Robustness of AI-generated text detectors

Osama Ahmed, Austin Phillips, and Ryan DeVries

Siamese Calibrated Reconstruction Network (SCRN)

Paper 1

Background

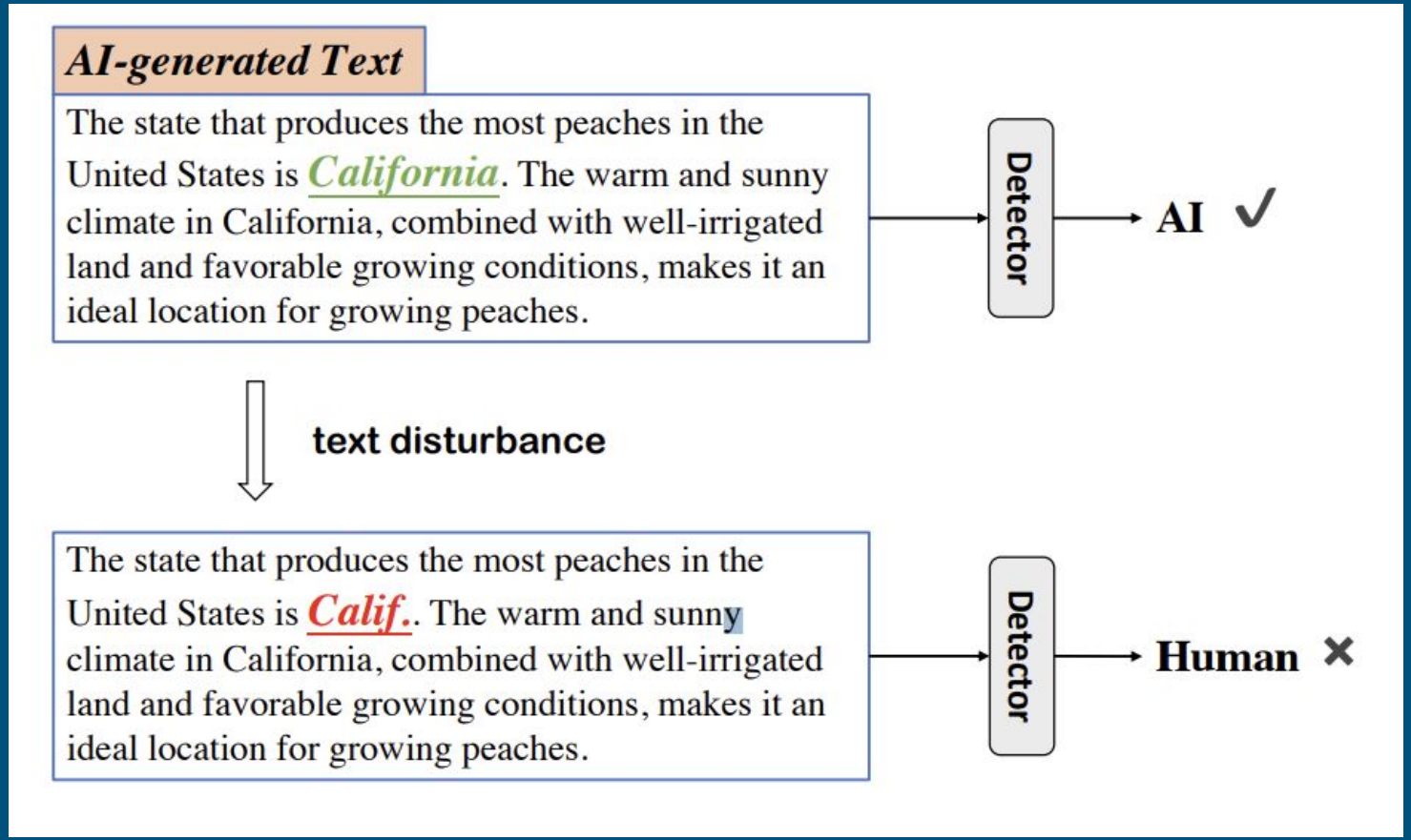
- LLMs have increasingly been used to mimic humans
  - ChatGPT, Deepseek, Cursor AI
- Concerns about misuse of AIGT
  - Bias, fake news, academic dishonesty, etc..
- AIGT have been developed in order to combat misuse

Metric-based vs Model-based

- Two categories of AIGT detection methods
- Metric-Based
  - Use LLM to generate scores (probability, rank, and entropy scores)
- Model-Based
  - Train detectors using supervised learning to classify text using labeled data

Problem with these approaches

- Both types are susceptible to adversarial perturbations
  - Perturbations are word substitutions or character swapping
- Depends on token level features
- AIGT detection should be based on high level features

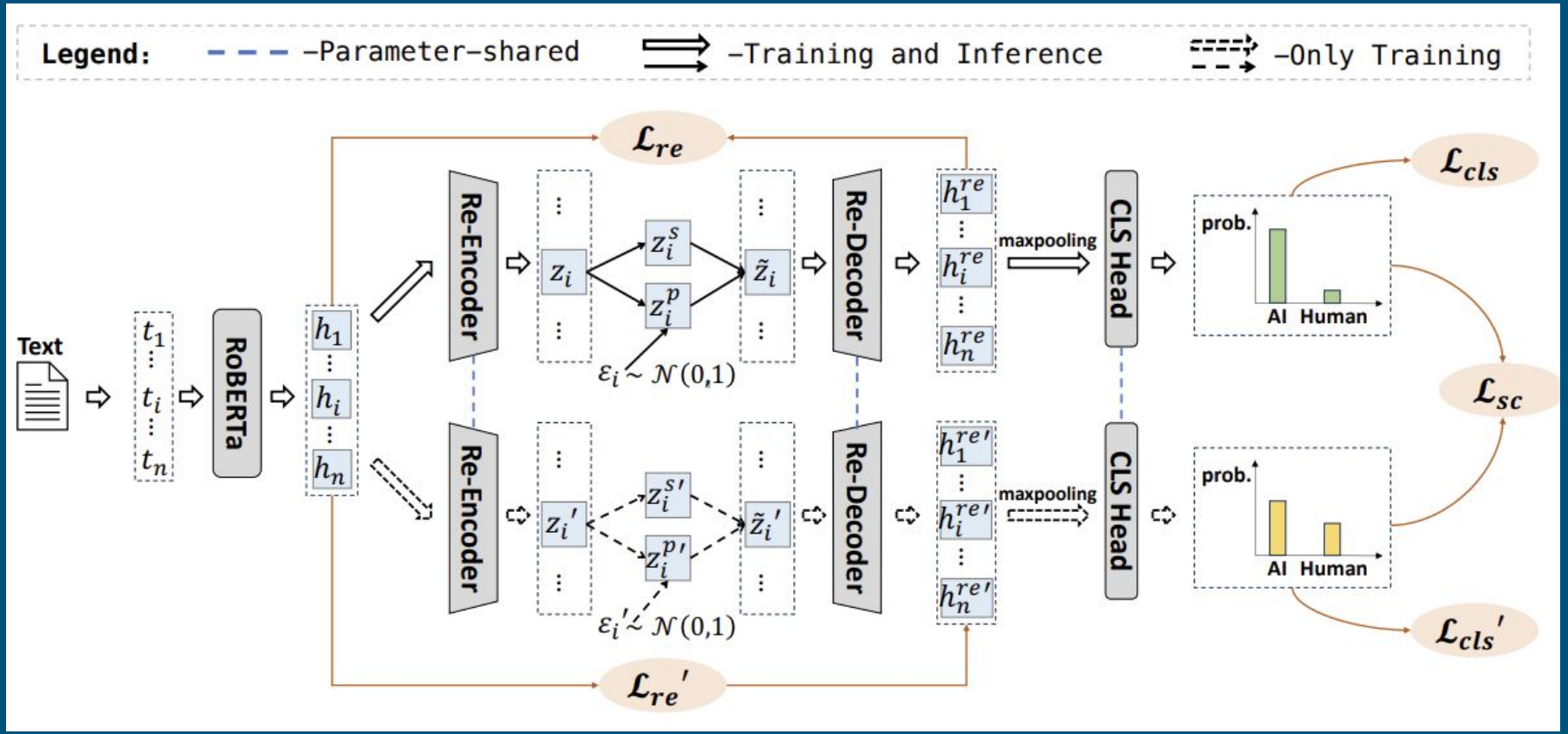

AI-generated Text
The state that produces the most peaches in the United States is California. The warm and sunny climate in California, combined with well-irrigated land and favorable growing conditions, makes it an ideal location for growing peaches.
Detector
AI
text disturbance
The state that produces the most peaches in the United States is Calif.. The warm and sunny climate in California, combined with well-irrigated land and favorable growing conditions, makes it an ideal location for growing peaches.
Detector
Human

Model Architecture

- 3 components to model
- Encoder
  - Pretrained RoBERTa model
- Reconstruction Network
- Classification Head
  - MaxPooling layer to extract features
  - MLP classifier
  - Classifcation loss was cross entropy

Reconstruction Network

- Representation received from RoBERTa encoder
- Representation mapped to a lower-dimensional space by a ReEncoder
  - Splits token representation into semantic and purburation terms
- Representation is reconstructed by the Re-Decoder


Legend:
-Parameter-shared
-Training and Inference
-Only Training
Text
RoBERTa
Re-Encoder
Re-Encoder
Re-Decoder
Re-Decoder
maxpooling
maxpooling
CLS Head
CLS Head
prob.
AI Human
prob.
AI Human
$\mathcal{L}_{re}$
$\mathcal{L}_{re}'$
$\mathcal{L}_{cls}$
$\mathcal{L}_{cls}'$
$\mathcal{L}_{sc}$
$\varepsilon_i \sim \mathcal{N}(0,1)$
$\varepsilon_i' \sim \mathcal{N}(0,1)$

Reconstruction Network

- Latent regularization:
  - $$\mathcal{L}_{\text{reg}}(x) = \frac{1}{n}\sum_{i=1}^{n}\left(\left\|z_i^{(s)}\right\|_2^2 + \left|z_i^{(p)}\right|^2 - \alpha \cdot \log\left(\left|z_i^{(p)}\right|\right)\right)$$
- Reconstruction Loss
  - $$\mathcal{L}_{\text{re}} = -\frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}}\left(\mathcal{L}_{\text{mse}}(x) + \beta \cdot \mathcal{L}_{\text{reg}}(x)\right)$$
- Still not enough to be robust against adversarial attack

Siamese Calibration

- Aim to minimize the symmetric KL divergence of two interference branches
  - Same input subject to independent random noise
- Average of $D_{KL}\left(P(x,\epsilon')\|P(x,\epsilon)\right)$ and $D_{KL}\left(P(x,\epsilon)\|P(x,\epsilon')\right)$
- Total loss during training: $\mathcal{L}_{\text{all}} = \lambda_1(\mathcal{L}_{\text{cls}} + \mathcal{L}'_{\text{cls}}) + \lambda_2(\mathcal{L}_{\text{re}} + \mathcal{L}'_{\text{re}}) + \lambda_3\mathcal{L}_{\text{sc}}$
- During interference, only a single branch is taken

Experimental Setup - Datasets

- Human ChatGPT Comparison Corpus (HC3)
  - Human vs chatGPT responses
- TruthfulQA
  - Testing truthfulness on misconceptions
- Ghostbuster
- SeqXGPT-Bench

Experimental Setup - Training

- Trained on 8 x 32GB NVIDIA V100 GPUs
- Used base versions of pre-trained Bert, RoBERTa, and DeBERT

Hyperparameters	Value
Batch Size	16
Training Epochs	2
Optimizer	AdamW
Learning Rate	1e-4
d	768
d^z	512
α	2.0
β	0.5
λ_1	0.5
λ_2	0.01
λ_3	0.5

Experimental Setup - Testing

- In-domain Robustness
  - Testing: HC3, Training: HC3
- Cross-domain Robustness
  - Testing: TruthfulQA, Training: HC3
- Cross-genre Robustness
  - Testing: Ghostbuster, Training: HC3
- Mixed-source Robustness
  - SeqXGPT-Bench

Experimental Setup - Testing Metrics

- OA - Original Accuracy
- AUA - Accuracy under attack
- ASR - Accuracy success rate
- ANQ - Average number of queries
  - higher = more robust
- Precision, Recall, F1
  - When no attack is done

	Methods	AI → Human				Human → AI				Overall			
		OA ↑	AUA ↑	ASR ↓	ANQ ↑	OA ↑	AUA ↑	ASR ↓	ANQ ↑	OA ↑	AUA ↑	ASR ↓	ANQ ↑
PWWS	Log-Likelihood	96.00	0.00	100.00	957.42	**100.00**	99.00	1.00	1223.54	98.00	49.50	49.49	1090.48
	Log-Rank	96.50	0.00	100.00	974.20	99.00	98.50	0.51	1233.61	97.75	49.25	49.62	1103.90
	Entropy	86.00	0.00	100.00	962.97	95.00	79.00	16.84	1112.75	90.50	39.50	56.35	1037.86
	GLTR	95.00	0.00	100.00	986.10	99.00	83.50	15.66	1136.52	97.00	41.75	56.96	1061.31
	SeqXGPT	99.50	11.00	88.94	1368.07	**100.00**	**100.00**	**0.00**	1224.88	99.75	55.50	44.36	1296.47
	BERT	**100.00**	1.50	98.50	1070.74	99.50	98.50	1.01	1211.18	99.75	50.00	49.87	1140.96
	RoBERTa	**100.00**	38.50	61.50	1332.60	**100.00**	99.50	0.50	1223.76	**100.00**	69.00	31.00	1278.18
	DeBERTa	**100.00**	3.00	97.00	1170.89	**100.00**	99.50	0.50	1223.53	**100.00**	51.25	48.75	1197.21
	ChatGPT-Detector	98.00	0.00	100.00	1074.98	**100.00**	99.50	0.50	1224.60	99.00	49.75	49.75	1149.79
	Flooding	**100.00**	23.00	77.00	1422.60	**100.00**	**100.00**	**0.00**	1225.00	**100.00**	61.50	38.50	1323.81
	RDrop	99.50	67.50	32.16	1585.15	**100.00**	**100.00**	**0.00**	**1225.12**	99.75	83.75	16.04	1405.08
	RanMASK	**100.00**	50.00	50.00	1562.84	**100.00**	**100.00**	**0.00**	1245.97	**100.00**	75.00	25.00	1404.40
	RMLM	**100.00**	73.50	26.50	1561.35	**100.00**	98.50	1.50	1216.52	**100.00**	86.00	14.00	1388.94
	SCRN	**100.00**	**94.50**	**5.50**	**1665.53**	**100.00**	**100.00**	**0.00**	1225.02	**100.00**	**97.25**	**2.75**	**1445.28**
Deep-Word-Bug	Log-Likelihood	96.00	0.00	100.00	109.20	**100.00**	**100.00**	**0.00**	306.24	98.00	50.00	48.98	207.72
	Log-Rank	96.50	0.00	100.00	110.93	99.00	99.00	**0.00**	308.03	97.75	49.50	49.36	209.48
	Entropy	86.00	0.00	100.00	108.82	95.00	92.50	2.63	295.70	90.50	46.25	48.90	202.26
	GLTR	95.00	0.00	100.00	114.79	99.00	98.50	0.51	306.09	97.00	49.25	49.23	210.44
	SeqXGPT	99.50	8.00	91.96	139.96	**100.00**	**100.00**	**0.00**	306.16	99.75	54.00	45.86	223.06
	BERT	**100.00**	12.50	87.50	152.30	99.50	98.50	1.01	282.92	99.75	55.50	44.36	217.61
	RoBERTa	**100.00**	53.00	47.00	293.59	**100.00**	**100.00**	**0.00**	302.58	**100.00**	76.50	23.50	298.08
	DeBERTa	**100.00**	32.00	68.00	171.36	**100.00**	**100.00**	**0.00**	295.56	**100.00**	66.00	34.00	233.46
	ChatGPT-Detector	98.00	13.50	86.22	161.07	**100.00**	**100.00**	**0.00**	301.62	99.00	56.75	42.68	231.34
	Flooding	**100.00**	35.00	65.00	175.48	**100.00**	**100.00**	**0.00**	275.71	**100.00**	67.50	32.50	225.59
	RDrop	99.50	60.50	39.20	367.74	**100.00**	**100.00**	**0.00**	306.09	99.75	80.25	19.55	336.92
	RanMASK	**100.00**	59.00	41.00	332.98	**100.00**	**100.00**	**0.00**	**315.78**	**100.00**	79.50	20.50	324.38
	RMLM	**100.00**	66.00	34.00	377.24	**100.00**	**100.00**	**0.00**	308.91	**100.00**	83.00	17.00	343.08
	SCRN	**100.00**	**87.50**	**12.50**	**437.50**	**100.00**	**100.00**	**0.00**	305.93	**100.00**	**93.75**	**6.25**	**371.72**
Pruthi	Log-Likelihood	96.00	1.00	98.96	9000.48	**100.00**	**100.00**	**0.00**	9519.34	98.00	50.50	48.47	9259.91
	Log-Rank	96.50	1.00	98.96	10084.75	99.00	99.00	**0.00**	9557.18	97.75	50.00	48.85	9820.96
	Entropy	86.00	0.00	100.00	7920.77	95.00	90.50	4.74	9085.92	90.50	45.25	50.00	8503.35
	GLTR	95.00	1.00	98.95	10321.49	99.00	95.00	4.04	9507.22	97.00	48.00	50.52	9914.36
	SeqXGPT	99.50	1.00	98.99	10505.46	**100.00**	**100.00**	**0.00**	9609.78	99.75	50.50	49.37	10057.62
	BERT	**100.00**	21.00	79.00	17460.63	99.50	98.00	1.51	9502.22	99.75	59.50	40.35	13481.42
	RoBERTa	**100.00**	57.00	43.00	20431.19	**100.00**	**100.00**	**0.00**	9561.62	**100.00**	78.50	21.50	14996.40
	DeBERTa	**100.00**	34.50	65.50	17338.08	**100.00**	99.50	0.50	9606.24	**100.00**	67.00	33.00	13472.16
	ChatGPT-Detector	98.00	36.50	62.76	18563.57	**100.00**	99.50	0.50	9591.88	99.00	68.00	31.31	14077.72
	Flooding	**100.00**	68.50	31.50	20823.59	**100.00**	**100.00**	**0.00**	9540.92	**100.00**	84.25	15.75	15182.26
	RDrop	99.50	69.00	30.65	20132.45	**100.00**	99.50	0.50	9516.64	99.75	84.25	15.54	14824.54
	RanMASK	**100.00**	68.50	31.50	21052.49	**100.00**	98.00	2.00	**9748.31**	**100.00**	83.25	16.75	**15400.40**
	RMLM	**100.00**	71.50	28.50	20949.12	**100.00**	98.00	2.00	9373.92	**100.00**	84.75	15.25	15161.52
	SCRN	**100.00**	**82.50**	**17.50**	**21122.83**	**100.00**	**100.00**	**0.00**	9540.20	**100.00**	**91.25**	**8.75**	15331.52

Cross-domain AIGT detection under PWWS attack

	Methods	AI → Human				Human → AI				Overall			
		OA ↑	AUA ↑	ASR ↓	ANQ ↑	OA ↑	AUA ↑	ASR ↓	ANQ ↑	OA ↑	AUA ↑	ASR ↓	ANQ ↑
PWWS	Log-Likelihood	67.00	0.00	100.00	381.32	97.00	97.00	0.00	89.96	82.00	48.50	40.85	235.64
	Log-Rank	72.00	0.00	100.00	374.13	95.50	95.00	0.52	89.63	83.75	47.50	43.28	231.88
	Entropy	43.50	0.00	100.00	433.91	98.00	80.00	18.37	83.96	70.75	40.00	43.46	258.94
	GLTR	66.50	0.00	100.00	376.47	88.50	56.50	36.16	79.80	77.50	28.25	63.55	228.14
	SeqXGPT	93.00	4.00	95.70	397.55	98.50	94.00	4.56	97.64	95.75	49.00	48.83	247.60
	BERT	80.00	0.00	100.00	384.23	99.50	99.50	0.00	89.24	89.75	49.75	44.57	236.74
	RoBERTa	90.50	6.50	92.82	410.38	75.50	74.00	1.99	**99.81**	83.00	40.25	51.51	255.10
	DeBERTa	91.50	1.00	98.91	381.80	98.00	97.50	0.51	88.78	94.75	49.25	48.02	235.29
	ChatGPT-Detector	**96.00**	1.00	98.96	364.00	98.50	88.50	10.15	85.93	**97.25**	44.75	53.98	224.96
	Flooding	90.00	0.00	100.00	387.93	78.00	74.50	4.49	101.59	84.00	37.25	55.65	244.76
	RDrop	89.50	6.00	93.30	429.06	89.00	73.50	17.42	91.10	89.25	39.75	55.46	260.08
	RanMASK	89.00	1.00	98.88	406.12	81.00	78.00	3.70	98.26	85.00	39.50	53.53	252.19
	RMLM	81.50	5.50	93.25	426.98	98.00	98.00	0.00	91.56	89.75	51.75	42.34	259.27
	SCRN	86.50	**40.50**	**53.18**	**551.20**	**99.50**	**99.50**	**0.00**	89.24	93.00	**70.00**	**24.73**	**320.22**

Cross-genre AIGT detection under PWWS attack

	Methods	AI → Human				Human → AI				Overall			
		OA ↑	AUA ↑	ASR ↓	ANQ ↑	OA ↑	AUA ↑	ASR ↓	ANQ ↑	OA ↑	AUA ↑	ASR ↓	ANQ ↑
PWWS	Log-Likelihood	62.00	0.00	100.00	2700.46	**97.50**	**96.50**	**1.03**	**6077.26**	79.75	48.25	39.50	4388.86
	Log-Rank	64.50	0.00	100.00	2734.98	97.50	95.50	2.05	6054.48	81.00	47.75	41.05	4394.73
	Entropy	77.50	0.00	100.00	2783.14	74.00	34.00	54.05	5352.47	75.75	17.00	77.56	4067.80
	GLTR	50.50	0.00	100.00	2696.80	97.50	67.50	30.77	5476.04	74.00	33.75	54.39	4086.42
	SeqXGPT	85.50	0.00	100.00	2712.97	88.00	65.50	25.57	5776.35	**86.75**	32.75	62.25	4244.66
	BERT	57.00	0.00	100.00	2692.61	95.50	75.00	21.47	5619.45	76.25	37.50	50.82	4156.03
	RoBERTa	82.00	0.00	100.00	2655.43	83.00	59.00	28.92	5522.05	82.50	29.50	64.24	4088.74
	DeBERTa	90.00	0.00	100.00	2763.66	77.50	53.50	30.97	5329.64	83.75	26.75	68.06	4046.65
	ChatGPT-Detector	58.50	0.00	100.00	2606.75	93.00	73.00	21.51	5827.88	75.75	36.50	51.82	4217.32
	Flooding	87.50	0.00	100.00	2733.18	82.50	58.00	29.70	5447.84	85.00	29.00	65.88	4090.51
	RDrop	95.00	10.00	89.47	3155.59	73.00	65.00	10.96	5973.84	84.00	37.50	55.36	4564.72
	RanMASK	67.00	2.00	97.01	2667.19	87.00	75.00	13.79	5433.82	77.00	38.50	50.00	4050.50
	RMLM	58.50	9.50	83.76	3397.99	92.00	72.50	21.20	5440.61	75.25	41.00	45.51	4419.30
	SCRN	**94.50**	**71.00**	**24.87**	**4419.16**	70.50	54.50	22.70	5725.79	82.50	**62.75**	**23.94**	**5072.48**

Mixed-source AIGT detection under PWWS attack

	Methods	AI → Human				Human → AI				Overall			
		OA ↑	AUA ↑	ASR ↓	ANQ ↑	OA ↑	AUA ↑	ASR ↓	ANQ ↑	OA ↑	AUA ↑	ASR ↓	ANQ ↑
PWWS	Log-Likelihood	72.00	0.50	99.31	1281.86	62.00	53.50	13.71	1667.91	67.00	27.00	59.70	1474.88
	Log-Rank	73.50	0.50	99.32	1286.24	62.50	56.00	10.40	1697.20	68.00	28.25	58.46	1491.72
	Entropy	63.00	0.00	100.00	1239.29	55.50	27.50	50.45	1396.39	59.25	13.75	76.79	1317.84
	GLTR	76.50	0.00	100.00	1260.99	67.50	19.00	71.85	1285.64	72.00	9.50	86.81	1273.32
	SeqXGPT	**96.50**	65.00	32.64	1867.81	96.00	70.00	27.08	1893.98	**96.25**	67.50	29.87	1880.90
	BERT	90.50	1.00	98.90	1204.52	90.00	59.00	34.44	1815.44	90.25	30.00	66.76	1509.98
	RoBERTa	95.50	64.50	32.46	1840.19	93.00	62.50	32.80	1729.72	94.25	63.50	32.63	1784.96
	DeBERTa	95.50	54.50	42.93	1764.94	96.00	80.00	16.67	1940.47	95.75	67.25	29.77	1852.70
	Flooding	96.00	60.50	36.98	1800.01	95.50	53.00	44.50	1610.45	95.75	56.75	40.73	1705.23
	RDrop	**96.50**	69.00	28.50	1819.95	95.00	70.00	26.32	1815.62	95.75	69.50	27.42	1817.78
	RanMASK	94.00	60.00	36.17	1784.11	86.00	71.00	17.44	1715.72	90.00	65.50	27.22	1749.92
	RMLM	91.00	69.00	24.18	1879.96	91.50	78.00	14.75	1986.50	91.25	73.50	19.45	1933.23
	SCRN	95.00	**87.00**	**8.42**	**1986.98**	**96.00**	**91.50**	**4.69**	**2099.91**	95.50	**89.25**	**6.54**	**2043.44**

Summary of Results

- Human -> AI attacks are harder than AI -> Human attacks
- SCRN is able to improve robustness against perturbations in at least 4 different real world settings
  - Drop Off in accuracy with non-perturbed data (OA)

Limitations

- All experiments done in english, did not explore multilingual corpora
- The paraphrasing attack was not considered as text perturbations and was not tested in the experiments

Paper 2: DIPPER

Introduction

- Robustness of detection algorithms for paraphrased AI-generated text is unclear
- Using DIPPER, can improve detection techniques for paraphrased text
- Paraphrasers must be external (if used by the base LLM, still susceptible to watermarking)

What Does Dipper Do?

- ***Di***scourse ***P***ara***p***hras***er*** (DIPPER) utilizes two techniques to evade detection
- Feature 1: Paraphrasing text in context
  - Paraphrases paragraph-length text (not sentence-length as many LLM's do)
  - Reorders content
  - Can use the user prompt
- Feature 2: Controlling Output Diversity
  - Existing paraphrasers lack output diversity
  - Provides control over lexical diversity and content reordering for the output

DIPPER (Visual Explanation)

- DIPPER (11B model) paraphrases AI-generated text by replacing watermarked tokens with semantically-equivalent benign tokens, undetectable by conventional watermark detectors

Why Does this Matter?

- Highlight the existing vulnerability of AI-content detectors to paraphrasing
- Prevent plagiarism
- Open-Source Contribution to the Research Community
  - The authors published all their code and work
  - They hope others will build off of their work and make more robust models

Retrieval Methods

- LLM API's save every output generated in a database
- When candidate text is given, it will compare the **semantic** representation to the output stored in the database
- Information Retrieval (IR) evaluates based on keyword matching and frequency
- Detection results
  - 97.3% of PG19 paraphrases
  - 80.4% of Wikipedia paraphrases
- Important to note: it is NOT comparing exact words and watermarking, just the meaning of the sentences and words themselves

Background on AI Methods

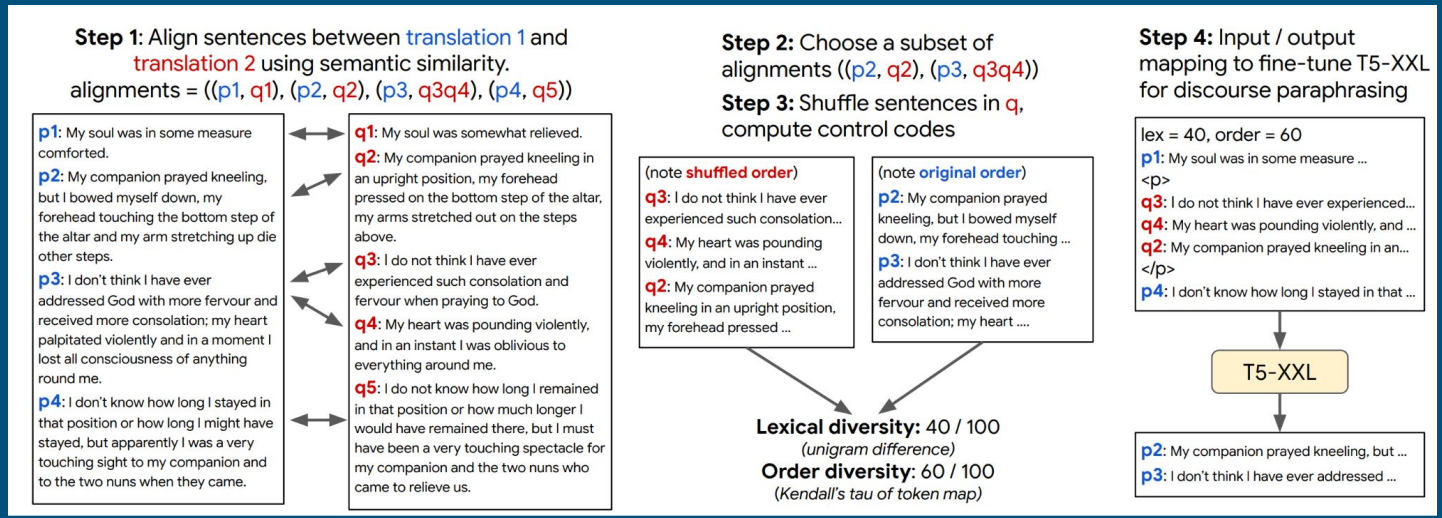
- Watermarking
  - Can be detected post-hoc
  - Imperceptible to human readers, has little effect on text quality, and hard to remove
- Statistical Outlier
  - Early methods: detect irregularities in entropy and perplexity
  - ChatGPT release inspired creation of closed-source GPTZero and DetectGPT (DetectGPT acknowledges AI text has higher LLM likelihood than meaningful perturbations)
- Classifier
  - Distinguishes human-written text and AI-generated text
  - OpenAI created a GPT model as a web interface
- Paraphrasing bypasses all of these techniques through altering statistical properties

Building Paraphraser Attacker

- Because traditional statistical properties will not bypass detection, context will be used for the attack
- Controllable context ability
- Paraphraser must be different from the watermarked model
- Utilizes translations of paragraphs in non-English novels and English novels and treats them as paraphrases
- At the paragraph level, so has ability to have external context and structural reordering

Building Paraphraser Attacker

- Step 1: Align Sentences
- Step 2: Choose sentence subset
- Step 3: Re-order
- Step 4: Map



Experiments Attacking with DIPPER

- Three evaluation metrics are of paramount importance
- Detection accuracy
  - True-positive rate
  - False-positive rate (fixed to 1%)
- Semantic similarity
  - Importance because if paraphrasing is effective, it will have the same meaning
  - Semantic similarity evaluated using P-SP from Wieting et al.
    - Robust against topically similar non-paraphrases
    - Using random paragraphs from same book, score is 0.09
    - Average human paraphrasing score is 0.76 (semantics preserved if it beats this)

Models and Datasets

- Base LMs
  - GPT2-XL (1.5B), OPT-13B, and text-davinci-003 from GPT-3.5 (175B)
  - 300 tokens long before passing to dipper
- Two types of generations tasks
  - Open-ended generation (LM generates continuation of two-sentence prompt)
  - Long-form question answering (LM answers question with 300-word answer)
  - Human-written text kept in testing set

Detection Algorithms and Process

- Detection algorithms
  - Watermarking
  - DetectGPT
  - GPTZero
  - OpenAI's text classifier
  - RankGenXL-all
- Paraphrasing AI-generated text
  - Pass prompts and responses for each task through DIPPER
  - Inputs are lexical and order controls
  - Truncate so all have same number of words (human, ai-generated, and paraphrased)
  - To preserve semantics, paraphrase three sentences at a time and only pass through once (to demonstrate effectiveness)

Experiments Attacking with DIPPER (Open)

Metric →	Sim ↑	Detection Accuracy ↓				
Detector →		Watermarks	DetectGPT	OpenAI	GPTZero	RankGen
GPT2-1.5B	-	100.0	70.3	21.6	13.9	**13.5**
+ DIPPER 20L	99.2	97.1	28.7	19.2	9.1	15.8
+ DIPPER 40L	98.4	85.8	15.4	17.8	7.3	18.0
+ DIPPER 60L	96.9	68.9	8.7	**13.3**	7.1	19.8
+ DIPPER 60L, 60O	94.3	**57.2**	**4.6**	14.8	**1.2**	28.5
OPT-13B	-	99.9	14.3	11.3	8.7	**3.2**
+ DIPPER 20L	99.1	96.2	3.3	11.8	5.4	5.2
+ DIPPER 40L	98.6	84.8	1.2	11.6	3.8	6.6
+ DIPPER 60L	97.1	63.7	0.8	**9.1**	6.3	9.3
+ DIPPER 60L, 60O	94.6	**52.8**	**0.3**	10.0	**1.0**	13.5
GPT-3.5-175B, davinci-003	-	-	26.5*	30.0	7.1	**1.2**
+ DIPPER 20L	97.6	-	12.5*	20.6	4.3	1.7
+ DIPPER 40L	96.7	-	8.0*	22.4	4.8	2.0
+ DIPPER 60L	94.2	-	7.0*	**15.6**	6.1	3.9
+ DIPPER 60L, 60O	88.4	-	**4.5***	**15.6**	**1.8**	7.3
Human Text	-	1.0	1.0	1.0	1.0	1.0

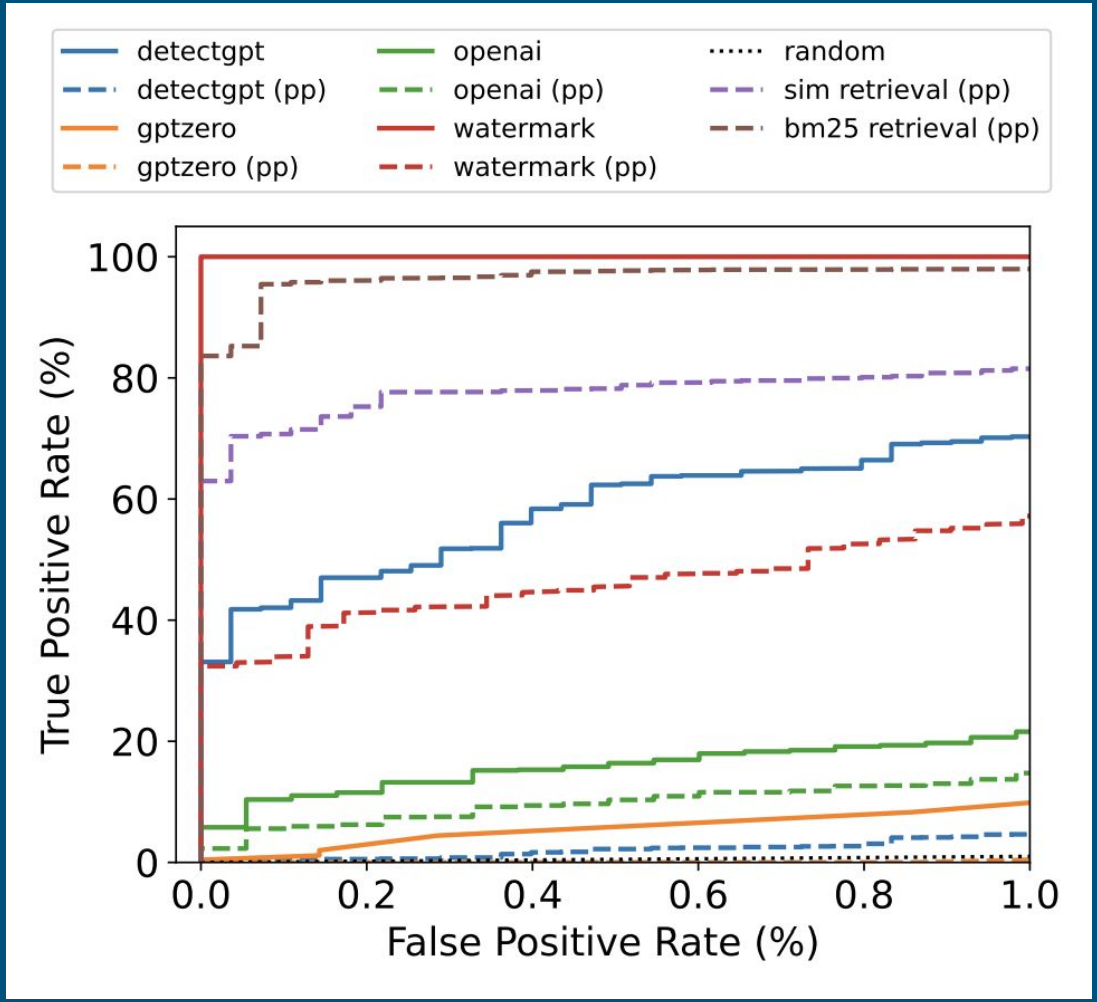
Results (Long-Form)

- Paraphrasing preserves semantic accuracy while significantly lowering detection rate
- Non-watermarking detectors generally ineffective
- ROC plots confirm trends at 1% false-positive rates

Metric →	Sim ↑	Detection Accuracy ↓		
		W.M.	D.GPT	O.AI
GPT2-1.5B	-	100.0	74.9	59.2
+ DIPPER 20L	99.5	98.9	45.7	35.3
+ DIPPER 40L	99.0	90.7	28.0	34.4
+ DIPPER 60L	97.5	71.1	15.8	**31.3**
+ 60L, 60O	96.2	**55.8**	**7.6**	32.7
OPT-13B	-	100.0	29.8	33.5
+ DIPPER 20L	99.6	98.3	15.0	24.5
+ DIPPER 40L	99.4	87.3	6.4	24.1
+ DIPPER 60L	96.5	65.5	3.2	**21.6**
+ 60L, 60O	92.9	**51.4**	**1.5**	**21.6**
GPT-3.5-175B davinci-003	-	-	67.0*	40.5
+ DIPPER 20L	99.9	-	54.0*	43.1
+ DIPPER 40L	99.8	-	36.0*	43.1
+ DIPPER 60L	99.5	-	23.0*	40.1
+ 60L, 60O	98.3	-	**14.0***	**38.1**
Human Text	-	1.0	1.0	1.0

ROC Curves

- Different detectors in varying colors
- Before paraphrasing solid
- After paraphrasing dashed



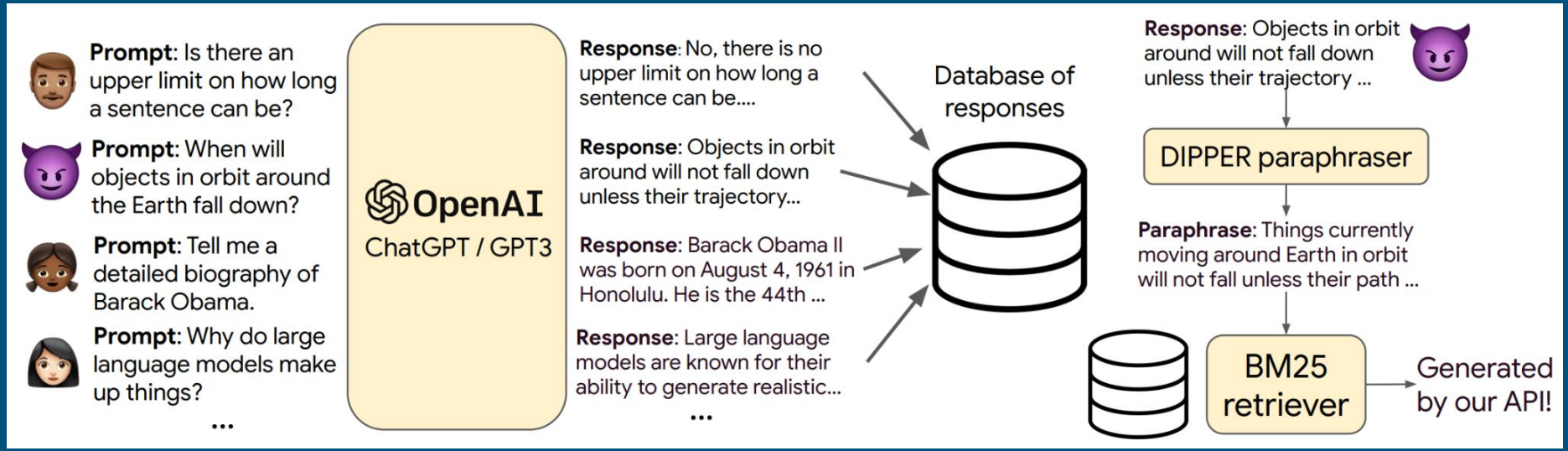
Alternative Paraphrasing Attacks

- Results displayed were after one paraphrasing iteration, to improve effectiveness can do so multiple times
- Use alternative paraphrasers to DIPPER which may prove more effective
- Use LLM’s to paraphrase certain areas
    - While this may be effective, it could also be prone to watermarking detection

Retrieval Defense Overview

- As previously discussed, LLM API's store generated text and prompts in a database
- Users can enter AI text as a query, then the interface searches to see if a sequence is semantically similar to the input
- Utilizes a semantic similarity scorer (e.g. P-SP or BM25)

Retrieval Defense Overview



Formulating Retrieval Defense

- Building the database
  - $x_1,...,x_N$ are set of prompts fed into API
  - $y_i = f_{LM}(x_i)$ as LLM output
  - $\mathbf{Y} = [\mathbf{y}_1,...,\mathbf{y}_N]$ is the constructed database through encoding all API outputs retrieval encoder $\mathbf{y}_i = f_{ret}(y_i)$
  - Database is dynamically updated and inaccessible except through the API

- Retrieving the database
  - y' is candidate text
  - $\mathbf{y}' = f_{ret}(y')$ is encoded vector
  - For a the interface client to know if y' was generated by the API f_{LM}, find the maximum similarity score:

$$\text{output} = \text{score} > T, \text{ where score} = \max_{i \in \{1,..N\}} \frac{\mathbf{y}' \cdot \mathbf{y}_i}{|\mathbf{y}'| \, |\mathbf{y}_i|}$$

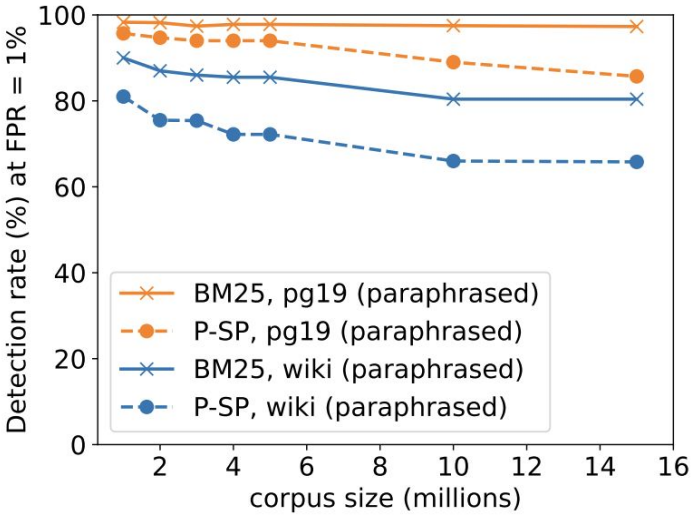
  - Non-paraphrased text will result in 1.0
  - Increasing T increases detection of paraphrased content, but also increases false-positive rate

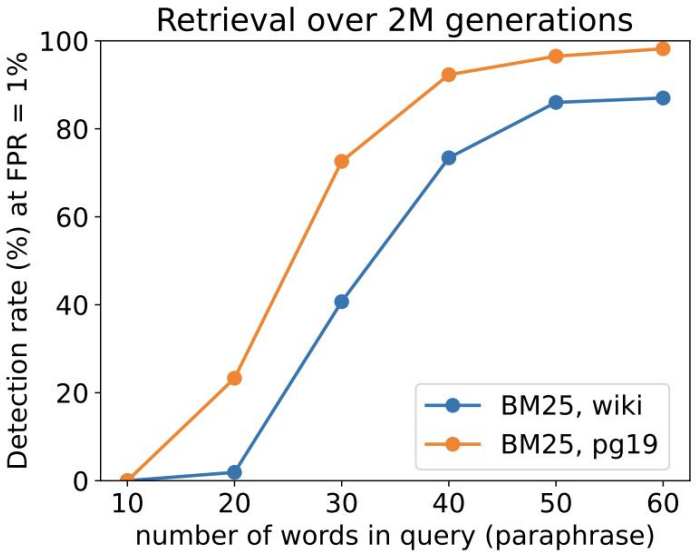
Retrieval versus Other Detectors

	GPT2-XL			OPT-13B			GPT-3.5 (davinci-003)		
	Original	+ 60L	+ 60 L,O	Original	+ 60L	+ 60 L,O	Original	+ 60L	+ 60 L,O
Watermark [2023a]	100.0	71.1	55.8	100.0	65.5	51.4	-	-	-
DetectGPT [2023]	74.9	15.8	7.6	29.8	3.2	1.5	1.0	0.0	0.0
OpenAI [2023a]	59.2	31.3	32.7	33.5	21.6	21.6	40.5	40.1	38.1
(Ours) Retrieval over corpus of 3K generations from model itself, with retriever:									
SP	100.0	95.6	87.7	100.0	94.8	85.3	100.0	94.2	85.1
BM25	100.0	99.2	97.8	100.0	99.3	97.3	100.0	98.6	96.2
(Ours) Retrieval over corpus of 9K generations pooled from all three models, with retriever:									
SP	100.0	88.9	75.4	100.0	89.6	76.4	100.0	93.8	84.6
BM25	100.0	98.3	95.2	100.0	98.5	94.4	100.0	98.5	96.0
(Ours) Retrieval over 43K ShareGPT responses + corpus of 3K generations from model itself, with retriever:									
SP	100.0	94.0	84.8	100.0	94.2	84.7	100.0	94.1	84.9
BM25	100.0	98.9	97.5	100.0	99.0	97.3	100.0	98.4	95.5

Retrieval versus Large Retrieval Corpus

- Retrieval is effective with 15M generation corpus size (left)
- Performs best with minimum 50 token query (right)





Retrieval: Scalability

- Store space requirements
  - Major LLMs have complex storage infrastructure
  - Only 5TB (compared to Google Search Index, 100,000TB)
- Computational requirements
  - 14-Core GPU, took 1s per retrieval (15M)
  - Extrapolating to 2B would take 130 s/retrieval
  - Fully parallelizable and likely would use a better GPU than a Macbook's
- Large database accuracy
  - Expensive to create from scratch, thus must use publicly available databases
  - Using 1B would be more effective, but hard to access (could use an LLM's private database)

Retrieval: Limitations

- API-Specific
  - Must know the applicable API (if DeepSeek used instead of OpenAI, OpenAI's API will proclaim its not paraphrased)
- Closed-Source LLMs
  - Open-source LLMs do not store generated outputs in a database like closed-source LLMs do
  - Watermarking also has a similar limitation
- Retrieval infrastructure
  - With an estimate of 2B entries per database every year, optimization must be applied
- Privacy Concerns
  - Potential risk of *all* user data being leaked

Retrieval: Limitations

- Data Memorization
  - Can result in false-positives, originally written by humans but then classified as AI-generated
  - Suggestion: API providers retrieve over the model's training set
- Large Databases
  - Causes a decrease in accuracy, but overall is rather minor (1% when scaling PG19 1 to 15M)
- Iterative Attacks (Access to Detectors)
- Lack of threshold, *T*, guarantee
- Short outputs

Paper 3: OUTFOX

Paper Overview

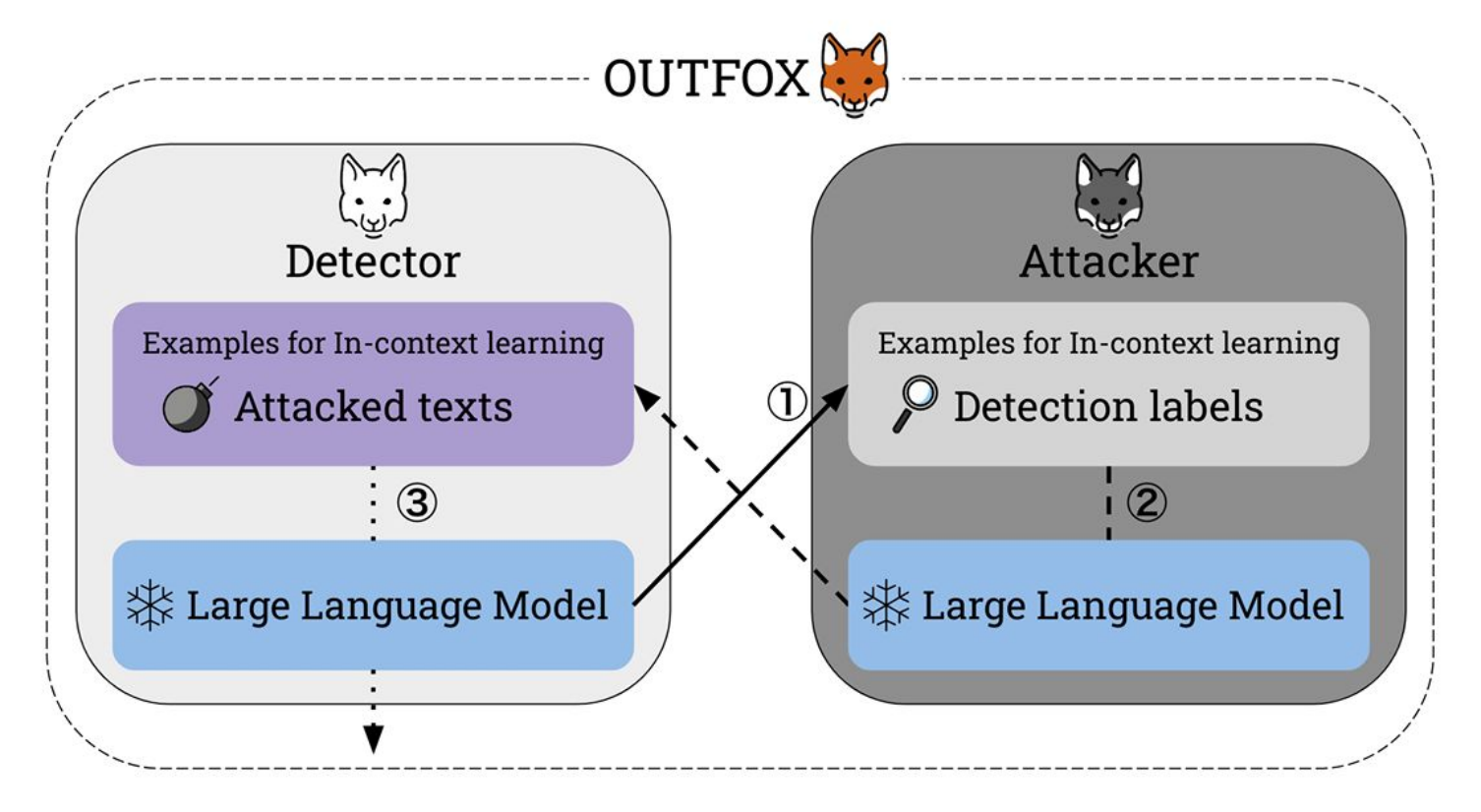
- Malicious users might attempt to deliberately evade the detectors based on detection results.
- Previous studies did not operate based on the assumption above.
- OUTFOX improves the robustness of LLM-generated-text detectors by allowing both the detector and the attacker to consider each other’s output.

Motivation

- Growing concerns about the potential misuse of LLMs, like in plagiarizing AI Generated text (AIGT).
- Existing AIGT detectors perform poorly against simple attacks like paraphrasing.
- This raises the risk that malicious users might exploit LLMs to create texts specifically designed to evade detection.

Methodology

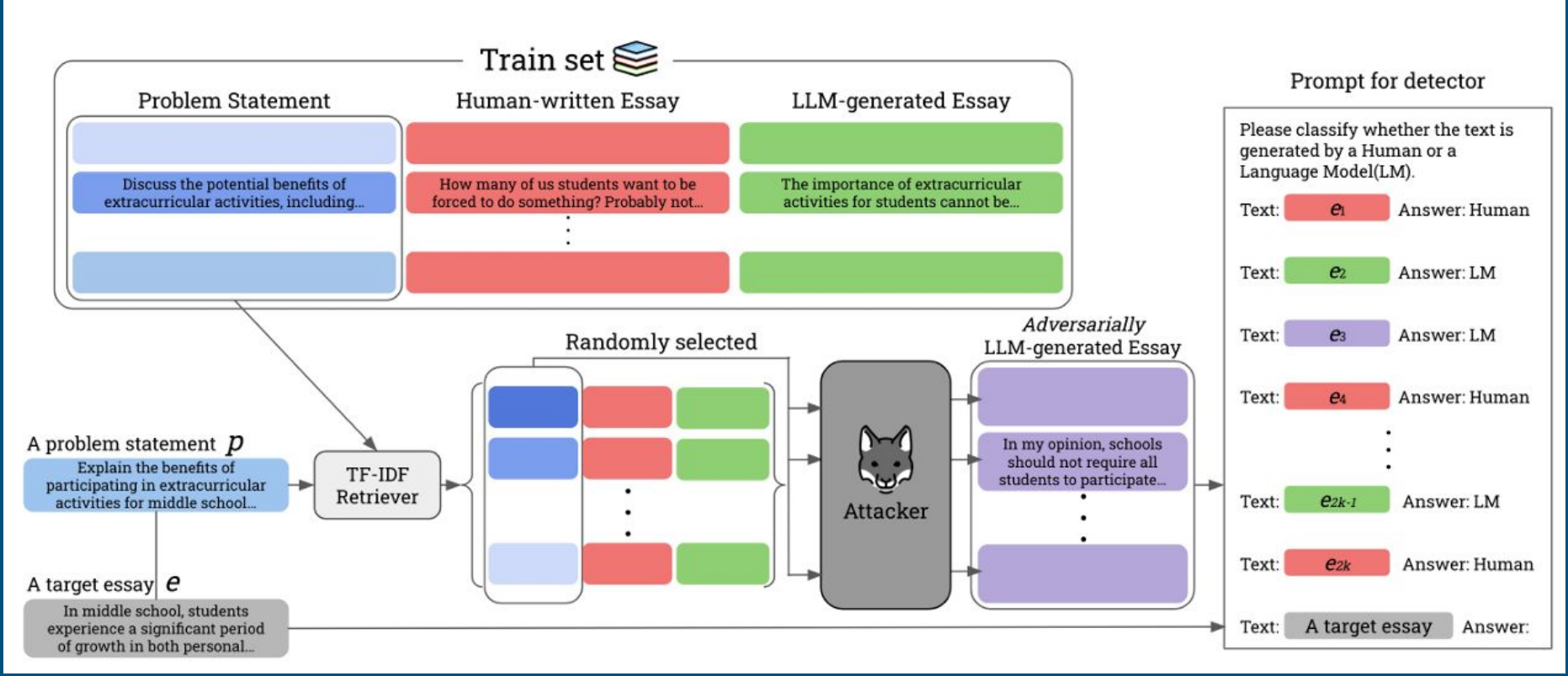
- OUTFOX Framework Overview:
  - Collaboration between a detector (identifying AI-generated essays) and an attacker (creating adversarial examples to bypass detection).
  - In-context learning to improve detection capabilities.
- Key innovation: The adversarial generation process makes the detector more robust and adaptable.



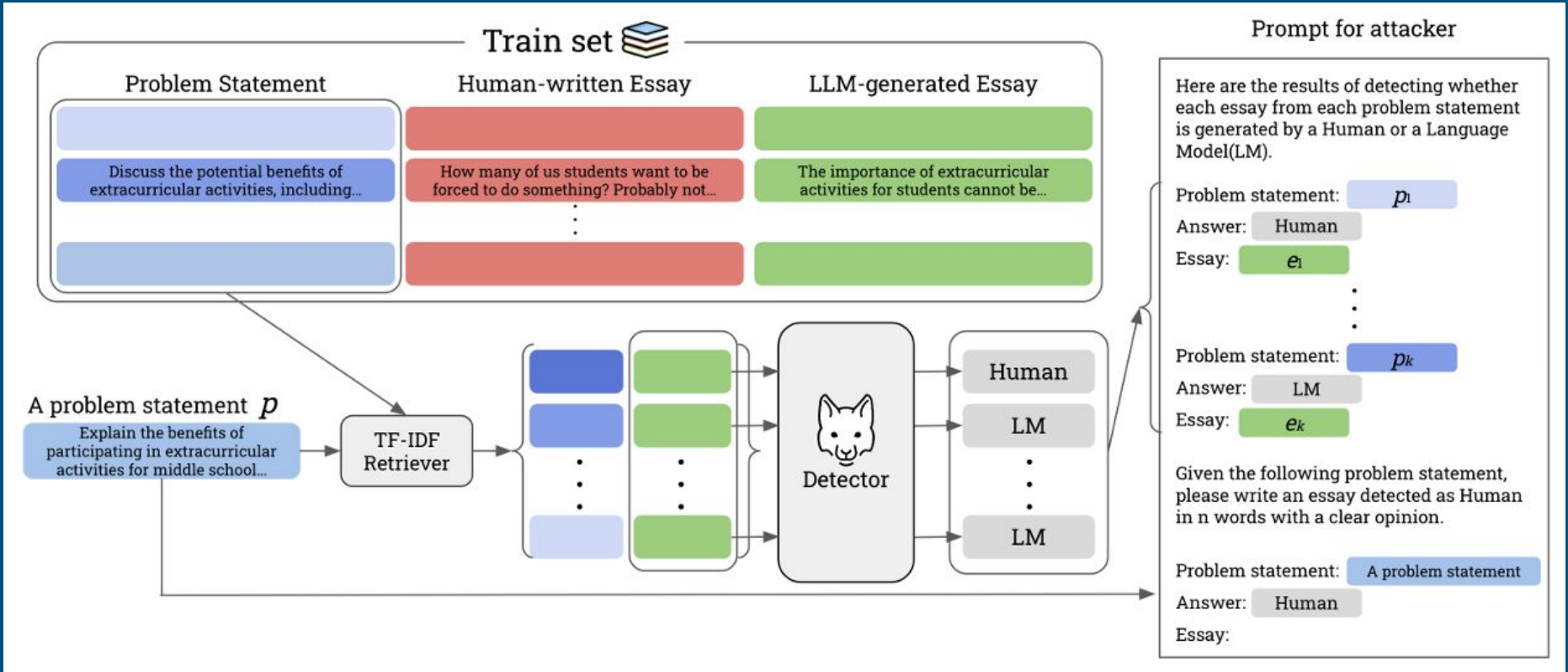
The authors propose OUTFOX, a novel framework designed to enhance the robustness and applicability of LLM-generated text detectors.

Constructing a Dataset to Detect LLM-Generated Essays

- Base Dataset: Argumentative essays from Maggie et al. (2022), written by 6th–12th grade U.S. students.
- Dataset Creation Process:
  - Generated pseudo-problem-statements using ChatGPT.
  - Instruction-tuned LLMs crafted essays based on these statements.
- Dataset Composition:
  - 15,400 triplets of essay problem statements, student-written essays, and LLM-generated essays.
  - Split: 14,400 (training), 500 (validation), 500 (test).
- Includes 500 adversarially attacked essays for evaluation.



OUTFOX detector: The detector utilizes the adversarially generated essays as examples for in-context learning to learn to detect essays from our OUTFOX attacker.



OUTFOX attacker: The attacker considers our OUTFOX detectors prediction labels as examples for in-context learning and adversarially generates essays that are harder to detect.

“Although the framework theoretically allows the detector and attacker to iteratively strengthen each other many times, we focus on only once.”

Attacker	Detector	Metrics (%) ↑			
		HumanRec	MachineRec	AvgRec	F1
DIPPER	w/o Attacks	98.6	66.2	82.4	79.0
	w/ DIPPER	98.2	79.6	**88.9**	**87.8**
	w/ OUTFOX	97.8	72.4	85.1	82.9
OUTFOX	w/o Attacks	98.8	24.8	61.8	39.4
	w/ DIPPER	98.6	20.8	59.7	34.0
	w/ OUTFOX	97.2	69.6	**83.4**	**80.7**

Comparison of the detection performances of our OUTFOX detector on attacked essays, with and without considering attacks.

Essay Generator	Detector	Metrics (%) ↑			
		HumanRec	MachineRec	AvgRec	F1
ChatGPT	w/o Attacks	99.0	94.0	**96.5**	**96.4**
	w/ DIPPER	99.2	87.8	93.5	93.1
	w/ OUTFOX	97.8	92.4	95.1	95.0
GPT-3.5	w/o Attacks	98.6	95.2	**96.9**	96.8
	w/ DIPPER	98.8	92.4	95.6	95.5
	w/ OUTFOX	97.6	96.2	**96.9**	**96.9**
FLAN-T5-XXL	w/o Attacks	98.8	68.2	83.5	80.5
	w/ DIPPER	99.2	72.0	**85.6**	**83.3**
	w/ OUTFOX	97.0	73.4	85.2	83.2

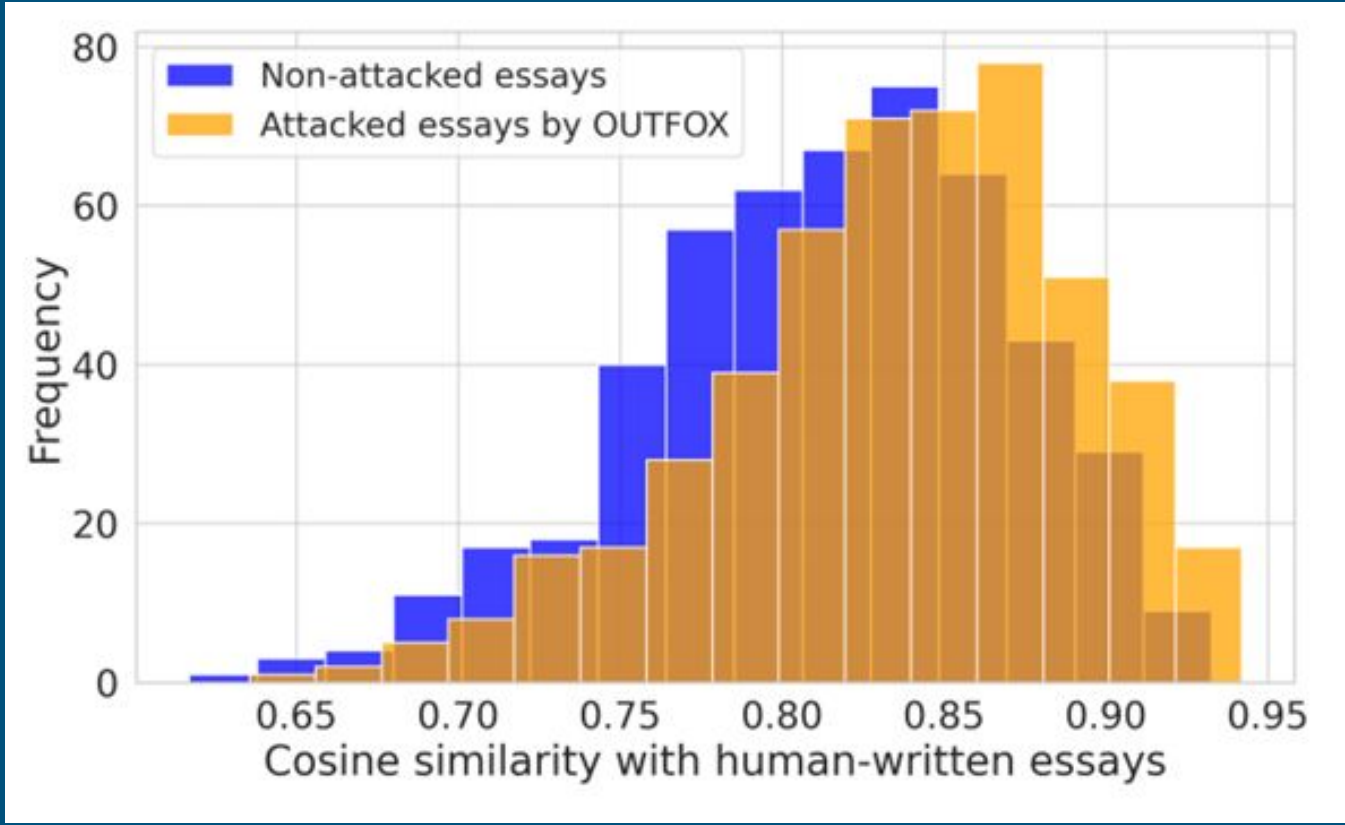
Comparison of the detection performances of the OUTFOX detector on non-attacked essays, with and without considering the attacks.

Detector	Attacker	Metrics (%) ↓			
		HumanRec	MachineRec	AvgRec	F1
RoBERTa-base	Non-attacked	93.8	92.2	93.0	92.9
	DIPPER	93.8	89.2	91.5	91.3
	OUTFOX	93.8	69.2	**81.5**	**78.9**
RoBERTa-large	Non-attacked	91.6	90.0	90.8	90.7
	DIPPER	91.6	97.0	94.3	94.4
	OUTFOX	91.6	56.2	**73.9**	**68.3**
HC3 detector	Non-attacked	79.2	70.6	74.9	73.8
	DIPPER	79.2	3.4	41.3	5.5
	OUTFOX	79.2	0.4	**39.8**	**0.7**
OUTFOX	Non-attacked	99.0	94.0	96.5	96.4
	DIPPER	98.6	66.2	82.4	79.0
	OUTFOX	98.8	24.8	**61.8**	**39.4**

Comparison of the detection performance of the detectors on ChatGPT-generated essays, before and after being attacked by DIPPER and OUTFOX.

Baseline type	Essay Generator	Detector	Metrics (%) ↑			
			HumanRec	MachineRec	AvgRec	F1
Statistical outlier methods	FLAN-T5-XXL	log $p(x)$	2.0	97.6	49.8	66.0
		Rank	28.8	86.2	57.5	67.0
		LogRank	12.0	90.6	51.3	65.0
		Entropy	39.4	80.4	59.9	66.7
		DetectGPT	29.8	76.2	53.0	61.9
		OUTFOX	97.0	73.4	**85.2**	**83.2**
Supervised classifiers	ChatGPT	RoBERTa-base	93.8	92.2	93.0	92.9
		RoBERTa-large	91.6	90.0	90.8	90.7
		HC3 detector	79.2	70.6	74.9	73.8
		OUTFOX	97.8	92.4	**95.1**	**95.0**
	GPT-3.5	RoBERTa-base	93.8	92.0	92.9	92.8
		RoBERTa-large	92.6	92.0	92.3	92.3
		HC3 detector	79.2	85.0	82.1	82.6
		OUTFOX	97.6	96.2	**96.9**	**96.9**

Comparison of the detection performances of the OUTFOX detector and prior approaches on non-attacked essays.



Cosine similarity distributions of non-attacked essays and the OUTFOX attacker-generated essays with human-written essays, respectively.

Conclusion

- OUTFOX Framework: Improves detector robustness against attacks through in-context learning.
- Key Findings:
  - Detector effectively learns to identify adversarial essays.
  - Minimal negative impact on detecting non-attacked texts.
  - Adversarial examples outperform previous methods in evading detection.
- Insights: Attacker-generated essays are semantically closer to human-written essays, enhancing attack success.
- Future Directions: Expand the framework to domains like fake news detection and academic paper analysis.

References

https://proceedings.neurips.cc/paper_files/paper/2023/file/575c450013d0e99e4b0ecf82bd1afaa4-Paper-Conference.pdf

https://arxiv.org/abs/2406.01179

OUTFOX: LLM-Generated Essay Detection Through In-Context Learning with Adversarially Generated Examples | Proceedings of the AAAI Conference on Artificial Intelligence