Counter Turing Test (CT^2): Investigating AI-Generated Text Detection for Hindi - Ranking LLMs based on Hindi AI Detectability Index (ADI_{hi})

Anonymous ACL submission

Abstract

The widespread adoption of large language models (LLMs) like GPTs, BARD, and others has raised concerns regarding the potential risks and repercussions linked to the misapplication of AI-generated text, necessitating increased vigilance. While these models are primarily trained for English, their extensive training on vast datasets covering almost the entire web equips them with capabilities to perform well in numerous other languages such as Hindi and Spanish. AI-generated text detection (AGTD) has emerged as a topic that has already received immediate attention in research, with some initial methods having been proposed, soon followed by the emergence of techniques to bypass detection. In this paper, we report our investigation on AGTD for the Hindi language: i) examined 16 large language models (LLMs) to evaluate their proficiency in generating Hindi text; introducing the AI-generated news article in Hindi (AG_{hi}) dataset, ii) thoroughly evaluated the effectiveness of four recently proposed AGTD techniques: ConDA, J-Guard, RADAR, and Intrinsic Dimension Estimation for detecting AI-generated Hindi text, iii) proposed Hindi AI Detectability Index (ADI_{hi}) which shows a spectrum to understand the evolving landscape of eloquence of AI-generated text in Hindi and efficacy of available AGTD techniques to counter adversarial use of LLMs for Hindi.

1 Introduction

AI-generated text detection is necessary for several reasons, primarily centered around addressing the challenges and potential risks associated with the widespread use of AI-generated content. Here are some key reasons why text detection for AI-generated content is crucial:

- Misinformation and Fake News: AI-generated text can be used to create and spread misinformation, fake news, or malicious content. Detecting such content is essential to prevent the dissemination of false or harmful information (Kreps et al., 2022).
- Online Manipulation: AI-generated text can be used for online manipulation, such as creating fake reviews, comments, or social media posts. Detection tools help in identifying and mitigating such manipulative activities (Chernyaeva et al., 2022).
- Phishing and Scams: Malicious actors may use AI-generated text to craft convincing phishing emails or messages. Detection tools can help identify and block such fraudulent attempts, protecting users from falling victim to scams (Basit et al., 2021).
- Maintaining Trust and Credibility: The proliferation of AI-generated content can erode trust in online information sources (Crothers et al., 2023). Detection mechanisms help maintain the credibility of online platforms and prevent users from being deceived by false or manipulated information.

In summary, as generative models are growing, we need comparable detection techniques. AI text detection is necessary to safeguard individuals, organizations, and society from the potential negative consequences of malicious or misleading content generated by AI systems. It plays a crucial role in maintaining the integrity of online communication and upholding ethical standards in the use of AI technologies. We are the first to conduct experiments for AI-generated news article generation and detection techniques for the Hindi language. Hindi is the fourth most-spoken first language in the world after Mandarin, Spanish, and English (Wikipedia, 2023). Taking inspiration from recent works of AI-generated text detection for English (Chakraborty et al., 2023) where they discussed 6 detection techniques namely watermarking, perplexity estimation, burstiness estimation, negative log curvature, and stylometric variation, we extend it to regional languages like Hindi and cover four new detection techniques that are suitable for multilingual AI-generated text detection for Hindi.

OUR CONTRIBUTIONS: A Counter Turing Test (CT^2) and AI Detectability Index for Hindi (ADI_{hi})

- Introducing the *Counter Turing Test (CT^2)* for Hindi, a benchmark that incorporates methods designed to provide a thorough assessment of the resilience of existing AGTD techniques in Hindi.
- Conducting a thorough examination of 16 LLMs to generate an AI-generated news article in Hindi. (AG_{hi}) dataset
- Presenting the *AI Detectability Index for Hindi (ADI_{hi})* as a metric for Language Models to assess whether their outputs can be identified as generated by artificial intelligence or not.
- Curated datasets and models will be made available with the MIT License, making it favorable for open-source research and commercial use.

2 Data Generation choices for detection techniques in regional languages

For exploring generation and detection models beyond English, we chose Hindi which is one of the 4th most popular languages in the world (Wikipedia, 2023). This section discusses our selected LLMs and elaborates on our data generation methods.

2.1 LLMs: Rationale and coverage

We chose a wide gamut of 16 LLMs that have exhibited exceptional results on a wide range of NLP tasks. They are: (i) GPT-4 (**?**); (ii) GPT-3.5 (Chen et al., 2023); (iii) GPT-2 (base, medium, large, xl) (Radford et al., 2019); (iv) BARD (Bard, 2023); (v) Bloom (560M, 3B, 7B) (Workshop and et al, 2023) (vi) Bloomz (560M, 1B, 3B, 7B) (Muennighoff et al., 2022); (vii) mGPT (1.3B) (Shliazhko et al., 2023); (viii) Mistral Instruct 7B (Jiang et al., 2023).

As the field is in a constant state of evolution, we acknowledge that this process will never reach its finality but instead will persist in its expansion. Therefore, we intend to maintain the Hindi leaderboard benchmark as an open platform for researchers, facilitating ongoing updates and contributions.

2.2 Criteria of Acceptance/Rejection for AI generated news articles

We experimented with a total 16 LLMs including variation in their parameter size. Through our experimentation and observation of the output, we rejected 13 models. Some of the outputs are present in fig 1. The criteria used to determine acceptance or rejection of a model are as follows:

Language Consistency: If the response is in English, the model is rejected.

Code-Switching: If the response starts in Hindi but later switches to English, the model is rejected.

Gibberish Output: Models that produce unintelligible or gibberish responses are rejected.

Parameter Size: Among the accepted models, the model with the highest parameter size is considered, as larger models are expected to produce higher-quality text.

Apart from these four cases, models are accepted. To provide transparency, we have retained the responses for 100 data points from BBC Hindi for the rejected models and will be releasing them. This dataset exemplifies why certain models were deemed unfit for inclusion due to their

Rejection criteria with Model	News Headline	Human written news article	AI-generated news article
Language Consistency (GPT2)	मधुमिता शुक्ला हत्याकांड: अमरमणि त्रिपाठी और उनकी पत्नी के जेल जाने की कहानी Madhumita Shukla murder case: Amarmani Tripathi and his wife go to jail	अगस्त 2023नौ मई, 2003 को लखनऊ के तक़रीबन सभी वरिष्ठ पुलिस अधिकारी एक ख़ास बैठक में व्यस्त थे. शाम के चार बजे से चल रही इस बैठक में अगले दिन यानी 10 मई को शहर में होने वाले चुप ताज़िया के जुलूस की तैयारियों पर चर्चा हो रही थी. मोहर्रम के दौरान निकलने वाले हज़ारों लोगों के इस जुलूस की सुरक्षा व्यवस्था लखनऊ पुलिस के लिए हमेशा से चुनौती रही थी. August 2023, On May 9, 2003, almost all senior police officers in Lucknow were busy in a special meeting. The meeting, which began at 4 pm, discussed preparations for the Chup Tazia procession to be held in the city on the next day, May 10. The security arrangements for this procession of thousands of people during Muharram were always a challenge for the Lucknow Police.	May 15, 2017 The article was written by a journalist who was not a journalist. The article was published on May 15 and was published in the Hindustan Times. In the article, the journalist said that the BJP government has been trying to get the Congress to give the Congress a platform to run the country. .@jamesharan_nagar @jameshaan_nao
Code-Switching (GPT2-XL)			मधुमिता शुक्ला हत्याकांड: अमरमणि त्रिपाठी और उनकी पत्नी के जेल जाने की कहानीं का क क । की सर्।ि।ी।\n\nThe article is written in Hindi and the headline is written as मु मारात शाही: स् म म १५॥\n\nIn the above example, the Hindi word म is used to indicate the noun म. The word श is used for the adjective शि. The adjective साम is also used for स. The noun क is used in the article to indicate a noun.
Gibberish Output (Mistral 7B instruct)			मधुमिता शुक्ला हत्याकांड: अमरमणि त्रिपाठी और उनकी पत्नी के जेल जाने की कहानी\n\nमधुमाता शुक्ला हत्याकांड वालों की उनको जेल जाने की कहानी है। उनकें नहीं उनको शुचिती हमारी पुरानों की संस्करण पहले अवश्यक होता है। उनको जेल जाने की कहानी बहुती है। आजकल समस्यों को अलग उनको जेल जाने को अलग नहीं किया जाता है।

Figure 1: This figure illustrates the outputs from the models that were rejected. The initial lines of the articles are displayed, with the content in blue text representing a translation for better understanding. As excluded models do not provide accurate Hindi responses, their translations are not presented.

inability to generate coherent and meaningful text. In summary, out of all the 16 LLMs tested for AI-generated news articles in Hindi, we have considered 3 models (BARD, GPT-3.5, GPT-4) which are generating news articles in Hindi as described in Fig. 2.

2.3 Dataset Curation Methodology and Statistics

The methodology employed for creating the dataset used in our research comprises human-written news articles and AI-generated responses from the headlines given as prompts to LLMs.

- **Human Written Articles**: The human-written articles dataset is derived from headlines sourced from BBC (BBC) and NDTV (NDTV) news source, encompassing various categories, including India, international affairs, sports, Bollywood, lifestyle, health, and more. Significantly, the absence of a comprehensive archive feature on these websites shaped our approach to dataset construction. In contrast to prior studies that extract headlines from specific periods, we chose to compile a varied set of headlines without being bound by temporal limitations.

<table>
<tr><th>Model</th><th>News Headline</th><th>Human written news article</th><th>AI-generated news article</th><th>AI-generated news article translated for better understanding</th></tr>
<tr><td>BARD</td><td rowspan="2">इंडिया गठबंधन की वे चुनौतियां, जिनसे बीजेपी को मिलती है राहत

Challenges of India alliance, gives relief to BJP</td><td rowspan="2">6 घंटे पहले केंद्र में भारतीय जनता पार्टी की सत्ता को कड़ी टक्कर देने के मक़सद से गठित हुए विपक्षी गठबंधन ‘इडिया’ की गुरुवार को मुंबई में बैठक शुरू होगी और यह शुक्रवार तक चलेगी. पटना और बेंगलुरु के बाद ये इस गठबंधन की यह तीसरी बैठक है.

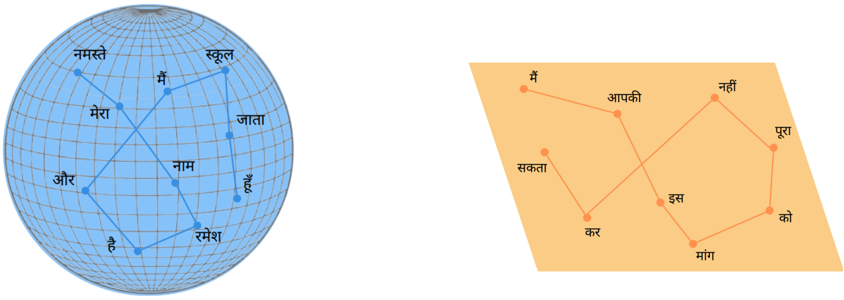
The meeting of the opposition alliance 'India', formed with the aim of giving a tough fight to the Bharatiya Janata Party's power at the center, will begin in Mumbai on Thursday and it will run till Friday. This is the third meeting of the alliance after Patna and Bengaluru.</td><td>2024 के लोकसभा चुनावों से पहले, विपक्षी दलों नै एक बड़े गठबंधन का गठन किया है, जिसका नाम है "इंडिया गठबंधन"। इस गठबंधन में 28 से अधिक दल शामिल हैं, जिनमें कांग्रेस, समाजवादी पार्टी, राष्ट्रवादी कांग्रेस पार्टी, तृणमूल कांग्रेस, आम आदमी पार्टी, शिवसेना, और बीएसपी शामिल हैं।</td><td>Ahead of the 2024 Lok Sabha elections, the opposition parties have formed a big alliance, which is named "India Coalition". This alliance includes more than 28 parties, including Congress, Samajwadi Party, Nationalist Congress Party, Trinamool Congress, Aam Aadmi Party, Shiv Sena, and BSP.</td></tr>
<tr><td>GPT2-small</td><td>Consider the given headline and write a news article for it in hindi: इंडिया गठबंधन की वे चुनौतियां, जिनसे बीजेपी को मिलती है राहत । कर हि का क क ॥ क, क. क. क, क) क (क) क The following is a list of the most common words in Hindi. 1. पिी 2. मे 3. हात 4. भी
The पे is a verb that means to be present.</td><td>Consider the given headline and write a news article for it in hindi. India Alliance challenges from which BJP gets relief.
gibberish text
The following is a list of the most common words in Hindi.
1. p 2. in 3. hand 4. also
The pay is a verb that means to be present.</td></tr>
</table>

Figure 2: This figure represents the outputs from LLMs for generating Hindi news articles using BARD (that is accepted) and GPT-2 small (that is rejected) respectively. The initial lines of the articles are displayed, with the content in blue text representing a translation for better understanding.

- **AI Generated Articles**: To obtain AI-generated responses, we employed state-of-the-art 16 Language Models (LLMs). The hyperparameters for reproducing the dataset are present in appendix A.1 and the code and dataset can be found at `http://bit.ly/3Ts0BwJ`. The headlines collected from the human-written articles were presented as prompts to these LLMs, which generated text responses. These responses were human-evaluated and each of these models was either accepted or rejected based on language consistency, code-switching, and gibberish output generation. Examples for each of it present in figure 1. Following meticulous evaluations, we selected three models for the curation of AI-generated articles, resulting in a total of 21,129 AI-generated news articles in Hindi from two Hindi news sources BBC and NDTV as discussed in table 1. In addition, we compiled responses from the rejected models for 100 data points each, thereby providing a valuable resource for future research endeavors.

3 Experiments

Recently, four methods and their combinations have been proposed for AI-generated text detection in Indian Languages : *(i) RADAR (Hu et al., 2023),*



(a) Human-written text

(b) AI-generated text

Figure 3: Intrinsic dimensions for Human written text and AI-generated text in Hindi are different. English translations for the Hindi sentences are provided for clarity. For human-written text, the translation is *Hello my name is Ramesh and I go to school*, and for AI-generated text, the translation is *I cannot fulfill this request of yours.* Note that the dimensions are symbolic and represent an abstract conceptualization (Tulchinskii et al., 2023)

.

Data Sources	Human Written News Articles	AI Generated News Articles
BBC	1762	5286
NDTV	5281	15843
Total	7043	21129

Table 1: Number of human-written and AI-generated news articles in Hindi.

(ii) Intrinsic Dimension Estimation (Tulchinskii et al., 2023), (iii) J-Guard (Kumarage et al., 2023), (iv) ConDA (Bhattacharjee et al., 2023). This paper focuses on critiquing their robustness and presents empirical evidence demonstrating their brittleness.

RADAR: Robust AI-text detector via adversarial learning (RADAR) (Hu et al., 2023) is a novel framework that employs adversarial training to enhance AI-text detection. RADAR's approach involves training a robust detector and a paraphraser, which generates text aimed at evading detection. Evaluation across various LLMs and datasets demonstrates RADAR's significant performance advantage, particularly in scenarios involving paraphrasing. We also observe RADAR's strong transferability across LLMs, further highlighting its potential for improved text detection capabilities, as exemplified with GPT-3.5-Turbo.

Intrinsic Dimension Estimation: Intrinsic Dimension estimation (Tulchinskii et al., 2023) introduces an invariant property for human-written text—namely, the intrinsic dimension of the underlying embedding manifold. This metric remains consistent across diverse text domains, varying human writer proficiency, and different languages. We present an example of it in fig 3.

J-Guard: J-Guard (Journalism Guided Adversarially Robust Detection of AI-generated News) (Kumarage et al., 2023) is an advanced framework designed to tackle the growing issue of AI-generated news, which can spread misinformation online. Unlike general AI text detection methods, J-Guard specializes in identifying AI-generated news with high reliability and improved resistance to adversarial attacks.

ConDA: The Contrastive domain adaptation framework (ConDA) addresses the problem of AI-generated text detection by framing it as an unsupervised domain adaptation task where the domains are different large language models(LLMs). The framework assumes that we have access to labeled source data and unlabeled target data. This framework blends standard domain adaptation techniques with the representation power of contrastive learning to learn domain invariant representations that are effective for the final unsupervised detection task. ConDA leverages the power of both, unsupervised domain adaptation and self-supervised representation learning for AI-generated text detection.

4 Results

This section discusses the results based on the models we discussed in the previous section. The results can be reproduced using the code and data present in http://bit.ly/3Ts0BwJ.

Evaluation for RADAR: To evaluate results from RADAR, we present accuracy, precision, recall, and F1 scores along with their classification rate. Classification rate is defined as the model's ability to accurately discern whether a given text is human-written or generated using a large language model. It is calculated as the percentage of correctly classified data points relative to the total number of data points.

BBC Dataset			
	BARD	GPT-3.5	GPT-4
Accuracy	72.211	37.089	37.486
Precision	14.634	0.000	2.985
Recall	50.0	0.000	0.795
F1-score	22.642	0.000	1.255
NDTV Dataset			
	BARD	GPT-3.5	GPT-4
Accuracy	53.380	48.959	49.205
Precision	79.024	12.838	25.862
Recall	9.203	0.360	0.852
F1-score	16.486	0.699	1.650

Table 2: RADAR exhibits a higher level of proficiency in distinguishing BARD responses compared to GPT-3.5/GPT-4 responses. Accuracy for BARD responses is 72.211% in the BBC Dataset and 53.380% in the NDTV dataset. These values significantly surpass the accuracy for GPT-3.5 and GPT-4 responses. It is crucial to note that the model fails to identify any GPT-3.5 responses in the BBC Dataset as AI-generated, resulting in 0% precision, recall, and F1-score.

BBC Dataset				
	Human-written	BARD	GPT-3.5	GPT-4
Classification rate	74.177	4.427	0.000	0.795
NDTV Dataset				
	Human-written	BARD	GPT-3.5	GPT-4
Classification rate	97.557	9.203	0.360	0.852

Table 3: RADAR results based on classification rate. The classification rate is defined as the % of data points belonging to a specific class that the model correctly classifies as either human-written or AI-generated. We observe that RADAR successfully classifies human-written text with a classification rate of 74.1% in the BBC Dataset and 97.5% in the NDTV Dataset. However, it exhibits limitations in accurately identifying AI-generated text as AI-generated. Notably, the classification rate drops to 0% in the case of GPT-3.5 in the BBC Dataset.

Observation from RADAR: Our experiments demonstrate BARD responses exhibit a higher level of detectability in both the BBC dataset and NDTV dataset by a significant difference as discussed in table 2 and table 3.

Evaluation for Intrinsic Dimensionality: Intrinsic Dimensionality uses a Maximum Likelihood estimate (MLE) to assess the data likelihood of the text to provide an estimate of the dimension of the given text, as well as the Persistent Homology Dimension estimator (PHD), operates within the realm of persistent homology, a mathematical framework that captures topological features across different spatial scales. By analyzing the topological features and their lifespans in the dataset, PHD provides a sample-efficient and noise-tolerant estimation of the dataset's dimension. We present the box plots for our results in appendix B.1.

Observation from Intrinsic Dimensionality: We note MLE and PHD between human text and responses generated by BARD. This discrepancy implies that BARD responses are detectable and distinct from human-written text. Conversely, both GPT-3.5 & GPT-4 responses exhibit MLE and PHD values identical to those of human text, posing a challenge in distinguishing these responses as AI-generated as demonstrated in table 4.

BBC Dataset				
	Human-written	BARD	GPT-3.5	GPT-4
MLE	10.016	7.272	9.796	9.541
PHD	6.967	3.120	6.882	7.002
NDTV Dataset				
MLE	9.592	7.061	9.549	9.416
PHD	6.781	3.105	6.720	6.900

Table 4: PHD and MLE values for various text sources. A greater discrepancy between PHD and MLE values of Human-written text and AI-generated text indicates higher detectability of AI-generated texts. Lower MLE and PHD values for BARD responses make it easier to identify them as AI-generated text compared to GPT-3.5 and GPT-4 responses.

Evaluation for J Guard: We showcase the performance metrics, including accuracy, precision, recall, and F1-score, to evaluate the model. The training process involves utilizing samples generated by the AI generator, and subsequently, the model is tested on an independent set comprising samples from the same AI generator. We adhere to the 7:2:1 train-test-validation split of the dataset, mirroring the approach employed by (Kumarage et al., 2023).

BBC Dataset			
	BARD	GPT-3.5	GPT-4
Accuracy	99.007	99.291	98.440
Precision	99.709	99.128	99.718
Recall	98.281	99.417	97.245
F1-score	98.990	99.272	98.466
NDTV Dataset			
	BARD	GPT-3.5	GPT-4
Accuracy	99.290	98.958	99.242
Precision	99.505	99.606	99.229
Recall	99.016	98.249	99.229
F1-score	99.260	98.923	99.229

Table 5: The table presents accuracy, precision, recall, and F1-score metrics for J-Guard framework trained on a specific dataset and subsequently tested on the same dataset. The framework demonstrates strong performance across the responses of all the considered Language Models (LLMs), with the lowest observed accuracy to be 98.440% in the case of GPT-4 responses in BBC Dataset.

Observarion from J Guard: We observe that the J-Guard framework classifies the texts efficiently when trained on samples from the same large language model as discussed in table 5.

Evaluation for ConDa: We present the accuracy, precision, recall, and F1-score for both datasets. ConDA uses unsupervised domain adaptation and self-supervised contrastive learning to effectively leverage labeled source domain and unlabeled target domain data. We utilize the best-performing pre-trained model with GPT-3 as the target generator i.e. the unlabelled target data is generated by GPT-3.

BBC Dataset			
	BARD	GPT-3.5	GPT-4
Accuracy	47.645	45.658	43.445
Precision	47.456	45.099	42.212
Recall	43.927	39.955	35.528
F1-score	45.623	42.371	38.582
NDTV Dataset			
	BARD	GPT-3.5	GPT-4
Accuracy	55.245	50.587	51.856
Precision	57.030	50.899	52.736
Recall	42.548	33.232	35.770
F1-score	48.736	40.211	42.627

Table 6: Our observations indicate that within the ConDA framework, detectability of BARD is marginally superior to that of GPT-3.5 and GPT-4, achieving accuracies of 47.645% and 55.245% in the BBC and NDTV Datasets, respectively.

Observation from ConDa: We observe that the model performs better at detecting BARD samples as compared to both the GPT models for both the BBC and NDTV datasets. However, the difference is more noticeable on the NDTV Dataset as present in table 6.

4.1 Overall Analysis on these models

In our investigation, we compare the efficacy of four AI-generated text detection models, RADAR, J Guard, ConDA, and Intrinsic dimension estimation. Our experiments reveal that both, RADAR and Intrinsic dimension estimation methods showed shortcomings in differentiating between AI-generated and human-written texts.

These methods demonstrate suboptimal results in the classification of texts indicating limitations in their ability to categorize the texts. On the other hand, the J-Guard framework demonstrates a significantly superior performance when compared to the other models. The ConDA framework performs better than the Intrinsic Dimension Estimation and RADAR methods in distinguishing between AI-generated and human-written texts. However, it still falls short of the performance achieved by the J-Guard model, which stands out as the most effective in our evaluations.

5 AI Detectability Index for Hindi (ADI_{hi})

As the landscape of language models evolves rapidly, with new models continually emerging, the long-term viability of existing AGTD techniques may face challenges. Continuous adaptation and innovation in AGTD approaches will be essential to effectively cope with the dynamic nature of advanced language models. For this purpose, we employ the AI Detectability Index formula proposed by (Chakraborty et al., 2023). The formula is as follows:

$$ADI_x = \frac{100}{U \times 2} * [\sum_{x=1}^{U} \{\delta_1(x) * \frac{\left(P_t - L_H^{plx}\right)}{\left(1 - \mu_H^{plx}\right)}\} + \{\delta_2(x) * \frac{\left(B_t - L_H^{brsty}\right)}{\left(1 - \mu_H^{brsty}\right)}\}] \quad (1)$$

where, $P_t = \frac{1}{U} * \{\sum_{x=1}^{U} \left(logp_u^{i} - logp_u^{i+1}\right)\}$ and $B_t = \frac{1}{U} * \{\sum_{x=1}^{U} \left(logp_u^{i+(i+1)+(i+2)} - logp_u^{(i+3)+(i+4)+(i+5)}\right)\}$.

The formulation of ADI_{hi} takes perplexity and burstiness as the foundation for ADI calculation. As asserted by (Chakraborty et al., 2023), alternative methods proposed for AGTD including negative log curvature(Mitchell et al., 2023), stylistic features(Lagutina et al., 2019; Neal et al., 2018), and classification are considered as derivate functions of perplexity and burstiness. Any pattern revealed by these alternative AGTD methods will be effectively encapsulated by the perplexity and burstiness. As discussed in the preceding sections, the task of distinguishing between human-written and AI-generated text has become increasingly challenging. For ADI calculation, the mean perplexity (μ_H^{plx}) and burstiness (μ_H^{brsty}) are derived from human-written texts and to enhance the comparison between the current text and human text, Le Cam's lemma (Cam, 1986-2012) is applied, utilizing precalculated values (L_H^{plx} and L_H^{brsty}) as discussed by (Chakraborty et al., 2023). To assess the overall contrast, a summation is employed over all the data points, denoted as U in the formulation. Ranking of the LLMs based on their detectability is essential for comparative measures. This is facilitated by utilizing multiplicative damping factors, $\delta_1(x)$ and $\delta_2(x)$, which are calculated based on $\mu \pm rank_x \times \sigma$. We calculate the initial value of ADI_{hi} for all the LLMs, considering $\delta_1(x)$ and $\delta_2(x)$ as 0.5. The mean (μ) and standard deviation (σ) derived from the initial ADI_{hi} are utilized to recalculate the ADI_{hi} for all the LLMs. Subsequently, the final ADIs are scaled between 0-100 and then ranked. We showcase the ADI spectrum for various Language Models (LLMs) in Fig 4.

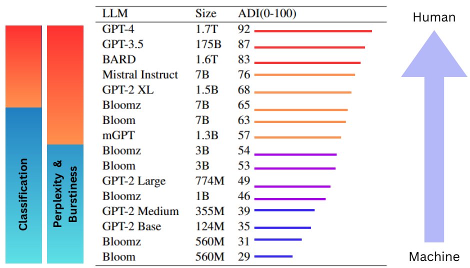


Figure 4: ADI across various LLMs.

6 Conclusion

In conclusion, our research contends that SOTA AGTD techniques are susceptible to fragility. We experimented with 16 distinct LLMs to create the dataset (AG_{hi}) and support the assertion. We introduce the "AI Detectability Index for Hindi" (ADI_{hi}), and we present a means to assess and rank LLMs based on their detectability levels. The excitement and success of LLMs have resulted in their extensive proliferation, and this trend is anticipated to persist regardless of the future course they take. In light of this, the CT^2 benchmark and the ADI_{hi} will continue to play a vital role in catering to the scientific community.

7 Discussion And Limitations

We address the critical issue of AI-generated text detection (AGTD) in the context of the Hindi language, leveraging insights gained from the widespread adoption of large language models (LLMs) trained primarily for English. Despite the valuable contributions, there are certain limitations inherent in this work as discussed in the following points.

- Generalization to Other Languages: The study primarily focuses on the Hindi language, and the findings may not be directly applicable to other languages with distinct linguistic characteristics. Future research could explore the extension of these insights to a broader range of languages.
- Evolution of LLMs: The rapidly evolving nature of LLMs raises the possibility that newer models, not included in the study, may exhibit different behaviors. As such, the generalizability of the findings to future LLMs may be limited.
- Dynamic AI-generated text detection Landscape: The research evaluates AGTD techniques based on the current state of detection methods. However, the dynamic nature of the AI-generated text detection methods suggests that new strategies may emerge, potentially impacting the long-term efficacy of the proposed techniques.
- Real-world Application Challenges: The controlled experimental setting may not fully capture the complexities of real-world applications. Future research could explore the challenges and nuances that arise in practical implementation scenarios.

8 Ethical Considerations

Our experiments reveal the constraints of AGTD methods in Hindi. It is crucial to note that while we envision ADI_{hi} as a tool for constructive purposes, there exists the potential for misuse by malicious entities, especially in generating AI-generated text like fake news that is indistinguishable from human-written content. We strongly caution against any such misuse of our findings.

References

Google AI Bard. 2023. An important next step on our ai journey. [Online; accessed 06-December-2023].

Abdul Basit, Maham Zafar, Xuan Liu, Abdul Rehman Javed, Zunera Jalil, and Kashif Kifayat. 2021. A comprehensive survey of ai-enabled phishing attacks detection techniques. *Telecommunication Systems*, 76:139–154.

BBC. Bbc news hindi. https://www.bbc.com/hindi. [Online; accessed 2023-11-12].

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

Lucien Le Cam. 1986-2012. Asymptotic methods in statistical decision theory.

Megha Chakraborty, SM Tonmoy, SM Zaman, Krish Sharma, Niyar R Barman, Chandan Gupta, Shreya Gautam, Tanay Kumar, Vinija Jain, Aman Chadha, et al. 2023. Counter turing test ctˆ 2: Ai-generated text detection is not as easy as you may think–introducing ai detectability index. *arXiv preprint arXiv:2310.05030*.

Xuanting Chen, Junjie Ye, Can Zu, Nuo Xu, Rui Zheng, Minlong Peng, Jie Zhou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. How robust is gpt-3.5 to predecessors? a comprehensive study on language understanding tasks. *arXiv preprint arXiv:2303.00293*.

O Chernyaeva, TH Hong, YK Park, YH Kim, and G Ren. 2022. Ai generating and detecting manipulated online customers reviews. pages 270–275.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Sarah Kreps, R Miles McCain, and Miles Brundage. 2022. All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation. *Journal of experimental political science*, 9(1):104–117.

Tharindu Kumarage, Amrita Bhattacharjee, Djordje Padejski, Kristy Roschke, Dan Gillmor, Scott Ruston, Huan Liu, and Joshua Garland. 2023. J-guard: Journalism guided adversarially robust detection of ai-generated news.

Ksenia Lagutina, Nadezhda Lagutina, Elena Boychuk, Inna Vorontsova, Elena Shliakhtina, Olga Belyaeva, Ilya Paramonov, and P.G. Demidov. 2019. A survey on stylometric text features. In *2019 25th Conference of Open Innovations Association (FRUCT)*, pages 184–195.

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

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

NDTV. Ndtv india. https://ndtv.in/. [Online; accessed 2023-11-12].

Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. 2018. Surveying stylometry techniques and applications. *ACM Computing Surveys (CSUR)*, 50(6):86.

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

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2023. mgpt: Few-shot learners go multilingual.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Serguei Barannikov, Irina Piontkovskaya, Sergey Nikolenko, and Evgeny Burnaev. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts.

Wikipedia. 2023. List of languages by number of native speakers — Wikipedia, the free encyclopedia. Article Link. [Online; accessed 06-December-2023].

BigScience Workshop and Teven Le Scao et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

A Appendix

This section provides supplementary material in the form of additional examples, implementation details, etc. to bolster the reader's understanding of the concepts presented in this work.

A.1 Hyperparameters for models

We list the hyperparameters employed in text generation for both included and excluded models. Various hyperparameters were applied to evaluate the rejected models, but their outcomes did not meet our criteria, leading to their exclusion from further consideration. Figure 5 provides a comprehensive overview of all the hyperparameters for the models.

<table>
<tr><th>Models</th><th>Hyperparameters</th></tr>
<tr><td>BARD</td><td>-</td></tr>
<tr><td>GPT-3.5</td><td rowspan="2">temperature: 1
max_tokens: 1000
frequency_penalty: 0</td></tr>
<tr><td>GPT-4</td></tr>
<tr><td>GPT-2 (base, medium, large, xl)</td><td>temperature: 0.4
length_penalty: 2.0
early_stopping: True</td></tr>
<tr><td>Bloom (560M, 3B, 7B)</td><td rowspan="2">temperature: 0.85
top_p: 0.95
repetition_penalty: 1.3</td></tr>
<tr><td>Bloomz (560M, 1B, 3B, 7B)</td></tr>
<tr><td>Mistral Instruct 7B</td><td>temperature: 0.5</td></tr>
<tr><td>MGPT 1.3B</td><td>temperature: 0.
no_repeat_ngram_size: 3
length_penalty: 1.5</td></tr>
</table>

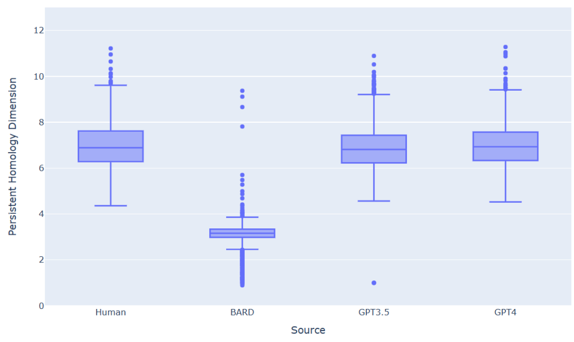
Figure 5: Hyperparameters used to generate text from different models. No hyperparameters are available for BARD as the data was collected directly from the BARD website.

B Results

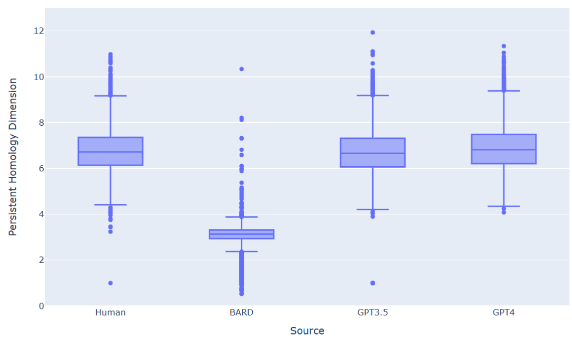
In this section, we discuss additional results from two of the AI-generated text detection techniques i.e. J-Guard and Intrinsic Dimension Estimation.

B.1 Results from Intrinsic Dimension Estimation

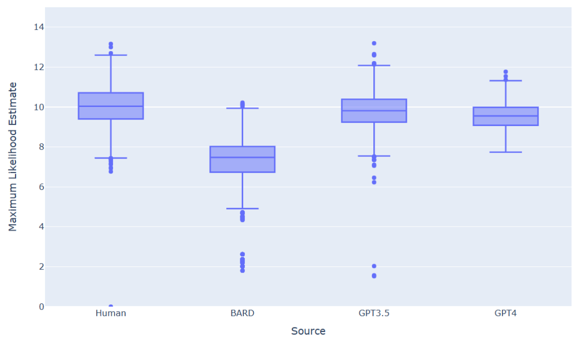
Here we present results for Intrinsic dimension estimation in the form of box plots.



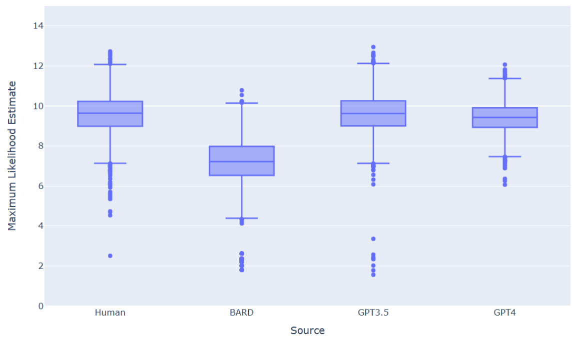
(a) PHD values for various text sources in the BBC Dataset



(b) PHD values for various text sources in the NDTV Dataset



(c) MLE values for various text sources in the BBC Dataset



(d) MLE values for various text sources in the NDTV Dataset

Figure 6: Maximum likelihood estimate(MLE) and persistent homology dimension(PHD) across datasets for a range of text sources.Maximum likelihood estimate refers to the approach of estimating the intrinsic dimension of a dataset by evaluating. The calculation of PHD involves leveraging persistent homology to asses the intrinsic dimension of the dataset. Persistent homology analyzes the topological features and structures present in the dataset at different scales to estimate the intrinsic dimension of the dataset. We observe that the PHD and MLE values for BARD responses are lower compared to the human-written texts. This significant difference makes it easier to distinguish between the human-written texts and BARD responses. However, the PHD and MLE values for GPT-3.5 and GPT-4 are identical to those of human-written text. These similar values pose a challenge in distinguishing between them.

B.2 Results from J-Guard

Here we present cross-domain performance metrics like accuracy, precision, and recall for the J-Guard framework. In this evaluation, the model undergoes training on a specific dataset and is subsequently tested on each distinct dataset. This method provides insights across various domains, exhibiting the model's ability to generalize to a dataset not encountered during the training phase.

	Testing Dataset							
Training Dataset			BBC			NDTV		
			BARD	GPT-3.5	GPT-4	BARD	GPT-3.5	GPT-4
	BBC	BARD	99.007	98.156	99.574	79.072	80.919	81.013
		GPT-3.5	98.440	99.291	99.149	76.752	79.640	79.403
		GPT-4	88.963	97.731	98.440	73.674	81.297	81.0.13
	NDTV	BARD	99.291	97.589	98.156	99.290	94.602	94.366
		GPT-3.5	99.433	99.574	99.574	99.006	98.958	99.432
		GPT-4	99.574	99.433	99.574	99.006	98.438	99.242

Table 7: Cross-domain accuracy for J-Guard framework: Cross-domain accuracy is calculated by training the model on one specific dataset and evaluating its performance on every datasets. We observe a drastic decrease in model accuracy when training on the BBC dataset and subsequently testing on the NDTV dataset. In contrast, when training on the NDTV dataset and then subsequently testing on the BBC dataset the model's accuracy remains relatively stable and in certain cases even increases.

	Testing Dataset							
Training Dataset			BBC			NDTV		
			BARD	GPT-3.5	GPT-4	BARD	GPT-3.5	GPT-4
	BBC	BARD	99.709	97.960	99.724	71.165	73.063	72.701
		GPT-3.5	99.706	99.128	98.904	69.259	71.296	71.069
		GPT-4	100.000	99.696	99.718	69.726	75.099	74.557
	NDTV	BARD	100.000	99.695	100.000	99.505	99.566	99.462
		GPT-3.5	100.000	100.000	100.000	99.800	99.606	99.613
		GPT-4	100.000	100.000	100.000	99.900	99.404	99.229

Table 8: Cross-domain precision for J-Guard framework: We note a significant decrease in precision, ranging from 69% to 75%, when the model is trained on the BBC dataset and evaluated on the NDTV dataset. Conversely, when the model is trained on the NDTV dataset and tested on the BBC dataset, precision consistently reaches almost 100%. This observation suggests that the model excels in distinguishing between human-written texts and AI-generated texts specifically when trained on the NDTV dataset and evaluated on the BBC dataset, but not vice versa.

	Testing Dataset							
Training Dataset			BBC			NDTV		
			BARD	GPT-3.5	GPT-4	BARD	GPT-3.5	GPT-4
	BBC	BARD	98.281	98.251	99.449	94.980	96.304	98.266
		GPT-3.5	97.135	99.417	99.449	92.913	97.374	97.977
		GPT-4	77.650	95.627	97.245	80.020	92.121	93.160
	NDTV	BARD	98.567	95.335	96.419	99.016	89.300	89.017
		GPT-3.5	98.854	99.125	99.176	98.130	98.249	99.229
		GPT-4	99.140	98.834	99.174	98.032	97.374	99.229

Table 9: Cross-domain recall of the J-Guard framework: It is noteworthy that in nearly all combinations, a recall score exceeding 90 is observed, with a few exceptions such as when the model is trained on BBC GPT-4 and tested on BBC BARD (77.650) and when the model is trained on BBC GPT-4 and tested on NDTV BARD (80.02). A higher recall score indicates the model's effectiveness in accurately identifying AI-generated text as such.

	Testing Dataset							
Training Dataset			BBC			NDTV		
			BARD	GPT-3.5	GPT-4	BARD	GPT-3.5	GPT-4
	BBC	BARD	98.990	98.108	99.586	81.366	83.089	83.572
		GPT-3.5	98.403	99.272	99.176	79.361	82.319	82.382
		GPT-4	87.419	97.619	98.466	74.519	82.744	82.827
	NDTV	BARD	99.279	97.466	98.177	99.260	94.154	93.950
		GPT-3.5	99.424	99.561	99.585	98.958	98.923	99.421
		GPT-4	99.568	99.413	99.585	98.957	98.378	99.229

Table 10: Cross-domain F1-score for J-Guard framework: A high F1 score indicates the model's proficiency in achieving a balance between precision and recall, particularly in accurately identifying AI-generated text.