mathematics

MDPI

Review

Enhancing the Robustness of AI-Generated Text Detectors: A Survey

Xin Liu , Yang Li and Kan Li *

School of Computer Science & Technology, Beijing Institute of Technology, No. 5, South Street, Zhongguancun, Beijing 100081, China; xinliu@bit.edu.cn (X.L); liyangsmu@bit.edu.cn (Y.L.)
* Correspondence: likan@bit.edu.cn

Abstract

In recent years, AI-generated text (AIGT) detection has attracted increasing attention, and some detectors demonstrate high accuracy in benchmark settings. However, the complexity and diversity of AIGT and counter-detection methods in real-world applications present substantial challenges for AIGT detection. Consequently, there is a growing demand for more robust AIGT detectors. This survey provides a systematic overview of existing research on enhancing the robustness of AIGT detectors. We categorize the focus of related literature into three key areas: text perturbation robustness, out-of-distribution (OOD) robustness, and AI–human hybrid text (AHT) detection robustness. For each area, we thoroughly summarize and analyze the corresponding robustness enhancement methods and additionally incorporate some approaches from other fields as a supplement. We also methodically organize relevant benchmark datasets, robustness evaluation methods, and metrics used to assess detectors' performance. Then, through experiments, we evaluate the robustness of several commonly used detectors. Experiments show that text perturbations, OOD text, and AHT all affect the performance of these detectors, revealing that there remains significant room for improvement in their robustness. Finally, we suggest promising future directions based on the current issues faced by AIGT detectors and the detection requirements in real-world scenarios. To the best of our knowledge, this is the first review focused specifically on the robustness of AIGT detection.

Keywords: large language models; AI-generated text detection; model robustness

MSC: 68T50

Academic Editor: Dan Vilenchik

Received: 28 May 2025
Revised: 26 June 2025
Accepted: 28 June 2025
Published: 30 June 2025

Citation: Liu, X.; Li, Y.; Li, K. Enhancing the Robustness of AI-Generated Text Detectors: A Survey. *Mathematics* **2025**, *13*, 2145. https://doi.org/10.3390/math13132145

1. Introduction

With the rapid advancement of artificial intelligence (AI), large language models (LLMs) [1–3], known for their outstanding language capabilities, have gradually become indispensable tools in people's daily life and work. In various daily contexts, people increasingly rely on LLMs to generate desired text content, such as question answering [4], story generation [5], and scientific writing [6]. While LLMs provide significant convenience, they also raise concerns about the misuse of AI-generated text (AIGT), including issues like fake news [7], academic integrity [8], and harmful content [9]. To regulate the use of AIGT, effectively distinguishing it from human-written text (HWT) has become a critical challenge. However, the high similarity between AIGT and HWT makes the accuracy of human judgment in AIGT detection only marginally better than random guessing [10]. Consequently, there is an urgent need for more advanced AIGT detection methods.

Currently, AIGT detection has attracted increasing attention, leading to the emergence of a variety of detection methods. These methods can be broadly categorized into three types [11]: zero-shot detectors, training-based detectors, and watermarking detectors. Zero-shot detectors distinguish between AIGT and HWT based on inherent text features, without relying on supervised training with large labeled datasets. They typically extract semantically poor statistical features from the text and compare them against predefined thresholds to infer the text source. A type of representative approach involves feeding the test sample into a surrogate LLM to obtain metrics such as the average log-likelihood, rank [12], log-rank [13], or entropy [14] of all tokens, which are then used as features for classification. Another notable technique is the perturbed-based method [13], which leverages the observation that AIGT often resides in regions where the log-probability function exhibits a negative curvature. This method uses a pretrained mask-filling model to generate semantically similar perturbations of the original text, and then compares the log probability of the original and perturbed texts to determine the text source. Training-based detectors utilize the powerful representational capabilities of neural networks to learn rich textual features from large-scale training data, thereby enabling accurate detection. A widely adopted strategy is fine-tuning pretrained language models, such as BERT [15], RoBERTa [16], DeBERTa [17], or XLNet [18], on AIGT detection datasets. Some methods further incorporate contrastive learning [19] or adversarial learning [20] to enhance the performance of the detectors. Watermarking detectors take a different approach by embedding algorithmically detectable patterns into the generated text, while striving to maintain the quality and diversity of LLM outputs. A representative approach divides the vocabulary into a green list and a red list [21]. During generation, the selection probability of green-list tokens is boosted, thereby increasing their proportion in the output. The origin of the text can then be determined by analyzing the proportion of tokens from the red and green lists.

Although current AIGT detectors have made great progress, they still face numerous challenges. First, their performance significantly deteriorates when exposed to various types of text perturbations, including character-level and word-level substitutions, deletions, insertions, and swaps [22], sentence-level alterations [23], text paraphrase [24], and adversarial attacks [25]. These perturbations deceive AIGT detectors by introducing subtle perturbations that are nearly imperceptible to humans, severely affecting detection accuracy. Second, AIGT detectors may underperform when confronted with out-of-distribution (OOD) data, such as out-of-domain text [26], multilingual text [27], and cross-LLM text [28]. The distributional differences between these OOD data and the detector's training data cause the model to struggle in generalizing to OOD data. Third, detectors often perform poorly when dealing with AI–human hybrid text (AHT) [29]. In daily life, AHT is more prevalent than pure AIGT, but most current detectors are designed to detect the latter, significantly limiting their practical applicability in real-world settings. These challenges emphasize the need for more robust AIGT detectors.

In recent years, some AIGT detection studies have begun to focus not only on improving detection accuracy but also on enhancing the robustness of detectors. However, to date, no comprehensive review has systematically compiled this body of research. This survey aims to fill this gap by providing a comprehensive discussion of research focused on enhancing the robustness of AIGT detectors. Unlike previous reviews on AIGT detection [10,30,31] that cover various types of detection methods, we specifically focus on methods aimed at strengthening the robustness of AIGT detectors. We begin by defining the AIGT detection task and outlining the scope of discussions on robustness in AIGT detection. We categorize the research on AIGT detection robustness into three main areas based on their focus: text perturbation robustness, OOD robustness, and AHT detection robustness. We then provide a detailed discussion of the representative methods employed

in each category. Notably, due to the limited research on enhancing the robustness of AIGT detectors, we also incorporate some methods from other research fields that can be directly used to enhance AIGT detection robustness, such as techniques for improving the robustness of text classification models. This allows us to present a more comprehensive framework for strengthening AIGT detection robustness. We also systematically organize the relevant benchmark datasets, robustness evaluation methods, and metrics used to assess detectors' performance. Then, we carry out experiments to evaluate the robustness of several commonly used AIGT detectors. Finally, we highlight promising future directions, considering the current challenges of AIGT detectors and the detection requirements in real-world scenarios, with the aim of advancing the development of more robust and effective AIGT detectors.

The main contributions of this survey are summarized as follows:

- We provide a systematic review of methods for enhancing the robustness of AIGT detectors. To the best of our knowledge, this is the first survey dedicated to the study of enhancing AIGT detector robustness.
- We conduct a comprehensive taxonomy of the literature on enhancing AIGT detector robustness, categorizing it into three main areas based on their focus: text perturbation robustness, OOD robustness, and AHT detection robustness.
- We systematically organize the relevant benchmark datasets, evaluation methods, and metrics used to assess AIGT detectors' robustness.
- We evaluate the robustness of several commonly used AIGT detectors through experiments, revealing that there is still significant room for improvement.
- We outline several promising future directions motivated by real-world demands, offering valuable insights for enhancing the robustness and practical applicability of AIGT detection.

The rest of the paper is organized as follows. In Section 2, we define AIGT detection and the research scope of AIGT detection robustness. In Sections 3–5, we systematically catalog the research methods aimed to improve the AIGT detector in text perturbation robustness, OOD robustness, and AHT detection robustness, respectively. Next, we provide a detailed organization of the benchmark datasets, evaluation methods, and metrics for AIGT detection robustness evaluation in Section 6. In Section 7, we evaluate the robustness of several commonly used AIGT detectors. Then, in Section 8, we provide promising future directions for enhancing AIGT detection robustness. Finally, we conclude the entire survey in Section 9.

2. Preliminaries

2.1. AIGT Detection

The AIGT detection task is commonly treated as a binary classification problem. Its core objective is to distinguish whether the input text is generated by AI or written by a human. Let $\mathcal{S}$ be the set of text sequences to be detected. Given an input sequence $s \in \mathcal{S}$ and an AIGT detector $\mathcal{D}$, the formal representation of this task can be formulated as follows:

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

The detector $\mathcal{D}$ provides a probabilistic approximation of the decision function from Equation (1).

2.2. Robustness in AIGT Detection

The concept of model robustness is widely discussed in the fields of machine learning and deep learning. A well-performing model should not only excel in accuracy, but also exhibit strong robustness. However, there is no unified definition of robustness, and its meaning varies depending on the context, community, and research field. Freiesleben et al. [32] defined robustness in machine learning as follows: The robustness target is said to be robust to the robustness modifier if relevant interventions in the modifier, as specified by the robustness domain, do not lead to greater changes in the target than specified by the target tolerance. In deep learning, robustness typically refers to the requirement that a network behaves smoothly, meaning that small input perturbations or slight modifications to the model should not result in significant fluctuations in the output of deep neural networks [33]. Jin et al. [34] described a robust solution as one whose performance only decreases gradually when design variables or environmental parameters are varied within a certain range. Although various researchers describe robustness in different ways, they all emphasize that it refers to the ability to handle complex and dynamic situations effectively.

In AIGT detection, the focus on detector robustness varies across different studies. Huang et al. [35] categorized resistance to adversarial perturbations as an aspect of detector robustness. Antoun et al. [26] attributed the performance of a detector on out-of-domain data to the robustness issue. In addition, Yang et al. [36] included the detection of texts created through human–machine collaborative authorship in real-world scenarios as part of the robustness discussion. Based on existing research on AIGT detection robustness, we define the robustness of an AIGT detector as an ability to handle situations beyond the ideal testing conditions. Specifically, we categorize the related research into three types according to their focus on robustness issues in AIGT detection, as illustrated in Figure 1. First, from a micro-level perspective on the form of text representation, we define the category of text perturbation robustness. It encompasses local perturbations, such as character-level and word-level insertions, deletions, swaps, and substitutions [37], as well as global paraphrase. Second, we define the category of OOD robustness from a macro-level perspective of text distribution characteristics. This category involves cross-LLM [28] and cross-domain [38] detection. Finally, we define AHT detection robustness based on the complex composition of texts in real-world scenarios. It mainly includes the detection of human-created text polished by AI, AI-generated text edited by humans, and alternating human- and AI-created text. In the following, we will review the relevant methods for enhancing the robustness of the three types mentioned above, and provide a systematic categorization of these methods.

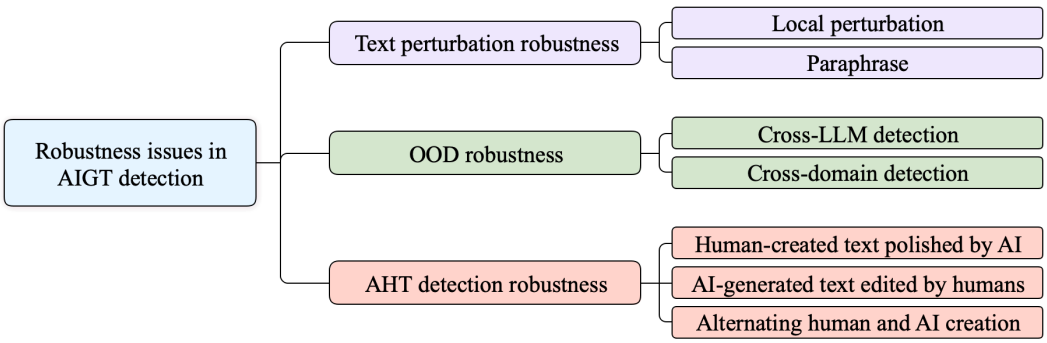

Figure 1. Taxonomy of the focus on robustness issues in AIGT detection.

3. Text Perturbation Robustness

Recent studies have demonstrated that current AIGT detectors are highly vulnerable to text perturbations. Even minor modifications, such as replacing a word with its abbre-

viation [35] or simply inserting an extra space [39], can significantly impact the detection results. Text perturbations refer to various operations that alter the original form of the text. Based on the scope of the perturbation, it can be further classified into local perturbation and global paraphrase. Local perturbation typically refers to modifying a small segment of text, which is usually difficult to perceive. It mainly includes character-level and word-level insertions, deletions, swaps, and substitutions. On the one hand, these perturbations may arise from unintentional spelling mistakes, while on the other hand, they can be deliberately introduced for adversarial purposes [40]. Although these perturbations may seem insignificant to humans, they can easily lead to misclassification by some detectors. Different from local perturbation, paraphrase may alter the overall form of the text. Text paraphrase is essentially a process of recombining the semantics and syntax of text [41]. It changes the overall expression of the text without altering its original meaning. This ability to generate diverse expressions while maintaining the original meaning makes paraphrase a key strategy for evading AIGT detection [42] has shown that recursive paraphrase can significantly reduce detection rates.

We summarize the existing methods for enhancing the text perturbation robustness of AIGT detectors and categorize them into five main types: data-augmentation-based methods, adversarial-training-based methods, noise-filtering-based methods, text-correction-based methods, and text-paraphrase-based methods, as shown in Table 1. Below, we will provide a detailed discussion of each category and present the corresponding representative approaches.

Table 1. Summary of text perturbation robustness enhancement methods.

Category	Method	Target Problem		Techniques
		Local Perturbation	Paraphrase	
Data augmentation-based methods	Wei et al. [37]	✓	✗	Synonym replacement, random insertion, random swap, random deletion.
	Karimi et al. [43]	✓	✗	Randomly inserting punctuation marks.
	Coulombe [44]	✓	✓	Transformation of syntactic trees.
	Guo et al. [45]	✓	✗	WordMixup, senMixup.
Adversarial training-based methods	Zhou et al. [25]	✓	✗	Word importance ranking, synonym replacement.
	Li et al. [46]	✓	✗	Targeted perturbation on embedding, generative adversarial network.
	Hu et al. [20]	✗	✓	Paraphraser, generative adversarial network.
Noise filtering-based methods	Cai et al. [47]	✓	✗	Unbiased watermark, Discarded Tokens.
	Wang et al. [48]	✓	✗	Anomaly filter.
	Huang et al. [35]	✓	✗	Reconstruction network, siamese calibration.
Text correction-based methods	Pruthi et al. [40]	✓	✗	Word recognition model.
	Zhou et al. [49]	✓	✗	Perturbation discriminator, embedding estimator.
	Li et al. [50]	✓	✗	Multimodal embedding, neural machine translation.

Table 1. *Cont.*

Category	Method	Target Problem		Techniques
		Local Perturbation	Paraphrase	
Text paraphrase-based methods	Krishna et al. [24]	✗	✓	Retrieval augmentation.
	Wu et al. [51]	✗	✓	Grammar correction model, Grammar Error Correction Score.
	Mao et al. [52]	✗	✓	Rewriting editing distance, prompt engineering.

3.1. Data-Augmentation-Based Methods

Data augmentation is a technique used to generates new samples by applying transformations to existing data. It enriches the training data and prevents the model from overfitting, and thus, enhances the model's robustness. This technique is widely used in fields such as computer vision (CV) [53], speech [54], and natural language processing (NLP) [55]. In AIGT detection, data augmentation can be used to generate perturbed samples, thereby improving the detector's robustness to text perturbations. Wei et al. [37] proposed easy data augmentation techniques, EDA, to improve the performance of text classifiers. EDA consists of four operations: synonym replacement, random insertion, random swap, and random deletion. It can effectively simulate text perturbation scenarios, thus enhancing the robustness of the detector. Karimi et al. [43] proposed an easier data augmentation method AEDA than EDA. This method only involves randomly inserting punctuation marks into the original text, keeping the order of the words while changing the position of each word in the sentence. Unlike the previous methods, which may affect the readability of the text, Coulombe [44] generates augmented samples through the transformation of syntactic trees, thereby maintaining sentence readability. In addition to altering the character composition and expression of the text, Guo et al. [45] proposed wordMixup and senMixup, which perform interpolation on word embeddings and sentence embeddings, respectively. Their studies show that these interpolation strategies are also effective data augmentation methods for sentence classification.

3.2. Adversarial-Training-Based Methods

Adversarial training has proven to be an effective method for defending against adversarial attacks [56]. Its core idea is to augment the training data by introducing adversarial examples in each training loop, thereby improving the model's robustness [57]. Adversarial training typically involves the following steps: generating adversarial samples based on the model's weaknesses, combining them with the original samples to form a new training dataset, and training the model using this augmented dataset. A critical challenge in adversarial training is how to generate high-quality adversarial examples. Zhou et al. [25] proposed a framework for generating adversarial examples based on word importance to bypass detection. They introduce a dual-aspect word importance ranking algorithm that integrates model gradients and perplexity from LLMs. Based on the word importance ranking, the most important words are replaced with their synonyms. Li et al. [46] proposed an adversarial framework named GREATER to train a robust AIGT detector based on Generative Adversarial Networks (GANs) [58]. The framework consists of an adversary GREATER-A and a detector GREATER-D. The GREATER-A is used to generate adversarial samples by replacing the important tokens based on a targeted perturbation on their embedding. The GREATER-D is trained using both the original data and adversarial examples. The two components are trained in a co-training manner, which allows them to continuously enhance each other's capabilities. These methods, which leverage word

importance to generate adversarial samples, are generally effective in strengthening the detector's resistance to local perturbations. However, these methods may alter the original meaning of the text and have limited effectiveness in improving the detector's ability to counter paraphrasing attacks. To resolve these issues, Hu et al. [20] proposed a robust AIGT detector called RADAR. It consists of a paraphraser and a detector. The goal of the paraphraser is to generate paraphrased samples that are undetectable by the detector while preserving the original meaning. The detector continuously improves its robustness against paraphrase attacks by adding the high-quality samples generated by the paraphraser to its training set.

3.3. Noise-Filtering-Based Methods

The goal of noise-filtering-based methods is to eliminate the noise introduced by perturbations, thus preventing it from interfering with the detection results. One type of method removes abnormal or extreme values from features that do not conform to the normal distribution, thereby improving the robustness of AIGT detectors. The key to such methods is how to identify outliers. Cai et al. [47] proposed a modification detection technique for unbiased watermark. They designed a metric named "discarded tokens" to measure the number of tokens not included in watermark detection. When a modification is made, this metric alters, providing evidence of the change. Then, to mitigate the impact of modifications, they dropped the modified tokens according to their proposed metric and use the log-likelihood ratio score to perform robust detection. Wang et al. [48] proposed an anomaly filter method to enhance the robustness of the DetectGPT method [13]. The DetectGPT identify AIGT based on the observation that AIGT is typically located in areas where the log-probability function shows negative curvature. However, text perturbations can significantly reduce the log probability of AIGT, causing it to deviate from the region of negative curvature. So, they prevent the top k% tokens with lowest probabilities from being masked and perturbed for each text when doing mask-filling, as well as from being involved in the computation of log probability in the DetectGPT. In addition to explicit outlier filtering, denoising in the latent space also contributes to improving the model's robustness. Huang et al. [35] proposed the Siamese Calibrated Reconstruction Network (SCRN) to handle adversarial perturbations. SCRN leverages a reconstruction network to add and remove noise from the hidden representation of each token, which helps learn a robust semantic representation against perturbations. It also includes a Siamese calibration technique to ensure equally confident predictions across different noises, thereby improving the robustness against adversarial perturbations.

3.4. Text-Correction-Based Methods

Unlike noise-filtering-based methods that directly remove noise, text-correction-based methods aim to restore noisy samples to their original clean state before perturbations by applying correction techniques, thus mitigating the impact of the perturbations. Pruthi et al. [40] proposed placing a word recognition model before the classifier to identify and correct adversarial misspellings, thereby improving the model's classification accuracy on perturbed samples. Their word recognition models are based on the RNN semicharacter architecture and introduce several novel backoff strategies to handle rare and unseen words. Zhou et al. [49] proposed a framework called DISP to identify and adjust malicious perturbations. They deployed a perturbation discriminator to identify potential perturbations, while an embedding estimator restores the embedding of each potential perturbed word based on its context. Finally, the recovered text is obtained through an approximate kNN search. Li et al. [50] proposed an adversarial defense framework, TEXTSHIELD, based on multimodal embedding and neural machine translation (NMT). They first used an

adversarial NMT model to correct input text, thus denoising adversarial perturbations. Then, they extracted the semantic, glyph, and phonetic features of the corrected text to handle glyph-based and phonetic-based perturbations. Finally, they fused these multimodal features to form a semantic-rich representation for robust classification.

3.5. Text-Paraphrase-Based Methods

Specialized paraphrasers and LLMs can be used to generate multiple paraphrased texts that are semantically identical but syntactically different from a single AIGT, presenting a significant challenge for detectors. However, from another perspective, a paraphraser can be used to create a corpus of paraphrased texts. If the text to be detected matches any entry in this corpus, it can be identified as a paraphrased text. Based on this idea of retrieval-augmented techniques [59,60], Krishna et al. [24] proposed using retrieval on previously generated sequences as a countermeasure to paraphrase attacks. They approach the problem from the perspective of API providers, first storing all the text generated by their LLM in a database. During detection, the text to be checked is compared for similarity with the content in the database. If the similarity is high, the text is considered likely to be AIGT. However, this method has several notable drawbacks: first, since the database is constructed from texts generated by a specific API, it can only detect text generated by that particular API and not content generated by other APIs; second, the API provider must maintain a large database; third, data privacy concerns must also be taken into account. Therefore, this method still has considerable room for improvement. Another type of text-paraphrase-based method finds that AIGT generally has higher quality compared to HWT from the perspective of LLMs. When both types of texts are modified, AIGT tends to have fewer changes than HWT. Based on this, Wu et al. [51] proposed a simple yet effective black-box zero-shot detector. They first used a grammar correction model to rewrite the grammatical errors in the text to be detected. Then, they calculated the Grammar Error Correction Score between the original and the rewritten text to determine whether it is AIGT. Mao et al. [52] identified AIGT by prompting the LLMs to paraphrase the text and calculating the rewriting editing distance between the original and rewritten versions. They designed three prompting methods based on invariance, equivariance, and output uncertainty measurement, respectively, and achieved improved performance across several benchmarks and state-of-the-art detectors. Since this type of methods leverages paraphrase operations, it is inherently robust against paraphrase attacks.

Below, we provide a comprehensive analysis of the aforementioned categories of text perturbation robustness enhancement methods. Data augmentation–based methods are generally more effective for handling local perturbations and are relatively easy to implement, as they do not require modifications to the detector architecture. However, these methods require carefully designed augmentation strategies to avoid introducing unnecessary noise, and the number of augmented samples must be seriously determined. In contrast, adversarial training methods can generate targeted adversarial samples tailored to the specific weaknesses of different models, avoiding the introduction of unnecessary noise. Even so, these methods are computationally expensive, often produce samples with poor interpretability, and can lead to overfitting on adversarial samples, causing a degradation in the detector's performance on clean samples. Noise-filtering-based methods demand a high level of accuracy in identifying noise. Incorrect filtering may result in the loss of important features, which could, in turn, negatively affect the detector's performance. Text-correction-based methods primarily focus on samples with spelling or grammatical errors, making them effective in handling scenarios where malicious local perturbations aim to evade detection. However, since HWT inherently contains more such errors than AIGT, text correction may blur the boundaries of these features between the two. Text-

paraphrase-based methods can effectively counter paraphrase attacks. Nevertheless, these methods often require multiple rewrites of the original text using LLMs, which increases the computational overhead.

4. OOD Robustness

AIGT detection's OOD robustness refers to the detector's ability to generalize and maintain strong performance on text distributions that differ from those encountered during training. This typically encompasses cross-LLM and cross-domain detection. In cross-LLM scenarios, the rapid development of LLMs means that new LLMs are constantly emerging, while existing ones are frequently updated. This requires current detectors to adapt efficiently to the detection of text generated by these new LLMs. In cross-domain detection, the limited training data available for the detector, coupled with the complexity and diversity of real-world data, means that the detector is likely to encounter text domains that were not included in the training set. Here, "domain" includes text genres, topics, and other related aspects. This necessitates that the detector not only performs well on data from familiar domains, but also generalizes effectively to text from unseen domains.

Below, we will introduce some representative methods for enhancing OOD robustness, including contrastive-learning-based methods, statistical-based methods, domain-invariant-feature-based methods, and prompt-based methods, as shown in Table 2.

Table 2. Summary of OOD robustness enhancement methods.

Category	Model	Target Problem		Techniques
		Unseen Domain	Unseen Model	
Contrastive learning-based methods	Bhattacharjee et al. [61]	✗	✓	Synonym replacement, minimizing the Maximum Mean Discrepancy.
	Bhattacharjee et al. [62]	✗	✓	Gradient reversal layer, domain classifier.
	Guo et al. [63]	✗	✓	Multi-level contrastive learning, Training-Free Incremental Adaptation.
	Cava et al. [19]	✗	✓	Triplet-network contrastive learning.
Statistical-based methods	Solaiman et al. [12]	✓	✗	Log-likelihood
	Mitchell et al. [13]	✓	✗	Log-Rank
	Gehrmann et al. [14]	✓	✗	Entropy
	Li et al. [64]	✗	✓	Contrastive features, linear classifier.
	Verma et al. [65]	✓	✓	Feature engineering, logistic regression classifier.
Domain-invariant feature-based methods	Tulchinskii et al. [66]	✓	✓	Intrinsic dimensionality.
	Kuznetsov et al. [67]	✓	✓	Removing detrimental linear subspaces, subspace decomposition, feature selection.
Prompt-based methods	Chen et al. [68]	✓	✗	Inverse prompt,
	Yu et al. [69]	✓	✗	Decoupling prompt and intrinsic characteristics.

4.1. Contrastive-Learning-Based Methods

Contrastive learning is primarily used to learn effective feature representations of data and has been widely applied in fields such as CV [70,71] and NLP [72,73]. It can effectively enhance the model's robustness, generalization ability, and learning capacity, particularly when dealing with few-shot data [61]. Contrastive learning improves the model's representation ability by increasing the similarity of representations between similar samples and decreasing the similarity of representations between dissimilar samples. Bhattacharjee et al. [61] proposed a Contrastive Domain Adaptation framework called ConDA, to adapt to the emergence of new LLMs. They used labeled data from the source LLM, while only unlabeled data is available from the target LLM. They employed synonym replacement to construct positive sample pairs and learn a better text representation by contrastive learning. Then, they mapped data from the source LLM and target LLM to the Reproducing Kernel Hilbert Space, and minimized the Maximum Mean Discrepancy to help the model better adapt to the data distribution of the new model. Bhattacharjee et al. [62] designed a domain generalization framework EAGLE to detect AIGT from unseen target LLMs. Similar to [61], they first constructed a perturbed version of each input and use contrastive learning to learn robust text representations. Differently, they achieved domain generalization through domain adversarial training. They added a gradient reversal layer and a domain classifier after their encoder to make the domain classifier predict domain labels correctly, while the encoder parameters are optimized to deceive the domain classifier, thus learning domain-invariant features. Guo et al. [63] designed a multi-task auxiliary, multi-level contrastive learning framework called DeTeCtive, to learn different writing styles of distinct LLMs. Their approach not only compared AIGT with HWT, but also contrasted text generated by different LLMs and across various LLM series. During model training, both the classification task and the learning of contrasting representations across different models were performed simultaneously. In addition, to better adapt their detector to newly emerged LLMs and avoid retraining the detector, they also designed a Training-Free Incremental Adaptation approach based on the K-Nearest Neighbors (KNN) algorithm. Cava et al. [19] proposed a triplet-network contrastive learning framework WhosAI for AIGT detection and attribution. They first constructed triplets consisting of an anchor sample and its positive sample and negative sample. The triplets were then fed into a triplet network, and the triplet loss was used to constrain the distance between positive pairs to be smaller than the distance between negative pairs. During inference, their detector employed an off-line manner in which the centroids for each category are precomputed. The query text was classified into the category corresponding to its nearest centroid. This method can be easily generalized to new LLMs by computing new centroids using AIGT from corresponding new LLMs.

4.2. Statistical-Based Methods

Statistical-based methods typically utilize surrogate language models, such as GPT-2 [74] and LLaMA [2], in conjunction with feature engineering to extract statistical features from the text. A fixed threshold or a simple machine learning classifier can then be applied to detect AIGT based on these features. Due to the relatively few or even no training parameters involved, they can be easily adapted to a new domain. Some zero-shot detectors extract statistical features such as log-likelihood [12], log-rank [13], or entropy [14] for AIGT detection. Specifically, the text to be detected is fed into the surrogate model, and statistical features above corresponding to each token in the text are obtained following the text generation process. Then, the average of these features across all tokens is calculated and compared to predefined thresholds to determine the text source. For example, AIGT typically exhibits higher average log-likelihood values than HWT. Therefore, when

the average log-likelihood of a text exceeds a certain threshold, the text is classified as AIGT. Since they do not rely on labeled data or model training, they exhibit excellent generalization ability. However, these methods generally have lower accuracy compared to training-based detectors. Other detectors combine the advantages of training-based approaches by incorporating machine learning classifiers, achieving more accurate detection, and offering scalability due to their fewer training parameters. Li et al. [64] proposed a detector named Sniffer to trace and detect AIGT. They first input the training data into several LLMs, and obtained and aligned token-level perplexity of the training data between different LLMs. Then, they constructed contrastive features using perplexity, including the percent-of-low-perplexity score, sentence-level perplexity, and the Pearson and Spearman correlation coefficient between different LLMs. These contrastive features were used to train a linear classifier to predict text origins. When faced with a new LLM, only a linear classifier needs to be retrained to transfer to the new model. Verma et al. [65] proposed the Ghostbuster detector with strong generalization ability across distribution shifts of text domains, prompts, and models. They first used several weaker language models than the target model to compute token probability vectors. They constructed 13 vector and scalar operations to generate features based on token probabilities, and defined a structured search space for feature selection. Finally, they trained a logistic regression classifier on the combination of probability-based features and seven additional features based on word length and the largest token probabilities. Their experimental results demonstrate that Ghostbuster exhibits excellent generalization capability.

4.3. Domain-Invariant-Feature-Based Methods

Domain-invariant features are characteristics that remain consistent across different domains or environments. These features are independent of the data distribution of a specific domain and can generalize across multiple domains, playing a vital role in improving the OOD robustness of detectors. Some studies uncover domain-invariant features by analyzing the embedding space of the text, and use these features for detection. Tulchinskii et al. [66] proposed an invariant feature for HWT, namely the intrinsic dimensionality of the manifold underlying the set of embeddings for a sample. They found that the average intrinsic dimensionality of fluent texts in alphabet-based languages tends to be around 9, while for Chinese, it is approximately 7. In contrast, the average intrinsic dimensionality of AIGT is approximately 1.5 lower for each language, with a clear statistical distinction between HWT and AIGT distributions. Based on this, they designed a score-based detector which uses intrinsic dimensionality estimation for text via persistent homology dimension to approximate the real dimension of a manifold, achieving stable performance across text domains, LLMs, and human writer proficiency levels. Kuznetsov et al. [67] explored the geometry of the embedding space in Transformer-based text encoders and demonstrated that removing detrimental linear subspaces enhances the training of a robust classifier by eliminating domain-specific spurious features. They investigate several subspace decomposition methods and feature selection methods, and their approach outperforms the state-of-the-art methods in cross-domain and cross-generator transfer.

4.4. Prompt-Based Methods

Most detection methods identify AIGT by learning classification features from training samples, which inherently depends on the distribution of the training data. In contrast, prompt-based methods do not directly rely on features of existing samples. Instead, they first reconstruct the prompt that generated the sample to be detected. This prompt is then used as input to an LLM to generate new text. The detection is ultimately based on comparing the differences between the regenerated text and the sample being evaluated.

Chen et al. [68] proposed the Inverse Prompt for AI Detection (IPAD) to achieve robust and interpretable detection results. It consists of a Prompt Inverter, which predicts the prompts that could have generated the input text, and a Distinguisher, which measures how well the input text aligns with the predicted prompt. They designed two versions of the Distinguisher: (1) one identifies AIGT by evaluating the alignment between the predicted prompt and the text to be detected and (2) the other performs detection by comparing the text generated from the predicted prompt with the text to be tested. The two methods both achieve good performance on OOD data. Yu et al. [69] proposed to decouple prompt and intrinsic characteristics for AIGT detection. They employed an auxiliary LLM to infer the prompt associated with the candidate text and used this predicted prompt to generate new text. Then, they compared the similarity between the candidate text and the regenerated text, using this similarity as a feature for detection.

Below, we will provide a comprehensive discussion of the aforementioned methods for enhancing OOD robustness. Contrastive-learning-based methods offer strong transferability and can effectively learn robust feature representations. However, they heavily depend on the design of negative samples, and the computation of positive and negative sample pairs increases the model's computational cost. Moreover, the effectiveness of contrastive learning is highly sensitive to the choice of hyperparameters. Statistical-based methods typically involve fewer training parameters, resulting in lower deployment and computational costs. However, this also limits their performance in practical applications. Domain-invariant-feature-based methods show strong domain adaptability, enabling effective cross-domain task performance without the need for large amounts of labeled data. However, the feature learning process is complex and may result in the loss of crucial domain-specific information. Prompt-based methods harness the powerful generative capabilities and extensive knowledge base of LLMs to improve detector performance in unseen domains. Nonetheless, this type of method is highly sensitive to prompts and requires meticulous prompt engineering.

5. AHT Detection Robustness

Many of the previous detectors were primarily designed to distinguish between pure AIGT and HWT [75,76], meaning the content to be detected has only one attribution. However, in real-world scenarios, there are many AHT, such as human-created text polished by AI, AI-generated text edited by humans, or alternating human and AI creation. Ref. [29] showed that existing detectors have difficulty identifying AHT, especially when handling subtle modifications and variations in writing styles, severely impacting the application of AIGT detectors in real-world scenarios. This hybrid text style severely blurs the boundary between characteristics of AIGT and HWT. Meanwhile, AI's writing features may also be overshadowed by human writing styles, posing a significant challenge for AIGT detection.

We categorize the current AHT detection methods into the following types: sequence-labeling-based methods, segmentation-based methods, AI–human-text-ratio-based methods, and multiclass-classification-based methods, as shown in Table 3. In the following, we will provide a detailed introduction to each type and its representative approaches.

Table 3. Summary of AHT detection robustness enhancement methods.

Category	Model	Techniques
Sequence labeling-based methods	Kadiyala et al. [77]	Multilingual transformer model.
	Wang et al. [78]	Convolutional network, self-attention layer.
Segmentation-based methods	Zeng et al. [79]	Segment detection, segment classification.
	Dugan et al. [80]	Boundary detection.
	Zeng et al. [81]	Boundary detection, prototype.
AI–human text ratio-based methods	Yang et al. [36]	Polish Ratio, Jaccard Distance, Levenshtein Distance.
Multiclass classification-based methods	Abassy et al. [82]	Adding two categories: machine-written, then machine-humanized texts, and human-written, then machine-polished texts.

5.1. Sequence-Labeling-Based Methods

This type of method transforms the original text classification task into a sequence labeling task. It performs binary classification on each token in the input text, and then uses a voting mechanism to select the category with the most tokens as the overall category of the text. Kadiyala et al. [77] used various multilingual transformer models, such as DeBERTaV3 [83] and Longformer [84], with/without additional LSTM [85] or CRF layers [86] to perform AIGT detection on human-LLM co-authored texts by binary token classification. Their methods performed well on texts from unseen domains, unseen generators, and non-native speakers, as well as texts with adversarial inputs. Wang et al. [78] proposed the Sequence X (Check) GPT detector, SeqXGPT. They first used several white-box LLMs to generate word-wise log probability lists of input texts. Then, they regarded these log probability lists as waves in speech signals, using convolutional networks and self-attention layers to extract local and contextualized features. After that, they trained a linear classifier on these contextualized features to predict the label of each word. Finally, they tallied the occurrences of each word label and chose the most frequent label as the final category for each sentence.

5.2. Segmentation-Based Methods

Segmentation-based methods split the detection task into two steps. First, it divides the AHT into multiple segments, each originating from a single source. Then, it classifies each segment individually into its corresponding source. Zeng et al. [79] introduced a two-step segmentation-based pipeline to deal with AHT. They first used a segment detection module to divide a hybrid text into segments, each of which comes from a single authorship. The segment detection module can use Transformer^2 [87], SegFormer [88], and any other text segmentation model. Then, a segment classification module was used to classify the segments obtained in the previous step. The segment classification module can leverage BERT, RoBERTa, and any other text classification model. Dugan et al. [80] formalized the AHT detection task as a boundary detection task to identify the boundary between AIGT and HWT within AHT. However, they only investigated human performance on this task and consider the scenario where there is a single boundary within the AHT. Based on

their idea, Zeng et al. [81] separated AIGT from HWT during the encoder training process. Then, they introduced prototypes, which is the mean of embedding vectors of a set of consecutive sentences, and calculated the distances between every two adjacent prototypes. They argued that a boundary exists between the most distant adjacent prototypes.

5.3. AI–Human-Text-Ratio-Based Methods

Another way to address AHT detection is to determine the degree of involvement of both AI and humans in a text, that is, the proportion of HWT and AIGT in the entire text. The more AI is involved, the larger the proportion of AIGT should be; conversely, the proportion of HWT should be higher. Yang et al. [36] constructed a novel dataset termed HPPT comprising pairs of human-written and ChatGPT-polished abstracts. Based on the sample pairs in the dataset, they proposed a new metric called Polish Ratio. It was used to quantify the extent of text modification before and after being revised with ChatGPT by calculating the Jaccard Distance and Levenshtein Distance between the sample pairs. Afterwards, they trained a regression model to predict the Polish Ratio of a text. In ideal conditions, the Polish Ratio of HWT should be close to 0, while that of AIGT should be close to 1. This method can not only detect AHT but also provide interpretable detection results.

5.4. Multiclass-Classification-Based Methods

This type of method transforms the original binary classification task into a multiclass classification task by breaking down the label space into more granular categories, and then constructs corresponding datasets to train an AHT detector. Abassy et al. [82] proposed a fine-grained AIGT detector, LLM-DetectAIve. They transformed the original binary classification AIGT detection task into a four-class classification, adding two categories: machine-written, then machine-humanized, and human-written, then machine-polished texts.They further constructed a relevant dataset and train a detector using pretrained classifiers. This fine-grained classification approach allows for the seamless transfer of certain existing pure AIGT detection methods to AHT detection with only training on the corresponding dataset. However, it requires careful construction of the dataset.

Below, we will provide a comprehensive analysis of the aforementioned methods for enhancing AHT detection robustness. Sequence-labeling-based methods allow for more fine-grained detection. However, they come with higher computational costs, as they require labeling each token in the sentence. Segmentation-based methods strike a balance between fine-grained detection and efficiency. However, they require careful segmentation, ensuring that the resulting segments retain semantic integrity. AI–human-text-ratio-based methods are relatively scarce. The key challenges are determining the ratio of AHT and constructing the corresponding labeled data. Multiclass-classification-based methods are relatively more intuitive and easier to implement, but they also require the collection of relevant labeled data.

6. Evaluation

In addition to designing effective methods to enhance the robustness of AIGT detection, it is equally important to evaluate the robustness of the detector. Effective evaluation methods, on one hand, can test the validity of an approach, and on the other hand, help identify the weaknesses of a detector, thereby enabling targeted improvements to the model. In this section, we will first introduce commonly used benchmark datasets for evaluating the robustness of AIGT detectors. Next, we will discuss some robustness evaluation methods, which can be used to test different aspects of detector robustness. Finally, we will introduce relevant evaluation metrics, which will provide a more intuitive quantitative comparison of performance across different detectors.

6.1. Benchmark Datasets

In this section, we systematically organize commonly used benchmark datasets in AIGT detection. Table 4 shows the summary of these datasets. These data may come from different LLMs, topics, and genres. They can be used not only to train AIGT detectors but also to evaluate their robustness. Although some datasets may not be directly use to evaluate the robustness of detectors, they can be combined with other techniques to achieve the goal of robustness evaluation. Specifically, by combining these datasets with certain perturbation methods, they can be used to assess the robustness of detectors against text perturbations. Using these datasets from different distributions for training and testing can help evaluate the robustness of detectors to OOD data. Furthermore, these datasets can serve as raw material, in combination with LLMs and prompt engineering, to generate AHT, helping evaluate the robustness of detectors on hybrid texts.

Table 4. Summary of benchmark datasets for evaluating the robustness of AIGT detectors.

Dataset	Size	Lauguage	Multi-LLMs	Multi-Domains	Robust Evaluation
TweepFake [89]	24 k	English	✓	✗	OOD
TuringBench [90]	200 k	English	✓	✗	OOD
HC3 [91]	123 k	English, Chinese	✗	✓	OOD
HC3 Plus [92]	143 k	English, Chinese	✗	✓	OOD
CHEAT [93]	50 k	English	✗	✗	-
OpenLLMText [94]	343 k	English	✓	✗	OOD
ArguGPT [95]	8 k	English	✓	✗	OOD
MAGE [28]	436 k	English	✓	✓	Perturbations, OOD
MGTBench [96]	21 k	English	✓	✓	Perturbations, OOD
IDMGSP [97]	29 k	English	✓	✗	OOD
MULTITuDE [98]	73 k	Arabic, Catalan, Chinese, Czech, Dutch, English, German, Portuguese, Russian, Spanish, Ukrainian	✓	✓	OOD
HANSEN [99]	21 k	English	✓	✗	OOD
M4 [27]	247 k	Arabic, Bulgarian, Chinese, English, Indonesian, Russian, Urdu	✓	✓	OOD
SeqXGPT-Bench [78]	36 k	English	✓	✓	OOD, AHT
Ghostbuster [65]	23 k	English	✓	✓	OOD
HPPT [36]	12 k	English	✗	✗	AHT
SnifferBench [64]	36 k	English	✓	✓	OOD
M4GT-Bench [100]	138 k	Arabic, Bulgarian, Chinese, English, German, Indonesian, Italian, Russian, Urdu	✓	✓	OOD, AHT
DetectRL [101]	234 k	English	✓	✓	Perturbations, OOD
GRiD [102]	6 k	English	✗	✗	-
MIXSET [29]	3.6 k	English	✓	✓	OOD, AHT
LLM-DetectAIve [82]	382 k	English	✓	✓	OOD, AHT
MAiDE-up [103]	20 k	Chinese, English, French, German, Italian, Korean, Romanian, Russian, Spanish, Turkish	✗	✗	OOD
HLU [104]	1 k	Urdu	✓	✓	-

6.2. Robustness Evaluation Methods

In addition to using existing datasets to evaluate the robustness of AIGT detectors, there are also some methods that can be used for robustness evaluation. Below, we will introduce common evaluation techniques for different aspects of detector robustness.

6.2.1. Text Perturbation Robustness Evaluation

The core of testing the robustness of the AIGT detector against text perturbations lies in constructing perturbed samples. We can add random perturbations to the original samples to create these samples. Text data augmentation methods [37] are a great choice for this purpose. These methods typically apply operations like synonym replacement, random insertion, random swap, and random deletion to words or characters in a text to expand the training dataset, thereby enhancing the model's performance. Here, we can apply them to construct perturbed samples to test the robustness of the detector. However, these methods are random in nature and may introduce some bias into the evaluation. Pruthi et al. [40] adopted a greedy strategy, using four operations, i.e., Swap, Drop, Keyboard, and Add, to apply 1-character and 2-character attacks on the text. Specifically, Swap refers to swapping two adjacent characters in a word, Drop refers to dropping an arbitrary character in a word, Keyboard simulates typing errors by replacing a character in a word with a neighboring key on the keyboard, and Add refers to inserting a random character into a word. For 1-character attacks, they test all potential perturbations mentioned above until a perturbed sample that changes the model's prediction is found. For 2-character attacks, they greedily fix the edit with the lowest confidence among the 1-character attacks, then apply all the allowed perturbations to the remaining words. This method alleviates the bias caused by random perturbations to some extent, but increases the time cost.

Adversarial perturbation methods play a significant role in evaluating model robustness. By exploiting the weaknesses of the detector, they can generate adversarial samples in a highly efficient and targeted manner. Ren et al. [105] proposed a synonym substitution algorithm based on word saliency and classification probability. Their method can minimize the classification accuracy with a very low word substitution rate. Gao et al. [22] proposed the DeepWordBug algorithm to generate adversarial samples in a black-box setting. They applied character-level transformations to the critical tokens to minimize the edit distance of the perturbation while still altering the original classification. Zhao et al. [106] exploited a generative adversarial network to generate sentence-level adversarial perturbations. Their method can generate natural and legible adversarial samples to deceive the classifier.

Paraphrase generation [107] can be used to evaluate the AIGT robustness against paraphrase attacks. It can rephrase a given text into different forms while preserving the original meaning. There are many specialized paraphrasers [108,109] that can be used to generate paraphrased samples. They are typically based on pretrained sequence-to-sequence models. In addition, text paraphrasing can also be performed by prompting LLMs. By generating texts in different styles, the robustness of the detector against text paraphrasing can be evaluated.

6.2.2. OOD Robustness Evaluation

Evaluating OOD robustness mainly involves selecting samples from distributions different from the AIGT detector's training set for evaluation. In general, texts from sources, topics, styles, or models, which differ from the detector's training set, can be selected for evaluation. It depends on which aspect of generalization ability is being emphasized in the evaluation. For example, texts generated by LLaMA [2] can be used to train a detector and ChatGPT generated texts can be utilized as the test set to evaluate the detector's robustness on an unseen model.

6.2.3. AHT Detection Robustness Evaluation

To evaluate AHT detection robustness, AHT can be constructed through the following ways: (1) AI generates the text, and humans make modifications; (2) humans create the text, and AI refines it; (3) AI and humans alternate in the creation process; and (4) humans write the beginning of the text, and AI continues it. These methods can all generate high-quality AHT to evaluate AIGT detectors.

6.3. Metrics

Metrics are crucial in model evaluation as they quantify model performance, providing a more intuitive understanding of the model's performance while also guiding model improvements. Next, we will introduce the commonly used evaluation metrics for assessing the robustness of AIGT detector.

We have summarized nine commonly used metrics for evaluating the robustness of AIGT detectors, which are accuracy, precision, recall, F1 score, AUROC, AUA, ASR, QC, and MR. Accuracy represents the proportion of correctly detected samples out of all samples, and is the most direct reflection of the detector's performance. A higher accuracy indicates a larger proportion of correctly detected samples by the model. Precision refers to the proportion of correctly identified samples within one class, either AIGT or HWT, out of all the samples predicted to belong to that class. The higher the precision, the better the detection performance for that particular class. Recall refers to the proportion of correctly detected samples within a specific class out of all the actual samples in that class. It is also used to measure the detector's ability to predict samples belonging to that class. The higher the recall, the more samples of that class are detected correctly. F1 score is the harmonic mean of precision and recall. It combines both metrics and is especially useful in cases of class imbalance, providing a better overall measure of the model's performance. A higher F1 score indicates better detector performance. AUROC evaluates the model's classification ability by plotting the relationship curve between the model's True Positive Rate and False Positive Rate, and calculating the area under the curve. A larger AUROC indicates a stronger detection ability of the detector. AUA [35] denotes the accuracy under attack. It is used to represent the performance of the detector on perturbed texts. The higher its value, the more robust the model is. ASR [110] is the attack success rate, which represents the proportion of adversarial samples that successfully deceive a detector out of all adversarial samples. The smaller its value, the stronger a detector's resistance to interference. QC [110] is the query count, which refers to the number of queries the attacker must make in order to search for a successful adversarial example. The larger its value, the greater the cost of evading detection, indicating that the detector is more robust. MR [110] denotes the modification rate, representing the proportion of words that have been perturbed in the adversarial text. The larger its value, the more modifications are needed to bypass the detection, indicating that the detector is more robust.

7. Experiment

Text perturbations, OOD texts, and AHT pose significant challenges to current AIGT detectors. To quantify the impact of them while revealing the limitations of current AIGT detectors regarding robustness, we design a series of experiments to evaluate several commonly used AIGT detectors, including Log-likelihood, Rank [12], Log-Rank [13], Entropy, GLTR [14], BERT [15], RoBERTa [16], and SCRN [35]. Since Log-likelihood, Rank, Log-Rank, Entropy, and GLTR rely on a surrogate LLM to compute relevant features, we follow the common practice and adopt GPT-2 [74] as the surrogate model. BERT and RoBERTa are fine-tuned using their respective pretrained base versions. Furthermore, for SCRN,

we employ the base version of RoBERTa as the encoder and train the whole framework following the configurations specified in [35].

7.1. Experiment Setup

To evaluate the robustness of AIGT detectors against text perturbations, we sampled 5000 balanced samples from the HC3 dataset [91] as the training set and 1500 balanced samples as the test set, which includes English responses from both human experts and ChatGPT to questions ranging from open-domain, financial, medical, legal, and psychological areas. Following the perturbation methods introduced by Pruthi et al. [40], as described in Section 6.2.1, we applied four types of character-level perturbations: Swap, Drop, Keyboard (Key), and Add, to individual characters (1-char) and two characters (2-char) in the text. Under both levels of perturbation, a detector is considered to make a correct prediction on a sample only if it correctly predicts all the perturbed variants generated from that sample; otherwise, the prediction is regarded as incorrect. Notably, unlike their greedy strategy, which perturbs all potential characters in sequence, we apply only one perturbation per word to improve computational efficiency. This approach strikes a balance between faithfully reflecting the detectors' true robustness and reducing computational costs.

To assess the detectors' OOD robustness and AHT robustness, we replace the previous test set with 1500 balanced samples drawn from the OOD dataset TruthfulQA [111], and another 1500 samples from the AHT dataset MIXSET [29], respectively. It is worth noting that all AHT data are considered as AIGT. Additionally, since the only difference between these two scenarios and the text perturbation robustness evaluation lies in the testing phase, with no variations in the training process, we do not retrain the detectors and instead reuse the same trained models employed in the text perturbation robustness evaluation.

7.2. Experiment Results

7.2.1. Detector Performance on Perturbed Text

Table 5 shows the accuracy of different detectors under text perturbations. It can be observed that under the normal scenario, most detectors achieve high detection accuracy, except for Rank. This indicates that most methods are capable of extracting effective classification features under standard conditions. However, the average probability rank of all tokens plays a relatively limited role in distinguishing between AIGT and HWT. Then, when faced with text perturbations, all detectors show varying degrees of performance degradation, demonstrating that text perturbations pose a significant challenge to all types of AIGT detectors. Besides, all types of perturbations have a certain degree of impact on the accuracy of all detectors. This further highlights the importance of enhancing the robustness of AIGT detectors against text perturbations. Among these methods, SCRN shows relatively strong robustness, achieving optimal performance under various disturbances. This is primarily because the model simulates text perturbations by adding noise to token-level hidden representations and then learns robust feature representations through the process of denoising. Even so, changing just a single character can still affect its detection results. For example, in Table 6, SCRN correctly identifies the original text as AIGT. However, by simply swapping the positions of the letters "i" and "n" in the word "meaning", the text is able to evade SCRN's detection. These results suggest that the robustness of current AIGT detectors remains to be improved.

Table 5. Accuracy (%) of different detectors under text perturbations. Bold font indicates the best accuracy under each test condition.

Model	Normal	Swap		Drop		Add		Key	
		1-Char	2-Char	1-Char	2-Char	1-Char	2-Char	1-Char	2-Char
Log-likelihood [12]	95.87	93.60	87.87	94.13	90.13	93.67	87.93	93.80	88.47
Rank [12]	63.93	34.33	14.27	39.60	14.40	30.33	15.00	30.40	14.40
Log-Rank [13]	97.33	94.33	88.13	95.07	90.33	94.67	88.40	94.53	88.87
Entropy [14]	92.73	86.27	76.87	87.13	79.60	86.47	77.47	86.53	77.40
GLTR [14]	97.07	93.67	88.47	93.47	89.33	93.60	89.13	93.53	88.73
BERT [15]	99.07	95.60	81.73	96.80	87.47	96.07	84.67	97.07	86.33
RoBERTa [16]	99.80	96.87	87.40	96.53	85.13	97.40	87.20	96.60	88.40
SCRN [35]	**99.87**	**99.00**	**96.93**	**99.07**	**97.80**	**99.07**	**97.60**	**98.93**	**97.87**

Table 6. An example illustrating the impact of a text perturbation on the SCRN detection result. Changing the blue text to the red text results in a change in the detection results.

Scenario	Text	Prediction
Original	MS-DRGs (Medicare Severity Diagnosis Related Groups) were implemented by the Centers for Medicare and Medicaid Services (CMS) in October 2007 as a part of the Inpatient Prospective Payment System (IPPS) for hospitals participating in the Medicare program in the United States. MS-DRGs are used to classify hospital stays into distinct groups that are clinically homogeneous, meaning that the patients within each group are expected to have a similar clinical course and resource utilization. MS-DRGs are used to determine the payment that hospitals receive for inpatient stays covered by Medicare.	AIGT
Perturbation	MS-DRGs (Medicare Severity Diagnosis Related Groups) were implemented by the Centers for Medicare and Medicaid Services (CMS) in October 2007 as a part of the Inpatient Prospective Payment System (IPPS) for hospitals participating in the Medicare program in the United States. MS-DRGs are used to classify hospital stays into distinct groups that are clinically homogeneous, meainng that the patients within each group are expected to have a similar clinical course and resource utilization. MS-DRGs are used to determine the payment that hospitals receive for inpatient stays covered by Medicare.	HWT

7.2.2. Detector Performance on OOD Text

Table 7 shows the performance of different detectors on OOD text. We record the accuracy, the detection precision, and recall for human (H.) and AI text, respectively, as well as the overall F1 score of these detectors. It can be observed that compared to the normal scenario in Table 5, which represents the in-distribution detection setting, the accuracy of most detectors drops noticeably in the OOD detection scenario. This indicates that the OOD robustness of the detectors still needs improvement. Even SCRN, which performed well under in-distribution conditions, shows a significant decline in performance on OOD text, suggesting that it may have overfitted to domain-specific features. In contrast, the Log-likelihood model, which previously shows moderate performance, achieves the best results in the OOD setting, with relatively high accuracy and F1 score. This is likely because the method does not rely on the tokens themselves but instead leverages the probability of each token during generation, thereby avoiding dependence on domain-specific textual

features. Additionally, no consistent pattern is observed across detectors in terms of the precision and recall for detecting AIGT and HWT, indicating that the detection difficulty for both types of texts is comparable in OOD scenarios.

Table 7. Performance (%) of different detectors on OOD text. Bold font indicates the highest value of each evaluation metric.

Model	Accuracy	Precision (H.)	Precision (AI)	Recall (H.)	Recall (AI)	F1 Score
Log-likelihood [12]	**89.07**	85.05	94.13	94.80	83.33	**88.40**
Rank [12]	62.27	83.46	61.64	43.07	91.47	73.64
Log-Rank [13]	86.87	81.89	93.68	94.67	79.07	85.76
Entropy [14]	77.47	69.51	96.40	**97.87**	57.07	71.69
GLTR [14]	82.47	80.25	85.04	86.13	78.80	81.80
BERT [15]	84.47	77.41	**96.41**	97.33	71.60	82.17
RoBERTa [16]	85.87	**93.67**	80.43	76.93	**94.80**	87.03
SCRN [35]	78.53	87.81	72.91	66.27	90.80	80.88

7.2.3. Detector Performance on AHT

Table 8 shows the performance of different detectors on AHT. Since the AHT test set contains only AIGT, we report evaluation metrics exclusively for this class, and all precision (AI) scores are 1. Additionally, as the accuracy of the detectors on this dataset is identical to the recall (AI), we omit the accuracy column from the table. It can be observed that the performance of these detectors on AHT data is far from satisfactory, with some evaluation metrics even falling below 50%, clearly demonstrating the limitations of current detectors in identifying AHT. However, as AI–human co-authored text is becoming increasingly common in daily life, this further highlights the urgent need for methods that enhance AHT detection robustness.

Table 8. Performance (%) of different detectors on AHT. Bold font indicates the highest value of each evaluation metric.

Model	Precision (AI)	Recall (AI)	F1 Score
Log-likelihood [12]	100.00	23.40	37.93
Rank [12]	100.00	**65.20**	**78.93**
Log-Rank [13]	100.00	19.47	32.59
Entropy [14]	100.00	29.47	45.52
GLTR [14]	100.00	18.87	31.74
BERT [15]	100.00	39.33	56.46
RoBERTa [16]	100.00	49.67	66.37
SCRN [35]	100.00	60.87	75.67

8. Future Directions

Adversarial Training. Unlike traditional data augmentation techniques, adversarial training generates adversarial samples tailored to address the specific weaknesses of different detectors and various stages of their training. It aims to deceive the detector with minimal alterations to the samples. This approach has proven effective in enhancing model robustness across domains like CV [112] and NLP [113], and is gradually being applied to the field of AIGT detection [20]. Moving forward, incorporating more diverse and efficient attack strategies into adversarial training can further strengthen AIGT detectors against diverse threats. Meanwhile, it is important to consider that as adversarial training advances, it may blur the classification boundaries between AIGT and HWT, which may influence the model's detection performance to original data. Therefore, designing adversarial training methods that strike a balance between improving robustness and maintaining classification performance is essential.

Robust feature representations. Currently, a variety of features are used to distinguish between AIGT and HWT, achieving decent results on texts from specific domains and generated by specific LLMs. However, most of these features are highly sensitive to text perturbations and domain shifts, making them easy to alter, and thus, affecting the accuracy of the detector. As a result, there is still a need to explore more robust features and feature representation methods to enhance the robustness of AIGT detectors.

Interpretability of Detection Results. The interpretability of detection results can help researchers understand the reasoning behind a model's decision making, enabling them to identify the causes of errors when the model makes mistakes and improve the model accordingly. It can also enhance the credibility of the detector and increase user acceptance. Additionally, it allows users to further assess the reliability of the detector's results by combining the interpretable predictions with their own analysis. Specifically, fine-grained AIGT detection can be considered, highlighting parts that exhibit clear AI characteristics and presenting them to users through visualization methods. Furthermore, aligning the detector's confidence with model performance and presenting the model's confidence to users can further strengthen the reliability of the detector.

Versatile AIGT detectors. Most AIGT detectors perform well when distinguishing pure AIGT or pure HWT, but their performance deteriorates when handling AHT. Although there have been some attempts to address AHT detection scenarios [79], there is still much progress to be made. Besides, most detectors perform better on longer texts but struggle with short text detection, which also requires further research. Moreover, in real-world scenarios, people not only expect detectors to identify AIGT but also assess the authenticity of the information within the AIGT. This necessitates detectors that can handle various types of texts and detection tasks simultaneously.

Effective Evaluation Benchmarks. Many studies on AIGT detectors involve dataset collection and the construction of benchmarks. However, they typically reflect ideal testing environments or specific contexts for which their detectors are optimized, leading to poor performance in real-world applications. Therefore, AIGT detection requires more data that aligns with real-world application scenarios, preferably sourced from data generated by people's actual use of LLMs. Additionally, more diverse evaluation methods and metrics need to be designed to effectively assess the performance and robustness of AIGT detectors. A unified evaluation environment and leaderboard also need to be established to facilitate more intuitive horizontal comparisons between different detectors.

Domain Adaptation for AIGT Detectors. The field of LLMs is evolving rapidly, with new models continuously emerging and existing ones being regularly updated, resulting in shifts in their output styles. When detectors face text generated by these emerging LLMs, they may struggle to accurately identify the content due to distributional discrepancies between the styles of the detector's training data and the text from these newer LLMs. However, retraining the detector from scratch is a resource-intensive process. Therefore, designing efficient domain adaptation methods that enable accurate identification of text from new LLMs using only a limited amount of labeled data is essential. This would significantly enhance the ability of AIGT detectors to keep up with the rapid updates and iterations of LLMs.

Noise Learning in AIGT Detection. As LLMs become more widely used in daily life, collecting pure HWT has become increasingly difficult, resulting in the inadvertent inclusion of AHT in the training corpus of detectors. This presents significant challenges for model training. How to effectively train AIGT detectors using corpora that contain noisy samples remains an area that requires further exploration. Therefore, integrating noise learning techniques with AIGT detection methods is of considerable practical significance.

Mathematical Perspectives on AIGT Detection. While many current AIGT detectors have shown decent empirical performance, most approaches lack a solid foundation in mathematical theory, which in turn limits their interpretability and trustworthiness. Investigating the underlying mathematical principles of AIGT detection, and designing detectors guided by rigorous theoretical frameworks, can not only provide empirical effectiveness but also offer provable performance guarantees and improved explainability. Moreover, such integration of theory and practice has the potential to enhance model robustness and inspire the development of more advanced and reliable AIGT detectors.

9. Conclusions

In recent years, researchers have made significant efforts to develop AIGT detectors to enhance the regulation of AIGT usage. However, due to the rapid development of LLMs, the quality of AIGT is continuously improving and the types of text are becoming increasingly diverse and complex, posing significant challenges to AIGT detection. To effectively handle these challenges encountered in real-world scenarios, as well as various attack methods designed to bypass detectors, AIGT detectors must not only achieve high accuracy under ideal detection conditions, but also possess strong robustness in complex detection environments. This paper systematically reviews the current research on enhancing the robustness of AIGT detectors, offering a comprehensive perspective on the current state of the field. We introduce the general definition of the AIGT detection task and categorize the robustness issues of AIGT detectors into three aspects: text perturbation robustness, OOD robustness, and AHT detection robustness. We provide an in-depth discussion of related methods and representative works aimed at enhancing each aspect of robustness. Additionally, we summarize the benchmark datasets, robustness evaluation methods, and metrics used to evaluate detectors' performance. We also conduct experiments to evaluate the robustness of several commonly used detectors, revealing that there is still significant room for improvement in their robustness. Finally, we propose potential future research directions based on the current issues faced by AIGT detectors and the detection requirements in real-world scenarios, providing valuable insights for building more robust and effective AIGT detectors.

Advances in AIGT detection are poised to play an increasingly crucial role in AI governance. Reliable detection of AI-generated content is vital for maintaining transparency, ensuring content authenticity, and preventing the misuse of generative models in sensitive areas such as news dissemination, education, and legal documentation. Furthermore, robust AIGT detection methods can support regulatory bodies in enforcing disclosure requirements and mitigating the risks of disinformation. Therefore, the development and deployment of more powerful and robust AIGT detectors could play a key role in building trustworthy and ethical AI systems.

Author Contributions: Conceptualization, X.L.; methodology, X.L.; formal analysis, X.L. and Y.L.; investigation, X.L.; resources, K.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L., Y.L., and K.L.; visualization, Y.L.; supervision, Y.L. and K.L.; project administration, K.L.; funding acquisition, X.L. and K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Beijing Natural Science Foundation (No. 4222037, L181010) and the China Scholarship Council (No. 202306030179).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35, Proceedings of the Annual Conference on Neural Information Processing Systems 2022 (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022*; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2022.
2. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971. [CrossRef]
3. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.T.; Li, X.; Lin, X.V.; et al. OPT: Open Pre-trained Transformer Language Models. *arXiv* **2022**, arXiv:2205.01068. [CrossRef]
4. Sung, M.; Feng, S.; Gung, J.; Shu, R.; Zhang, Y.; Mansour, S. Structured List-Grounded Question Answering. In Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025), Abu Dhabi, United Arab Emirates, 19–24 January 2025; Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B.D., Schockaert, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 8347–8359.
5. Ma, Y.; Qiao, Y.; Liu, P. MoPS: Modular Story Premise Synthesis for Open-Ended Automatic Story Generation. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), Bangkok, Thailand, 11–16 August 2024; Volume 1: Long Papers; Ku, L., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 2135–2169. [CrossRef]
6. Bernardino, M.; Cargnelutti, R.; Garcia, R.D.S.; Silva, W. The Use of ChatGPT in Improving and Reviewing Scientific Paper Writing: An Exploratory Study. *IEEE Rev. Iberoam. Tecnol. Aprendiz.* **2024**, *19*, 120–128. [CrossRef]
7. Zheng, X.; Luo, M.; Wang, X. Unveiling Fake News with Adversarial Arguments Generated by Multimodal Large Language Models. In Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025), Abu Dhabi, United Arab Emirates, 19–24 January 2025; Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B.D., Schockaert, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 7862–7869.
8. Hysaj, A.; Freeman, M.; Hamam, D. Using AI Tools to Enhance Academic Writing and Maintain Academic Integrity. In *Social Computing and Social Media, Proceedings of the 16th International Conference (SCSM 2024), Held as Part of the 26th HCI International Conference (HCII 2024), Washington, DC, USA, 29 June–4 July 2024*; Coman, A., Vasilache, S., Eds.; Proceedings, Part II; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2024; Volume 14704, pp. 57–66. [CrossRef]
9. Peterson-Salahuddin, C. Repairing the harm: Toward an algorithmic reparations approach to hate speech content moderation. *Big Data Soc.* **2024**, *11*, 20539517241245333. [CrossRef]
10. Wu, J.; Yang, S.; Zhan, R.; Yuan, Y.; Wong, D.F.; Chao, L.S. A Survey on LLM-generated Text Detection: Necessity, Methods, and Future Directions. *arXiv* **2023**, arXiv:2310.14724. [CrossRef]
11. Yang, X.; Pan, L.; Zhao, X.; Chen, H.; Petzold, L.R.; Wang, W.Y.; Cheng, W. A Survey on Detection of LLMs-Generated Content. In Proceedings of the Findings of the Association for Computational Linguistics (EMNLP 2024), Miami, FL, USA, 12–16 November 2024; Al-Onaizan, Y., Bansal, M., Chen, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 9786–9805.
12. Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Wang, J. Release Strategies and the Social Impacts of Language Models. *arXiv* **2019**, arXiv:1908.09203.
13. Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C.D.; Finn, C. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. In Proceedings of the International Conference on Machine Learning (ICML 2023), Honolulu, HI, USA, 23–29 July 2023; Proceedings of Machine Learning Research; Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J., Eds.; PMLR: New York, NY, USA, 2023; Volume 202, pp. 24950–24962.
14. Gehrmann, S.; Strobelt, H.; Rush, A.M. GLTR: Statistical Detection and Visualization of Generated Text. In Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August 2019; Volume 3: System Demonstrations; Costa-jussà, M.R., Alfonseca, E., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 111–116. [CrossRef]
15. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, MN, USA, 2–7 June 2019; Volume 1: Long and Short Papers; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186. [CrossRef]
16. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
17. He, P.; Liu, X.; Gao, J.; Chen, W. Deberta: Decoding-Enhanced Bert with Disentangled Attention. In Proceedings of the 9th International Conference on Learning Representations (ICLR 2021), Virtual Event, Austria, 3–7 May 2021.

18. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.G.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32, Proceedings of the Annual Conference on Neural Information Processing Systems 2019 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019*; Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 5754–5764.
19. Cava, L.L.; Costa, D.; Tagarelli, A. Is Contrasting All You Need? Contrastive Learning for the Detection and Attribution of AI-generated Text. In *Proceedings of the ECAI 2024—27th European Conference on Artificial Intelligence, 19–24 October 2024, Santiago de Compostela, Spain—Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*; Endriss, U., Melo, F.S., Bach, K., Diz, A.J.B., Alonso-Moral, J.M., Barro, S., Heintz, F., Eds.; Frontiers in Artificial Intelligence and Applications; IOS Press: Amsterdam, The Netherlands, 2024; Volume 392, pp. 3179–3186. [CrossRef]
20. Hu, X.; Chen, P.; Ho, T. RADAR: Robust AI-Text Detection via Adversarial Learning. In *Advances in Neural Information Processing Systems 36, Proceedings of the Annual Conference on Neural Information Processing Systems 2023 (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023*; Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S., Eds.; ACM, Inc.: New York, NY, USA, 2023.
21. Kirchenbauer, J.; Geiping, J.; Wen, Y.; Katz, J.; Miers, I.; Goldstein, T. A Watermark for Large Language Models. In Proceedings of the 40th International Conference on Machine Learning (ICML 2023), Honolulu, HI, USA, 23–29 July 2023; Proceedings of Machine Learning Research; Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J., Eds.; PMLR: New York, NY, USA, 2023; Volume 202, pp. 17061–17084.
22. Gao, J.; Lanchantin, J.; Soffa, M.L.; Qi, Y. Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SP Workshops 2018), San Francisco, CA, USA, 24 May 2018; IEEE Computer Society: Los Alamitos, CA, USA, 2018; pp. 50–56. [CrossRef]
23. Peng, X.; Zhou, Y.; He, B.; Sun, L.; Sun, Y. Hidding the Ghostwriters: An Adversarial Evaluation of AI-Generated Student Essay Detection. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 10406–10419. [CrossRef]
24. Krishna, K.; Song, Y.; Karpinska, M.; Wieting, J.; Iyyer, M. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36, Proceedings of the Annual Conference on Neural Information Processing Systems 2023 (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023*; Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S., Eds.; ACM, Inc.: New York, NY, USA, 2023.
25. Zhou, Y.; He, B.; Sun, L. Humanizing Machine-Generated Content: Evading AI-Text Detection through Adversarial Attack. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC/COLING 2024), Torino, Italy, 20–25 May 2024; Calzolari, N., Kan, M., Hoste, V., Lenci, A., Sakti, S., Xue, N., Eds.; ELRA and ICCL: Torino, Italy, 2024; pp. 8427–8437.
26. Antoun, W.; Mouilleron, V.; Sagot, B.; Seddah, D. Towards a Robust Detection of Language Model-Generated Text: Is ChatGPT that easy to detect? In *Actes de CORIA-TALN 2023, Proceedings of the Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles, TALN 2023—Volume 1: Travaux de Recherche Originaux—Articles Longs, Paris, France, 5–9 June 2023*; Servan, C., Vilnat, A., Eds.; ATALA: Paris, France, 2023; pp. 14–27.
27. Wang, Y.; Mansurov, J.; Ivanov, P.; Su, J.; Shelmanov, A.; Tsvigun, A.; Whitehouse, C.; Afzal, O.M.; Mahmoud, T.; Sasaki, T.; et al. M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2024), St. Julian's, Malta, 17–22 March 2024; Volume 1: Long Papers; Graham, Y., Purver, M., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 1369–1407.
28. Li, Y.; Li, Q.; Cui, L.; Bi, W.; Wang, Z.; Wang, L.; Yang, L.; Shi, S.; Zhang, Y. MAGE: Machine-generated Text Detection in the Wild. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), Bangkok, Thailand, 11–16 August 2024; Volume 1: Long Papers; Ku, L., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 36–53. [CrossRef]
29. Zhang, Q.; Gao, C.; Chen, D.; Huang, Y.; Huang, Y.; Sun, Z.; Zhang, S.; Li, W.; Fu, Z.; Wan, Y.; et al. LLM-as-a-Coauthor: Can Mixed Human-Written and Machine-Generated Text Be Detected? In Proceedings of the Findings of the Association for Computational Linguistics (NAACL 2024), Mexico City, Mexico, 16–21 June 2024; Duh, K., Gómez-Adorno, H., Bethard, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 409–436. [CrossRef]
30. Yang, Z.; Feng, Z.; Huo, R.; Lin, H.; Zheng, H.; Nie, R.; Chen, H. The Imitation Game revisited: A comprehensive survey on recent advances in AI-generated text detection. *Expert Syst. Appl.* **2025**, *272*, 126694. [CrossRef]
31. Valiaiev, D. Detection of Machine-Generated Text: Literature Survey. *arXiv* **2024**, arXiv:2402.01642.
32. Freiesleben, T.; Grote, T. Beyond generalization: A theory of robustness in machine learning. *Synthese* **2023**, *202*, 109. [CrossRef]
33. Liu, J.; Jin, Y. A comprehensive survey of robust deep learning in computer vision. *J. Autom. Intell.* **2023**, *2*, 175–195. [CrossRef]

34. Jin, Y.; Sendhoff, B. Trade-off between performance and robustness: An evolutionary multiobjective approach. In Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Faro, Portugal, 8–11 April 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 237–251.
35. Huang, G.; Zhang, Y.; Li, Z.; You, Y.; Wang, M.; Yang, Z. Are AI-Generated Text Detectors Robust to Adversarial Perturbations? In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), Bangkok, Thailand, 11–16 August 2024; Volume 1: Long Papers; Ku, L., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 6005–6024. [CrossRef]
36. Yang, L.; Jiang, F.; Li, H. Is ChatGPT Involved in Texts? Measure the Polish Ratio to Detect ChatGPT-Generated Text. *arXiv* **2023**, arXiv:2307.11380. [CrossRef]
37. Wei, J.W.; Zou, K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. 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 2019), Hong Kong, China, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 6381–6387. [CrossRef]
38. Zhang, K.; Hei, X.; Fei, R.; Guo, Y.; Jiao, R. Cross-Domain Text Classification Based on BERT Model. In *Database Systems for Advanced Applications, Proceedings of the DASFAA 2021 International Workshops—BDQM, GDMA, MLDLDSA, MobiSocial, and MUST, Taipei, Taiwan, 11–14 April 2021*; Proceedings; Lecture Notes in Computer Science; Jensen, C.S., Lim, E., Yang, D., Chang, C., Xu, J., Peng, W., Huang, J., Shen, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12680, pp. 197–208. [CrossRef]
39. Cai, S.; Cui, W. Evade ChatGPT Detectors via A Single Space. *arXiv* **2023**, arXiv:2307.02599. [CrossRef]
40. Pruthi, D.; Dhingra, B.; Lipton, Z.C. Combating Adversarial Misspellings with Robust Word Recognition. In Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August 2019; Volume 1: Long Papers; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 5582–5591. [CrossRef]
41. Jia, X.; Mao, Z.; Zhang, Z.; Lv, Q.; Wang, X.; Wu, G. Syntax-controlled paraphrases generation with VAE and multi-task learning. *Comput. Speech Lang.* **2025**, *89*, 101705. [CrossRef]
42. Sadasivan, V.S.; Kumar, A.; Balasubramanian, S.; Wang, W.; Feizi, S. Can AI-Generated Text be Reliably Detected? *arXiv* **2023**, arXiv:2303.11156. [CrossRef]
43. Karimi, A.; Rossi, L.; Prati, A. AEDA: An Easier Data Augmentation Technique for Text Classification. In Proceedings of the Findings of the Association for Computational Linguistics (EMNLP 2021) Virtual Event/Punta Cana, Dominican Republic, 16–20 November 2021; Moens, M., Huang, X., Specia, L.,Yih, S.W., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2748–2754. [CrossRef]
44. Coulombe, C. Text Data Augmentation Made Simple By Leveraging NLP Cloud APIs. *arXiv* **2018**, arXiv:1812.04718.
45. Guo, H.; Mao, Y.; Zhang, R. Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *arXiv* **2019**, arXiv:1905.08941.
46. Li, Y.; Zhang, Z.; Li, C.; Shen, C.; Liu, X. Iron Sharpens Iron: Defending Against Attacks in Machine-Generated Text Detection with Adversarial Training. *arXiv* **2025**, arXiv:2502.12734. [CrossRef]
47. Cai, Y.; Wang, Y.; Hu, D.; Gu, C. Modification and Generated-Text Detection: Achieving Dual Detection Capabilities for the Outputs of LLM by Watermark. *arXiv* **2025**, arXiv:2502.08332. [CrossRef]
48. Wang, Y.; Feng, S.; Hou, A.B.; Pu, X.; Shen, C.; Liu, X.; Tsvetkov, Y.; He, T. Stumbling Blocks: Stress Testing the Robustness of Machine-Generated Text Detectors Under Attacks. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), Bangkok, Thailand, 11–16 August 2024; Volume 1: Long Papers; Ku, L., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 2894–2925. [CrossRef]
49. Zhou, Y.; Jiang, J.; Chang, K.; Wang, W. Learning to Discriminate Perturbations for Blocking Adversarial Attacks in Text Classification. 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 2019), Hong Kong, China, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4903–4912. [CrossRef]
50. Li, J.; Du, T.; Ji, S.; Zhang, R.; Lu, Q.; Yang, M.; Wang, T. TextShield: Robust Text Classification Based on Multimodal Embedding and Neural Machine Translation. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 2020), Boston, MA, USA, 12–14 August 2020; Capkun, S., Roesner, F., Eds.; USENIX Association: Berkeley, CA, USA, 2020; pp. 1381–1398.
51. Wu, J.; Zhan, R.; Wong, D.F.; Yang, S.; Liu, X.; Chao, L.S.; Zhang, M. Who Wrote This? The Key to Zero-Shot LLM-Generated Text Detection Is GECScore. In Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025), Abu Dhabi, United Arab Emirates, 19–24 January 2025; Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B.D., Schockaert, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 10275–10292.
52. Mao, C.; Vondrick, C.; Wang, H.; Yang, J. Raidar: GeneRative AI Detection viA Rewriting. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024), Vienna, Austria, 7–11 May 2024.

53. Buslaev, A.V.; Parinov, A.; Khvedchenya, E.; Iglovikov, V.I.; Kalinin, A.A. Albumentations: Fast and flexible image augmentations. *arXiv* **2018**, arXiv:1809.06839. [CrossRef]
54. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In Proceedings of the 20th Annual Conference of the International Speech Communication Association, Interspeech 2019, Graz, Austria, 15–19 September 2019; Kubin, G., Kacic, Z., Eds.; ISCA: Singapore, 2019; pp. 2613–2617. [CrossRef]
55. Feng, S.Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; Hovy, E.H. A Survey of Data Augmentation Approaches for NLP. In Proceedings of the Findings of the Association for Computational Linguistics (ACL/IJCNLP 2021), Online Event, 1–6 August 2021; Volume ACL/IJCNLP 2021, Findings of ACL; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 968–988. [CrossRef]
56. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018; Conference Track Proceedings; 2018.
57. Bai, T.; Luo, J.; Zhao, J.; Wen, B.; Wang, Q. Recent Advances in Adversarial Training for Adversarial Robustness. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI 2021), Virtual Event/Montreal, Canada, 19–27 August 2021; Zhou, Z., Ed.; 2021; pp. 4312–4321. [CrossRef]
58. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
59. Siriwardhana, S.; Weerasekera, R.; Kaluarachchi, T.; Wen, E.; Rana, R.; Nanayakkara, S. Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. *Trans. Assoc. Comput. Linguist.* **2023**, *11*, 1–17. [CrossRef]
60. Cai, D.; Wang, Y.; Liu, L.; Shi, S. Recent Advances in Retrieval-Augmented Text Generation. In Proceedings of the SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G., Eds.; ACM: New York, NY, USA, 2022; pp. 3417–3419. [CrossRef]
61. Bhattacharjee, A.; Kumarage, T.; Moraffah, R.; Liu, H. ConDA: Contrastive Domain Adaptation for AI-generated Text Detection. In Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP 2023), Nusa Dua, Bali, 1–4 November 2023; Volume 1: Long Papers; Park, J.C., Arase, Y., Hu, B., Lu, W., Wijaya, D., Purwarianti, A., Krisnadhi, A.A., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 598–610. [CrossRef]
62. Bhattacharjee, A.; Moraffah, R.; Garland, J.; Liu, H. EAGLE: A Domain Generalization Framework for AI-generated Text Detection. *arXiv* **2024**, arXiv:2403.15690. [CrossRef]
63. Guo, X.; He, Y.; Zhang, S.; Zhang, T.; Feng, W.; Huang, H.; Ma, C. DeTeCtive: Detecting AI-generated Text via Multi-Level Contrastive Learning. In *Advances in Neural Information Processing Systems 38, Proceedings of the Annual Conference on Neural Information Processing Systems 2024 (NeurIPS 2024), Vancouver, BC, Canada, 10–15 December 2024*; Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2024.
64. Li, L.; Wang, P.; Ren, K.; Sun, T.; Qiu, X. Origin Tracing and Detecting of LLMs. *arXiv* **2023**, arXiv:2304.14072.
65. Verma, V.; Fleisig, E.; Tomlin, N.; Klein, D. Ghostbuster: Detecting Text Ghostwritten by Large Language Models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2024), Mexico City, Mexico, 16–21 June 2024; Volume 1: Long Papers; Duh, K., Gómez-Adorno, H., Bethard, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 1702–1717. [CrossRef]
66. Tulchinskii, E.; Kuznetsov, K.; Kushnareva, L.; Cherniavskii, D.; Nikolenko, S.I.; Burnaev, E.; Barannikov, S.; Piontkovskaya, I. Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts. In *Advances in Neural Information Processing Systems 36, Proceedings of the Annual Conference on Neural Information Processing Systems 2023 (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023*; Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2023.
67. Kuznetsov, K.; Tulchinskii, E.; Kushnareva, L.; Magai, G.; Barannikov, S.; Nikolenko, S.I.; Piontkovskaya, I. Robust AI-Generated Text Detection by Restricted Embeddings. In Proceedings of the Findings of the Association for Computational Linguistics (EMNLP 2024), Miami, FL, USA, 12–16 November 2024; Al-Onaizan, Y., Bansal, M., Chen, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 17036–17055.
68. Chen, Z.; Feng, Y.; He, C.; Deng, Y.; Pu, H.; Li, B. IPAD: Inverse Prompt for AI Detection—A Robust and Explainable LLM-Generated Text Detector. *arXiv* **2025**, arXiv:2502.15902.

69. Yu, X.; Qi, Y.; Chen, K.; Chen, G.; Yang, X.; Zhu, P.; Shang, X.; Zhang, W.; Yu, N. DPIC: Decoupling Prompt and Intrinsic Characteristics for LLM Generated Text Detection. In *Advances in Neural Information Processing Systems 38, Proceedings of the Annual Conference on Neural Information Processing Systems 2024 (NeurIPS 2024), Vancouver, BC, Canada, 10–15 December 2024*; Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2024.
70. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R.B. Momentum Contrast for Unsupervised Visual Representation Learning. *arXiv* **2019**, arXiv:1911.05722.
71. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G.E. A Simple Framework for Contrastive Learning of Visual Representations. In Proceedings of the 37th International Conference on Machine Learning (ICML 2020), Virtual Event, 13–18 July 2020; Proceedings of Machine Learning Research; PMLR: New York, NY, USA, 2020; Volume 119, pp. 1597–1607.
72. Zhang, Y.; Zhang, R.; Mensah, S.; Liu, X.; Mao, Y. Unsupervised Sentence Representation via Contrastive Learning with Mixing Negatives. In Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022), Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence (IAAI 2022), The Twelveth Symposium on Educational Advances in Artificial Intelligence (EAAI 2022), Virtual Event, 22 February–1 March 2022; AAAI Press: Washington, DC, USA, 2022; pp. 11730–11738. [CrossRef]
73. Zhou, K.; Zhang, B.; Zhao, W.X.; Wen, J. Debiased Contrastive Learning of Unsupervised Sentence Representations. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022), Dublin, Ireland, 22–27 May 2022; Volume 1: Long Papers; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 6120–6130. [CrossRef]
74. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; Open and Efficient. *OpenAI Blog* **2019**, *1*, 9.
75. Su, J.; Zhuo, T.Y.; Wang, D.; Nakov, P. DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection of Machine-Generated Text. In Proceedings of the Findings of the Association for Computational Linguistics (EMNLP 2023), Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 12395–12412. [CrossRef]
76. Bao, G.; Zhao, Y.; Teng, Z.; Yang, L.; Zhang, Y. Fast-DetectGPT: Efficient Zero-Shot Detection of Machine-Generated Text via Conditional Probability Curvature. In Proceedings of the The Twelfth International Conference on Learning Representations (ICLR 2024), Vienna, Austria, 7–11 May 2024.
77. Kadiyala, R.M.R.; Pullakhandam, S.; Mehreen, K.; Sharma, D.; Gupta, S.; Purbey, J.; Srivastava, A.; TippaReddy, S.; Bobbili, A.R.; Chandrashekhar, S.T.; et al. Robust and Fine-Grained Detection of AI Generated Texts. *arXiv* **2025**, arXiv:2504.11952.
78. Wang, P.; Li, L.; Ren, K.; Jiang, B.; Zhang, D.; Qiu, X. SeqXGPT: Sentence-Level AI-Generated Text Detection. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 1144–1156. [CrossRef]
79. Zeng, Z.; Liu, S.; Sha, L.; Li, Z.; Yang, K.; Liu, S.; Gasevic, D.; Chen, G. Detecting AI-Generated Sentences in Human-AI Collaborative Hybrid Texts: Challenges, Strategies, and Insights. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI 2024), Jeju, Republic of Korea, 3–9 August 2024 ; pp. 7545–7553.
80. Dugan, L.; Ippolito, D.; Kirubarajan, A.; Shi, S.; Callison-Burch, C. Real or Fake Text?: Investigating Human Ability to Detect Boundaries between Human-Written and Machine-Generated Text. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023), Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence (IAAI 2023), Thirteenth Symposium on Educational Advances in Artificial Intelligence (EAAI 2023), Washington, DC, USA, 7–14 February 2023; Williams, B., Chen, Y., Neville, J., Eds.; AAAI Press: Washington, DC, USA, 2023; pp. 12763–12771. [CrossRef]
81. Zeng, Z.; Sha, L.; Li, Y.; Yang, K.; Gasevic, D.; Chen, G. Towards Automatic Boundary Detection for Human-AI Collaborative Hybrid Essay in Education. In Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2024), Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence (IAAI 2024), Fourteenth Symposium on Educational Advances in Artificial Intelligence (EAAI 2014), Vancouver, BC, Canada, 20–27 February 2024; Wooldridge, M.J., Dy, J.G., Natarajan, S., Eds.; AAAI Press: Washington, DC, USA, 2024; pp. 22502–22510. [CrossRef]
82. Abassy, M.; Elozeiri, K.A.; Aziz, A.; Ta, M.N.; Tomar, R.V.; Adhikari, B.; Ahmed, S.E.D.; Wang, Y.; Afzal, O.M.; Xie, Z.; et al. LLM-DetectAIve: A Tool for Fine-Grained Machine-Generated Text Detection. *arXiv* **2024**, arXiv:2408.04284. [CrossRef]
83. He, P.; Gao, J.; Chen, W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv* **2021**, arXiv:2111.09543.
84. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* **2020**, arXiv:2004.05150.
85. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
86. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H.S. Conditional Random Fields as Recurrent Neural Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, 7–13 December 2015; pp. 1529–1537. [CrossRef]

87. Lo, K.; Jin, Y.; Tan, W.; Liu, M.; Du, L.; Buntine, W.L. Transformer over Pre-trained Transformer for Neural Text Segmentation with Enhanced Topic Coherence. In Proceedings of the Findings of the Association for Computational Linguistics (EMNLP 2021), Virtual Event/Punta Cana, Dominican Republic, 16–20 November 2021; Moens, M., Huang, X., Specia, L., Yih, S.W., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 3334–3340. [CrossRef]
88. Bai, H.; Wang, P.; Zhang, R.; Su, Z. SegFormer: A Topic Segmentation Model with Controllable Range of Attention. In Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2023), Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence (IAAI 2023), Thirteenth Symposium on Educational Advances in Artificial Intelligence (EAAI 2023), Washington, DC, USA, 7–14 February 2023; Williams, B., Chen, Y., Neville, J., Eds.; AAAI Press: Washington, DC, USA, 2023; pp. 12545–12552. [CrossRef]
89. Fagni, T.; Falchi, F.; Gambini, M.; Martella, A.; Tesconi, M. TweepFake: About Detecting Deepfake Tweets. *arXiv* **2020**, arXiv:2008.00036. [CrossRef] [PubMed]
90. Uchendu, A.; Ma, Z.; Le, T.; Zhang, R.; Lee, D. TURINGBENCH: A Benchmark Environment for Turing Test in the Age of Neural Text Generation. In Proceedings of the Findings of the Association for Computational Linguistics (EMNLP 2021), Virtual Event/Punta Cana, Dominican Republic, 16–20 November 2021; Moens, M., Huang, X., Specia, L., Yih, S.W., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2001–2016. [CrossRef]
91. Guo, B.; Zhang, X.; Wang, Z.; Jiang, M.; Nie, J.; Ding, Y.; Yue, J.; Wu, Y. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv* **2023**, arXiv:2301.07597. [CrossRef]
92. Su, Z.; Wu, X.; Zhou, W.; Ma, G.; Hu, S. HC3 Plus: A Semantic-Invariant Human ChatGPT Comparison Corpus. *arXiv* **2023**, arXiv:2309.02731. [CrossRef]
93. Yu, P.; Chen, J.; Feng, X.; Xia, Z. CHEAT: A Large-scale Dataset for Detecting ChatGPT-writtEn AbsTracts. *arXiv* **2023**, arXiv:2304.12008. [CrossRef]
94. Chen, Y.; Kang, H.; Zhai, V.; Li, L.; Singh, R.; Raj, B. Token Prediction as Implicit Classification to Identify LLM-Generated Text. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 13112–13120. [CrossRef]
95. Liu, Y.; Zhang, Z.; Zhang, W.; Yue, S.; Zhao, X.; Cheng, X.; Zhang, Y.; Hu, H. ArguGPT: Evaluating, understanding and identifying argumentative essays generated by GPT models. *arXiv* **2023**, *arXiv:2304.07666*. [CrossRef]
96. He, X.; Shen, X.; Chen, Z.; Backes, M.; Zhang, Y. MGTBench: Benchmarking Machine-Generated Text Detection. In Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security (CCS 2024), Salt Lake City, UT, USA, 14–18 October 2024; Luo, B., Liao, X., Xu, J., Kirda, E., Lie, D., Eds.; ACM: New York, NY, USA, 2024; pp. 2251–2265. [CrossRef]
97. Mosca, E.; Abdalla, M.H.I.; Basso, P.; Musumeci, M.; Groh, G. Distinguishing Fact from Fiction: A Benchmark Dataset for Identifying Machine-Generated Scientific Papers in the LLM Era. In Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023), Toronto, ON, Canada, 14 July 2023; pp. 190–207.
98. Macko, D.; Móro, R.; Uchendu, A.; Lucas, J.S.; Yamashita, M.; Pikuliak, M.; Srba, I.; Le, T.; Lee, D.; Simko, J.; et al. MULTITuDE: Large-Scale Multilingual Machine-Generated Text Detection Benchmark. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 9960–9987. [CrossRef]
99. Tripto, N.I.; Uchendu, A.; Le, T.; Setzu, M.; Giannotti, F.; Lee, D. HANSEN: Human and AI Spoken Text Benchmark for Authorship Analysis. In Proceedings of the Findings of the Association for Computational Linguistics (EMNLP 2023), Singapore, 6–10 December 2023; Bouamor, H., Pino, J., Bali, K., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 13706–13724. [CrossRef]
100. Wang, Y.; Mansurov, J.; Ivanov, P.; Su, J.; Shelmanov, A.; Tsvigun, A.; Afzal, O.M.; Mahmoud, T.; Puccetti, G.; Arnold, T.; et al. M4GT-Bench: Evaluation Benchmark for Black-Box Machine-Generated Text Detection. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), Bangkok, Thailand, 11–16 August 2024; Volume 1: Long Papers; Ku, L., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 3964–3992. [CrossRef]
101. Wu, J.; Zhan, R.; Wong, D.F.; Yang, S.; Yang, X.; Yuan, Y.; Chao, L.S. DetectRL: Benchmarking LLM-Generated Text Detection in Real-World Scenarios. In *Advances in Neural Information Processing Systems 38, Proceedings of the Annual Conference on Neural Information Processing Systems 2024 (NeurIPS 2024), Vancouver, BC, Canada, 10–15 December 2024*; Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J.M., Zhang, C., Eds.; Curran Associates Inc.: Red Hook, NY, USA, 2024.
102. Qazi, Z.; Shiao, W.; Papalexakis, E.E. GPT-generated Text Detection: Benchmark Dataset and Tensor-based Detection Method. In Proceedings of the Companion ACM on Web Conference 2024 (WWW 2024), Singapore, 13–17 May 2024; Chua, T., Ngo, C., Lee, R.K., Kumar, R., Lauw, H.W., Eds.; ACM: New York, NY, USA, 2024; pp. 842–846. [CrossRef]
103. Ignat, O.; Xu, X.; Mihalcea, R. MAiDE-up: Multilingual Deception Detection of GPT-generated Hotel Reviews. *arXiv* **2024**, arXiv:2404.12938. [CrossRef]

104. Ali, I.; Atuhurra, J.; Kamigaito, H.; Watanabe, T. HLU: Human Vs LLM Generated Text Detection Dataset for Urdu at Multiple Granularities. In Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025), Abu Dhabi, United Arab Emirates, 19–24 January 2025; Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B.D., Schockaert, S., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 3495–3510.
105. Ren, S.; Deng, Y.; He, K.; Che, W. Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. In Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Florence, Italy, 28 July–2 August 2019; Volume 1: Long Papers; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 1085–1097. [CrossRef]
106. Zhao, Z.; Dua, D.; Singh, S. Generating Natural Adversarial Examples. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018; Conference Track Proceedings; 2018.
107. Ogasa, Y.; Kajiwara, T.; Arase, Y. Controllable Paraphrase Generation for Semantic and Lexical Similarities. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC/COLING 2024), Torino, Italy, 20–25 May 2024; Calzolari, N., Kan, M., Hoste, V., Lenci, A., Sakti, S., Xue, N., Eds.; ELRA and ICCL: Torino, Italy, 2024; pp. 3927–3942.
108. Razaq, A.; Shah, B.; Khan, G.; Alfandi, O.; Ullah, A.; Halim, Z.; Rahman, A.U. Improving paraphrase generation using supervised neural-based statistical machine translation framework. *Neural Comput. Appl.* **2025**, *37*, 7705–7719. [CrossRef]
109. Ouahrani, L.; Bennouar, D. Paraphrase Generation and Supervised Learning for Improved Automatic Short Answer Grading. *Int. J. Artif. Intell. Educ.* **2024**, *34*, 1627–1670. [CrossRef]
110. Wang, Z.; Liu, Z.; Zheng, X.; Su, Q.; Wang, J. RMLM: A Flexible Defense Framework for Proactively Mitigating Word-level Adversarial Attacks. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023), Toronto, Canada, 9–14 July 2023; Volume 1: Long Papers; Rogers, A., Boyd-Graber, J.L., Okazaki, N., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 2757–2774. [CrossRef]
111. Lin, S.; Hilton, J.; Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv* **2021**, arXiv:2109.07958.
112. Pasquadibisceglie, V.; Appice, A.; Castellano, G.; Malerba, D. JARVIS: Joining Adversarial Training With Vision Transformers in Next-Activity Prediction. *IEEE Trans. Serv. Comput.* **2024**, *17*, 1593–1606. [CrossRef]
113. Feng, X.; Gu, T.; Liu, X.; Chang, L. Learning from Mistakes: Self-correct Adversarial Training for Chinese Unnatural Text Correction. In Proceedings of the AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; Walsh, T., Shah, J., Kolter, Z., Eds. AAAI Press: Washington, DC, USA, 2025; pp. 23887–23895. [CrossRef]