AIDER: a Robust and Topic-Independent Framework for Detecting AI-Generated Text

Jiayi Gui[1], Baitong Cui[1], Xiaolian Guo[1], Ke Yu[1‡], Xiaofei Wu[1]
[1]School of Artificial Intelligence, Beijing University of Posts and Telecommunications
‡**Correspondence:** yuke@bupt.edu.cn

Abstract

The human-level fluency achieved by large language models in text generation has intensified the challenge of distinguishing between human-written and AI-generated texts. While current fine-tuned detectors exist, they often lack robustness against adversarial attacks and struggle with out-of-distribution topics, limiting their practical applicability. This study introduces **AIDER**, a robust and topic-independent AI-generated text detection framework. AIDER leverages the ALBERT model for topic content disentanglement, enhancing transferability to unseen topics. It incorporates an augmentor that generates robust adversarial data for training, coupled with contrastive learning techniques to boost resilience. Comprehensive experiments demonstrate AIDER's significant superiority over state-of-the-art methods, exhibiting exceptional robustness against adversarial attacks with minimal performance degradation. AIDER consistently achieves high accuracy in non-augmented scenarios and demonstrates remarkable generalizability to unseen topics. These attributes establish AIDER as a powerful and versatile tool for LLM-generated text detection across diverse real-world applications, addressing critical challenges in the evolving landscape of AI-generated content.

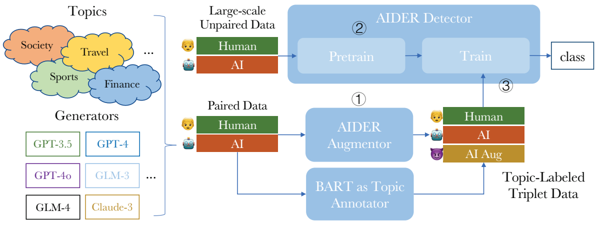


Figure 1: Overview of the AIDER framework. AIDER is designed to detect texts generated by various AI generators and across diverse topics. The framework consists of three main steps: In **Step 1**, AI-generated texts are augmented in the Augmentor module. In **Step 2**, a large corpus of human-written and AI-generated texts is used to train the model based on ALBERT in the first stage of the Detector module. In **Step 3**, paired human-written, AI-generated, and AI-augmented texts go through further training with ALBERT in the second stage of the Detector module to predict the final class.

1 Introduction

AI-generated text detection is the task of distinguishing AI-generated from human-written text. The rapid advancement of large language models (LLMs) like GPT(Achiam et al., 2023; Floridi and Chiriatti, 2020), Claude(Anthropic, 2023), Mistral(Jiang et al., 2023), GLM(GLM et al., 2024; Zeng et al., 2022), and Llama(Touvron et al., 2023) has made this task increasingly challenging. This raises concerns in academic integrity, journalistic authenticity, social media distortion, and legal credibility.

Developing effective methods for AI-generated text detection has become a crucial research area, with researchers exploring various approaches including: watermarking(Christ et al., 2024; Kirchenbauer et al., 2023), zero-shot detectors(Mitchell et al., 2023; Bhattacharjee and Liu, 2024), and fine-tuned classifiers(Hu et al., 2023; Guo et al., 2023).

However, existing detectors face significant challenges on two fronts. Firstly, they are highly vulnerable to various attack techniques, including paraphrasing (Krishna et al., 2024), adversarial (Hu et al., 2023), and prompt (Wu et al., 2023) attacks. These methods can drastically reduce detection accuracy by altering text structure or exploiting model weaknesses while preserving content meaning, leading to an arms race between detectors and evasion techniques. Secondly, these detectors struggle with transferability to unseen topics (Li et al., 2024). These vulnerabilities arise from different sources: attack techniques exploit the detectors' lack of exposure to adversarial samples, while topic transferability issues stem from limited generalization across diverse content domains. Ideally, the intrinsic style of expressing the same content

Proceedings of the 31st International Conference on Computational Linguistics, pages 9299–9310
January 19–24, 2025. ©2025 Association for Computational Linguistics

should distinguish AI from human texts. However, semantic variations introduce significant randomness in real-world scenarios, making it imperative to extract latent characteristics beyond mere semantics. Together, these challenges significantly hinder the effectiveness of AI-generated text detection in real-world scenarios, potentially leading to widespread undetected AI-generated content across various fields.

To address the limitations of existing AI-generated text detectors, we propose **AIDER**, a robust and topic-independent framework for detecting AI-generated content. The workflow of AIDER are illustrated in Figure 1. AIDER is designed to overcome vulnerabilities to various attack techniques and enhance generalization across diverse topics, ensuring broad practical applicability. To implement this comprehensive approach, AIDER's architecture consists of two key components: a **detector** and an **augmentor**. The **detector**, built on the lightweight ALBERT(Lan et al., 2019) model, employs a two-stage training paradigm. The first stage discerns fundamental differences between human-written and AI-generated texts, while the second stage introduces a contrastive label prediction module with triplet loss. A topic disentanglement module is incorporated to focus on generation-specific features, enhancing detection accuracy across varied topics. The **augmentor**, powered by large language models (LLMs), generates challenging samples that are fed into the detector's second-stage training. This synergistic process enhances the detector's robustness against a wide spectrum of evasion methods. By continuously exposing the detector to these challenging samples, the detector strengthens its resilience to various attack techniques, ensuring maintained accuracy in real-world scenarios. To support the development and evaluation of AIDER, we introduce the **AIGen** dataset, comprising approximately 24,000 human-written, AI-generated, and AI-augmented triplets. This comprehensive dataset ensures thorough training and evaluation across diverse topics and generation models.

2 Related Works

AI-Generated Text Detectors Contemporary methodologies for post-hoc AI-generated text detection can be broadly classified into three categories: watermark-based detectors, zero-shot-based detectors, and fine-tuned detectors(Ghosal et al., 2023). Watermark-based detectors (Kirchenbauer et al., 2023; Christ et al., 2024; Zhao et al., 2023) require access to the language model to embed signals in generated text, limiting their effectiveness when the text source is unknown; Zero-shot-based detectors like DetectGPT (Mitchell et al., 2023) and GLTR (Gehrmann et al., 2019) use statistical features to differentiate between human and AI-generated text. However, they require access to model prediction distributions and show performance inconsistencies across models (Ghosal et al., 2023). LLM-based zero-shot detectors like OUTFOX (Koike et al., 2024; Bhattacharjee and Liu, 2024) face instability and bias issues; Fine-tuned detectors (Solaiman et al., 2019; Guo et al., 2023; Hu et al., 2023) are classifiers trained on labeled datasets. They encounter challenges in data collection and reduced effectiveness with larger models (Gambini et al., 2022). This study employs a fine-tuned approach for detection under black-box settings, where target generator sources and parameters are unknown, mirroring real-world scenarios where detectors must operate without prior knowledge of the text's origin or the specific AI models used in its generation.

Attacking and Defense against AI-Generated Text Detectors A significant challenge for all detectors is the effectiveness of paraphrasing attacks, which can significantly impair their performance (Christ et al., 2024; Krishna et al., 2024). For example, Krishna et al. (2024) introduced the DIPPER paraphraser, which decreased detection accuracy by up to 90% compared to its accuracy in the absence of attacks. Consequently, various attack techniques have been developed, including adversarial attacks and prompt attacks (Ghosal et al., 2023; Wu et al., 2023). As a defense, research has increasingly focused on enhancing robustness against such attacks (Krishna et al., 2024; Hu et al., 2023; Koike et al., 2024). Despite advancements in adversarial defense, challenges persist in robustness, vulnerability to recursive attacks, and cost-efficiency. AIDER addresses these issues through a novel augmentation framework that generates samples encompassing multiple attack types and simulates recursive attacks via cyclical settings. This approach enhances the robustness of the fine-tuned detector component. Moreover, AIDER offers improved cost-effectiveness in both training and inference by eliminating the need for LLM interaction during the training phase.

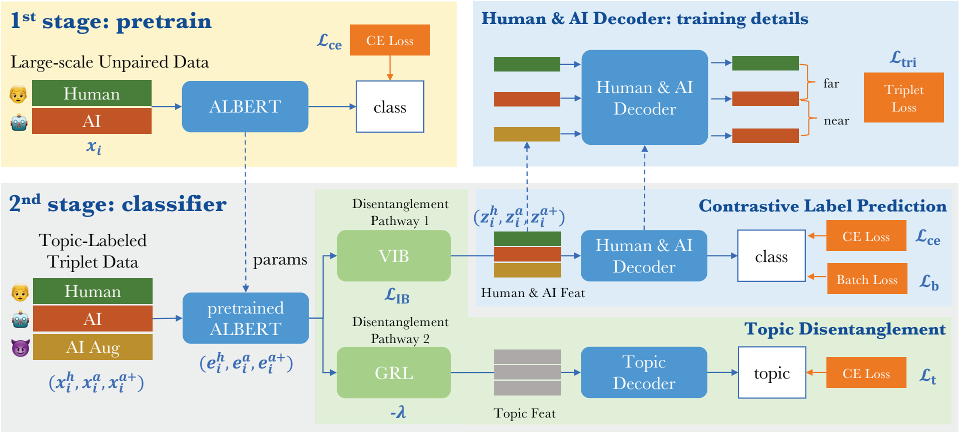

1st stage: pretrain
Large-scale Unpaired Data
Human
AI
x_i
ALBERT
class
$\mathcal{L}_{\text{ce}}$
CE Loss
Human & AI Decoder: training details
Human & AI Decoder
far
near
$\mathcal{L}_{\text{tri}}$
Triplet Loss
2nd stage: classifier
Topic-Labeled Triplet Data
Human
AI
AI Aug
(x_i^h, x_i^a, x_i^{a+})
params
pretrained ALBERT
(e_i^h, e_i^a, e_i^{a+})
Disentanglement Pathway 1
VIB
$\mathcal{L}_{\text{IB}}$
Disentanglement Pathway 2
GRL
-λ
(z_i^h, z_i^a, z_i^{a+})
Human & AI Feat
Contrastive Label Prediction
Human & AI Decoder
class
CE Loss
$\mathcal{L}_{\text{ce}}$
Batch Loss
$\mathcal{L}_{\text{b}}$
Topic Disentanglement
Topic Feat
Topic Decoder
topic
CE Loss
$\mathcal{L}_{\text{t}}$

Figure 2: The architecture of the proposed detector of AIDER, highlighting the topic disentagnlement module and the contrastive label prediction module.

Transferability on Unseen Topics Most detectors mentioned above are tailored to specific topics, such as news(Hanley and Durumeric, 2024; Zellers et al., 2019), scientific content(He et al., 2023; Liang et al., 2024) and academic texts(Koike et al., 2024; Yu et al., 2023). Their ability to transfer detection capabilities to out-of-distribution topics remains uncertain, posing a significant practical challenge(Li et al., 2024). To address these issues, we proposes a novel topic-disentanglement module in the detector of AIDER designed to enhance transferability to unseen topics. This approach is bolstered by the diverse range of topics included in our AIGen dataset. Prior to this study, no research has specifically focused on incorporating topic-disentanglement to improve detector transferability across various domains.

3 The Detector of AIDER

3.1 Holistic Architecture of Detector

The AI-generated text detector employs a **two-stage training paradigm**. It comprises two crucial modules: the **contrastive label prediction module** and the **topic disentanglement module**, as illustrated in Figure 2. The detector is built upon ALBERT(Lan et al., 2019), a lightweight variant of BERT(Devlin et al., 2018) that reduces model complexity while maintaining high performance.

In the first stage, ALBERT is pretrained on an aggregated dataset of approximately 500,000 unpaired human-AI texts sourced from publicly available datasets: Deepfake(Li et al., 2024), CHEAT(Yu et al., 2023) and MGTBench(He et al., 2023). These data undergo rigorous pre-processing and self-checking procedures to eliminate duplicates and remove extraneous information. The primary objective of this stage is to classify text as either human-written or AI-generated, enabling the model to discern fundamental differences between the two. This classification objective is achieved using cross entropy loss $\mathcal{L}_{\text{ce}}$. The weights of the final layer are preserved for fine-tuning in the second stage of training.

The second stage leverages the ALBERT model from the first stage utilizing the AIGen dataset introduced in Section 5. Given a triplet of input texts (x_i^h, x_i^a, x_i^{a+}), where x_i^h represents human-written text, x_i^a denotes AI-generated text, and x_i^{a+} is an augmented version of the AI-generated text, the triplet is processed through the ALBERT model to obtain respective representations e_i^h, e_i^a, and e_i^{a+}. These embeddings are subsequently fed into the topic disentanglement module and the contrastive label prediction module. The topic label in topic disentanglement module is determined using a zero-shot classification method[1] based on BART(Lewis et al., 2019), utilizing 12 predefined categories. This label encapsulates the topic of the triplet, aiming to facilitate the learning of features unrelated to the topic. This approach is particularly valuable given that the classified texts (whether generated by humans or AI) exhibit a wide variety of topics, which are independent of the text's origin.

3.2 Topic Disentanglement Module

The **topic disentanglement module** is engineered to disentangle topic-related information from the embeddings. This is achieved by processing the embeddings e_i^h, e_i^a, and e_i^{a+} through two parallel

[1] https://huggingface.co/facebook/bart-large-mnli

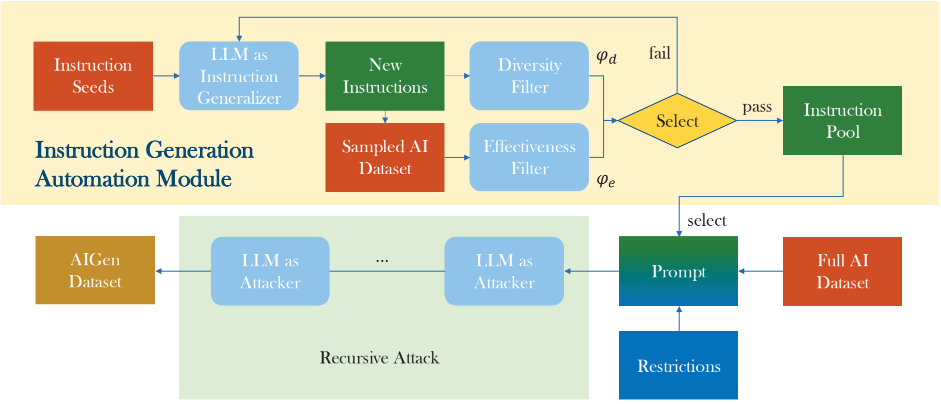


Figure 3: Workflow of the augmentor in AIDER framework, showcasing the auto generation of instruction and the augmentation process.

pathways: a Gradient Reversal Layer (GRL)(Ganin et al., 2016) and a Variational Information Bottleneck (VIB)(Alemi et al., 2022) as demonstrated in Figure 2.

In the GRL pathway, the embeddings e_i^h, e_i^a, and e_i^{a+} traverse the GRL before being fed into a multilayer perceptron (MLP) for topic label prediction of the input triplets. This process yields outputs g_i^h, g_i^a, and g_i^{a+}. The GRL functions as an identity transform during forward propagation but inverts the gradient direction during backpropagation:

$$\frac{\partial \mathcal{L}_\mathrm{t}}{\partial e_i^m} = -\lambda \frac{\partial \mathcal{L}_\mathrm{t}}{\partial g_i^m}, \quad m \in h, a, a+ \tag{1}$$

where $\mathcal{L}_\mathrm{t}$ is the topic classification loss which is computed using the cross-entropy loss function based on the output of the MLP, and λ represents a scaling factor.

Concurrently, in the VIB pathway, embeddings are processed to derive a latent representation z. This representation is optimized to maximize mutual information with task-relevant information (human or AI) while minimizing mutual information with the input. The VIB loss $\mathcal{L}_\mathrm{IB}$, reparameterization output of VIB z_i and total loss of the first pathway $\mathcal{L}_\mathrm{total}$, are formulated as:

$$\mathcal{L}_\mathrm{IB} = -0.5 \sum_{i=1}^{N} (1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2) \tag{2}$$

$$\begin{gathered} z_i^m = \mu_i^m + \sigma_i^m \odot \epsilon_i^m, \\ \epsilon_i^m \sim \mathcal{N}(0, I), \quad m \in h, a, a+ \end{gathered} \tag{3}$$

$$\mathcal{L}_\mathrm{total} = \mathcal{L}_\mathrm{c} + \beta \mathcal{L}_\mathrm{IB} \tag{4}$$

Here, $\mathcal{L}_\mathrm{IB}$ calculation employs Kullback-Leibler divergence(Kullback and Leibler, 1951) where N denotes embeddings e dimensionality, σ_i^2 and μ_i represent variance and mean of the i-th dimension, respectively. z_i^m signifies post-VIB latent embeddings for triplet texts, with ϵ_i sampled from a standard normal distribution and $\odot$ indicating element-wise multiplication. $\mathcal{L}_\mathrm{c}$ refers to Equation 7, while β balances information preservation of classification-related features and compression of topic-related features in the first pathway.

The synergistic application of GRL and VIB facilitates the model's learning of features that are both insensitive to topic labels (via GRL) and primarily relevant to the AI/human generation task (via VIB). This disentanglement process enables the **contrastive label prediction module** to learn robust representations that are topic-invariant and transferable across the triplet texts. Consequently, the detector's ability to generalize to unseen topics and effectively identify AI-generated texts is significantly enhanced.

3.3 Contrastive Label Prediction Module

The **contrastive label prediction module** plays a crucial role in enhancing the detector's discriminative power through contrastive learning. This module incorporates three loss components: triplet loss $\mathcal{L}_{tri}$, cross-entropy loss $\mathcal{L}_{ce}$, and balance loss $\mathcal{L}_b$. The triplet loss, serving as the core component, ensures that embeddings of the same class (AI-generated and AI-augmented) are proximal, while those of different classes (human-written and AI-generated) are well-separated:

$$\mathcal{L}_\mathrm{tri} = \max(0, d(z_i^h, z_i^a) - d(z_i^a, z_i^{a+}) + \alpha) \tag{5}$$

where $d(\cdot, \cdot)$ denotes the Euclidean distance be-

Dataset	AI	Size	Domains	Attack Types
IMDB(Maas et al., 2011)	×	1,000	Movies Review	×
SemEval(Xu et al., 2014)	×	1,000	Social Media Posts	×
SubjQA(Bjerva et al., 2020)	×	2,000	Books, Electronics, Grocery, Movies, Restaurants, TripAdvisor	×
Writing-Prompts(Fan et al., 2018)	×	2,000	Stories	×
OUTFOX(Koike et al., 2024)	✓	1,000	Argumentative Essays	×
HC3(Guo et al., 2023)	✓	1,000	Reddit, Open_QA, Wiki, Finance, Medicine	×
Daigt-v2[2]	✓	1,000	Essays, Academic Papers	×
AIGen (ours)	✓	24,000	Society, Science, Technology, Education, Politics, Sports, Finance, Entertainment, Books, Relationships, World, Health	prompt, paraphrase, adversarial

Table 1: AIGen datasets and details of its sources. AI marks if the dataset is a human-AI paired dataset or not

tween two embeddings and α is a margin parameter.

Complementing the triplet loss, the cross-entropy loss $\mathcal{L}_{\text{ce}}$ predicts the class labels (human or AI) of the input texts. To address the class imbalance inherent in the triplet structure of the training data, a novel balance loss $\mathcal{L}_b$ is introduced. This loss penalizes the model for predicting all samples as a single class within a batch:

$$\mathcal{L}_{\text{b}} = \left| \frac{1}{B} \sum_{i=1}^{B} \hat{y}_i - 0.5 \right| \qquad (6)$$

where B denotes the batch size, and $\hat{y}_i$ represents the predicted probability of the text being AI-generated. By minimizing $\mathcal{L}_{\text{b}}$, the model is encouraged to make balanced predictions, avoiding a bias towards one class over the other. This approach prevents the model from becoming trapped in local optima.

The overall loss for the module is computed as a weighted sum of these three components:

$$\mathcal{L}_{\text{c}} = \lambda_1 \mathcal{L}_{\text{tri}} + \lambda_2 \mathcal{L}_{\text{ce}} + \lambda_3 \mathcal{L}_{\text{b}} \qquad (7)$$

where λ_1, λ_2, and λ_3 are hyperparameters that control the relative importance of each loss component, allowing for fine-tuning of the model's learning focus.

4 The Augmentor of AIDER

The AIDER's augmentor presents an innovative approach to addressing robustness challenges through LLM-driven data augmentation. As illustrated in Figure 3, this comprehensive technique generates diverse adversarial scenarios. The framework efficiently processes AI-generated texts into augmented versions, simulating attacks (Ghosal et al., 2023) while expanding the dataset. Users can fine-tune parameters to produce high-quality, varied attack prompts, with control over data quantity, attack type distribution, and recursive attack frequency. The system also incorporates mechanisms to ensure dataset diversity and effectiveness.

The main process of the augmentor is made of three parts as shown in Figure 3: 1)**Attacker:** This component utilizes an LLM as its core. The framework is designed to accommodate various LLMs, including both open-source and proprietary models, which can be seamlessly integrated. Notably, it supports configurable recursive attacks on the same AI-generated text, allowing for multiple layers of augmentation. 2)**Restrictions:** These are essential guidelines implemented to mitigate the inherent instability of LLMs and to align the augmentation process more closely with specific requirements. These restrictions serve as guardrails to ensure the generated attacks remain relevant and controlled. Further details can be found in Appendix B. 3)**Prompt of Instructions:** The augmentor employs a seed instruction set of approximately 30 attack instructions, encompassing a wide range of attack types. To expand this set, the study implements the self-instruct(Wang et al., 2023) method. Each newly generated instruction undergoes a filtering process based on diversity φ_d and effectiveness φ_e metrics:

$$\varphi_d = 1 - \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(d(I_{\text{new}}, I_i) < T\right) \qquad (8)$$

$$\varphi_e = \frac{1}{M} \sum_{j=1}^{M} \mathbb{I}(f(x_j^{\text{aug}}) > \theta) \qquad (9)$$

where $d(I_{\text{new}}, I_i)$ is the Euclidean distance between newly generated instruction and the i-th instruction

Generator	Detector	Metrics(%)↑		
		AvgRec	Recall	F1
FLAN-T5-XXL	log $p(x)$	49.8	**97.6**	66.0
	Rank	57.5	86.2	67.0
	LogRank	51.3	90.6	65.0
	Entropy	59.9	80.4	66.7
	DetectGPT	50.8	71.6	59.3
	DIPPER	85.6*	72.0	83.3*
	OUTFOX	85.2	73.4	83.2
	AIDER(ours)	**86.7**	86.8*	**86.6**
ChatGPT	RoBERTa-base	93.0	92.2	92.9
	RoBERTa-large	90.8	90.0	90.7
	HC3 detector	74.9	70.6	73.8
	DIPPER	93.5*	87.8	93.1
	OUTFOX	**95.1**	92.4*	**95.0**
	AIDER(ours)	92.7	**98.0**	93.2*
GPT-3.5	RoBERTa-base	92.9	92.0	92.8
	RoBERTa-large	92.3	92.0	92.3
	HC3 detector	82.1	85.0	82.6
	DIPPER	95.6*	92.4	95.5*
	OUTFOX	**96.9**	**96.2**	**96.9**
	AIDER(ours)	91.6	95.9*	92.2

Table 2: Comparison of methods on OUTFOX(non-augmented) dataset using different generators. **Bold** marks the model ranking first, * marks the model ranking second.

Attacker	Detector	Metrics (%) ↑		
		AvgRec	Recall	F1
DIPPER	DIPPER	**88.9**	79.6*	**87.8**
	OUTFOX	85.1*	72.4	82.9
	HC3 detector	41.3	3.4	5.5
	AIDER (ours)	82.8	**82.9**	83.2*
OUTFOX	DIPPER	59.7	20.8	34.0
	OUTFOX	83.4*	69.6*	80.7*
	HC3 detector	39.8	0.4	0.7
	AIDER (ours)	**91.3**	**99.9**	**92.1**
		ΔAttack (%) ↓		
		AvgRec	Recall	F1
DIPPER	DIPPER	**4.6**	**8.2**	**5.3**
	OUTFOX	10.0	10.0*	12.1
	HC3 detector	33.6	67.2	68.3
	AIDER (ours)	9.9*	15.1	10.0*
OUTFOX	DIPPER	33.8	67.0	59.1
	OUTFOX	11.6*	22.8*	14.3*
	HC3 detector	34.1	70.2	73.1
	AIDER (ours)	**1.4**	**-1.9**	**1.2**

Table 3: Comparison of methods on OUTFOX(augmented) ChatGPT dataset using different attackers. ΔAttack is the difference in metrics between the results without the attack (shown in Table 2) and with the attack (shown in the current table).

in the existing pool, N is the total number of instructions in the pool. M is the number of samples in the GPT-generated test set, x_j^{aug} is the j-th augmented sample using the new instruction, $f()$ is the ZeroGPT API[3] that outputs a probability of being human-generated. T and θ are predefined thresholds for diversity and effectiveness respectively. A higher φ_d and φ_e value indicates higher diversity and effectiveness.

Filtered instructions are added to the seed set, forming a pool from which instructions are selected for the main process of augmentation. The selection method employs a stratified random sampling approach, where instructions are randomly selected from different attack types with approximately equal probability for each type.

5 Constructed AIGen Dataset

This study introduces the augmentation-based AIGen dataset, encompassing diverse topical domains as demonstrated in Table 1. The AIGen dataset amalgamates data from two primary sources: existing human-AI datasets (OUTFOX, HC3, and Daigt-v2) and established human-written datasets (IMDB, SemEval, SubjQA, and WritingPrompts). For the latter, we employed varied prompting strategies, sampling methods (temperature, top-p), and LLM configurations (e.g., GPT and GLM series) to generate diverse AI-generated texts. All entries underwent augmentation using AIDER's augmentor, powered by GPT-4 Turbo, applying various attack types including prompt-based, paraphrase-based, and adversarial approaches. To further enhance the dataset's complexity and robustness, 20% of the augmented texts were subjected to recursive attacks, with the number of iterations randomly varying between 2 and 3. AIGen dataset is a robust and diverse corpus, providing a valuable resource for future research.

6 Experiments

6.1 Experimental Settings

Dataset and Evaluation Metrics Experiments are conducted on the test set of OUTFOX(Koike et al., 2024), which consists of 1,500 argumentative essays written by native English students. The dataset includes texts generated by three widely-used target generators: Flan-T5-XXL, ChatGPT, and GPT-3.5. Only the texts generated by ChatGPT contain augmented (i.e., attacked) data pairs. The evaluation metrics employed in these experiments are F1 score, AI Recall (the recall specific to AI-generated text), and Average Recall (AvgRec), which is calculated as the mean of both human

[3] https://zerogpt.net/api-integration

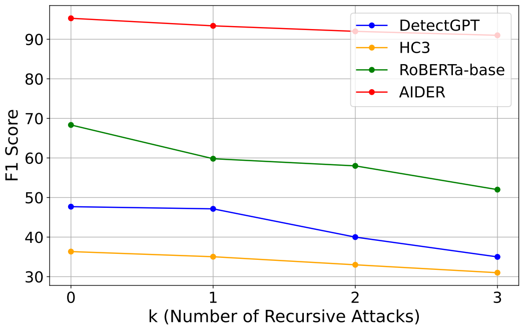


Figure 4: Detection performance demonstration of 4 models on different settings of recursive attacks on our AIGen dataset.

recall and AI recall.

Baselines This study conducts multiple experiments on both unsupervised baselines: GLTR(Gehrmann et al., 2019), DetectGPT(Mitchell et al., 2023); and supervised baselines: HC3 Detector(Guo et al., 2023), RoBERTa(Solaiman et al., 2019); and also two baselines with attackers: OUTFOX(Koike et al., 2024), DIPPER(Krishna et al., 2024) to evaluate the performance of augmentor framework. Further details can be found in Appendix C. In both Table 2 and Table 3, this study aligns with the OUTFOX dataset settings. In these evaluations, statistical approaches like GLTR are applied exclusively to generator FLAN-T5-XML, as they require access to model logits, which are unavailable for ChatGPT and GPT-3.5. OUTFOX is omitted from Table 4 because it is specifically designed and tested on essay (education)-related data, not addressing other topics. Similarly, DIPPER is excluded since it is trained solely on books and tailored for augmentation scenarios. This approach ensures that the evaluation remains relevant and accurate for the specific contexts and datasets each model is designed to handle.

Implementation Details The AIGen dataset was split into a 4:1 train-test ratio for implementation. Random seeds of all experiments are set to 42. The detector is trained for 20 epochs and 5 epochs for the first stage and second stage respectively. The learning rate was set to 2e-5, the batch size to 16 in both stages. AdamW and CosineAnnealingLR are used as learning rate optimizer and scheduler in both stages. The max length is set to 512 in all ALBERT-related settings. The weights of the loss functions are $\lambda_1 = 0.5$ for the triplet loss, $\lambda_2 = 0.3$ for the cross-entropy loss, $\lambda_3 = 0.2$ for the balance loss, and $\beta = 0.38$. The thresholds in augmentor are $T = 0.75$ and $\theta = 0.60$.

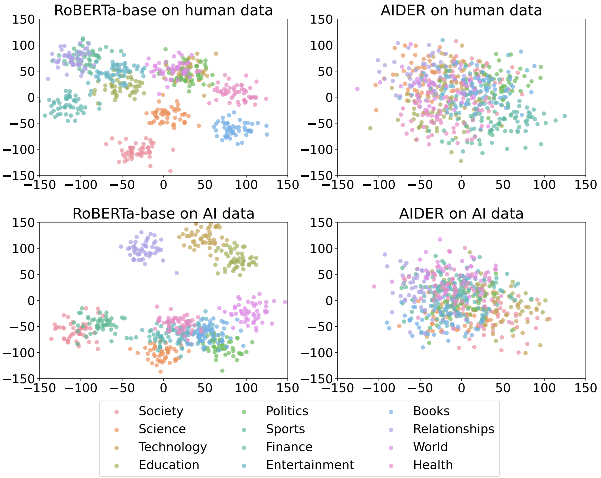


Figure 5: t-SNE visualizations of topic-specific representations: a comparison of AIDER and RoBERTa-base on human-written vs. GPT-4-Turbo(AI)-generated texts.

6.2 Results and Analysis

Comparison with Baselines This experiment evaluates the proposed detector's performance against existing methods on the non-augmented OUTFOX dataset. As shown in Table 2, AIDER outperforms other approaches for FLAN-T5-XXL-generated essays, surpassing the second-best method by 1.1% in F1-score and 3.3% in average recall (AvgRec). AIDER's high AvgRec of 86.7% likely stems from its extensive pre-training on diverse human-written and AI-generated texts. For ChatGPT and GPT-3.5-generated essays, AIDER exceeds supervised classifiers but falls short of OUTFOX and DIPPER. This performance discrepancy may be attributed to the contrastive training focusing more on augmented data than original AI-generated texts. The augmentation effect, where increased focus on augmented data slightly lowers performance on non-augmented data, explains these differences. Our aim is to create a classifier capable of handling diverse AI-generated texts, justifying this trade-off. Statistical approaches, while achieving high recall, tend to overclassify human-written essays as LLM-generated, resulting in lower AvgRec. Overall, AIDER demonstrates robust performance in identifying non-augmented LLM-generated essays across various generators compared to baseline methods.

Model	Environment		Internet		Fashion	
	F1	AvgRec	F1	AvgRec	F1	AvgRec
RoBERTa-base	45.92	44.27	65.56	61.80	48.11	42.05
HC3 Detector	41.25	52.86	48.07	50.46	38.49	48.62
DetectGPT	43.81	50.50	51.18	59.85	42.70	54.09
AIDER (Ours)	**89.36**	**88.92**	**93.28**	**94.07**	**90.55**	**89.32**

Table 4: Performance comparison of 4 models on 3 unseen topics from AIGen dataset

Robustness against Adversarial Attacks This experiment evaluates detector robustness by comparing OUTFOX, DIPPER, HC3 detector, and AIDER on augmented OUTFOX datasets (Table 3). AIDER achieves the highest recall (82.9%) and second-best F1 (83.2%) on DIPPER-augmented data, while outperforming all baselines on OUTFOX-augmented data with significant improvements in F1 (+11.4%) and AvgRec (+7.9%) compared to OUTFOX. Despite performance drops (ΔAttack) relative to the non-augmented setting (Table 2), AIDER shows minimal degradation, particularly under OUTFOX augmentation. Interestingly, AIDER's Recall improves in the OUTFOX augmentation scenario, possibly due to similarities between OUTFOX's strategy and AIDER's GPT-based approach. In contrast, methods like the HC3 detector exhibit substantial performance deterioration. Figure 4 further illustrates AIDER's consistent high F1 scores across varying attack rounds(k), while other models (DetectGPT, HC3, RoBERTa-base) experience significant declines due to their lack of training on recursive attack samples. These results collectively demonstrate AIDER's superior robustness and effectiveness in detecting LLM-generated texts, even in challenging adversarial scenarios.

Performance on Unseen Topics Table 4 demonstrates AIDER's superior performance on three unseen topics from the AIGen dataset, consistently achieving the highest F1 scores (89.36% to 93.28%) and AvgRec across all topics. In contrast, HC3 Detector, RoBERTa-base, and DetectGPT exhibit considerably lower performance, particularly struggling with "Environment" and "Fashion" topics. To further investigate this performance disparity, Figure 5 presents t-SNE visualizations of learned features from both AIDER and RoBERTa-base on the AIGen dataset. RoBERTa-base (left column) displays well-defined clusters corresponding to different topics, indicating a tendency to learn topic-specific features. Conversely, AIDER (right column) demonstrates significant overlap among topics for both human-written and AI-generated data, suggesting its ability to extract more topic-invariant features. The combination of quantitative results and qualitative visualizations clearly illustrates AIDER's superior ability to learn generalized, less topic-biased representations, enabling robust performance across various text generation sources and unseen topics, a crucial advantage in real-world applications where topic diversity is prevalent.

Method	Non-Augmented Data		
	F1	Recall	AvgRec
log $p(x)$ (GLTR)	32.60	58.90	58.85
Rank (GLTR)	35.33	58.42	57.59
LogRank (GLTR)	29.27	58.45	57.90
Entropy (GLTR)	41.91	59.36	59.36
DetectGPT	47.69	57.02	57.02
HC3 Detector	36.34	56.00	63.41
RoBERTa-base	68.34*	63.00*	63.41*
AIDER w/o 1-ST	78.05	80.52	80.76
AIDER w/o TD	84.11	86.02	84.43
AIDER w/o CL	87.44	89.01	88.10
AIDER (ours)	**95.30**	**95.26**	**95.26**
	Augmented Data		
	F1	Recall	AvgRec
log $p(x)$ (GLTR)	32.52	58.85*	58.08*
Rank (GLTR)	33.18	57.59	56.42
LogRank (GLTR)	27.65	57.90	54.45
Entropy (GLTR)	39.96	58.51	58.01
DetectGPT	47.14	57.02	55.57
HC3 Detector	35.04	55.93	56.37
RoBERTa-base	59.81*	56.37	56.37
AIDER w/o 1-ST	90.89	90.94	91.01
AIDER w/o TD	84.62	87.07	84.82
AIDER w/o CL	79.25	81.68	82.17
AIDER (ours)	**93.39**	**93.33**	**93.00**

Table 5: Comparison of methods with and without data augmentation. **1-ST** means the first stage training, **TD** means the topic disentanglement module, **CL** means contrastive label prediction module.

Effectiveness of AIGen Dataset This experiment evaluates various baselines on the AIGen dataset to assess its effectiveness in challenging LLM-generated text detection models. As shown in Table 5, AIDER achieves state-of-the-art performance, followed by the supervised classifier RoBERTa-base. Notably, unsupervised methods demonstrate greater robustness to distribution shifts in augmented data compared to supervised approaches, likely due to their focus on statistical patterns rather than reliance on specific training data. A comparison between the OUTFOX dataset (Table 2) and AIGen (Table 5) reveals that AIGen presents increased difficulty for both augmented and non-augmented data. This heightened challenge underscores the effectiveness of the AIGen augmentor framework in creating more complex examples for LLM-generated text detection, thereby pushing the boundaries of model performance and robustness.

Ablation Study The bottom part of Table 5 presents an ablation study of AIDER, examining the impact of its key components: first stage training (1-ST), topic disentanglement module (TD), and contrastive label prediction module (CL). The removal of 1-ST results in a significant performance decline on non-augmented data, indicating its importance in distinguishing intrinsic linguistic patterns and styles. Conversely, the absence of the CL module leads to a more substantial drop in performance on augmented data, highlighting its role in learning complex patterns within AI-generated texts subject to adversarial perturbations or stylistic variations. Notably, the TD module's removal causes a marked decline in detecting both non-augmented and augmented data, underscoring its critical function in mitigating the influence of topic-related features and enhancing AIDER's ability to discern fine-grained differences between human-written and AI-generated texts.

7 Conclusions and Future Works

This paper presents AIDER, a novel framework for detecting AI-generated text. By leveraging augmentor and contrastive learning , AIDER enhances robustness against adversarial attacks. By leveraging topic disentanglement module, AIDER become generalized on diverse topical contents. Experimental results highlight AIDER's ability to outperform state-of-the-art methods significantly under adversarial and topic-varied conditions. Future research will focus on refining data augmentation strategies by integrating more sophisticated attack techniques, exploring advanced attack models, and investigating cross-lingual detection capabilities.

8 Limitations

While the proposed AIDER framework demonstrates superior performance, robustness against adversarial attacks with minimal performance degradation, and high generalization capability across various topics, it has two major limitations: (1)This study aims to incorporate different sources of target generators (i.e., GPTs, GLMs, and Claudes) to enhance practical applicability. However, the generalization capability of detecting texts generated by out-of-distribution models has not been adequately addressed in AIDER's design, making it vulnerable to new and unseen generators. (2)To ensure the authenticity of human-written texts, this study employs an automatic approach, collecting data from periods before the emergence of LLMs, rather than utilizing human verification. This strategy introduces a data shift from outdated human-written texts to contemporary human-written texts, potentially affecting the model's performance on new human-written texts due to temporal discrepancies.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under the Grant No. 62371057; and the 111 Project of China under the Grant No. B08004.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2022. Deep variational information bottleneck. In *International Conference on Learning Representations*.

Anthropic. 2023. Introducing claude.

Amrita Bhattacharjee and Huan Liu. 2024. Fighting fire with fire: can chatgpt detect ai-generated text? *ACM SIGKDD Explorations Newsletter*, 25(2):14–21.

Johannes Bjerva, Nikita Bhutani, Behzad Golahn, Wang-Chiew Tan, and Isabelle Augenstein. 2020. Subjqa: A dataset for subjectivity and review comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Miranda Christ, Sam Gunn, and Or Zamir. 2024. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694.

Margherita Gambini, Tiziano Fagni, Fabrizio Falchi, and Maurizio Tesconi. 2022. On pushing deepfake tweet detection capabilities to the limits. In *Proceedings of the 14th ACM Web Science Conference 2022*, pages 154–163.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL).

Soumya Suvra Ghosal, Souradip Chakraborty, Jonas Geiping, Furong Huang, Dinesh Manocha, and Amrit Singh Bedi. 2023. Towards possibilities & impossibilities of ai-generated text detection: A survey. *arXiv preprint arXiv:2310.15264*.

Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *differences*, 14:18.

Hans WA Hanley and Zakir Durumeric. 2024. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 18, pages 542–556.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. MGTBench: Benchmarking Machine-Generated Text Detection. *CoRR abs/2303.14822*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, 36:15077–15095.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21258–21266.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. Mage: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, et al. 2024. Monitoring ai-modified content at scale: A case study on the impact of chatgpt on ai conference peer reviews. *arXiv preprint arXiv:2403.07183*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

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

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics*.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F Wong, and Lidia S Chao. 2023. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724*.

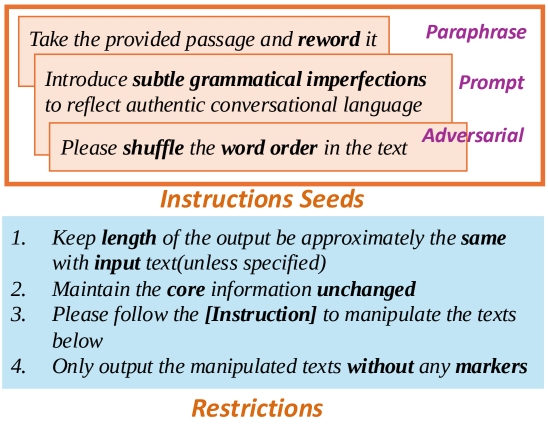
Wei Xu, Alan Ritter, Chris Callison-Burch, William B. Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from Twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.

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

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

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text. In *The Twelfth International Conference on Learning Representations*.



*Take the provided passage and **reword** it*

*Introduce **subtle grammatical imperfections** to reflect authentic conversational language*

*Please **shuffle** the **word order** in the text*

1. *Keep **length** of the output be approximately the **same** with **input** text(unless specified)*
2. *Maintain the **core** information **unchanged***
3. *Please follow the **[Instruction]** to manipulate the texts below*
4. *Only output the manipulated texts **without** any **markers***

Figure 6: Illustration of restrictions and different attack types in augmentor of AIDER.

A Details of Pre-training Dataset

Table 6 contains details of data used in the first stage of AIDER's detector.

B Illustration of Restrictions and Instruction Seeds in Augmentor

Figure 6 shows the restrictions used in AIDER's augmentor and examples of differenct attack types in instructions seeds. For further clarification, current attacks can be categorized into three main types(Wu et al., 2023):

1. **Paraphrase-Based Attacks**. These attacks typically involve using paraphrasing techniques, such as DIPPER, to alter the distribution of generated AI text.

2. **Prompt-Based Attacks**. These attacks leverage advanced prompting techniques to improve the quality and effectiveness of generated text, presenting challenges to detectors trained with simple prompt-generated text.

3. **Adversarial-Based Attacks**. These attacks modify textual features through operations such as random shuffling, deformation, word swapping, and misspelling.

C Details of Baselines

The introduction of each of the baseline is listed:

- **GLTR**(Gehrmann et al., 2019): an unsupervised method that utilizes four statistical measures based on token-wise log probabilities, average token rank, token log-rank, and predictive entropy to distinguish AI-generated text from human-written text.

Domain	Dataset	Sou.	Gen.	H-Size	AI-Size	Description
News, Story, Question, Arfgument, Scientific Topic…	Deepfack(Li et al., 2024)	7	27	150,858	281,824	a comprehensive benchmark dataset designed to assess the proficiency of AI-generation detectors amidst real-world scenarios
Academic Paper	CHEAT(Yu et al., 2023)	1	1	15,395	46,185	large-scale ChatGPT-written Abstract dataset
Essay, News	MGTBench(He et al., 2023)	5	7	3,000	21,000	datasets of different machine-generated text (MGT) detection methods.

Table 6: Aggregated pretraining datasets and their details. H-Size is the size of Human-written texts; AI-Size is the size of AI-generated texts; Gen. is the number of generators; Sou. is the number of generators' sources.

- **DetectGPT**(Mitchell et al., 2023): an unsupervised approach that leverages a proxy language model to compute log probabilities of generated tokens, hypothesizing that minor perturbations to AI-generated text result in negative curvature of the log-likelihood curve, serving as a discriminative feature for classification.

- **HC3 Detector**(Guo et al., 2023): a supervised model that employs a RoBERTa-base architecture trained on a mix of full-text and split sentences from the HC3 corpus for one epoch.

- **RoBERTa**(Solaiman et al., 2019): a supervised classifier from OpenAI that aims to detect texts generated by the 1.5B-parameter GPT-2 model, available in two versions: base and large, with different model sizes.

- **OUTFOX**(Koike et al., 2024): a framework based on LLM that improves detector robustness via adversarial in-context learning between detector and attacker.

- **DIPPER**(Krishna et al., 2024): a document-level paraphraser and detector that can control output diversity in terms of vocabulary and content re-ordering.