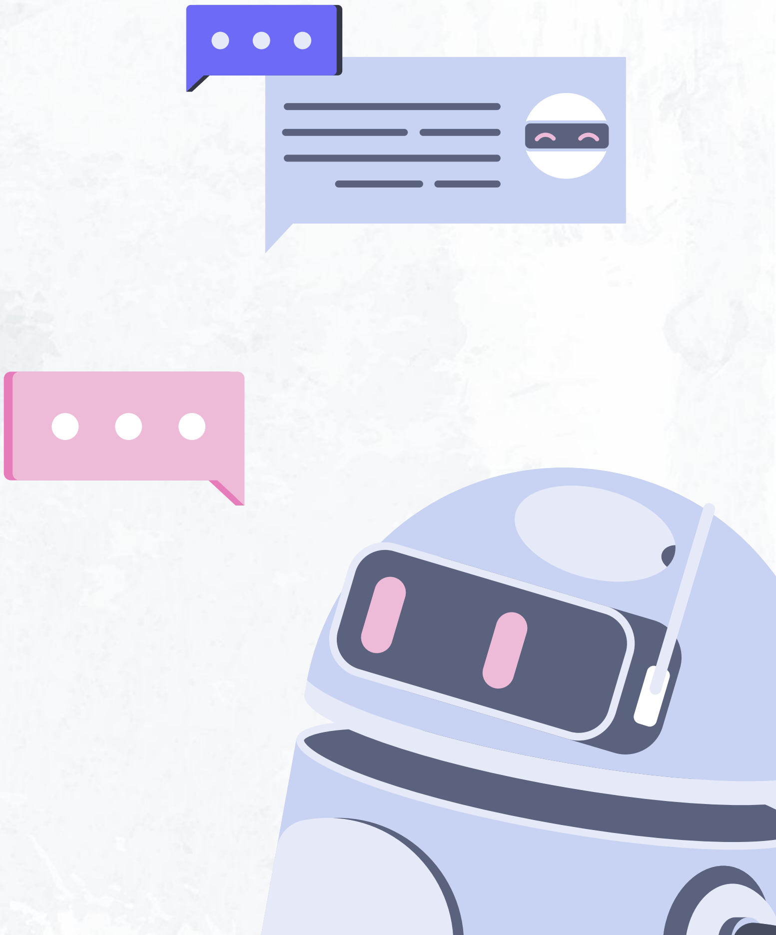
Group 05

AI Generated Text Detection



(AI)

19001746 : - Vignagajan
19001762 : - Vinothini
19000685: - Kavishan

Table of contents

01 →

Introduction

(AI)

AI Generated Text

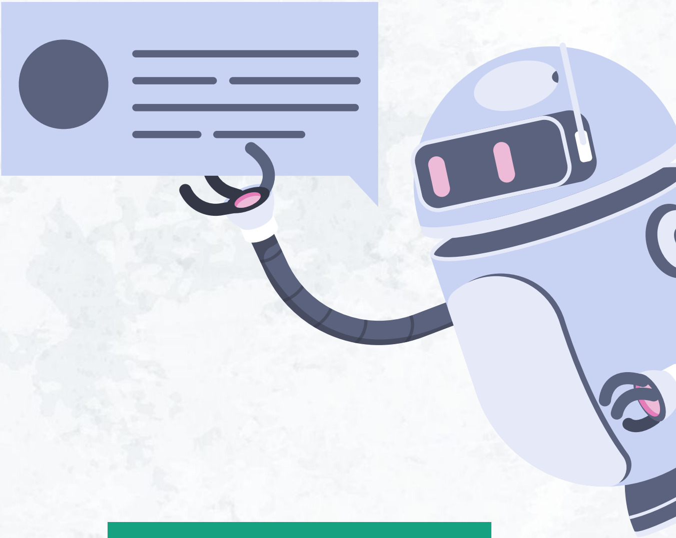
AI generated content have been recently popular after publicly available LLMs(ChatGPT) which produce high quality text. But there are two main problems.

(a) Misinformation

AI generated text doesn’t have credibility because it doesn’t give any references about the sources where the information is retrieved from.

(b) Fraudulent activities

AI to create fake content at little cost, and experts say the output can do a better job fooling the public than human-created content.



AI Generated Text VS Human Generated Text

AI generated text have some following features which could be used to distinguish from human text.

- **Formulaic structure**
- **Specific patterns (watermarks)**
- **Low Perplexity**

02 →

Traditional Approaches

Using Statistical Metrics

These metrics provide quantitative insights into the linguistic characteristics of a text, and their values can often reveal patterns indicative of automated text generation

(a) **Entropy** → Higher entropy of next word prediction suggest greater diversity, while lower values indicate more predictability.

(b) **Perplexity** → Lower perplexity because they are designed to optimize word prediction, making them more predictable and coherent.

(c) **n-gram frequency** → Machine-generated texts may have a higher frequency of specific n-grams because they can inadvertently replicate patterns present in their training data.

Detecting Fake Content with Relative Entropy Scoring

- Markovian n-gram language models represent sequences of words. For instance, with a 3-gram model, the probability of a sequence of k > 2 words is given by:

  $$p(w1 \ldots wk) = p(w1)p(w2|w1) \cdots p(wk|wk{-}2wk{-}1)$$

- They used perplexities computed and the detection performed by **thresholding these perplexities**, where the threshold is tuned on some development data.

		3-gram model			4-gram model		
		newsp	euro	wiki	newsp	euro	wiki
pw5	2k	0.70	0.76	0.26	0.70	0.78	0.28
	5k	**0.90**	0.89	0.39	**0.90**	0.85	0.37
pw10	2k	0.31	0.50	0.21	0.30	0.51	0.17
	5k	0.43	0.65	0.30	0.42	0.67	0.29
ws10	2k	0.85	**0.94**	0.44	0.81	0.95	0.51
	5k	**0.97**	**0.97**	0.71	**0.96**	0.95	0.73
ws25	2k	**1.00**	**0.99**	0.79	**1.00**	**0.99**	**0.99**
	5k	**0.97**	**1.00**	0.80	**0.98**	**1.00**	**0.98**
ws50	2k	**1.00**	**1.00**	**0.90**	**1.00**	**1.00**	**1.00**
	5k	**1.00**	**1.00**	**0.91**	**1.00**	**1.00**	**1.00**
lm2	2k	**0.95**	0.88	0.83	**0.95**	0.87	**0.97**
	5k	**0.96**	**0.92**	**0.90**	**0.94**	**0.96**	**0.97**
lm3	2k	0.39	0.25	0.20	0.45	0.27	0.29
	5k	0.56	0.25	0.21	0.60	0.30	0.38
lm4	2k	0.46	0.25	0.28	0.48	0.28	0.41
	5k	0.60	0.25	0.21	0.66	0.29	0.44
spam	2k			**1.00**			**1.00**

Lavergne, T., Urvoy, T. and Yvon, F., 2008. Detecting Fake Content with Relative Entropy Scoring. Pan, 8(27-31), p.4.

Detecting Fake Content with Relative Entropy Scoring

- They used entropy-based detector uses a similar strategy to score n-grams according to the semantic relation between their first and last words.
- This is done by finding **useful n-grams**, ie. n-grams, that can help detect fake content
- Using the entropy scoring, if the score is higher, the text is AI generated and lower its AI generated.
- This scoring method is not scalable because this scoring is expensive

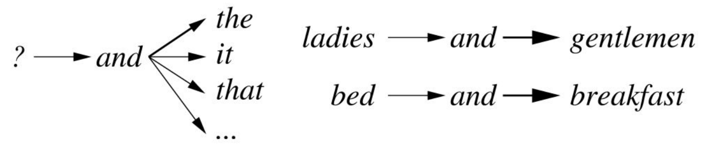
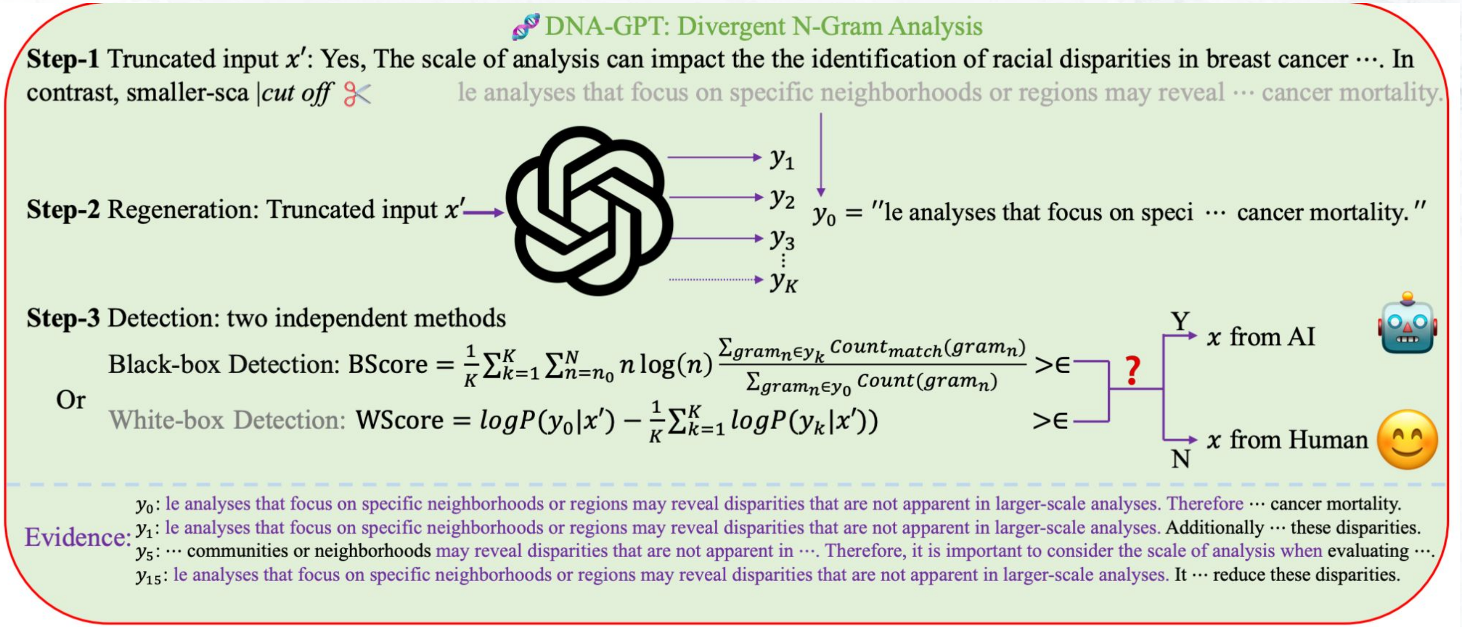


Figure 4. Examples of useful n-grams. “and” has many possible successors, “the” being the most likely; in comparison, “ladies and” has few plausible continuations, the most probable being “gentlemen”; likewise for “bed and”, which is almost always followed by “breakfast”. Finding “bed and the” in a text is thus a strong indicator of forgery.

$$KL(p(\cdot|h)||p(\cdot|h')) = \sum_{w} p(w|h) log \frac{p(w|h)}{p(w|h')}$$

Lavergne, T., Urvoy, T. and Yvon, F., 2008. Detecting Fake Content with Relative Entropy Scoring. *Pan*, *8*(27-31), p.4.

DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text



Yang, X., Cheng, W., Petzold, L., Wang, W.Y. and Chen, H., 2023. DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text. *arXiv preprint arXiv:2305.17359*.

DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text

- They conducted extensive experiments on the most advanced LLMs from OpenAI, including **text-davinci-003, GPT-3.5-turbo, and GPT-4, as well as open-source models such as GPT-NeoX-20B and LLaMa-13B**.
- Training free flexible strategy
- The entire method depends on GPT models, therefore scalability of this method is questionable for future

Yang, X., Cheng, W., Petzold, L., Wang, W.Y. and Chen, H., 2023. DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text. *arXiv preprint arXiv:2305.17359*.

03 →

Existing Tools

(AI)

Impact of Generative AIs

In November 2022, ChatGPT was launched.Within two months of its launch, it had over 100 million subscribers and was labelled “the fastest growing consumer app ever”

Testing of Detection Tools for AI-Generated Text

Testing of Detection Tools for AI-Generated Text

Debora Weber-Wulff (University of Applied Sciences HTW Berlin, Germany, weberwu@htw-berlin.de), (corresponding author)
Alla Anohina-Naumeca (Riga Technical University, Latvia, alla.anohina-naumeca@rtu.lv)
Sonja Bjelobaba (Uppsala University, Sweden, sonja.bjelobaba@crb.uu.se),
Tomáš Foltýnek (Masaryk University, Czechia, foltynek@fi.muni.cz)
Jean Guerrero-Dib (Universidad de Monterrey, Mexico, jean.guerrero@udem.edu.mx),
Olumide Popoola (Queen Mary University of London, UK, O.Popoola@qmul.ac.uk),
Petr Šigut (Masaryk University, Czechia, petrsigut10@gmail.com),
Lorna Waddington (University of Leeds, UK, l.l.waddington@leeds.ac.uk)

https://arxiv.org/ftp/arxiv/papers/2306/2306.15666.pdf

Research

- Can existing detection tools reliably differentiate between human-written text and AI generated text?
- Do machine translation and content obfuscation techniques affect the detection of AI-generated text?

References

The following 14 detection tools were tested

- Check For AI (https://checkforai.com)
- Compilatio (https://ai-detector.compilatio.net/)
- Content at Scale (https://contentatscale.ai/ai-content-detector/)
- Crossplag (https://crossplag.com/ai-content-detector/)
- DetectGPT (https://detectgpt.ericmitchell.ai/)
- Go Winston (https://gowinston.ai)
- GPT Zero (https://gptzero.me/)
- GPT-2 Output Detector Demo (https://openai-openai-detector.hf.space/)
- OpenAI Text Classifier (https://platform.openai.com/ai-text-classifier)
- PlagiarismCheck (https://plagiarismcheck.org/)
- Turnitin (https://demo-ai-writing-10.turnitin.com/home/)
- Writeful GPT Detector (https://x.writefull.com/gpt-detector)
- Writer (https://writer.com/ai-content-detector/)
- ZeroGPT (https://www.zerogpt.com/)

Dataset

- human-written (01-Hum)
- human-written in a non-English language with a subsequent AI/machine Translation to English (02-MT)
- AI-generated text (03-AI and 04-AI)
- AI-generated text with subsequent human manual edits (05-ManEd)
- AI-generated text with subsequent AI/machine paraphrase (06-Para)

Evaluation

Human-written (NEGATIVE) text (docs 01-Hum & 02-MT), and the tool says that it is written by a:		
[100 - 80%) human	True negative	TN
[80 - 60%) human	Partially true negative	PTN
[60 - 40%) human	Unclear	UNC
[40 - 20%) human	Partially false positive	PFP
[20 - 0%] human	False positive	FP
AI-generated (POSITIVE) text (docs 03-AI, 04-AI, 05-ManEd & 06-Para), and the tool says it is written by a:		
[100 - 80%) human	False negative	FN
[80 - 60%) human	Partially false negative	PFN
[60 - 40%) human	Unclear	UNC
[40 - 20%) human	Partially true positive	PTP
[20 - 0%] human	True positive	TP

[or] means inclusive (or) means exclusive

Results

Accuracy of the detection tools (binary approach)

Tool	01-Hum	02-MT	03-AI	04-AI	05-ManEd	06-Para	Total	Accuracy	Rank
Check For AI	9	0	9	8	4	2	32	59%	6
Compilatio	8	9	8	8	5	2	40	74%	2
Content at Scale	9	9	0	0	0	0	18	33%	14
Crossplag	9	6	9	7	4	2	37	69%	4
DetectGPT	9	5	2	8	0	1	25	46%	11
Go Winston	7	7	9	8	4	1	36	67%	5
GPT Zero	6	3	7	7	3	3	29	54%	8
GPT-2 Output Detector Demo	9	7	9	8	5	1	39	72%	3
OpenAI Text Classifier	9	8	2	7	2	1	29	54%	8
PlagiarismCheck	7	5	3	3	1	2	21	39%	13
Turnitin	9	9	8	9	4	2	41	76%	1
Writeful GPT Detector	9	7	2	3	2	0	23	43%	12
Writer	9	7	4	4	2	1	27	50%	10
ZeroGPT	9	5	7	8	2	1	32	59%	6
Average	**94%**	**69%**	**63%**	**70%**	**30%**	**15%**			

ACC_bin = (TN + TP) / (TN + PTN + TP + PTP + FN + PFN + FP + PFP + UNC)

Accuracy of the detection tools(semi-binary approach)

Tool	01-Hum	02-MT	03-AI	04-AI	05-ManEd	06-Para	Total	Accu-racy	Rank
Check For AI	9	3.5	9	8	4	2.5	36	67%	6
Compilatio	8.5	9	8.5	8	5.5	2	41.5	77%	2
Content at Scale	9	9	0	0	0	0	18	33%	14
Crossplag	9	6	9	7	4.5	2	37.5	69%	5
DetectGPT	9	6.5	5.5	8	2	1.5	32.5	60%	10
Go Winston	7.5	7.5	9	8	4.5	1.5	38	70%	4
GPT Zero	6	3	7.5	8	5.5	5.5	35.5	66%	8
GPT-2 Output Detector Demo	9	7	9	8	5	1.5	39.5	73%	3
OpenAI Text Classifier	9	8.5	3.5	7.5	3.5	1.5	33.5	62%	9
PlagiarismCheck	8	6.5	4	4.5	2	2.5	27.5	51%	13
Turnitin	9	9	8.5	9	4.5	2.5	42.5	79%	1
Writeful GPT Detector	9	7.5	5	4.5	2.5	0.5	29	54%	12
Writer	9	7	4.5	5	3	1.5	30	56%	11
ZeroGPT	9	6.5	7	8	3	2.5	36	67%	6
Average	**95%**	**77%**	**71%**	**74%**	**39%**	**22%**			

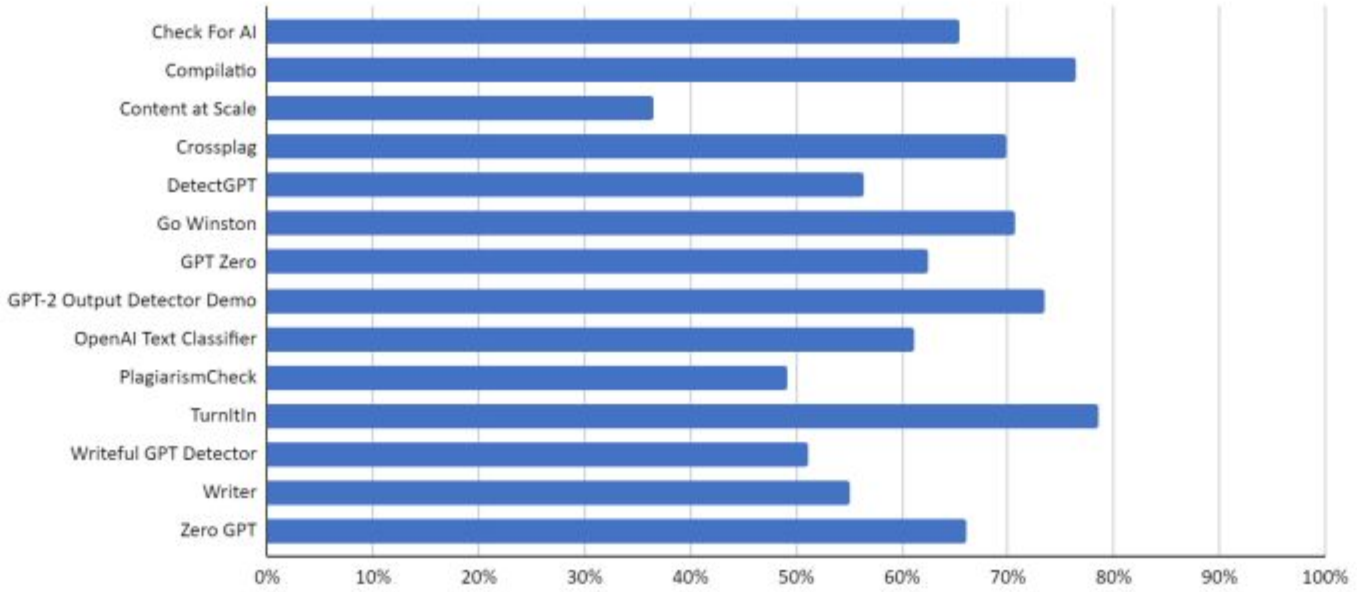
ACC_SEMIBIN = (TN + TP + 0.5 × PTN + 0.5 × PTP) / (TN + PTN + TP + PTP + FN + PFN + FP + PFP + UNC)

Logarithmic approach to accuracy evaluation

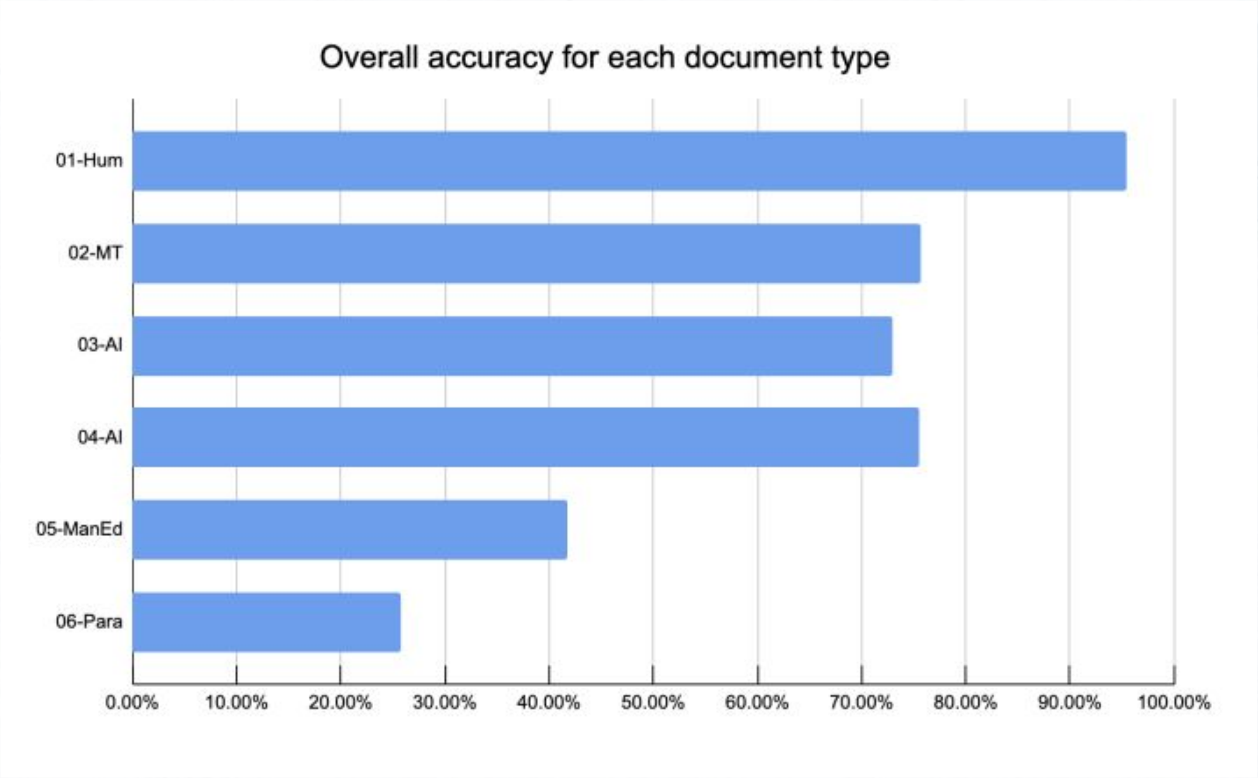
Positive case	Negative case	Score
FN	FP	1
PFN	PFP	2
UNC	UNC	4
PTP	PTN	8
TP	TN	16

Tool	01-Hum	02-MT	03-AI	04-AI	05-ManEd	06-Para	Total	Accu-racy	Rank
Check For AI	144	62	144	129	74	54	607	70%	7
Compilatio	136	144	136	132	91	40	679	79%	2
Content at Scale	144	144	23	24	17	18	370	43%	14
Crossplag	144	99	144	115	76	40	618	72%	6
DetectGPT	144	108	88	129	38	36	543	63%	10
Go Winston	124	124	144	130	79	45	646	75%	4
GPT Zero	102	60	121	128	89	89	589	68%	8
GPT-2 Output Detector Demo	144	114	144	129	84	35	650	75%	3
OpenAI Text Classifier	144	136	67	124	67	48	586	68%	9
PlagiarismCheck	128	108	76	82	50	53	497	58%	12
Turnitin	144	144	136	144	81	53	702	81%	1
Writeful GPT Detector	144	122	81	76	50	20	493	57%	13
Writer	144	117	83	84	53	35	516	60%	11
ZeroGPT	144	108	120	132	65	54	623	72%	5
Average	**96%**	**79%**	**75%**	**77%**	**45%**	**31%**			

Overall accuracy for each tool calculated as an average of all approaches discussed



Overall accuracy for each document type (calculated as an average of all approaches discussed)

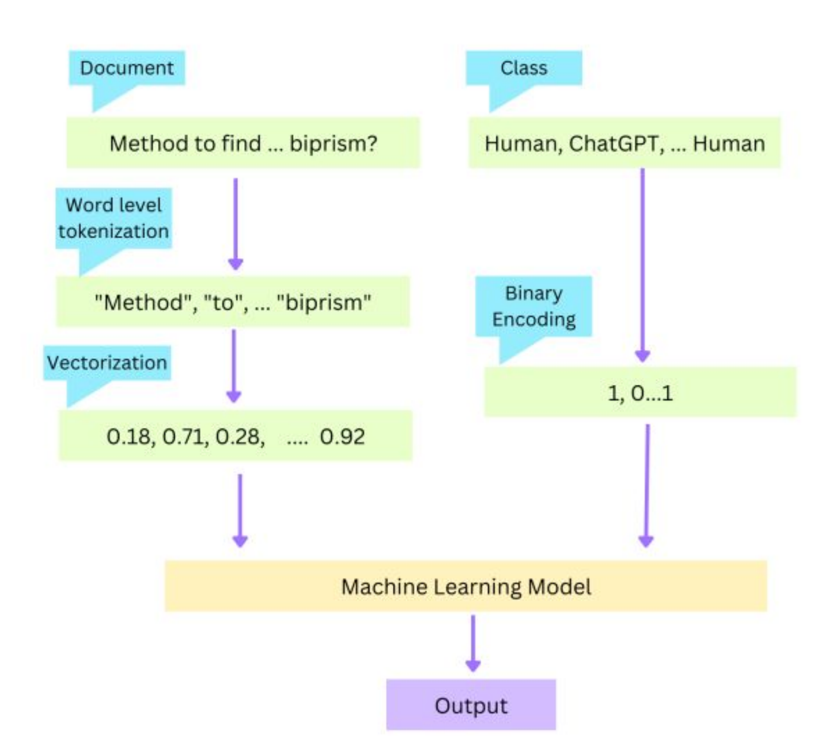


04 →

ML based Approaches

(AI)

Classifier models



Comparison among basic supervised models

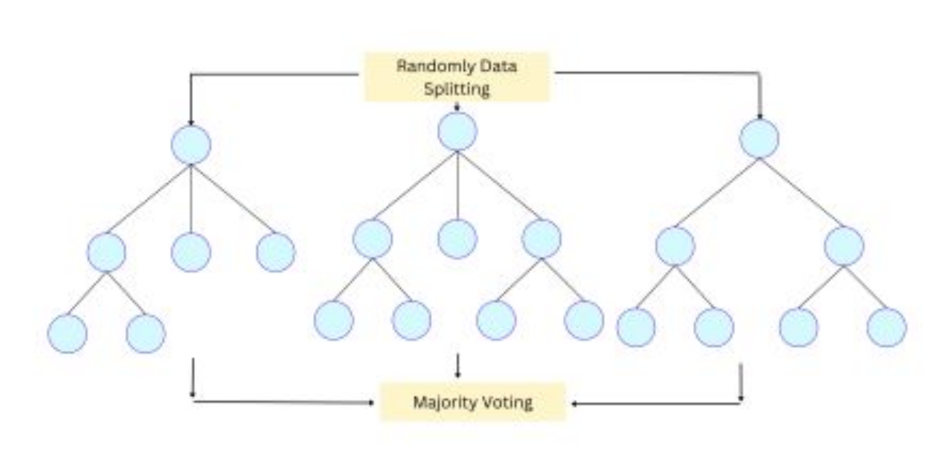
PERFORMANCE OF DIFFERNET CLASSIFIERS

Model	Accuracy	Precision	Recall	F1-Score	MCC
Logistic Regression	0.74	0.73	0.73	0.73	0.48
Support Vector Machines	0.75	0.75	0.71	0.73	0.50
Decision Tree	0.63	0.75	0.79	0.67	0.29
K- Nearest Neighbor	0.69	0.67	0.68	0.67	0.37
Random Forest	0.76	0.73	0.81	0.76	0.53
AdaBoost	0.71	0.68	0.74	0.71	0.43
Bagging Classifier	0.74	0.71	0.75	0.73	0.47
Gradient Boosting	0.71	0.66	0.78	0.72	0.42
Multi-layer Perceptron	0.72	0.73	0.72	0.72	0.43
Long Short-Term Memory	0.73	0.73	0.77	0.75	0.46
Extremely Randomized Trees	**0.77**	**0.74**	**0.78**	**0.76**	**0.54**

Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning (Islam, N., Sutradhar, D., Noor, H., Raya, J.T., Maisha, M.T. and Farid, D.M., 2023. Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning. *arXiv preprint arXiv:2306.01761*.)

Extremely Randomized Trees model

- Uses Term Frequency-Inverse Document Frequency (**TF-IDF**) vectorizer
- TF-IDF is to emphasize the important words
- ERTC is an ensemble algorithm that is based on decision tree.
- At testing, it takes majority voting for prediction.
- the corpus generated by **GPT-3.5**
- Published in **26 May 2023**



Limitations

- **Computationally Intensive:** Constructing **multiple decision trees during the training** phase and **performing majority voting during testing** can be computationally expensive, especially when dealing with a large number of trees and features.

- **Hyperparameter Tuning:** As mentioned, there are several hyperparameters to tune, such as the **number of decision trees** (in this case, 50), **splitting criteria** (gini), and others. Finding the optimal set of hyperparameters can be time-consuming and requires expertise.

CHATGPT OR HUMAN? DETECT AND EXPLAIN. EXPLAINING DECISIONS OF MACHINE LEARNING MODEL FOR DETECTING SHORT CHATGPT-GENERATED TEXT

- The main 2 steps :

  1.have used the **DistilBERT** which is pre-trained for the sequence classification task to do classification of chatgpt generated text

  2.**SHAP** for Explaining Model's Decisions (SHAP can help identify which words or phrases in a given text are the most important in determining the model's output)

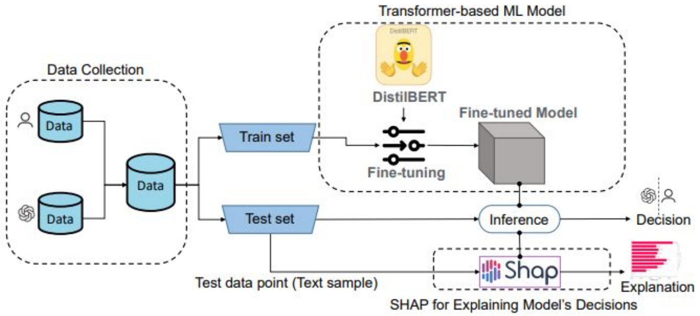


Figure 1: Schematic representation of the study design and building blocks.

Mitrović, S., Andreoletti, D. and Ayoub, O., 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852*.

CHATGPT OR HUMAN? DETECT AND EXPLAIN. EXPLAINING DECISIONS OF MACHINE LEARNING MODEL FOR DETECTING SHORT CHATGPT-GENERATED TEXT

- Published in **30 Jan 2023**
- Achieves accuracy of **98%** corpus for chatgpt Query text
- Achieves accuracy of **79%** corpus for chatgpt rephrase text
- generated by **GPT-3.5**

Observations

- ChatGPT tends to **describe experiences rather than expressing feelings**.
- ChatGPT, unlike humans, **refrains from using personal pronouns** (**no personal pronouns, no expressions of feelings**)
- ChatGPT tends to **use uncommon (unusual) words.**
- Aggressive language and **rude vocabs are never used by ChatGPT**.
- ChatGPT vocabulary is **much more formal**
- misses colloquial terms and abbreviations (e.g. it never uses "&" instead of "and").
- ChatGPT quite repeats itself. A majority of reviews in the ChatGP Tquery dataset, starts with "**the restaurant**", "**this restaurant**" or contains the word "**restaurant**".
- **contain atypical words** or language constructs (e.g. "inattentive")

Limitations

- Transformer based model discriminates better when the text is generated based on customer queries and not by rephrasing original human texts.
- This model is applicable when the **text is short.**

05 →

Proposed Method

Intuition

To improve the accuracy, we have to consider the occurrence of the **unique attributes(UA)** of AI generated text based on the observations from SHAP in DistilBERT. The attributes are,

- no personal pronouns
- no expressions of feelings
- use uncommon (unusual) words
- no rude vocabs
- more formal words
- no colloquial terms and abbreviations (e.g. it never uses “&” instead of “and”).

By using the property Unique Attributes, we can improve the model.

Research Questions & Objectives

RQ: How can we quantify attribute information into the model?
RO: We have to define metric to identify existence of unique attributes

RQ: How can we achieve better performance?
RO: We use the existing best model with modifications to improve overall performance

RQ: How can we incorporate unique attribute information into the model?
RO: We have to train the model with loss that depicts this information

Derivation

We have to quantify above experiments in favour of increasing the discriminating power of DistillBert.

Step 1: Calculate the frequency of unique attributes
Why? : *Quantify the existence of unique attributes*

Step 2: Learnable λi coefficients.
Why? : *These coefficients will be useful in identify contribution of each attribute category for the learning task. We don't know the relationships, therefore we learn and update through a training process.*

Step 3: Normalize the UAFS score by dividing by m and n
Why? : For machine learning purposes.

UAFS

Cosider lists, $UAC = \{attc_1, attc_2, attc_3, ..., attr_m\}$, $UAL = \{attr_{11}, attr_{12}, attr_{13}, ..., attr_{mn}\}$

$$UAFS = \frac{1}{m \cdot n \cdot C_{text}} \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j \cdot C(attr_{ji}) \qquad (1)$$

UAC : Unique Attribute Category List

UAL : Unique Attribute List

$UAFS$: Unique Attribute Frequency Score

$attr_{ji}$: Attribute of attribute category $attc_j$

λ_j : Learnable cofficient of $attc_j$

m : Cardinality of UAC

n : Cardinality of UAL

C_{text} : Number of words in the text

UAFL

We will derive the loss as follows,

$$UAFL = 1 - UAFS \quad (2)$$

$UAFL$: Unique Attribute Frequency Loss

By integrating the above loss with the existing approach mentioned, **DistilBert** during training and we can evaluate the model using the **UAFS** score with the learned parameters.

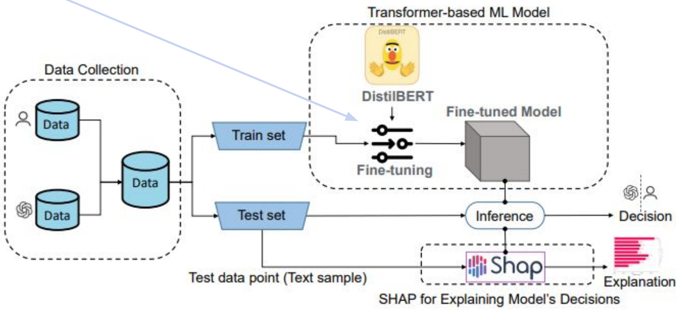


Figure 1: Schematic representation of the study design and building blocks.

Evaluation Methods

- Perplexity-based Classification

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

- ML-based Classification

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

- **Calibrate with state of the art long text detection tools (turnitin (78%)**

Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning (Islam, N., Sutradhar, D., Noor, H., Raya, J.T., Maisha, M.T. and Farid, D.M., 2023. Distinguishing Human Generated Text From ChatGPT Generated Text Using Machine Learning. *arXiv preprint arXiv:2306.01761*.)

Limitations

- UAC and UAL creation needs human effort.
- Initially, we will apply this loss for the DistillBERT on **short text**.
- We need extensive experiments on short text before moving to long text. Then only we can move to long text
- Training will be harder or model needs to trained for long time to

Thanks!

CREDITS: This presentation template was created by **Slidesgo** and includes icons by **Flaticon**, infographics & images by **Freepik** and content by **Eliana Delacour**