancy, it faces issues such as insufficient data, insufficient coverage of the domain, and insufficient sample complexity for good detectors. The most popular datasets used to train and evaluate detectors for LLM-generated texts will be introduced in this section. A frequently utilized dataset is HC3 (Human ChatGPT Comparison Corpus), which contains parallel human - and ChatGPT-generated responses across several academic and general topics. HC3 has become a benchmark metric for many due to its balanced design that integrates several different domains. HC3 is , however, for English-only, limiting its relevance to cross-lingual settings. Also, M4 (Multidomain, Multimodal, Multilingual, and Multiscale) is another primary dataset that includes texts from various LLMs in many languages. This approach can be used to test robustness across the various domains and models ; however is limited due to a lack of low-resource languages. The TuringBench dataset is similar, as it collects outputs from various text generators. MULTITuDE and RAID (Robust AI-generated Text Identification Dataset) are datasets designed for multilingual and adversarial evaluation to test detector robustness against paraphrasing and translation .Table 6 summarizes datasets widely adopted in AI-generated text detection research.

Table 7 **Datasets widely adopted in AI-generated text detection research**

Datasets	Language	Size	Data Description
HC3 [75]	English, Chinese	48,644	answers from human experts and ChatGPT in areas of finance, law, medicine, and technology.
MAGE [76]	English	436,606	A large dataset covers various domains like news, social media, academic writing, and online forums.
M4GT-Bench [77]	Multilingual	138,464	domains like news, scientific articles, and social media
CHEAT [78]	English	35,304	The dataset focuses on the academic domain
M4 [79]	Multilingual	247,000	A benchmark dataset created by multiple LLMs across various domains (e.g., news, social media, and academic).
RAID [80]	English	8,087,788	a large-scale benchmark from 11 LLMs across diverse writing styles. It introduces adversarial perturbations and mixed-content challenges to evaluate the robustness of AI-text detectors.
GPABench2 [81]	English	2,800,000	A benchmark dataset focusing on academic writing
GRiD [82]	English	6,513	A benchmark dataset for detecting GPT-generated text
RU-AI [83]	English	1,470,000	A large-scale multimodal dataset includes text, image, and audio pairs
The Pile [84]	English	1,392,522	A large and varied domain—such as academic papers, books, code, legal documents, scientific literature, web text, forums, and more
Ghostbuster [85]	English	38.420	Benchmarks in Student Essays, Creative Writing, and News Articles.
PAN [86]	English	361,579	Data from multiple domains, essays, news, and fiction
LLM-DetectAIve[87]	English	303,110	Four categories: 1) human-written, 2) machine-generated, 3) machine-written then machine-humanized, 4) human-written then machine-polished.
Arabic Fake News Dataset [88]	Arabic	1,500	News -Real & Fake, Human & AI-generated
XGLUE [89]	Multilingual	1.5M	Text classification, paraphrase identification, QA, and news classification
MULTITuDE [90]	Multilingual	250k	Binary classification (human vs. AI), model attribution, cross-lingual transfer, and cross-domain

7. DETECTION TOOLS FOR AIGTD

With the rapid advancement of LLMs and their growing application in numerous fields, the demand for AIGTD is increasing rapidly, resulting in the release of several online tools to the public. The illegal growth of AIGTD brings many challenges, including academic dishonesty, false information, and realness of online content. As a result, it is now essential for researchers, educators, and content creators to identify machine-generated text. Several studies assess these tools. [91,92] The paper tests 14 detection tools, evaluating their accuracy and reliability. The results indicated an accuracy of 50%-76%. The performance is further degraded by machine translation and content obfuscation techniques. According to the paper, tools meant to detect academic dishonesty won't work. There is a need for preventive pedagogical strategies instead. [93] Study evaluates the accuracy of five tools. The Copyleaks AI Content Detector and OpenAI's text classifier performed the best. All five tools failed to detect AI-generated content accurately and consistently in different languages. It can make it difficult to identify AI plagiarism in academic writing. Due to the rapid advancement of artificial intelligence (AI), the performance results in the studies available may become dated quickly. The earlier studies were conducted before the launch of the upgraded AI-based text generator tool ChatGPT in November 2022. They will not reflect the actual performance of the tools. This part compares some of these leading tools based on supported languages, whether they are free/paid, and the detection methodology used by them, as summarized in Table 8.

Table 8 Available tools for AIGTD

Tool	Key Features	Supported languages	Accessibility	Robustness to Paraphrasing &Translation	Accuracy [93]	Accuracy [94]	Accuracy [80]
Originality AI	Paraphrase plagiarism, real-time scanning, and editorial tools	30+	Paid	Low drops under machine translation & obfuscation	97.09	100	85
CopyLeaks	Multilingual, plagiarism check, API, accuracy, reporting.	30+	Paid	Medium- unstable under rewriting	-	100	-
Crosspalg	fast, simple, analytic	English	Free	Low sensitivity to paraphrasing & length	-	69	-
GPTZero	Perplexity-based, multi-level analysis, accurate	English	Free	Low	63.77	57	67
ZeroGPT	Instant, multilingual, granular analysis	30+	Paid	Low	-	83	66
Sapling	Fast, detailed, integrated, helpful	English	Free	Low	66.66	33	-
Winston AI	processing, detailed reports, team collaboration	English	Paid	Medium			71
Quillbot	Multilingual, detailed, integrated, free-tier, structured	20	Free	Low	-	-	-
Grammarly	Grammar-integrated, simple, percentage-based, citation,	English	Free	Low	-	-	-

8. ANALYSIS AND DISCUSSION

Various approaches have been proposed for AI-generated text detection, each offering complementary strengths and facing distinct limitations. Statistical and stylometric methods are computationally efficient, interpretable, and effective in low-data or zero-shot settings by leveraging measurable linguistic and stylistic cues; however, they are sensitive to paraphrasing, post-editing, short texts, and domain or genre shifts, with diminishing effectiveness as modern language models increasingly resemble human writing patterns. Transformer-based methods address some of these limitations by capturing deep semantic and contextual representations and achieving high detection accuracy across domains and writing styles, yet they require large-labeled datasets and substantial computational resources, exhibit limited interpretability, and remain vulnerable to adversarial transformations, particularly in low-resource languages.

Watermarking-based methods provide a scalable and low-cost solution for verifying the provenance of AI-generated text and supporting accountability and copyright protection, but their effectiveness depends on control over the generation model, lacks standardization, and can be weakened by paraphrasing, translation, or deliberate watermark removal. To overcome the individual weaknesses of single approaches, hybrid methods integrate multiple detection signals, improving robustness and reducing false positives and false negatives; nevertheless, they introduce higher computational complexity and system overhead, and may still be circumvented by coordinated adversarial attacks.

The performance of the four main detection approaches (Statistical, Transformer-based, Watermarking, and Hybrid methods) based on previous studies is shown in Figure 5. As can be seen, hybrid models show a consistently high detection accuracy, from 95% to 99%, which signifies their strength. The combined strengths of statistical clues and deep contextual embeddings allow this improvement to take place. Like the CNN-based models, the performance of transformer-based methods not only remained high but also stable (approximately 90-94%). Nevertheless, their precision varied slightly by experiment, implicating some sensitivity to dataset or prompt shift. Watermarking methods were able to achieve good accuracy, ranging from 82 to 90%, and were able to show good detection when the text was not much altered. In tests assessing user-readability features, slightly lower ratings were given to outputs from them relative to competitors. On the other hand, Statistical methods achieved the worst performance and the most variation in performance (77-87%) because they were based on surface-level features such as perplexity and burstiness that are not effective against generative models. In general, hybrid architecture seems to yield the best trade-offs and scalability as detection frameworks, combining the interpretability of conventional features and deep learning's representational power. The conclusion of our taxonomy shows that future research on AI-text detection should focus on robust and generalized hybrid and transformer-based approaches.

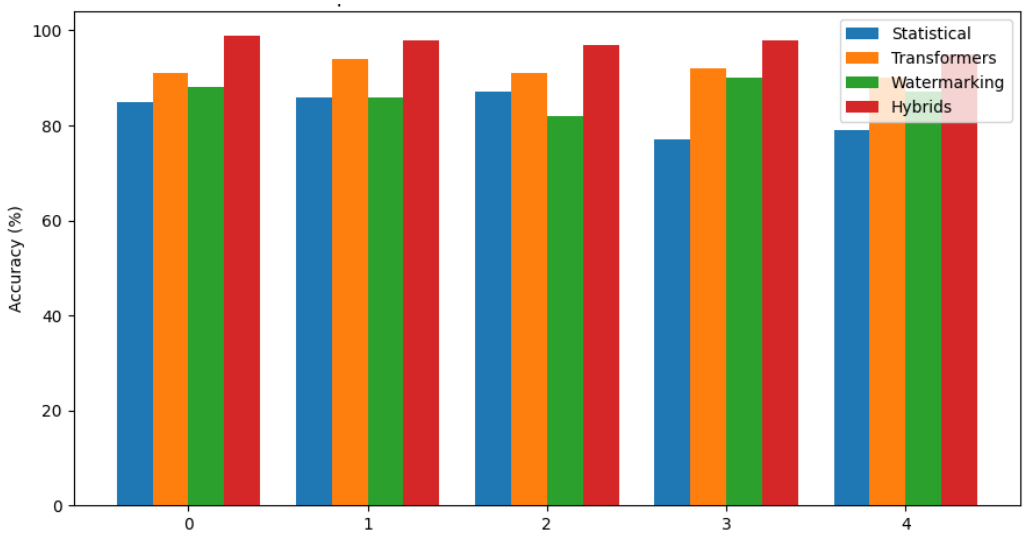

FIGURE 5 Comparative performance of AIGTD approaches.

An analysis of the datasets summarized in Table 7 reveals noticeable differences concerning their scale, domain diversity, language coverage, and real-life relevance. These factors directly impact the performance and generalizability of AI-generated text detection systems.

When it comes to dataset size, large-scale resources like RAID, GPABench2, RU-AI, and The Pile provide millions of samples, allowing training models with larger capacity and evaluation to be more stable. Nevertheless, their large size often leads to domain imbalance with certain text types (e.g., news or academic writing) being more prevalent in the corpus. The smaller datasets, such as HC3, CHEAT, and Ghostbuster, are well defined but have limited sample diversity and provide a more controlled environment, which negatively affects their robustness and real-world deployment evaluation.

Datasets like M4, M4GT-Bench, and The Pile, which provide text samples in multiple domains such as news, academic writing, and social media, are credible indicators of cross-domain evaluation. Many benchmarks, however, still have an academic bias. For instance, HC3, CHEAT, GPABench2, PAN . Detectors trained on such data overfit academic writing style, consequently degrading performance on informal or creative text. The ecological validity of reported results is limited by this domain skew.

Most of the popular datasets are in English. From a language perspective, data coverage in most reports is through scarce resources. Examples include M4, M4GT-Bench, and MULTITuDE/RAID. Within multilingual datasets, low-resource languages are underrepresented, often comprising shallow or noisy samples. The Arabic Fake News dataset precisely highlights this gap, as it remains small and domain-specific, indicating that larger, balanced multilingual corpora may be needed for reliable detection of fakeness beyond English.

The data is synthetic, which is another important limitation. A lot of datasets are built with completely AI-generated texts, whereas many real-world scenarios involve human-AI hybrid or post-edited content. Datasets like LLM-DetectAIve alleviate this problem by introducing humanized or machine-polished text, but these cases are still very rare in benchmarks. This discrepancy explains why detectors are accurate in benchmark settings but fail when paraphrasing, rewriting, and mixed authorship are used.

Overall, the analysis shows that the choice of dataset has a strong effect on detection performance, as reported accuracy and robustness claims. Neural and transformer-based detectors are preferred in the case of larger datasets. Smaller and cleaner datasets, on the other hand, inflate performance estimates under largely controlled conditions. The lack of balanced multilingual, multi-domain, and adversarially realistic datasets remains a major bottleneck in AIGTD research. Consequently, future progress depends not only on improved detection models but also on the construction of diverse, unbiased, and realistically curated datasets, particularly for low-resource languages such as Arabic.

An analysis of the tools summarized in Table 8, such as Originality AI and Copyleaks achieve the highest reported scores under controlled conditions, with some cases reaching near-perfect accuracy. However, when texts go through machine translation and paraphrasing, these metrics degrade sharply, revealing a huge gap between benchmark performance and real-world robustness. In contrast, commonly used free tools like GPTZero, Crossplag, and Sapling demonstrate moderate to low accuracy, often below 70, making them unreliable for high-stakes decisions

In terms of robustness, all tested tools have low resistance to paraphrasing and translation attacks. Many tools that are branded as multilingual don’t behave very well once the text has been rewritten in a cycle. This limitation is manifested, in particular, for perplexity-based detectors, such as GPTZero, whose performance is sensitive to length and surface fluency.

When it comes to cross-language reliability, most tools are essentially English-centric. While many platforms claim to support more than 30 languages, the fact that they do not report accuracy consistently across the languages makes them susceptible to unfairness and bias. According to the table, multilingual support is, in essence, support that is nominally there but is certainly not empirical support, especially about low-resource languages.

The tools' accessibility and transparency set them apart further. Paid systems usually yield better accuracy; however, they are black-box models that do not provide full insight into their detection decisions as well as error sources. While free tools are more accessible, they compromise on reliability and robustness. Furthermore, none of the tools evaluated provides systematic reporting of false positives or false negatives, which is a prerequisite for responsible deployment in education or in the courts.

Overall, the comparison shows that the present AIGTD tools must be treated as assistive indicators and not as definitive detectors. In spite of high reported accuracy, robustness, multilingual reliability, or transparency is not guaranteed. These results amplify the importance of standardized benchmarking, adversarial evaluation, and explainable detection before being trusted in real-world applications.

9. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

Recent improvements in AIGTD are facing numerous ongoing issues affecting scalability, robustness, and generalization [95]. Used methods tend to be vulnerable to adversarial modifications [96], through paraphrasing, style transfer, machine translation, etc. Detection accuracy can drop by more than 45% [97]. Cross-domain and cross-model generalization is still limited since detectors trained on specific domains or generation models often fail to perform well on unseen architectures or text types. Most AIGTD research focuses on English, leaving low-resource languages underexplored. This is problematic since LLMs such as GPT-4 and LLaMA now generate outputs in many languages. Detectors trained mainly on English often misclassify human-written texts in under-resourced languages, raising ethical concerns. Initial efforts, such as the RAID corpus and the Arabic subset of M4, provide some coverage; however, broader multilingual benchmarks are still needed. Some promising directions are multilingual fine-tuning (e.g., XLM-R, mBERT), zero-or few-shot transfer learning, and synthetic dataset generation through translation or paraphrasing. It is vital to expand these resources to enhance detection for additional languages. Also, the high computational cost associated with transformer-based approaches hinders practical deployment in real-time and resource-constrained settings, and biases towards specific models as well as their reduced performance in low-resource languages [98]. To address these limitations, future research should prioritize the following areas.

1. Adversarial Robustness and Generalization: Develop detectors that resist paraphrasing, style obfuscation, and translation attacks and are adaptable across models and domains without extensive retraining.
2. Multilingual and Low-Resource Detection: Using transfer learning and zero-shot techniques will help expand coverage of morphologically rich and underrepresented languages.
3. Explainable AI (XAI): The design of interpretable models that give humans understandable reasons for classification decisions, thus establishing trust and transparency.
4. Real-Time and Lightweight Systems: The creation of efficient detectors for low-latency use cases like chat, social-networking, and cheating detection, including deployment in edge devices.
5. Hybrid and Ensemble Frameworks: The combination of statistical, transformer-based, as well as watermarking and human-in-the-loop methods can improve accuracy and resilience in diverse contexts.
6. Benchmarking and Dataset Expansion: builds large-scale, diverse, and publicly accessible benchmarks and datasets with standardized evaluation metrics for a fair comparison of detection systems.

7. Ethics, Privacy, and Provenance Tracking: It involves the design of a detector that is privacy-preserving and legally compliant. Moreover, that involves watermarking and provenance block chaining for detecting authenticity. Through innovative, ethical, and sustainable solutions to these challenges, the field can progress towards scalable, multilingual, and trustworthy detection systems that can effectively combat AI-generated content and its growing sophistication.

10. CONCLUSIONS

According to this review, the AIGTD has made significant progress so far, but existing techniques are still facing serious issues with respect to scalability, robustness, multilingually, and robustness to the new generation. There are various techniques available. Statistical, transformer-based, watermarking, and hybrid techniques are included. Nonetheless, no single strategy works well across all areas and situations. There are many different varieties of evasion techniques and architectures to generate text. Therefore, there is an urgent need for a broader-based detection network that can be explained. In the future, progress will depend on establishing detection mechanisms that are multilingual and low-resource. The future development of vision systems will depend on benchmark datasets, real-time lightweight systems, and ethical frameworks that uphold privacy and remain consistent with the norms of society. Besides, detection together with watermarking and provenance tracking provides the promise of authenticating the integrity of the content. By addressing these shortcomings, the field can develop scalable, transparent, and robust solutions that can tackle increasingly sophisticated AIs producing text.

FUNDING

None

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their efforts.

CONFLICTS OF INTEREST

The authors declare no conflict of interest

REFERENCES

[1] OpenAI et al., “GPT-4 Technical Report,” Mar. 2024, [Online]. Available: http://arxiv.org/abs/2303.08774

[2] H. Touvron et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models,” Jul. 2023, [Online]. Available: http://arxiv.org/abs/2307.09288

[3] C. Raffel et al., “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” Sep. 2023, [Online]. Available: http://arxiv.org/abs/1910.10683

[4] P. Denny, H. Khosravi, A. Hellas, J. Leinonen, and S. Sarsa, “Can We Trust AI-Generated Educational Content? Comparative Analysis of Human and AI-Generated Learning Resources,” Jul. 2023, [Online]. Available: http://arxiv.org/abs/2306.10509

[5] L. Yan et al., “Practical and ethical challenges of large language models in education: A systematic scoping review,” Br. J. Educ. Technol., Jan. 2024, doi: 10.1111/bjet.13370.

[6] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, “From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy,” IEEE Access, 2023, doi: 10.1109/ACCESS.2023.3300381.

[7] D. I. Adelani, H. Mai, F. Fang, H. H. Nguyen, J. Yamagishi, and I. Echizen, “Generating Sentiment-Preserving Fake Online Reviews Using Neural Language Models and Their Human- and Machine-based Detection,” Dec. 2019, [Online]. Available: http://arxiv.org/abs/1907.09177

[8] A. Majeed and S. O. Hwang, “Reliability Issues of LLMs: ChatGPT a Case Study,” IEEE Reliability Magazine, vol. 1, no. 4, pp. 36–46, 2024, doi: 10.1109/MRL.2024.3420849.

[9] H. Yu and Y. Guo, “Generative artificial intelligence empowers educational reform: Current status, issues, and prospects,” Frontiers in Education, 2023, doi: 10.3389/feduc.2023.1183162.

[10] S. Suvra Ghosal, S. Chakraborty, J. Geiping, F. Huang, and A. Singh Bedi, “A Survey on the Possibilities & Impossibilities of AI-generated Text Detection.”

[11] Y. Mo, H. Qin, Y. Dong, Z. Zhu, and Z. Li, “Large Language Model (LLM) AI text generation detection based on transformer deep learning algorithm.”

[12] M. Maktabdar Oghaz, L. Babu Saheer, K. Dhame, and G. Singaram, "Detection and classification of ChatGPT-generated content using deep transformer models," Front Artif Intell, vol. 8, 2025, doi: 10.3389/frai.2025.1458707.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding." [Online]. Available: https://github.com/tensorflow/tensor2tensor

[14] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.11692

[15] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," Mar. 2020, [Online]. Available: http://arxiv.org/abs/1910.01108

[16] J. Wu, S. Yang, R. Zhan, Y. Yuan, L. S. Chao, and D. F. Wong, "A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license," Computational Linguistics, vol. 51, no. 1, 2025, doi: 10.1162/coli.

[17] H. Alshammari, A. El-Sayed, and K. Elleithy, "AI-Generated Text Detector for Arabic Language Using Encoder-Based Transformer Architecture," Big Data and Cognitive Computing, vol. 8, no. 3, p. 32, Mar. 2024, doi: 10.3390/bdcc8030032.

[18] E. N. Crothers, N. Japkowicz, and H. L. Viktor, "Machine-Generated Text: A Comprehensive Survey of Threat Models and Detection Methods," IEEE Access, vol. 11, pp. 70977–71002, 2023, doi: 10.1109/ACCESS.2023.3294090.

[19] A. Vaswani et al., "Attention Is All You Need."

[20] M. A. K. Raiaan et al., "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," IEEE Access, vol. 12, pp. 26839–26874, 2024, doi: 10.1109/ACCESS.2024.3365742.

[21] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators," Mar. 2020, [Online]. Available: http://arxiv.org/abs/2003.10555

[22] Y. Pan, L. Pan, W. Chen, P. Nakov, M.-Y. Kan, and W. Y. Wang, "On the Risk of Misinformation Pollution with Large Language Models," Oct. 2023, [Online]. Available: http://arxiv.org/abs/2305.13661

[23] M. Fore, S. Singh, C. Lee, A. Pandey, A. Anastasopoulos, and D. Stamoulis, "Unlearning Climate Misinformation in Large Language Models," May 2024, [Online]. Available: http://arxiv.org/abs/2405.19563

[24] C. Ihekweazu, B. Zhou, and E. A. Adelowo, "The Use of Artificial Intelligence in Academic Dishonesty: Ethical Considerations", [Online]. Available: https://iscap.us/proceedings/

[25] A. Iyengar and A. Kundu, "Large Language Models and Computer Security," in 2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), IEEE, Nov. 2023, pp. 307–313. doi: 10.1109/TPS-ISA58951.2023.00045.

[26] M. Adamec and M. Turčaník, "Development of Malware Using Large Language Models," in 2024 New Trends in Signal Processing (NTSP), IEEE, Oct. 2024, pp. 1–5. doi: 10.23919/NTSP61680.2024.10726304.

[27] R. Marrone, D. Cropley, and K. Medeiros, "How does narrow AI impact human creativity?" Creativity Research Journal, 2024, doi: 10.1080/10400419.2024.2378264.

[28] S. Wang et al., "Unique Security and Privacy Threats of Large Language Model: A Comprehensive Survey," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.07973

[29] C. O'Hagan, "Generative AI: UNESCO study reveals alarming evidence of regressive gender stereotypes," https://www.unesco.org/en/articles/generative-ai-unesco-study-reveals-alarming-evidence-regressive-gender-stereotypes?utm_source=chatgpt.com.

[30] Z. Xu and V. S. Sheng, "Detecting AI-Generated Code Assignments Using Perplexity of Large Language Models," 2024. [Online]. Available: www.aaai.org

[31] G. P. Georgiou, "Differentiating between human-written and AI-generated texts using linguistic features automatically extracted from an online computational tool."

[32] F. Amirjalili, M. Neysani, and A. Nikbakht, "Exploring the boundaries of authorship: a comparative analysis of AI-generated text and human academic writing in English literature," Front Educ (Lausanne), vol. 9, 2024, doi: 10.3389/feduc.2024.1347421.

[33] A. P. Singh, R. Saxena, and S. Saxena, "The Human Touch in the Age of Artificial Intelligence: A Literature Review on the Interplay of Emotional Intelligence and AI," Asian Journal of Current Research, vol. 9, no. 4, pp. 36–50, Sep. 2024, doi: 10.56557/ajocr/2024/v9i48860.

[34] D. Beresneva, "Computer-generated text detection using machine learning: A systematic review," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2016, pp. 421–426. doi: 10.1007/978-3-319-41754-7_43.

[35] G. Jawahar, M. Abdul-Mageed, and L. V. S. Lakshmanan, "Automatic Detection of Machine Generated Text: A Critical Survey," Nov. 2020, [Online]. Available: http://arxiv.org/abs/2011.01314

[36] M. Dhaini, W. Poelman, and E. Erdogan, "Detecting ChatGPT: A Survey of the State of Detecting ChatGPT-Generated Text," Sep. 2023, [Online]. Available: http://arxiv.org/abs/2309.07689

[37] J. Wu, S. Yang, R. Zhan, Y. Yuan, L. S. Chao, and D. F. Wong, "A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions," 2025, doi: 10.1162/coli.

[38] C. Vasilatos, M. Alam, T. Rahwan, Y. Zaki, and M. Maniatakos, "HowkGPT: Investigating the Detection of ChatGPT-generated University Student Homework through Context-Aware Perplexity Analysis," Mar. 2025, [Online]. Available: http://arxiv.org/abs/2305.18226

[39] F. Habibzadeh, "GPTZero performance in identifying artificial intelligence-generated medical texts: a preliminary study," The Korean Academy of Medical Sciences, vol. 38, no. 38, 2023.

[40] S. Gehrmann, H. Strobelt, and A. M. Rush, "GLTR: Statistical Detection and Visualization of Generated Text." [Online]. Available: http://gltr.io.

[41] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature."

[42] M. Gallé, J. Rozen, G. Kruszewski, and H. Elsahar, "Unsupervised and Distributional Detection of Machine-Generated Text," Nov. 2021, [Online]. Available: http://arxiv.org/abs/2111.02878

[43] J. Su, T. Yue Zhuo, D. Wang, and P. Nakov, "DetectLLM: Leveraging Log-Rank Information for Zero-Shot Detection of Machine-Generated Text." [Online]. Available: https://github.

[44] T. Kumarage, J. Garland, A. Bhattacharjee, K. Trapeznikov, S. Ruston, and H. Liu, "Stylometric Detection of AI-Generated Text in Twitter Timelines," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.03697

[45] L. Fröhling and A. Zubiaga, "Feature-based detection of automated language models: Tackling GPT-2, GPT-3 and Grover," PeerJ Comput Sci, vol. 7, pp. 1–23, 2021, doi: 10.7717/PEERJ-CS.443.

[46] C. Opara, "StyloAI: Distinguishing AI-Generated Content with Stylometric Analysis," May 2024, [Online]. Available: http://arxiv.org/abs/2405.10129

[47] U. Chakraborty, J. Gheewala, S. Degadwala, D. Vyas, and M. Soni, "Safeguarding Authenticity in Text with BERT-Powered Detection of AI-Generated Content," in 2024 International Conference on Inventive Computation Technologies (ICICT), 2024, pp. 34–37. doi: 10.1109/ICICT60155.2024.10544590.

[48] P. Javaji, P. S. Sreeya, and S. Rajesh, "Detection of AI Generated Text With BERT Model," in 2024 2nd World Conference on Communication & Computing (WCONF), 2024, pp. 1–6. doi: 10.1109/WCONF61366.2024.10692072.

[49] Y. Zhou and J. Wang, "Detecting AI-Generated Texts in Cross-Domains," in DocEng 2024 - Proceedings of the 2024 ACM Symposium on Document Engineering, Association for Computing Machinery, Inc, Aug. 2024. doi: 10.1145/3685650.3685673.

[50] D. Gifu and C. Silviu-Vasile, "Artificial Intelligence vs. Human: Decoding Text Authenticity with Transformers," Future Internet, vol. 17, no. 1, Jan. 2025, doi: 10.3390/fi17010038.

[51] J. M. Gakpetor, M. Doe, M. Y.-S. Damoah, D. D. Damoah, J. K. Arthur, and M. T. Asare, "AI-Generated and Human-Written Text Detection Using DistilBERT," in 2024 IEEE SmartBlock4Africa, 2024, pp. 1–7. doi: 10.1109/SmartBlock4Africa61928.2024.10779494.

[52] G. Bao, Y. Zhao, Z. Teng, L. Yang, and Y. Zhang, "Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature," Dec. 2024, [Online]. Available: http://arxiv.org/abs/2310.05130

[53] K. Taguchi, Y. Gu, and K. Sakurai, "The Impact of Prompts on Zero-Shot Detection of AI-Generated Text," 2024. [Online]. Available: https://github.com/kaito25atugich/Detector.

[54] M. AL-Smadi, "IntegrityAI at GenAI Detection Task 2: Detecting Machine-Generated Academic Essays in English and Arabic Using ELECTRA and Stylometry," Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.05476

[55] K. Aggarwal, S. Singh, Parul, V. Pal, and S. S. Yadav, "A Framework for Enhancing Accuracy in AI Generated Text Detection Using Ensemble Modelling," in 2024 IEEE Region 10 Symposium (TENSYMP), IEEE, Sep. 2024, pp. 1–8. doi: 10.1109/TENSYMP61132.2024.10752173.

[56] M. K. Mobin and M. S. Islam, "LuxVeri at GenAI Detection Task 1: Inverse Perplexity Weighted Ensemble for Robust Detection of AI-Generated Text across English and Multilingual Contexts," Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.11914

[57] D. Nguyen, K. M. N. Naing, and A. Joshi, "Stacking the Odds: Transformer-Based Ensemble for AI-Generated Text Detection," Oct. 2023, [Online]. Available: http://arxiv.org/abs/2310.18906

[58] S. Sarkar, M. F. Babar, M. M. Hassan, M. Hasan, and S. K. K. Santu, "Processing Natural Language on Embedded Devices: How Well Do Transformer Models Perform?," in ICPE 2024 - Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering, Association for Computing Machinery, Inc, May 2024, pp. 211–222. doi: 10.1145/3629526.3645054.

[59] C. Meaney, W. Hakimpour, S. Kalia, and R. Moineddin, "A Comparative Evaluation Of Transformer Models For De-Identification Of Clinical Text Data," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2204.07056

[60] J. Tian, C. Fang, H. Wang, and Z. Wang, "BEBERT: Efficient and Robust Binary Ensemble BERT," Feb. 2023, doi: 10.1109/ICASSP49357.2023.10096223.

[61] J. Hao, J. Qiang, Y. Zhu, Y. Li, Y. Yuan, and X. Ouyang, "Post-Hoc Watermarking for Robust Detection in Text Generated by Large Language Models." [Online]. Available: https://github.com/AlfredWatson/RSFPH-WTGBBLM

[62] T. Munyer, A. A. Tanvir, A. Das, and X. Zhong, "DeepTextMark: A Deep Learning-Driven Text Watermarking Approach for Identifying Large Language Model Generated Text," IEEE Access, vol. 12, pp. 40508–40520, 2024, doi: 10.1109/ACCESS.2024.3376693.

[63] H. Wang, J. Li, and Z. Li, "AI-Generated Text Detection and Classification Based on BERT Deep Learning Algorithm."

[64] C. Guggilla, B. Roy, T. R. Chavan, A. Rahman, and E. Bowen, "AI Generated Text Detection Using Instruction Fine-tuned Large Language and Transformer-Based Models," 2025.

[65] T. Fagni, F. Falchi, M. Gambini, A. Martella, and M. Tesconi, "TweepFake: About detecting deepfake tweets," PLoS One, vol. 16, no. 5 May, May 2021, doi: 10.1371/journal.pone.0251415.

[66] C. Gu, C. Huang, X. Zheng, K.-W. Chang, and C.-J. Hsieh, "Watermarking Pre-trained Language Models with Backdooring," Feb. 2023, [Online]. Available: http://arxiv.org/abs/2210.07543

[67] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A Watermark for Large Language Models." [Online]. Available: https://www.github.com/

[68] A. B. Hou et al., "SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation," Apr. 2024, [Online]. Available: http://arxiv.org/abs/2310.03991

[69] E. Lucas and T. C. Havens, "GPTs Don't Keep Secrets: Searching for Backdoor Watermark Triggers in Autoregressive Language Models," 2023.

[70] X. Liu, Z. Zhang, Y. Wang, H. Pu, Y. Lan, and C. Shen, "COCO: Coherence-enhanced machine-generated text detection under low resource with contrastive learning," [Online]. Available: https://github.com/YichenZW/Coh-

[71] J. Blake, A. S. M. Miah, K. Kredens, and J. Shin, "Detection of AI-Generated Texts: A Bi-LSTM and Attention-Based Approach," IEEE Access, vol. 13, pp. 71563–71576, 2025, doi: 10.1109/ACCESS.2025.3562750.

[72] Y. Zhang et al., "Enhancing Text Authenticity: A Novel Hybrid Approach for AI-Generated Text Detection," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.06558

[73] A. , M. S. P. Yadav, "Classifying AI vs. Human Content: Integrating BERT and Linguistic Features for Enhanced Classification," Oper. Res, vol. 6, 2025.

[74] X. Guo et al., "DeTeCtive: Detecting AI-generated Text via Multi-Level Contrastive Learning." [Online]. Available: https://github.com/heyongxin233/DeTeCtive

[75] Z. Wang et al., "How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection", doi: 10.48550/arXiv.2301.07597.

[76] Y. Li et al., "MAGE: Machine-generated Text Detection in the Wild," May 2024, [Online]. Available: http://arxiv.org/abs/2305.13242

[77] Y. Wang et al., "M4GT-Bench: Evaluation Benchmark for Black-Box Machine-Generated Text Detection," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2402.11175

[78] P. Yu, J. Chen, X. Feng, and Z. Xia, "CHEAT: A Large-scale Dataset for Detecting ChatGPT-writtEn AbsTracts," Feb. 2024, [Online]. Available: http://arxiv.org/abs/2304.12008

[79] Y. Wang et al., "M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2305.14902

[80] L. Dugan et al., "RAID: A Shared Benchmark for Robust Evaluation of Machine-Generated Text Detectors," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2405.07940

[81] Z. Liu, Z. Yao, F. Li, and B. Luo, "On the Detectability of ChatGPT Content: Benchmarking, Methodology, and Evaluation Through the Lens of Academic Writing," in CCS 2024 - Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security, Association for Computing Machinery, Inc, Dec. 2024, pp. 2236–2250. doi: 10.1145/3658644.3670392.

[82] Z. Qazi, W. Shiao, and E. E. Papalexakis, "GPT-generated Text Detection: Benchmark Dataset and Tensor-based Detection Method," in WWW 2024 Companion - Companion Proceedings of the ACM Web Conference, Association for Computing Machinery, Inc, May 2024, pp. 842–845. doi: 10.1145/3589335.3651513.

[83] L. Huang, Z. Zhang, Y. Zhang, X. Zhou, and S. Wang, "RU-AI: A Large Multimodal Dataset for Machine-Generated Content Detection," Association for Computing Machinery (ACM), May 2025, pp. 733–736. doi: 10.1145/3701716.3715306.

[84] L. Gao et al., "The Pile: An 800GB Dataset of Diverse Text for Language Modeling," Dec. 2020, [Online]. Available: http://arxiv.org/abs/2101.00027

[85] Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein, "Ghostbuster: Detecting Text Ghostwritten by Large Language Models," arXiv preprint arXiv:2305.15047, 2023.

[86] J. Bevendorff et al., "Overview of the 'Voight-Kampff' Generative AI Authorship Verification Task at PAN and ELOQUENT 2025," 2025.

[87] M. Abassy et al., "LLM-DetectAIve: a Tool for Fine-Grained Machine-Generated Text Detection," Mar. 2025, [Online]. Available: http://arxiv.org/abs/2408.04284

[88] H. Himdi, N. Zamzami, F. Najar, M. Alrehaili, and N. Bouguila, "Arabic Fake News Dataset Development: Humans and AI-Generated Contributions," IEEE Access, vol. 13, pp. 62234–62253, 2025, doi: 10.1109/ACCESS.2025.3556376.

[89] Y. Liang et al., "XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 6008–6018. doi: 10.18653/v1/2020.emnlp-main.484.

[90] D. Macko et al., "MULTITuDE: Large-Scale Multilingual Machine-Generated Text Detection Benchmark," Oct. 2023, doi: 10.18653/v1/2023.emnlp-main.616.

[91] D. Weber-Wulff et al., "Testing of detection tools for AI-generated text," International Journal for Educational Integrity, vol. 19, no. 1, Dec. 2023, doi: 10.1007/s40979-023-00146-z.

[92] C. Chaka, "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, vol. 6, no. 2, pp. 94–104, Jun. 2023, doi: 10.37074/jalt.2023.6.2.12.

[93] A. Akram, "An Empirical Study of AI-Generated Text Detection Tools," 2023.

[94] W. H. Walters, "The Effectiveness of Software Designed to Detect AI-Generated Writing: A Comparison of 16 AI Text Detectors," Open Information Science, vol. 7, no. 1, Jan. 2023, doi: 10.1515/opis-2022-0158.

[95] A. Bhattacharjee, R. Moraffah, J. Garland, and H. Liu, "EAGLE: A Domain Generalization Framework for AI-generated Text Detection," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.15690

[96] Y. Zhou, B. He, and L. Sun, "Humanizing Machine-Generated Content: Evading AI-Text Detection through Adversarial Attack," Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.01907

[97] S. Chakraborty, A. S. Bedi, S. Zhu, B. An, D. Manocha, and F. Huang, “On the Possibilities of AI-Generated Text Detection,” Oct. 2023, [Online]. Available: http://arxiv.org/abs/2304.04736

[98] Y. Jiang, J. Hao, M. Fauss, and C. Li, “Detecting ChatGPT-generated essays in a large-scale writing assessment: Is there a bias against non-native English speakers?,” Comput Educ, vol. 217, p. 105070, 2024, doi: https://doi.org/10.1016/j.compedu.2024.105070.