The Pennsylvania State University The J. Jeffrey and Ann Marie Fox Graduate School

#### TEXT AUTHORSHIP IN THE AGE OF LARGE LANGUAGE MODELS

A Dissertation in Information Sciences and Technology by Saranya Venkatraman

 $\ensuremath{\textcircled{}}$  2024 Saranya Venkatraman

Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

December 2024

The dissertation of Saranya Venkatraman was reviewed and approved by the following:

Dongwon Lee Professor, College of Information Sciences and Technology Dissertation Advisor Chair of Committee

Ting-Hao (Kenneth) Huang Associate Professor, College of Information Sciences and Technology

Amulya Yadav Associate Professor, College of Information Sciences and Technology

Rui Zhang Assistant Professor, Department of Computer Science and Engineering

Carleen Maitland Professor, College of Information Sciences and Technology Program Head

## Abstract

Tremendous progress in text generation by Large Language Models (LLMs) has led to an exponential rise in both the quality and quantity of LLM-generated texts. We are now surrounded by texts that are written entirely or enhanced and edited by autoregressive models. These texts appear in many contexts, ranging from dialog turns in an interactive session with ChatGPT to academic articles summarized by an LLM, a news article generated entirely by a model on social media, and so on. The ubiquity and high quality of such texts have made tracking and detecting their presence a task of growing and urgent importance. Particularly, there are developing concerns about copyright infringement, privacy, malicious use, intellectual property (IP) rights, and academic integrity that require active efforts to identify and trace LLM texts.

In this thesis, we study machine-generated texts through four authorship-related tasks: (1) Human v/s Machine-generated Text evaluation: We first study if machines have human-text-like traits as measured by psycholinguistics-based measures. To do this, we turn to the Uniform Information Density (UID) principle that states that humans tend to distribute information or surprisal evenly or smoothly in language production. We analyze if machine-generated texts follow similar surprisal patterns and find that the answer depends on the decoding strategy used, with some settings generating more "human-like" surprisal distributions than others. But overall, we find that machines distribute surprisal differently than humans. Building upon this, we move to the next task, (2) Machine-generated text detection and Authorship Attribution: We develop "GPT-who", an authorship attributor that uses surprisal-based features to identify if a text is human-written or machine-generated, and also predicts the exact author LLM. We then study the reverse problem of (3) Authorship Obfuscation where the goal is to obfuscate or hide an author's identity by preserving semantics but altering the writing style such that it cannot be traced back to the original author. To do this, we present "ALISON", an obfuscation method that perturbs individual authors' syntactic patterns.

Beyond single-authored texts, we explore (4) multi-LLM collaborative text generation by creating "CollabStory", a benchmark dataset containing over 35k creative stories generated jointly by up to 5 state-of-the-art open-source LLMs. We do this to study how authorship-related tasks evolve when multiple authors are present in a text, in light of unifying frameworks such as vLLM and LangChain that have enabled this oncoming scenario. Through these novel methods and datasets, this dissertation advances our understanding of authorship in the evolving landscape of LLM-generated texts and provides practical tools for addressing emerging challenges.

## **Table of Contents**

List of	Figures	vii
List of	Tables	х
Acknow	Acknowledgments	
Chapte	er 1	
$\mathbf{Intr}$	oduction	1
1.1	Motivation for studying LLM authorhship	2
1.2	Background	7
1.3	RQ 1 & 2: Machine-Generated Text Evaluation	11
1.4	RQ 3: Authorship Attribution	12
1.5	RQ 4: Authorship Obfuscation	12
1.6	RQ 5: Collaborative Generation	13
Chapte	er 2	
Mao	chine-Generated Text Evaluation: Are machine-generated texts	
	human-like?	14
2.1	Introduction	14
2.2	Related Work	15
	2.2.1 Language production in humans	15
	2.2.2 Response generation in machines	15
	2.2.3 The UID principle and decoding algorithms.	16
2.3	Experimental Details	18
	2.3.1 Model & dataset	18
	2.3.2 Uniform Information Density score	19
	2.3.3 Information density of decoding algorithms	20
	2.3.4 Response evaluation	20
2.4	Results	22
	2.4.1 Information density of model responses	22
	2.4.2 UID score & automatic metrics	25
	2.4.3 UID score & human Judgments	27
2.5	Discussion	31

#### Chapter 3

Aut	horshi	p Attribution using Information Density-based features 3	3
3.1	Introd	uction	33
3.2	Relate	d Work	35
	3.2.1	Uniform Information Density (UID)	35
	3.2.2	Machine-Generated Text Detection	36
3.3	Metho	d: GPT-who $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	37
	3.3.1	UID-based features	39
	3.3.2	Baselines & Detectors	13
3.4	Result	s	4
	3.4.1	UID Signatures of Authors	4
	3.4.2	Machine Text Detection Performance	4
	3.4.3	Running Time	19
3.5	Conch	usion	50
3.6	Discus	sion	<i>5</i> 0
Chapte	er 4		
Aut	horshi	p Obfuscation using syntactic n-gram patterns 5	4
4.1	Introd	uction $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	<b>5</b> 4
	4.1.1	White-Box Attacks	68
	4.1.2	Black-Box Attacks	68
4.2	Propo	sed Method: ALISON	31
	4.2.1	Problem Formulation	31
	4.2.2	One-Time Stylistic Internal AA Classifier Training	31
	4.2.3	Model Architecture	;3
	4.2.4	Internal Classifier Training	;3
	4.2.5	Replacement Phrase Generation via Masked PLM 6	;4
	4.2.6	Text Obfuscation Process: One N-Gram at a Time	;5
4.3	Exper	imental Setup $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$	55
	4.3.1	Datasets	56
	4.3.2	Evaluation Metrics	37
4.4	Result	s	74
	4.4.1	Obfuscation Success	74
	4.4.2	Ablation of Interpretability-Based Replacement	'5
	4.4.3	Computational Complexity	'5
	4.4.4	Semantic Preservation	7
	4.4.5	Overall Text Quality	7
	4.4.6	Efficiency: Time Complexity	34
	4.4.7	Qualitative Analysis	35
4.5	Discus	$\sin \alpha$	37

#### Chapter 5

Col	laborative Generation: <b>CollabStory</b> a multi-LLM story benchmark	88
5.1	Introduction	89
5.2	Related Work	91
5.3	Methodology	92
	5.3.1 CollabStory: Dataset Creation	92
	5.3.2 LLM prompting	94
	5.3.3 Dataset Cleaning	95
	5.3.4 Post-processing and filtering	98
5.4	Dataset Analysis	99
	5.4.1 Story Continuity	102
5.5	Authorship Analysis: Extending PAN tasks for multi-LLM scenario	104
	5.5.1 Task 1: Is a story written by multiple authors or not?	104
	5.5.2 Task 2: How many authors have written a story?	105
	5.5.3 Task 3: Authorship Verification	105
	5.5.4 Task 4: Authorship Attribution	106
5.6	Conclusion	108
5.7	Discussion	110
Chapte	er 6	
Con	nclusion	116
6.1	Human and machine-generated texts have differences in surprisal distri-	-
	butions	116
6.2	Surprisal as a means to detect machine-generated text	117
6.3	Syntactic patterns can help obfuscate authorship	118
6.4	Collaborative writing through LLM prompting	119
6.5	Open Challenges	120
	6.5.1 Datasets that showcase LLM abilities	120
	6.5.2 Evaluation Measures and Human Alignment	120
	6.5.3 Detection and Obfuscation of multi-lingual texts	121
	6.5.4 Need for interpretable methods	121
6.6	Ethical Considerations	121
Biblio	graphy	122
-12108	5- ~r	

### List of Figures

- 2.1 Information density distribution of human and GPT-2 generated responses for the Persona-Chat dataset. A smaller width of the distribution or more values closer to 0 implies more uniform responses, and a wider spread (such as for the human response) indicates more non-uniform responses.
  15
- 2.2 Histogram of **UID Scores** of responses generated using different decoding algorithms. The farther the UID score from 0, the less uniform or more non-uniform the response. Human-generated reference text (left-top) has a higher frequency of non-uniform responses as compared to any model setting as can be seen from the wider spread of scores away from 0. Also, as the values of p and k increase (left to right), the information density distribution slowly approaches reference text-like non-uniformity. . . . . 16

## 2.3 Surprisal at every token in candidate responses to the same dialog history, color-coded with human-annotated **interesting** scores. Plots (*left to right*) are arranged in increasing order of uniformity (i.e. variance along y-axis). Less uniform the surprisal (left-most), better the score. . . . . 16

2.4	Our dataset contains 4 candidate responses for every dialog history, along	
	with human annotations for 3 qualitative measures	21

2.5	Screenshots of our MTurk study interface for collecting human judgments on 4 candidate responses per dialogue history, along 3 quality measures	22
2.6	Instructions and examples from MTurk study	23
2.7	Frequency of responses (Yes/Somewhat/No) for each qualitative measure in our human annotated dataset.	26
20	Histomana of average contance currentical for reasoning generated using	

2.8 Histograms of **average sentence surprisal** for responses generated using different decoding settings and human-generated reference text (left-top). 27

3.1	GPT-who leverages psycholinguistically motivated representations that capture authors' information signatures distinctly, even when the corresponding text is indiscernible.	34
3.2	An example of UID span feature extraction that selects the most uniform and non-uniform segments from the token surprisal sequence. As can be seen in this example, two texts that read well can have very different underlying information density distributions in a given context. UID features capture these hidden statistical distinctions that are not apparent in their textual form.	38
3.3	<b>GPT-who</b> uses token probabilities of articles to extract UID-based features. A classifier then learns to map UID features to different authors, and identify the author of a new unseen article.	39
3.4	Distribution of UID Scores of 20 authors from the TuringBench dataset grouped (dotted line) by architecture type. LMs that share architectures tend to distribute UID scores similarly.	42
3.5	Human and machine-text pairwise comparisons: Comparing UID Score (Variance) distributions	45
4.1	An example of ALISON successfully obfuscating a text by changing its style while preserving semantics.	55
4.2	ALISON: Our proposed obfuscation pipeline.	58
4.3	An example of extracting POS trigrams.	62
4.4	Average time taken per obfuscation by different methods for increasing average length of articles.	76
4.5	Distribution of author-wise contributions to label entropy post-obfuscation.	79
4.6	Normalized Author Label Entropy on Blog for Avengers	80
4.7	Normalized Author Label Entropy on Blog for ${\sf ALISON}$	80
4.8	Effect of varying $L$ on obfuscation success and semantic preservation	81
4.9	Effect of varying $c$ on obfuscation success and semantic preservation	82

5.1	<b>CollabStory</b> contains over $32k$ creative stories written collaboratively by up to 5 LLMs. Each story segment is generated by a single author, that then passes the narrative baton to the next, completing the storyline part by part in a sequential manner.	89
5.2	N on the X-axis denotes the number of authors, and $N=1(H)$ and $N=1(M)$ correspond to the human-written and machine-generated single-authored texts, respectively. All other texts (N >=2) are multi-LLM generated. Y-axis shows the values of the measure shown in each subplot as mentioned in the headings. For all measures, we show the average and standard deviation for N going from 1 to 5. For all measures except vocabulary richness (3rd column, 1st row), increasing the number of authors (N) does not lead to statistically significant deviations from the human text distribution.	101
5.3	To evaluate the continuity of story parts, we sample two potential continu-	

## List of Tables

1.1	An overview of the authorship tasks, research questions and motivation for each task presented in this thesis.	11
2.1	Common decoding strategies in text generation using language models	17
2.2	Part A: Pearson's correlation coefficient $(r)$ between UID score and Length, BLEU, chrF, and METEOR metrics of dialog responses generated using different decoding settings.	18
2.3	Part B: Pearson's correlation coefficient $(r)$ between UID score and BertScore, BLEURT, RoBERTa, and SacreBLEU metrics of dialog responses generated using different decoding settings	18
2.4	Pearson's $r$ between <b>UID score and and human judgments</b> of qualita- tive measures for dialog responses bucketed by surprisal [Surprisal interval = the ranges of surprisal values used for bucketing responses, n = number of responses in each surprisal interval, *p-value < .05]	25
2.5	Pearson's correlation coefficient $(r)$ between UID score and average sentence surprisal (all $p < 0.01$ )	26
2.6	Pearson's correlation coefficient $(r)$ of <b>UID score and surprisal with</b> human judgments of qualitative metrics (* $p$ <0.01)	27
2.7	Pearson's $r$ between <b>surprisal and human judgments</b> of qualitative measures for dialog responses bucketed by surprisal [Surprisal interval = the ranges of surprisal values used for bucketing responses, n = number of responses in each surprisal interval, *p-value < .05]	28
2.8	Examples of dialogue histories followed by 4 response candidates arranged by increasing UID score i.e. from more non-uniform to uniform responses and their corresponding human judgment scores.	31

3.1	Max. & Min. UID spans ablation study: Setting a span length of N=20 to- kens maximized performance (F1 score) across large-scale datasets (N>30 leads to subsequently lower and eventually consistent performance). It can be seen that our min/max features tremendously impact performance against randomly sampled or no span features at all.	41
3.2	Test Set Performance (F1 Scores) for Task 3 across different domains	46
3.3	Test Set Performance (F1 score) for TuringBench dataset	47
3.4	Test Set Performance (F1 score) for TuringBench dataset	48
3.5	Test Set Performance (F1 score) for InTheWild dataset. ITW refers to the LongFormer-based detector trained by Li et al. (2023) specifically for this benchmark. Acronyms used in this table are as follows: ID = In- distribution, OD = Out-of-distribution, DS-MS = Domain-specific Model- specific, CD-MS = Cross-domains Model-specific, DS-CM = Domain- specific Cross-models, CD-CM = Cross-domains Cross-models, UM = Unseen Models, UD = Unseen Domains	48
3.6	Test Set Performance (F1 score) for ArguGPT dataset. <sup>*</sup> denotes results reported in [1].	49
3.7	Average Running time over 6 test beds from the InTheWild dataset	49
4.1	Obfuscation success across various models. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.	68
4.2	Obfuscation success across various models. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.	69
4.3	Obfuscation success across various models. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.	70

4.4	Obfuscation success across various models. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.	71
4.5	Statistics of the one-time training runtime and the average inference time per one sample for all methods.	76
4.6	Perplexity of post-obfuscation texts measured using LLaMA2-7B (lower is better).	77
4.7	Obfuscation Success across four machine text detection models. $\ldots$ .	78
4.8	Semantic Preservation across four machine text detection models	78
4.9	Sample sentences from the Blog Authorship Corpus pre- and post- obfus- cation	83
5.1	Comparison of CollabStory with other existing collaborative creative story datasets. Here, "M-M" $\rightarrow$ "Machine-Machine", "H-H" $\rightarrow$ "Human-Human", where "H" $\rightarrow$ "Human" and "M" $\rightarrow$ Machine. Ours is the largest dataset to present multi-LLM or machine-machine collaborative generation	90
5.2	Summary of Data Collection Statistics	92
5.3	Prompt templates for different parts of the story. {n} here denotes the number of target words for each author.	94
5.4	Example of a 5-part LLM story from CollabStory dataset for the prompt: "Years ago, you promised your firstborn to a witch. Since then, despite your best efforts, you can't seem to give him away. The witch is starting to get pretty mad."	98
5.5	Comparison of descriptive features for articles with number of authors $N \in [1, 5]$ , as compared with human-written single-author stories for the same prompts. Here, "H" $\rightarrow$ "Human"; "M" $\rightarrow$ Machine. * represents statistical significance (p<0.01) compared to human-written stories	99
5.6	Descriptive Statistics or Features for stories generated by different authors for different parts of the stories. Here, "K" represents the part of the story written, i.e. $K=1$ corresponds to the first part of the story, $K=2$ referees to the second part, and so on.	100

5.7	Continuity evaluation of pairs of story parts using GPT-40 using different negative sample sources (either from within the same story or not). This table presents the percentage of story parts for which either only the correct part was evaluated as having continuity, or the negative sample (incorrect) or both	102
5.8	Performance is shown as F1-scores. AVG denotes average F1-score for each $N$ . For each $N \ge 2$ we evaluate the classifiers on their ability to distinguish the stories from those written with $N = 1$ . Best performing method is in <b>bold</b> and second highest <u>underlined</u> for each N	105
5.9	All scores are F1-scores. AVG denotes average F1-score for each $N$ . This task is particularly challenging with only the single-authored stories $(N = 1)$ being correctly classified. For all multi-authored texts, BERT and RoBERTa perform better than others. Best performing method is in <b>bold</b> and second highest <u>underlined</u> for each N	106
5.10	All scores are F1-scores and AVG denotes average F-1 scores for each $N$ for the task of detecting authorship boundaries between sentence pairs. We see that generally, performance is slightly higher for $N = 2$ across all classifiers. Best performing method is in <b>bold</b> and second highest <u>underlined</u> for each N.	106
5.11	F1-scores for identifying the author of story parts across articles written by different numbers of authors. The 5 columns show each of the labels or authors. AVG denotes average F1-scores across all authors. Best performing method is in <b>bold</b> and second highest <u>underlined</u>	107
5.12	Example 1: As can be seen from these two consecutive story parts, the text written by Gemma is not necessarily different from that of other authors (in the second column). Thus, from our manual inspection, we were unable to attribute any noticeable features to the higher authorship attribution performance for Gemma.	109
5.13	Example 2: As can be seen from these two consecutive story parts, the text written by Gemma is not necessarily different from that of other authors (in the second column). Thus, from our manual inspection, we were unable to attribute any noticeable features to the higher authorship attribution performance for Gemma.	110

5.14	Example 3: As can be seen from these two consecutive story parts, the	
	text written by Gemma is not necessarily different from that of other	
	authors (in the second column). Thus, from our manual inspection, we	
	were unable to attribute any noticeable features to the higher authorship	
	attribution performance for Gemma.	11
5.15	Real-world implications of the tasks involved in understanding LLM-LLM	
	collaboration for writing tasks	12

## Acknowledgments

This dissertation was supported by the National Science Foundation (NSF) under Award No. #1820609, #1950491, and #2131144. I would also like to thank the McDonnell Foundation (Award No. #1822813) for supporting my work on the ClassInSight project. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the awarding agencies.

I am forever indebted to my advisor, Prof. Dongwon Lee, for taking me on at a very challenging time and inspiring me to pursue a new research direction. In a very short span, his encouraging mentorship style and genuine support have transformed the course of this dissertation and my academic experience. I am also sincerely thankful to my amazing dissertation committee members—Dr. Amulya Yadav, Dr. Kenneth Huang, and Dr. Rui Zhang—for their insightful input, encouragement, and invaluable contributions to this work.

I remain eternally grateful to my first academic advisor, Dr. David Reitter, who inspired me to pursue a career in research. He shaped the foundational ideas of this dissertation, and his continued mentorship and unwavering support has profoundly shaped me as a researcher and person. I am also indebted to Dr. He He for graciously hosting me at her awesome lab at NYU. It was a privilege to learn from her, and my time at her lab indelibly enriched my academic path. I am also grateful to all the collaborators, the ClassInSight team and my internship mentors at Cadence, Samsung Research and Google for their contributions to my research and beyond. Working with such amazing researchers has been a highlight and an invaluable experience.

I have had the unique privilege of learning from the best and kindest labmates at two labs: my fellow PIKErs- Adaku (who brought me to PIKE), Jooyoung, Jason, Michi, Wooyong, Nafis, Mahjabin, Aashish, Xianren and Delvin, as well as folks at NYU - Dan, Richard, Vishakh, Nitish and Nick. Interacting with you all every day at the lab has been the biggest source of joy. I feel fortunate to have been a part of such a unique and nurturing lab culture. I cannot overstate the role played by my chosen family, role models, and soul sisters, Agnese and Ana, who have been with me since my first day in the US and who inspire me with their courage and kindness. Words cannot express my gratitude to Teju, Adaku, and Sahiti, who always have my back through thick and thin, and I am honored to have their friendship. My time at Penn State was most enriched due to the wonderful friends I made here- Lee, Si, Kirsten, Pranav, Ruchi, Sharad, Adway, Meghna, Nishanti, Rohitha, Drew, Erick, Lanre, Evan, Zixin, Rich, Sam, Prasanna, Tiffany, Raphael, Niki, Lennora, Medhavi, Prateek, Sian and Stephanie. Each one of you have shown me a kindness and affection that have made my time in State College truly memorable. I feel fortunate to have Pauline in my life, for her relentless support and unlimited cookies that got me through the final years of grad school. Shreya and Aishu's lifelong friendship and constant support have been my main pillars of strength through everything. I would also like to thank my amazing friends from home who have shaped who I am and whom I miss dearly - Ruchi, Jigyasa, Abhishek, Aseem, Dipanshu, Ritwik, and Himanshu.

I am so thankful to my PA family- Eera, Enya, Arpita, and Anurag for taking me in and being my home away from home. I am grateful to my family, uncles, aunts, and the most loving sisters, Aarti, Preeti, and Aditi, for their unconditional support. My parents, Sita and Venkatraman, and my grandmother Jembagam, who raised me, are the driving force and reason for everything I have ever attempted to accomplish. Lastly, I would like to thank my partner, Shaurya, whose steadfast support and calming presence have empowered me to persevere without fear. His encouragement, patience, and humor have sustained me through the most difficult moments, and his presence has filled these past years with joy and inspiration. To all who have contributed to this work, I extend my deepest gratitude—this dissertation is a testament to your invaluable support and belief in me.

## Dedication

For my parents, Sita and Venkatraman, and my grandmother, Jembagam.

# Chapter 1 Introduction

Generative Large Language Models (LLMs) are rapidly transforming the landscape of real-world applications, seamlessly integrating into our daily lives and reshaping how we perceive and interact with textual data. From virtual assistants that anticipate our needs and recommend products on retail websites such as Amazon, to intelligent content creation tools that enhance productivity, LLMs are becoming indispensable in industries like healthcare, education, and entertainment. As they evolve, their ability to generate human-like text, solve complex problems, and provide personalized experiences are redefining the boundaries of what machines can achieve.

LLMs now generate content across a wide array of contexts, including social media posts, bot-generated responses on online platforms such as Reddit, academic publications, creative writing, and more [2–9]. The rapid advancements in both the quality and the capabilities of LLMs have resulted in a substantial proliferation of machine-generated texts across digital spaces. The volume of machine-generated texts on the internet has grown exponentially and brought with it the serious implication of how humans are often unable to detect such texts and perceive them to be written by humans or verified sources [10–13]. This surge has introduced novel challenges and concerns regarding the use, origin, and implications of such texts.

This growth in the capabilities, quality, and acceptance of such machine-generated texts in everyday life has been accompanied by an increasing set of challenges for the Natural Language Processing (NLP) research community. A particularly crucial aspect of these emerging challenges involves understanding authorship in the context of LLMs. In this context, authorship extends beyond traditional notions and encompasses the broader task of detecting, tracking, and critically analyzing the presence of machine-generated texts. The use of LLMs is not limited to generating standalone content but increasingly involves their participation in collaborative or assistive roles in writing tasks, where they function as writing assistants or even co-authors in human-machine interactions.

This raises questions about how to attribute authorship when LLMs contribute to text, whether as part of a machine-in-the-loop writing process or in cases where the text has been revised or paraphrased by different models. The complexity of these collaborative relationships between humans and machines introduces a need for more sophisticated methods to discern the role of LLMs in generating specific portions of text. Additionally, it necessitates a deeper understanding of how LLMs are being integrated into various writing tasks, including creative writing, academic work, and professional documentation, and how this impacts the overall quality and originality of the produced content. As LLMs become more embedded in routine writing workflows, the NLP community must develop frameworks that allow for the transparent and accurate assessment of machine contributions. These frameworks must not only address the technical aspects of detecting LLM-generated text but also engage with the broader ethical and practical implications of using LLMs as writing collaborators. This includes evaluating the effectiveness of LLMs in enhancing writing processes and understanding the potential risks, such as the introduction of bias, misinformation, fake news proliferation, threats to privacy, or loss of originality, which might arise from relying on these models [2-9, 14].

#### 1.1 Motivation for studying LLM authorhship

The challenges arising from this ever-increasing usage and presence of machine-generated texts are manifold, complex, and often entangled. One way of organizing such growing concerns around machine-generated texts is by defining and understanding different authorship-related tasks that each address different aspects of LLM authorship, which would, in turn, address the challenges presented so far. In this thesis, we study four such tasks that capture ongoing challenges brought upon by the LLM era as follows:

• Evaluation: As LLMs increasingly contribute to the volume of textual data being consumed at scale, it is imperative to examine whether such models enhance the quality of text output and facilitate improvements in terms of speed of writing, accuracy in terms of the overall objective of the writing task, and effectiveness. The task of evaluating machine-generated texts refers to the operationalization

and study of different measures and metrics of text quality [15–17]. Quality here refers to different types of measurable properties of text such as readability [18], coherence [19], lexical diversity [20], semantic quality [21], and grammatical correctness [22]. Most measures of such quality can be categorized as either human evaluation measures or automated metrics. Human evaluation measures require some data collection form human participants or experts rating or qualitatively assessing the text manually [23–25]. Automatic metrics constitute evaluation measures that can be calculated based on the text generated or being evaluated using some operationalizations such as lexical counts, perplexity using a trained language model, rule-based systems, and deep learning-based methods [26–28]. The overarching goal of all such metrics is to gauge the quality of the text as measured along different properties to assess the "goodness" of the text.

In the context of machine-generated texts, the evaluation is focused on measures of whether the machine-generated text is of similar quality as the human-written text. This evaluation is crucial to understand and evaluate if machine-generated texts are more and more human-like and if they contribute meaningfully to the task at hand. Thus, the choice of evaluation measures matters in informing the type of implications for any writing task. For example, for an academic writing assistant, measures of factual correctness, grounding, and logical fallacies might be more crucial to gauge task success. On the other hand, a creative story-writing task might benefit more from measures of creativity, plot twists, diversity, and depth of characters. Thus, the writing task and domain often inform the choice of relevant evaluating parameters. In the case of machine-generated texts and their comparison to human-authored texts, the human-written samples are treated as references or "ground-truth" examples that models should replicate. Hence, comparative studies of such evaluation measures for the two types of texts can help gain insights into the general extent to which machine-generated texts follow human performance in writing tasks and help quantify and compare LLM abilities.

Thus, evaluation not only has implications in understanding the general state of LLM-authorship but can also help inform directions for further enhancements by identifying aspects and task criteria that aren't satisfied. With an expansion of LLM tasks, the methods of evaluating texts must also evolve and reflect the complexity and sophistication of machine-generated texts. For example, current LLMs are

all grammatically accurate; hence, reporting and comparing grammatically might lead to the wrongful inference that LLM-generated texts are at par with humangenerated texts. This can only be counteracted by including updated measures to capture the traits that are still challenging for LLMs, such as factuality and coherence. Thus, as LLMs get better, so should the methods of evaluating their qualities and comparisons with human performance.

**Attribution**: This authorship-centered task broadly involves identifying a text's author or mapping any given text to its true writer. Attribution can be done in two main settings: the machine-text detection setting or the reverse turing test, where the author can be one of two authors (machines or humans), or the multi-class extension that entails predicting exactly which human or machine (LLM) authored a text. Attribution is a task of growing urgency to the the progressive improvements in LLM text qualities that have made them increasingly indiscernible to both human readers as well as automated methods developed for this task [29,30]. The attribution task is one of the most crucial authorship-related tasks due to its implications and potential impacts on how society interacts with and perceives machine-generated texts at large. One such critical concern in the rapidly expanding and highly competitive LLM market is the issue of content ownership, and the verification of creative authorship. With LLMs frequently engaging in the paraphrasing and editing of human-generated texts, a crucial challenge arises: not only distinguishing machine-generated content from human-authored text, but additionally identifying the specific contributions.

For instance, consider a scenario in which some parts of a human-written text is paraphrased by an LLM, with the latter introducing factual inaccuracies or misinformation. In such cases, being able to discern the contributions of each author is essential, particularly for identifying potentially malicious actors or detecting the infiltration of unreliable content into critical domains such as journalism and public discourse. This capability has far-reaching implications, including its potential use in detecting perjury and preventing the adversarial manipulation of information, thereby safeguarding the integrity and credibility of public and media discourse. Furthermore, accurately identifying LLM-generated content is crucial to preventing unjust claims of intellectual property (IP) rights. In a competitive landscape, where the monetization of LLM technologies is increasingly pursued, the ability to attribute text accurately to its source prevents undue claims of ownership and reinforces the defense against erroneous or fraudulent IP assertions. This issue is particularly salient in academic contexts, where the use of closed-source LLMs by students or researchers without appropriate declaration of content ownership raises ethical concerns related to plagiarism and academic integrity.

The ability to trace LLM authorship also has significant implications in cases where the output of one LLM is misrepresented as the product of another, with the intent of exaggerating the capabilities of a given model. Such misrepresentation could be employed to attract investment or support for LLM applications in high-stakes fields such as healthcare, education, or creative industries, where the perceived quality and reliability of the model are critical factors in securing financial backing.

• **Obfuscation**: This task refers to the process of hiding an author's identity by changing the overlying text such that it preserves the content or meaning of the original text while changing the phrasing of it in such a way that it cannot be mapped back to the original author. This often involved shifting the style of presenting the content so as to hide an author's stylistic signatures. The motivation for obfuscation is the opposite to that of the attribution, with both tasks trying to undo the efficacy of the other. While attribution is a means to safeguard against identity theft in some ways, obfuscation is meant to encourage identity theft or shift so as to protect the original author. This task setting is crucial in settings with critical data. For example, in the case of political discourse, it is essential to be able to disguise the identity of an author or potential whistle-blower to protect the writer from any retaliation for making restricted information public. For law enforcement, too, incriminating evidence that needs to be presented anonymously is often subject to obfuscation so as to erase any remaining traces of stylistic nuances that might give away the author's or source's identity.

More generally, obfuscation can also be thought of as a means to "depersonalize" critical data by removing any signs of which organization or LLM authored a text. This is of interest to the LLM market competitors who might use obfuscation as a means to avoid being marked as a source of misinformation, hallucinations, or fake news that can be of high impact. The obfuscation process can help provide some legal blanket in such scenarios to organizations trying to avoid lawsuits or IP-related legalities. Furthermore, there is evidence to suggest that the knowledge

of authorship can often skew how the content is perceived. This has implications for reducing biases in the inferred legitimacy, quality or trustfulness of a text based on just its source. For example, if a news article is obfuscated to emulate the style of a Republican journalist or publisher, this might increase the trust of that content by readers who are familiar with their style.

On similar lines, individual writers might also seek to avoid personal bias by ensuring their work is judged solely on its content rather than their reputation or past work. In literary and academic contexts, authorship obfuscation can serve as a form of creative or scholarly experimentation, exploring how style and identity influence reader perceptions and stylistic consistency. Similarly, authors may alter their style to adopt fictional personas, creating distinct voices for characters. In regions with restricted free speech, authors may also obfuscate their identities to evade censorship or government retaliation, enabling the spread of controversial ideas without facing persecution. There is also a direct incentive in trying to circumvent authorship attributors that are responsible for moderating and removing machinegenerated texts in social forums means for human interactions such as Reddit and Twitter. Thus, obfuscation has implications for free speech protection, privacy, digital surveillance, and content moderation.

• Collaboration: Most methodologies built to analyze and understand machinegenerated texts make the broad assumption that the texts are strictly written by a single author end-to-end or homogenous in authorship, style, and the result of a continuous writing process. However, there is a shift in this paradigm driven by tools such as CoAuthor [8], CoPoet [31], ChatGPT [32], and other tools that enable human-machine collaborative writing. One step further, there are frameworks such as vLLM, HuggingFace, and LangChain that are now enabling simultaneous calls to multiple LLMs and seamless plug-and-play of LLMs from different organizations, both open and closed-source. Both of these scenarios are giving rise to an emerging increase in the amount of multi-authored or collaboratively written texts. However, this shift in real-world data has not been reflected equally by methods that address authorship-related concerns listed previously.

There is a need to adapt current machine-text evaluations, attributors, and obfuscators to this multi-author setting so that they do not fail from the assumptions of single-authored texts. However, a significant roadblock in such efforts has been the lack of data to train or even fine-tune existing approaches. Thus, there is a need for open-source datasets that capture the interactions of multiple LLMs in solving different types of writing tasks collaboratively via sequential prompting and generation. Such datasets can, in turn, help adapt existing methods to work in the multi-author setting and automate new tasks that arise, such as the detection of author boundaries, multi-author attribution, and evaluation.

In this thesis, we focus on the aforementioned authorship-related tasks that provide a deeper understanding of and insight into the properties of machine-generated texts, their detection, obfuscation, and collaborative generation. In the next sections we formalize each of these tasks and provide an overview of the methods and datasets developed as follows:

Machine-generated texts can be evaluated in many ways. This evaluation can range from particular properties, such as grammatically, to more complex evaluations, such as assessing the truthfulness of the text generated. In this dissertation, we aim to evaluate if **machine-generated texts are human-like**. To do this, we first define "human-likeness" using a psycholinguistic theory that delineates a particular property that humans follow in their language production. This theory is called the Uniform Information Density [33–37] theory and is crucial to both the evaluation as well as attribution portions of this thesis. We introduce this psycholinguistic principle in the next section:

#### 1.2 Background

In 1948, in his seminal paper A Mathematical Theory of Communication, Claude Shannon introduced Information Theory, a theory of communication [38]. According to this theory, the purpose of communication is to be able to transmit a message (out of all possible messages) across a noisy channel in a way such that the receiver is able to interpret the message at their end despite the noise in the channel [38]. Thus, information theory delineates how information should be transmitted across a channel or medium such that it is received as was intended by the sender while constructing the message or information to be shared. This theory has made inroads in how humans communicate via the channel of language. Specifically, during dialog, interlocutors are in this "information theoretic" scenario wherein one speaker says an utterance that is received and interpreted by the other participant in the dialog. The first one then again interprets the response of this second speaker, the conversation builds up and flows back and forth based on the mutual understanding that comes from this exchange of messages. There is an important aspect here of selecting a message or a particular way of constructing an utterance from all possible choices. When communicating, speakers are actively making choices such as what words to use and how to form the syntactical structure of the utterance. The purpose of this selection remains to be able to convey a meaning clearly and succinctly. It is here that the principle of Uniform Information Density (UID) positions itself in human dialog. But before a discussion of the role of UID in dialog, it is important to establish certain concepts to establish the basis of the theory. To begin with, Shannon defines the information content of a word as being directly related to its probability in a given context. Less predictable words have more information and more predictable words have less information. For example, in the following sentence:

#### "The Sun rises in the East."

the word "*East*" is very highly predictable from a semantic standpoint because the sentence is conveying a known concept. Thus, given its context ("The", "Sun", "rises", "in", "the") this word ("East") has high predictability, and thus less information content according to Information Theory. Shannon further goes on to associate information content with surprisal i.e. the amount of surprise that a word raises in a given context. Going back to the aforementioned example, the word "East" has very little surprisal as it is not surprising to read that the Sun rises on the East. Thus, it can be said that low probability is associated with high surprisal. These two thus go hand in hand with high information content. Conversely, high probability is accompanied by low surprisal and subsequently, lower information content.

When speakers engage in language use for the purpose of communication, they replicate the information theoretic concept of information transmission in a noisy channel of some limited bandwidth [38]. Shannon's definition of information content of a component or unit (n) is given by the inverse logarithm of its probability (p(n)) i.e.

$$I(n) = -logp(n) \tag{1.1}$$

Information Theory states that information exchange is optimized when information travels across the (noisy) channel at a uniform rate i.e. the amount of information transmitted should remain uniform per time close to the channel's Shannon (information) capacity. In dialog, the way UID is maintained is often through means such as modulating word length, choice of vocabulary, use of contractions and so on. Thus, a violation of UID can then be computationally detected when the amount of information content (as per Shannon's definition) per linguistic unit (sentence length/number of words) goes up or down all of a sudden at the onset of a word or sentential element. For language, the implications of this information optimal information exchange have manifested as a strategy that speakers use during language production. Frank and Jaeger's corpus-based study demonstrated that maintaining UID is one strategy that speakers use to regulate and optimize production [33]. The study shows how American English speakers tend to use a contraction ("you're") as opposed to the lengthened variant ("you are") when expressing elements of low and high information content, respectively [33]. Their study thus shows that one would use shorter elements for lower amounts of information and longer elements/sub sequences for expressing higher amounts of information. Thus, in a way keeping the information rate in the 'channel' close to uniform - which is the premise of Shannon's Information Theory.

[34] extended upon their work and addressed the often quoted problem with writtentext based studies in dialog and production by basing their analysis on spoken dialog so as to capture the nature of conversation as a 'joint activity' between two speakers [34]. They report that UID is consistent at the inter and intra-sentential levels. In their analysis, the information content drops at the beginning of a new topic (topic shift) and then eventually grows as the topic is being discussed more, and eventually lowers again as the topic ends. They also examine the variation in the contribution to information density from speakers playing different roles (topic initiator v/s topic responder), specifically in the context of the introduction of a new topic in dialog. They find that topic initiators and topic responders have decreasing and increasing entropy, respectively [34, 39].

In another work concerning UID in production, Jaeger and Levy investigated if syntactic reduction is a means by which speakers uniformly distribute information in utterances [40]. They found that speakers chose not to omit an optional function word at the onset of a less predictable phrase, but that they were more likely to omit the same word at the beginning of a more predictable phrase. Further, a corpus study [35] consolidated previous results that in choices that pertain to the syntactic structures that a speaker might pick, they are regulating those choices as per UID: the speaker is actively trying to make choices that would distribute the information that needs to be conveyed in a uniform manner across the linguistic (speech) signal. This is also another pivotal study of the foray of UID in production at the level of grammar structures for monolinguals. They used the Penn Treebank and Switchboard corpus of conversations in a generalized linear mixed model study of the effect of information density on syntactic structure preferences. They specifically found the effect that speakers were less likely to produce a complementizer "that" the lower the information density of the complement clause onset. This paper [35], thus, established the role of UID in choices at the syntactic level during production.

In Manin's work, he puts forth an interesting analysis of the relationship between word length and its predictability [41]. Specifically, he finds from corpus analysis, that the logarithm of the probability of a word given context (unpredictability) is linearly dependent on the length of the word. This is interpreted as a pointer to the notion that natural language is prone to 'even out' information rate such that longer words are less predictable and hence carry more information. In another study of the correlation of word length with information content, [36] find that speakers pick the shorter version of a word ("chimp") as opposed to the longer counterpart ("chimpanzee") in more predictive (lower information) and less predictive (higher information) contexts, respectively [36]. They attribute this finding to the fact that speakers are in fact smoothing out information distribution and making 'information-theoretically efficient' choices thereby lending even further support to the notion that word length is correlated with Shannon's idea of information content than their absolute frequency in usage [36].

In an expansive study across ten languages, Piantadosi et al. studied the relationship between word length and the information content of the words according to Shannon's definition [42]. They found, for the 10 languages, that information content was a considerably more significant predictor of word length than frequency, given the variation in the degrees of morphological inflections in them [42]. Finally, [37] studied the usage of 'less informative' expressions as a means of conveying meanings with higher predictability in a large-scale scale experiment that directly assessed comprehenders' ease of predicting the referent in an unfolding utterance [37]. They found, from an information theoretic analysis, that speakers tended to refer to highly predictable referents with short words. They found that pronouns were used for highly predictable referents while proper names were used for less predictable referents. This study also lends us an interesting perspective on the relative accessibility of pronouns v/s proper names given different contexts.

Authorship Task	Description	Research Question	Motivation
Evaluation	To assess texts according to some measure	Do machines distribute surprises like humans?	To understand if ma- chines have come close to human performance in writing tasks
Attribution	To predict the author of any text	Can surprisal help iden- tify the author of an arti- cle?	To safeguard against pla- giarism and academic in- tegrity violations
Obfuscation	To hide the author's style from a text	Can syntactic patterns be leveraged to hide author- ship?	To protect whistleblowers and eliminate biases
Collaboration	To have multiple LLMs solve writing tasks to- gether	How can sequential prompting be used to generate collaborative stories?	To enable research efforts in multi-LLM scenario

Table 1.1: An overview of the authorship tasks, research questions and motivation for each task presented in this thesis.

#### 1.3 RQ 1 & 2: Machine-Generated Text Evaluation

In light of UID theory, we refine our aim of evaluating if machine-generated texts are human-like by asking the following research question (RQ1): Do machines distribute information like humans do? Since UID theory tells us how humans distribute information or surprisal in their language generation, we aim to operationalize the surprisals distributions of both human and machine-generated texts. A comparative study of the two text distributions is presented in Chapter 2. The UID principle states that humans spread information uniformly or smoothly when planning utterances. We operationalize this theoretical aspect of human language to define "human-likeness" using surprisals derived from a language model's learned probability distribution. This enables us to measure and mechanistically evaluate the differences between humans and different machine-generation settings (i.e., decoding strategies) by comparing their surprisal distributions. We find that humans and machines do, in fact, spread surprisals differently in their texts.

We also ask a second research question (RQ2): Does more human-like text

**correlate with better text quality?** To answer this question, we use our UID operationalization as the score for human likeness and measure the correlation of this score with different measures of quality, both automated metrics as well as human evaluation scores. We conduct a large-scale human evaluation study to evaluate text generated using different decoding strategies as well as the reference human-generated text for a dialog response generation task on qualitative measures of how interesting, relevant, and engaging responses are.

#### 1.4 RQ 3: Authorship Attribution

Attribution refers to the task of automatically identifying the author of a given text. Since we learn from our evaluation study (RQ1) that humans and machines differ in how they distribute surprisal or information, we then study if these differences can help differentiate between human and machine-generated texts (i.e., the Automated Turing Test) and identify authorship. Our research question in the attribute setting is (**RQ3**): **Can surprisal distribution help indemnity the author of an article?** To this end, we propose GPT-who, an information-density-based text authorship attributor in Chapter 3. We developed a means to utilize the surprisal measures as part of a fast authorship attributor and found that these surprisal differences are distinguished between authors. Thus, UID theory is a useful tool in analyzing and identifying machine-generated texts, and it captures how machines are not yet human-like as per surprisal distributions.

#### 1.5 RQ 4: Authorship Obfuscation

While automated methods to detect machine-generated texts are on the rise, there is also a need to counteract detectors or solve the opposite problem of hiding or protecting authorship. For example, in the case of whistleblowers, journalism/free speech protection, there is often critical information to be made public or to be shared while needing to protect any tracing to the author or source. There is also the case of privacy protection, where an author might not want to be known, or in the case of high-risk or personal data, where de-personalization is required in the form of authorship style change. In some instances, hiding the authorship of information can also help reduce perceived biases or any partial analysis of the content. This leads us to the third task of authorship obfuscation (Chapter 4). To obfuscate authorship means hiding the identity or making the author's identity hard to identify based on the obfuscated text. We hypothesize that how a text is presented is the source of its style as opposed to what the text is meant to convey and that this "how" is captured by syntactic patterns. Thus, we ask the research question (**RQ4**): **Can syntactic patterns help obfuscate authorship?** We present a novel obfuscation technique called ALISON, which identifies the most frequently used and important syntactic patterns of each author and changes the corresponding text. We define syntactic patterns as parts-of-speech (POS) n-gram sub-sequences. Using syntax to guide the text obfuscation process works and provides evidence for our hypothesis that author style is sufficiently captured by an author's syntactic choices to phrase the information instead of the content or semantics of the information.

#### 1.6 RQ 5: Collaborative Generation

Finally, we move to a forward-looking and upcoming scenario that has not yet been addressed by most authorship-related methods, including evaluators, detectors, and obfuscators that always assume that every text is generated homogeneously by one author. Moving beyond such single-author settings, we take into consideration rapid developments in unifying frameworks such as HuggingFace, vLLM, and langChain that have seamlessly enabled the collective and interoperable use of multiple LLM authors in the same text. However, we first need a dataset that captures this multi-LLM authorship to build and adapt current methods to such an emerging scenario (Chapter 5). This brings us to the research question (**RQ5**): To generate a collaborative text benchmark using sequential prompting. We present CollabStory, a corpus of creative stories co-authored by multiple different LLMs, and elaborative generation.

By examining these four areas summarized in Table 1.1, this thesis aims to provide a comprehensive framework for understanding the current state of authorship-related tasks in the content of LLM-generated texts. This research not only sheds light on the technical challenges but also considers the broader implications for society, including the ethical and legal aspects of authorship in the age of machine-generated content.

## Chapter 2 Machine-Generated Text Evaluation: Are machine-generated texts human-like?

#### 2.1 Introduction

The Uniform Information Density (UID) hypothesis states that humans distribute information in their utterances evenly for optimal communication [35, 43]. Consequently, language generation has benefitted from UID-based objectives and regularization [44, 45]. Specifically, [46] argued that UID can be optimized for machine translation using beam search. Yet, the effect of different decoding algorithms on information density distributions of generated text are unknown, as is UID's broader role in neural response generation in the special case of dialogue models. Here, we investigate (i) if different decoding algorithms follow the UID principle, and (ii) if following the UID principle is beneficial for dialogue response generation, and (iii) collect human annotations of qualitative measures for multiple candidate responses to dialog histories generated using different decoding algorithms to study the relationship of dialog response quality and UID. We operationalize UID as the variance of surprisal and measure its correlation with automatic metrics (e.g., BLEU, METEOR, BERTScore) as well as human judgments on qualitative measures of response quality and find that adherence to UID correlates negatively with human judgments when the responses have very low/high surprisal.



Figure 2.1: Information density distribution of human and GPT-2 generated responses for the Persona-Chat dataset. A smaller width of the distribution or more values closer to 0 implies more uniform responses, and a wider spread (such as for the human response) indicates more non-uniform responses.

#### 2.2 Related Work

#### 2.2.1 Language production in humans

Spreading information content evenly in utterances is a marker of optimally strategized responses, and humans follow this UID principle as a means to state their thoughts clearly and to make themselves intelligible [33,47]. The probability of a sentence has been associated with the cognitive load it incurs [48]. As a means to avoid salient variations in the information content (surprisal, i.e., negative log probability) of responses, speakers maintain UID through linguistic choices such as that at the phonetic [49], syntactic [35] and lexical level [36].

#### 2.2.2 Response generation in machines

While large-scale pre-trained language models provide a rich prior for dialogue response generation, the choice of decoding algorithm used at the time of generation is crucial for the quality of generated responses [50–54]. While vanilla sampling often tends to produce incoherent text, greedy decoding leads to safe and repetitive responses. More recently, top-p/nucleus [50] and top-k sampling [55] are used to tune values of p/k to



Figure 2.2: Histogram of **UID Scores** of responses generated using different decoding algorithms. The farther the UID score from 0, the less uniform or more non-uniform the response. Human-generated reference text (left-top) has a higher frequency of non-uniform responses as compared to any model setting as can be seen from the wider spread of scores away from 0. Also, as the values of p and k increase (*left to right*), the information density distribution slowly approaches reference text-like non-uniformity.



Figure 2.3: Surprisal at every token in candidate responses to the same dialog history, color-coded with human-annotated **interesting** scores. Plots (*left to right*) are arranged in increasing order of uniformity (i.e. variance along y-axis). Less uniform the surprisal (left-most), better the score.

balance the diversity-quality trade-off [51, 56].

#### 2.2.3 The UID principle and decoding algorithms.

Both the UID principle and decoding algorithms (examples in Table 2.1) can be seen as guiding mechanisms for dialogue response production in humans and generation in ma-

Decoding Strategy	Description
Greedy	Always pick most likely token
Vanilla	Sample from Language Model's probability distribution
$\mathrm{Top}\text{-}k$	Sample from $k$ most probable tokens
Top- $p$ /Nucleus	Sample from $p$ portion of the probability mass

Table 2.1: Common decoding strategies in text generation using language models.

chines, respectively. UID's role in machine-generated dialogue is not well understood, with previous work mainly focused on machine translation and language modeling [45, 46, 57]. To address this gap, we present a comparative study of decoding methods to develop a deeper understanding of the role of UID in dialogue response generation.

It is thus a natural question to ask if dialogue models leverage the UID principle to improve generation quality. Despite its significance in human dialogue [33,47], its integration with models of human language production is not yet clear. Decoding algorithms in generative language models suggest a competition account that assumes that multiple words are available for selection (in the case of the common autoregressive models), although unlike in some psychological accounts, they do only compete by their relative probabilities. UID could, of course act on available representations differently, and this work makes no claim with respect to alternative accounts of how UID is implemented from the psycholinguistic perspective.

Information density is a measure derived from the probabilities of the generated text under the model's learnt distribution. lends itself to the idea that the information density patterns of different sampling settings would vary from each other and also from that of the gold standard or human-generated reference text. We examine if and how the information density patterns of common sampling methods differ. Further, we investigate if the differences in the response quality of the sampling/decoding methods could be explained by their information density patterns.

Generation Type	Length	BLEU	chrF	METEOR
p = 0.3	10	.00	.14	.12
p = 0.5	05	.03	.13	.10
p = 0.6	04	.06	.14	.13
p = 0.8	10	.03	.06	.05
p = 0.9	11	00	.03	.04
Greedy	14	.01	.14	.13
k = 10	04	.15	.03	.05
k = 20	05	.14	.05	.06
k = 50	09	.01	.03	.03
k = 100	07	.04	.00	.02

Table 2.2: Part A: Pearson's correlation coefficient (r) between UID score and Length, BLEU, chrF, and METEOR metrics of dialog responses generated using different decoding settings.

Generation Type	BertScore	BLEURT	RoBERTa	SacreBLEU
p = 0.3	.17	.17	0.19	.13
p = 0.5	.18	.17	.2	.15
p = 0.6	.01	.06	.01	.00
p = 0.8	.18	.16	.2	.15
p = 0.9	.16	.15	.19	.14
Greedy	.06	.05	.06	.06
k = 10	.07	.08	.07	.07
k = 20	.05	.04	.06	.04
k = 50	.06	.03	.03	.05
k = 100	.11	.08	.08	.08

Table 2.3: Part B: Pearson's correlation coefficient (r) between UID score and BertScore, BLEURT, RoBERTa, and SacreBLEU metrics of dialog responses generated using different decoding settings.

#### 2.3 Experimental Details

#### 2.3.1 Model & dataset

We use the fine-tuned GPT-2 [58] model provided by HuggingFace and use their data preprocessing and response generation scripts<sup>1</sup>. We used the Persona-Chat [59] data split

<sup>&</sup>lt;sup>1</sup>https://github.com/huggingface/transfer-learning-conv-ai

provided by the ConvAI2 challenge  $[60]^2$ . We then generated responses for 7500 dialogue histories randomly picked from 7801 validation set examples using vanilla, top-p, top-k sampling and greedy decoding.

#### Decoding algorithms

Vanilla sampling randomly picks the next token from the model's probability distribution, including many long-tail samples. Top-k samples from the k most probable tokens; Greedy decoding is Top-k = 1 decoding, always selecting the most probable next token. Top-p (Nucleus) sampling selects the next token from the top p portion of the probability mass.

#### 2.3.2 Uniform Information Density score

We measure UID as the variance of the surprisal (negative log likelihood) of each token in the response [45, 46, 61]. This measure is able to capture any sudden variations in the surprisal of the tokens in the sentence. UID Score is formulated as follows: the dialogue model learns a conditional probability p parameterized by  $\theta$  to predict the next token  $(y_t)$  in the sentence. The surprisal (u) of the next token  $y_t$  is,

$$u(y_t) = -\log(p_\theta(y|x, y < t)),$$
(2.1)

for  $t \ge 1$  where  $y_0 = \langle EOS \rangle$ , t = time step, and x = dialogue context. Higher the surprisal, lower its probability and vice-versa. Thus, surprisal indicates how unexpected or surprising a token is in a given context. Average surprisal of a sentence (y) is defined as,

$$\mu(y) = \frac{1}{|y|} \sum_{t} (u(y_t))$$
(2.2)

Finally, the *UID score* of a sentence (y) is defined as the negative normalized variance of the surprisal:

UIDscore
$$(y) = -\frac{1}{|y|} \sum_{t} (u(y_t) - \mu)^2$$
 (2.3)

<sup>2</sup>https://github.com/DeepPavlov/convai/tree/master/2018
From this formulation, a perfectly uniform sentence would have a variance equal to 0 (i.e. the surprisal of every token in the sentence is equal). Since we take the negative of the variance, the higher the absolute value of UID score, the more non-uniform its information density.

#### 2.3.3 Information density of decoding algorithms

For all the model and human generated responses to dialogue histories taken from the Persona-Chat validation set, we calculate and plot histograms of UID scores as shown in Figure 2.2 so as to gauge how different decoding algorithms distribute information in their model responses. We also visualize the histogram of average surprisals for the same set of candidate responses as shown in figure 2.8.

#### 2.3.4 Response evaluation

#### Automatic metrics

We measure the quality of responses using length (number of tokens), BLEU<sup>3</sup> [62], METEOR<sup>3</sup> [27], character level F-score (chrF)<sup>3</sup> [63], BLEURT<sup>4</sup> [64], a RoBERTa [65] based text similarity score<sup>5</sup> [66], BERTscore<sup>4</sup> [28] and SacreBLEU<sup>4</sup> [67].

#### Human evaluation

To study the effect of adherence to UID on the perceived quality of generated responses beyond n-gram, reference-based, and learned automatic metrics, we collected human judgments along 3 measures – **related** (to the dialogue history), **furthering** (if a response keeps the conversation going/is encouraging for the dialogue partner) and **interesting** (if the response provides engaging/new information). We provide screenshots of the task interface (Figure 2.5 and instructions in Figure 2.6.

Raters were selected based on the criteria that they be located in the US, and had attempted a minimum of 500 HITS at an accepted work rate greater than 97% on MTurk. We asked raters on MTurk to answer if a candidate response satisfied each of the

<sup>&</sup>lt;sup>3</sup>https://github.com/nltk/nltk/tree/develop/nltk/translate

<sup>&</sup>lt;sup>4</sup>https://github.com/huggingface/datasets/tree/master/metrics

<sup>&</sup>lt;sup>5</sup>https://github.com/UKPLab/sentence-transformers/blob/master/docs/usage/semantic\_textual\_similarity.md



 nice. i love to do stuff like that. i also swim.
 Top-k
 {.,,,.}
 {.,,..}

 are you a skater or do you do it daily?
 Greedy
 {.,,..}
 {.,,..}

 Figure 2.41. Our dataget contains 4 condidate regroupses for every dialog bigtory, along

Figure 2.4: Our dataset contains 4 candidate responses for every dialog history, along with human annotations for 3 qualitative measures.

qualitative measures (interesting, furthering and related) and gave them three response options: "Yes", "Somewhat" and "No". In a pilot study of 360 responses, we also included a measure for fluency. All of the responses were rated "Yes" by majority vote and we removed this measure from further analysis as all the generations in this study were fluent as indicated by the pilot study and from our observation. Figure 2.4 shows an overview of the data format collected in this study.

For correlation calculations, we assign integer score values to each of the three response options as 3 for "Yes", 2 for "Somewhat" and 1 for "No". Thus, the higher the score, the better the response is rated. Following the pilot study, for 194 dialogue histories, we showed the raters 4 candidate dialogue responses (total of 776 dialogue responses) and collected ratings on all \*3\* measures from \*3\* raters per dialogue history. In all, we obtained a total of 776\*3, i.e., 2328 total response-rating pairs. To calculate the score for each response along every measure, we take the mean of all ratings as the score. For cases where at least 2 out of 3 raters agree, we take majority vote as the final score. This constituted (2018 out of 2328) 86.68% of all the ratings collected. We show the overall distribution of qualitative scores for all the response-rating pairs in Figure 2.7. We verified the rater responses by checking if they were rating human-generated responses highly as those came from a trusted source (Persona-Chat). We also manually inspected Instructions Examples

Please read the following conversation history & rate the responses that follow as if you were a participant in this conversation:

Speaker A: its all tedious at first dear , but i know it will get better for you .

Speaker B: i guess it just really want to be out on my red bike .

Speaker A: sounds like a true joy . i'm retired now , so you are inspiring me to try that .

Speaker B: awesome . maybe we could go out riding together .

Speaker A: i would love that . life is too short to miss out on making new friends .

Rate each of the 4 candidate responses (1st column) on the 3 quality measures (2nd column onwards) based on how well each response satisfies the quality description.

You will be entering a total of 12 responses. Fill the responses one row at a time, i.e. first read the response in a row, and rate it on all 3 qualities before moving to the next row.

		Quality Measures & Description	
Candidate Responses	Related Is it on topic with the conversation history?	Furthering Does it encourage the conversation to continue?	Interesting Does it present engaging or new information?
ne to. life is a great tool to have!	Yes	Yes	Yes
	Somewhat	Somewhat	Somewhat
	No	No	No
hat is true. you guys should come ride with me!	Yes	Yes	Yes
	Somewhat	Somewhat	Somewhat
	No	No	No
ou are right. that is true. i am just happy to have someone vho loves me.	Yes Somewhat No	Yes Somewhat No	Yes Somewhat No
have to go, talk to you later.	Yes	Yes	Yes
	Somewhat	Somewhat	Somewhat
	No	No	No

Figure 2.5: Screenshots of our MTurk study interface for collecting human judgments on 4 candidate responses per dialogue history, along 3 quality measures.

a random subset of dialog history-candidate response sets and found the results to be in accordance with our intuitions.

# 2.4 Results

## 2.4.1 Information density of model responses

We plot the histograms of UID scores computed for all of the generated responses in Figure 2.2. The information densities of human-generated responses have a wider spread than responses produced by the models. Overall, the human-generated reference text has more non-uniform sentences than all model-generated responses. We notice a very high and narrow peak in the case of greedy decoding. This is not surprising as responses sampled using greedy search maximize the probability of the next token (minimize

#### 1. Read the given conversation history carefully.

# 2. Then, rate the quality of 4 candidate responses as potential next responses to the conversation history along 3 quality measures (12 responses in total):

<u>Note:</u> Respond as though you are a participant in the conversation. For example, do not mark a response as uninteresting due to personal preference. Instead, consider how a person in the conversation might find it.

Quality Measure	Description
Related	Does the response follow the conversation history's general topic and is a valid continuation of the dialogue?
Furthering	Does the response encourage the conversation to keep moving forward? This might be through a question or a response that can be easily followed-up on.
Interesting	Does the response present new or engaging information?

The following table contains <u>examples of candidate responses</u> corresponding to each rating option (Yes/Somewhat/No) for all 3 quality measures <u>for the given conversation history</u>:

#### **Conversation History:**

Speaker A: I do enjoy trying out different cuisines. Speaker B: Oh, nice. What's your favorite food? Speaker A: I like Peruvian food quite a lot. What about you?

Quality Measure	Rating	Response Example
	Yes	Oh, I recently made Ceviche at home! But I'm more into Mexican street food.
Interesting	Somewhat	I like Peruvian too.
	No	Nothing much.
	Yes	What do you like about it?
Furthering	Somewhat	I'm more into Mexican street food actually.
	No	I'm not into it.
	Yes	I do like it, but it's so hard to cook at home.
Related	Somewhat	Me too! What else do you enjoy?
	No	I go for runs all the time.

Figure 2.6: Instructions and examples from MTurk study.

surprisal). Consequently, such responses would have very low surprisal at almost every word, hence lower variance. Vanilla sampling uses the probability distribution learned from the training data, which might be why it is also closer to the validation set (reference text) distribution. With increase in p and k, we see that the information density distribution spreads across a larger range and includes more non-uniform responses, slowly approaching that of the reference text.

The UID scores distribution for top-p and top-k sampling can thus be thought of as lying between the two extremes of greedy (k=1) where most responses are very uniform to vanilla (k=size of vocabulary) where responses are relatively non-uniform due to random spikes in surprisal. Thus, the larger the portion of the model's learned distribution being accessed by the decoding algorithm, the more non-uniform or reference/human-like the responses generated using them in terms of information density.

We find that i. vanilla sampling follows the information density distribution of the human-generated text or the UID principle; ii. Greedy sampling leads to generations that are more uniform than the vanilla distribution. For top-p/k sampling, as the value of p/k increases, the generations progressively become non-uniform, gradually approaching the vanilla distribution; iii. UID does not correlate with n-gram/reference-based automatic metrics, though they do correlate with human judgements along measures of relatedness and interestingness. Thus, providing evidence that UID does capture nuances about generation quality that go beyond automatic metrics that are insufficient to measure task performance for open-ended dialogue. Finally, we also propose a speaker-specific UID regularizer that attunes the information density of model generations for multi-turn dialogue.

Higher values of p and k have shown to produce better quality text than lower cut-off criteria. In addition, these two sampling methods have also been known to produce better text than greedy or vanilla sampling. The differences in the UID distributions and the known differences in their generation quality thus need a further exploration of the relationship of UID score with quality metrics. Thus it is crucial to study the effect of UID while controlling for surprisal.

For the task of dialogue generation on PersonaChat, we find that nucleus sampling and top-k sampling do not follow the UID principle at higher values as argued for the task of machine translation in [46]. This confirms that it cannot be assumed that truncated probability distributions follow the same information distribution patterns as ground truth or human-generated data. For example, we see that vanilla sampling has a UID score histogram most similar to the reference text. This can be attributed to the fact that GPT-2 was fine-tuned on human-human dialogue and vanilla sampling utilizes GPT-2's learnt probability distribution as it is. We see, in contrast to our expectation, that the reference text isn't as uniformly distributed as other types of generation, and the information density curves exhibited by the sampling distributions rely on the distribution

		Pearson's $r$ between			
		UID score and qualitative metric			
Surprisal interval	n	Related	Furthering	Interesting	
(0.8, 1.2)	24	.17	03	30*	
(1.2, 1.6)	64	.12	.08	13	
(1.6, 2.0)	91	.05	23*	07	
(2.0, 2.4)	109	04	13	00	
(2.4, 2.8)	111	06	21*	05	
(2.8, 3.2)	105	02	.01	10	
(3.2, 3.6)	99	23*	10	.19	
(3.6, 4.0)	66	.03	05	09	
(4.0, 4.4)	42	33	22	09	
(4.4, 4.8)	24	14	61*	.04	
(4.8, 5.2)	12	33	14	$54^*$	
(5.2, 5.6)	13	98*	64	38	

Table 2.4: Pearson's r between **UID score and and human judgments** of qualitative measures for dialog responses bucketed by surprisal [Surprisal interval = the ranges of surprisal values used for bucketing responses, n = number of responses in each surprisal interval, \*p-value < .05]

of fine-tuning data, and thus UID might not be an inherent quality of any sampling method. It is also key to note that the task of attuning sampling methods is necessary in the cases where the truncated probability distribution no longer follows the UID distributions of the reference text or the vanilla distribution.

#### 2.4.2 UID score & automatic metrics

We present the correlation between UID scores and automatic metrics calculated for the generated dialogue responses in Tables 2.2 and 2.3. UID scores have a weak correlation with RoBERTa-based similary scores for two settings of nucleus sampling. Other than that, UID scores are not correlated with automatic metrics of response generation. We take this to be an indication that if UID scores do capture any aspect of response quality, it goes beyond what is measured by such metrics and might provide for a better evaluation criteria.

Generation Type	Pearson's $r$
Reference Text	69
Greedy	23
p = 0.3	43
p = 0.5	50
p = 0.6	56
p = 0.8	65
p = 0.9	68
k = 10	40
k = 20	45
k = 50	56
k = 100	63
k = 200	65
k = 500	69
Vanilla	74

Table 2.5: Pearson's correlation coefficient (r) between UID score and average sentence surprisal (all p < 0.01)



Figure 2.7: Frequency of responses (Yes/Somewhat/No) for each qualitative measure in our human annotated dataset.

	Pearson's $r$			
Quality	UID Score	Surprisal		
Related	.01	13*		
Furthering	.03	10*		
Interesting	04	01		

Table 2.6: Pearson's correlation coefficient (r) of **UID score and surprisal with** human judgments of qualitative metrics (\*p < 0.01)



Figure 2.8: Histograms of **average sentence surprisal** for responses generated using different decoding settings and human-generated reference text (left-top).

#### 2.4.3 UID score & human Judgments

Motivated by the fact that UID score is derived from surprisal, we test if surprisal is a confounding factor and find that, indeed, UID scores were highly correlated with average surprisal (Table 2.5). To tease apart the effect of UID scores on response quality, we controlled for surprisal by grouping or bucketing responses into 12 intervals of surprisals (within a range of 0.4 units as shown in the first column on Table 2.4). Within these intervals, surprisal had no correlation with generation quality (Table 2.7). Once we control for surprisal i.e. analyse dialog responses with similar surprisals but varying UID scores, we observe that UID scores negatively correlate with human judgments, to varying degrees of strength, for responses in very low or high surprisal intervals (see Table 2.4). Thus, for the extremities of the surprisal range, UID scores indicate that better rated responses are non-uniform.

			Pearson's	r
Surprisal interval	n	Related	Furthering	Interesting
(0.8, 1.2)	24	03	04	00
(1.2, 1.6)	64	10	16	.08
(1.6, 2.0)	91	.05	.14	.10
(2.0, 2.4)	109	14	08	27*
(2.4, 2.8)	111	12	.05	.09
(2.8,  3.2)	105	02	.06	00
(3.2,  3.6)	99	13	.12	.01
(3.6, 4.0)	66	.02	06	.06
(4.0,  4.4)	42	01	00	.06
(4.4, 4.8)	24	.20	.34	.23
(4.8, 5.2)	12	13	37	12
(5.2, 5.6)	13	.60	.83	.76

Table 2.7: Pearson's r between **surprisal and human judgments** of qualitative measures for dialog responses bucketed by surprisal [Surprisal interval = the ranges of surprisal values used for bucketing responses, n = number of responses in each surprisal interval, \*p-value < .05]

Dialogue history Speaker A: hi, i m susie. how are you?				
Reference Text	i'm good. just got back from some vol- unteering. how are you doing?	-1.99	3	
Candidate 1	hi. i am very good. just enjoying my favorite pastime. how are you?	-1.43	2	
Candidate 2	i am great! i volunteer at a soup kitchen and watch movies with my kids.	-1.19	2	
Candidate 3	i'm doing well. how are you?	-0.18	1	
	Dialogue history			

Speaker A: that's cool. my dad made us italian food tonight.
Speaker B: oh nice, i love italian food. my favorite is the garlic
bread. what is yours?

Speaker A: pasta, my son loves pizza though!

Response	Response text	UID	Interesting
type		Score	
Candidate 1	wow, my son took us and my two daugh- ters to a super fast food joint the other day	-4.83	3
Candidate 2	that is too bad, i like the cheesy stuff.	-2.08	2
Reference	do you have a favorite genre of movies?	-1.73	2
Text			
Candidate 3	i also like to stay home with my son.	-0.88	1
	Dialogue history		

Speaker B: i like that too, have to eat lots of carbs for my training.

Speaker A: hello i hope your sunday is great , what is your favorite kind of music?

Speaker B: hey there . been a relaxed sunday . yours ? music eclectic.

Speaker A: my sunday has been exciting ! i enjoy death metal.

Speaker B: death metal . cool . i spent the morning volunteering. Speaker A: i volunteer too , at the local pool to be a swim coach.

Response	Response text	UID	Furthering
type		Score	
Candidate 1	nice. such a nice day. how long have vou been coaching?	-9.99	3
Reference Text	that is great! we both volunteer! mine is rescuing bunnies.	-7.59	2
Candidate 2	that's cool. i donate my pay to the local	-3.89	2
Candidate 3	sick sick. beautiful color, navy blue is my favorite.	-2.79	1
	Dialogue history		

Speaker A: have you heard about the juggalos? weird.

Response	Response text	UID	Furthering
type		Score	
Reference	what are those? do they juggle balls?	-6.88	3
Text			
Candidate 1	no i have not. i like that band myself	-5.19	2
Candidate 2	we will not tell. do you like comics?	-4.27	2
Candidate 3	i have not heard of them. i've dolls too.	-2.8	1
	Dialogue history		
Speaker A:	hello, how are you today?		
Speaker B:	good. yourself? listening to classical	music.	
Speaker A:	that is cool! i just like to skateboard!	!	
Response	Response text	UID	Related
type		Score	
Reference	can you do a kick flip?	-14.11	3
Text			
Candidate 1	haha. i love skating too, but i have a	-11.28	3
	truck to keep as well.		
Candidate 2	nice. i love to do stuff like that. i also	-10.25	3
	swim.		
Candidate 3	are you a skater or do you do it daily?	-9.05	2

**Dialogue history** 

Speaker A: i hope it is a healthy snack. i am doing well, just moved here.

Speaker B: it is. i am vegan. just moved here too! where from? Speaker A: i too am vegan from germany. have you seen lafer! lichter! lecker! on tv?

Speaker B: oh yes! i love shows like that and watched it back home a lot.

Response type	Response text	UID Score	Related
Reference	they do amazing things with the hum-	-6.86	3
Text	mus. where is back home for you?		

Candidate 1	you must have a lot of fun watching	-4.53	2
	them.		
Candidate 2	they have the best new vegan cookbooks,	-2.93	3
	but i am more adventurous.		
Candidate 3	i do love the sky diving, too! i have seen	-2.47	1
	the first few seasons.		

Table 2.8: Examples of dialogue histories followed by 4 response candidates arranged by increasing UID score i.e. from more non-uniform to uniform responses and their corresponding human judgment scores.

# 2.5 Discussion

Humans tend to follow the Uniform Information Density (UID) principle by distributing information evenly in utterances. We study if decoding algorithms implicitly follow this UID principle, and under what conditions adherence to UID might be desirable for dialogue generation. We generate responses using different decoding algorithms with GPT-2 on the Persona-Chat dataset and collect human judgments on their quality using Amazon Mechanical Turk (see detailed examples of dataset collected in Table 2.8). We find that (i) surprisingly, model-generated responses follow the UID principle to a greater extent than human responses, and (ii) decoding algorithms that promote UID do not generate higher-quality responses. Instead, when we control for surprisal, non-uniformity of information density correlates with the quality of responses with very low/high surprisal. Our findings indicate that encouraging nonuniform responses is a potential solution to the "likelihood trap" problem (quality degradation in very high-likelihood text). Our dataset containing multiple candidate responses per dialog history along with human-annotated quality ratings is available at: https://huggingface.co/datasets/saranya132/dialog uid gpt2. We find that (i) UID correlates with human judgments on how interesting and furthering the generated responses are when the surprisal of the responses is too high/low; and, (ii) surprisingly, decoding algorithms generate responses that are more uniform text than the humangenerated reference text.

Contrary to our expectations, we find non-uniformity to be a more desirable property in machine-generated responses. Overall, UID scores and surprisal do not correlate with human judgments (Table 2.6). But when controlled for surprisal, we observe that UID score is correlated with human judgments for certain intervals (examples in Figure 2.3 and Table 2.8). Our results suggest that optimizing UID to generate uniform text might not be the right objective for regularizing decoding algorithms. Instead we find that non-uniform information density could be a potential solution to the "likelihood trap" problem according to which models generate lower quality text (as per human judgments) when sampling from the extremities of their likelihood space [68]. Consequently, we suggest that decoding algorithms be tuned to follow the information density patterns of human-generated non-uniform data when generating responses outside of the "safe" likelihood space.

# Chapter 3 Authorship Attribution using Information Density-based features

# 3.1 Introduction

The recent ubiquity of Large Language Models (LLMs) has led to more assessments of their potential risks. These risks include its capability to generate misinformation [69,70], memorized content [71], plagiarized content [72], toxic speech [73], and hallucinated content [74,75]. To mitigate these issues, researchers have proposed automatic and human-based approaches to distinguish LLM-generated texts (i.e., machine-generated) from human-written texts [69,76–78].

Automatically detecting machine-generated texts occurs in two settings- *Turing Test* (TT) which is the binary detection of human vs. machine; and *Authorship Attribution* (AA) which is the multi-class detection of human vs. several machines (e.g., GPT-3.5 vs. LLaMA vs. Falcon) [79]. While the TT problem is more rigorously studied, due to the wide usage of different LLMs, in the future, it will be imperative to build models for the AA tasks to determine which LLMs are more likely to be misused. This knowledge will be needed by policymakers when they inevitably institute laws to guard the usage of LLMs.

To that end, we propose GPT-who, the first psycholinguistically-inspired supervised domain-agnostic task-independent multi-class statistical-based detector. GPT-who calculates interpretable Uniform Information Density (UID) based features from the statistical distribution of a piece of text and automatically learns the threshold (using Logistic Regression) between different authors. Such features are calculated using the surprisals of tokens in a text, for example, the variance of surprisals, the mean, and so on (elaborated



Figure 3.1: GPT-who leverages psycholinguistically motivated representations that capture authors' information signatures distinctly, even when the corresponding text is indiscernible.

in Section 3.3.1). However, these techniques have limitations: (1) supervised learning approaches utilize deep learning or feature-based techniques which are computationally heavy and require decently sized datasets to perform well; (2) unsupervised techniques are all binary statistical-based approaches that require heuristic techniques to define thresholds used to distinguish these two main authors (LLM vs. Human). To showcase the detection capabilities of GPT-who, we use 4 large LLM benchmark datasets: Turing-Bench [79], GPABenchmark [80], ArguGPT [1], and Deepfake Text in-the-wild [81]. We find that GPT-who outperforms state-of-the-art statistical detectors and is at par with task and domain-specific fine-tuned LMs. This performative gain is consistent across benchmark datasets, types of LLMs, writing tasks, and domains.

In addition to improved detection performance, GPT-who is computationally inexpensive as it eliminates the need for any LLM fine-tuning. It utilizes a freely available off-the-shelf LM to compute token probabilities, followed by logistic regression using a small set of carefully crafted and theoretically motivated UID features. GPT-who also provides a means to interpret and understand its prediction behaviors due to the rich feature space it learns from. UID-based features enable observable distinctions in the surprisal patterns of texts, which help in understanding GPT-who's decision-making on authorship (Figure 3.1).

We also analyze the UID feature distributions of different LLMs and human-generated texts for the 4 datasets and find that humans distribute information more unevenly and diversely than models. In addition, UID features are reflective of differences in LLM architectures or families such that models that share architectures have similar UID distributions within but not outside their category. We find that UID-based features are a consistent predictor of authorship. Even when there are no glaring differences between uniform and non-uniform text, the differences in UID distributions are easily detectable and a powerful predictor of authorship, since they capture patterns that go beyond the lexical, semantic, or syntactic properties of text. Our work indicates that psycholinguistically-inspired tools can hold their ground in the age of LLMs and a simpler theoretically-motivated approach can outperform complex and expensive uninterpretable black-box approaches for machine text detection.

# 3.2 Related Work

#### 3.2.1 Uniform Information Density (UID)

Shannon's Information Theory states that information exchange is optimized when information travels across the (noisy) channel at a uniform rate [38]. For language production, this uniform rate of information content is the basis of the UID hypothesis that posits that humans prefer to spread information evenly, avoiding sharp and sudden peaks and troughs in the amount of information conveyed per linguistic unit. The information content or "surprisal" of a word is inversely proportional to its probability in a given context. Less predictable words have more surprisal while highly predictable words convey lower information. For example, in the sentence "I enjoy listening to vinyl records", the word "records" is highly predictable from a semantic standpoint given prior words such as "listening" and "vinyl". Thus, given its context, "records" has high predictability, and thus less information content or surprisal according to Information Theory. Formally, Shannon's definition of information content or Surprisal of a component or unit (n) is given by the inverse logarithm of its probability (p(n)) i.e.

$$Surprisal(n) = -\log p(n) \tag{3.1}$$

UID in human language production has been studied by measuring the amount of information content per linguistic unit (number of words) or by studying any sudden changes in surprisal at the onset of a word or sentential element [39,40]. A rich body of work in psycholinguistics has led to the finding that, in language production, humans try to spread information content or surprisal evenly and maintain UID through their lexical, syntactic, phonological, and semantic choices [33–36].

Frank and Jaeger's corpus-based study demonstrated that humans tend to use shorter elements for lower amounts of information and longer elements/sub-sequences for expressing higher amounts of information [33]. Thus, in a way keeping the information rate close to uniform. [34] extended upon this work and reported that UID is consistent at the inter and intra-sentential levels [34, 39]. [40] found that speakers chose not to omit an optional function word at the onset of a less predictable phrase, but that they were more likely to omit the same word at the beginning of a more predictable phrase. [35] and [36] consolidated previous findings that humans regulate their choices as per UID, actively distributing the information that needs to be conveyed evenly across the linguistic signal.

[37] studied the usage of 'less informative' expressions as a means of conveying meanings with higher predictability in a study that directly assessed comprehenders' ease of predicting the referent in an unfolding utterance [37] and found that speakers referred to highly predictable referents with short words. Thus, in language, humans try to spread information content or surprisal evenly and maintain UID through their lexical, syntactic, phonological, and semantic choices.

#### 3.2.2 Machine-Generated Text Detection

Large Language Models (LLMs) such as GPT-3.5, GPT-4 [82], LLaMA [83], Falcon [84], have the capacity to generate human-like-quality texts, which can be easily construed as human-written [85–87]. However, while such LLMs are remarkable, it, therefore, makes them susceptible to malicious use. These include the generation of toxic and harmful content, like misinformation and terrorism recruitment [69, 75, 79]. Due to such potential for misuse, we must develop techniques to distinguish human-written texts from LLM-generated ones to mitigate these risks.

To mitigate this potential for misuse of LLMs, researchers have developed several types of automatic detectors. These techniques include supervised [69, 70, 79, 88–90] and

unsupervised approaches [78,91–94]. These supervised approaches tend to be stylometric, deep learning, and ensemble-based models, while most unsupervised approaches are statistical-based detectors [77,95].

More recently, due to the increased ubiquity of LLMs, we need more interpretable, and less deep learning-based models. Deep learning models have been shown to be the most susceptible to adversarial perturbations than others [76]. To that end, we propose a supervised statistical-based technique, that calculates UID-based features of a given text and uses a classical machine learning model to automatically decide thresholds.

# 3.3 Method: GPT-who

We propose a psycholinguistically-motivated statistical-based machine-generated text detector GPT-who that uses a GPT-based language model to predict who the author of an article is. GPT-who works by exploiting a densely information-rich feature space motivated by the UID principle. UID-based representations are sensitive to intricate "fluctuations" as well as "smoothness" in the text. Specifically, operationalizations of UID are aimed at capturing the evenness or smoothness of the distribution of surprisal per linguistic unit (tokens, words), as stated by the UID principle.

For example, in Figure 3.2, we show sequences of tokens that correspond to the highest and lowest UID score spans within an article. Here, the differences between the two segments of texts might not be obvious at the linguistic level to a reader, but when mapped to their surprisal distributions, the two segments have noticeably distinct surprisal spreads as can be seen by the peaks and troughs i.e. variance of token surprisals along the y-axis about the mean (dotted line). Most approximations of this notion of "smoothness" of information spread and UID, thus, formulate it as the variance of surprisal or as a measure of the difference of surprisals between consecutive linguistic units [45, 46, 61, 96].

In measuring the distribution of surprisal of tokens, UID-based features can capture and amplify subtle information distribution patterns that constitute distinct information profiles of authors. Using just an off-the-shelf language model to calculate UID-based features, GPT-who learns to predict authorship by means of a simple classifier using UID representations. In addition, as these features can be directly mapped to their linguistic



Figure 3.2: An example of UID span feature extraction that selects the most uniform and non-uniform segments from the token surprisal sequence. As can be seen in this example, two texts that read well can have very different underlying information density distributions in a given context. UID features capture these hidden statistical distinctions that are not apparent in their textual form.

token equivalents, GPT-who offers a more interpretable representation of its detection behavior, unlike current black-box statistical detectors, as illustrated in Figure 3.2. The use of a psycholinguistically motivated representation also enables us to better interpret the resulting representation space. It can capture surprisal distributions indicative of and commonly occurring in human-written or machine-generated text. GPT-who is one of the first text detectors that focus on informing a simple classifier with theoretically motivated and intuitive features, as it only requires a fixed-length UID-based representation of length 44 and learns to predict authorship based on just these features, without the need for the full text or any LM fine-tuning in the process (See GPT-who's complete pipeline in Figure 3.3).



Figure 3.3: GPT-who uses token probabilities of articles to extract UID-based features. A classifier then learns to map UID features to different authors, and identify the author of a new unseen article.

#### 3.3.1 UID-based features

We use the 3 most widely used measures of UID scores as defined in previous works [45, 46, 61, 96] as follows: We first obtain the conditional probability p of each token  $(y_t)$  in an article using a pre-trained LM (GPT2-XL). The surprisal (u) of a token  $y_t$  is,

$$u(y_t) = -\log(p(y|y < t)),$$
(3.2)

for  $t \ge 1$  where  $y_0 = \langle BOS \rangle$ , and t = time step.

The lower the probability of a token, the higher its surprisal and vice-versa. Thus, surprisal indicates how unexpected a token is in a given context.

1. Mean Surprisal ( $\mu$ ) of an article (y) of length |y| in number of tokens is defined

as follows:

$$\mu(y) = \frac{1}{|y|} \sum_{t} (u(y_t))$$
(3.3)

2. UID (Variance) score or global UID score of an article (y) is calculated as the normalized variance of the surprisal:

$$UID(y) = \frac{1}{|y|} \sum_{t} (u(y_t) - \mu)^2$$
(3.4)

From this formulation, a perfectly uniform article would have the same surprisal at every token and hence 0 UID (variance) score.

3. UID (*Difference*) score or local UID score of an article (y) is calculated as the average of the difference in surprisals of every two consecutive tokens  $\mu(y_{t-1})$  and  $\mu(y_t)$ :

$$UID(y) = \frac{1}{|y| - 1} \sum_{t=2}^{|y|} abs(\mu(y_t) - \mu(y_{t-1}))$$
(3.5)

4. UID (*Difference*<sup>2</sup>) score is defined as the average of the squared difference in surprisals of every two consecutive tokens  $\mu(y_{t-1})$  and  $\mu(y_t)$ :

$$UID(y) = \frac{1}{|y| - 1} \sum_{n=2}^{|y|} (\mu(y_t) - \mu(y_{t-1}))^2$$
(3.6)

From this formulation, both local measures of UID capture any sudden bursts of unevenness in how information is dispersed in consecutive tokens of the articles.

Maximum and minimum UID spans In addition to previously used approximations of UID, we also craft a new set of features using the most and least uniform segments of an article. Our intuition for this feature is to focus on the extremities of the UID distribution in an article, as the most and least uniform spans would be the most expressive and distinct sequences from a UID perspective. All other spans or segments in an article necessarily lie in between these two extremities. Thus taking account of these two spans would ensure coverage of the whole range of surprisal fluctuations within an article. Thus, for each article, we calculate UID (variance) scores for all spans of consecutive tokens of a fixed length using a sliding window approach. We tuned this

	Random	No Spans	$\operatorname{Span}$	Length (	(N) of M	[in/Max	UID spans
Human v.	UID spans	rio spans	N=4	N=10	N = 15	N=20	N=30
GPT-1	0.75	0.76	0.99	0.99	0.98	1.00	<u>0.99</u>
$GPT-2\_small$	0.62	0.64	0.75	0.82	0.88	0.88	0.85
$GPT-2\_medium$	0.63	0.63	0.73	0.80	0.88	0.87	0.84
GPT-2_large	0.65	0.62	0.73	0.79	0.88	0.88	0.83
GPT-2_xl	0.65	0.61	0.72	0.80	<u>0.88</u>	0.89	0.85
GPT-2_PyTorch	0.55	0.64	0.83	0.84	0.87	0.85	0.86
GPT-3	0.63	0.69	0.71	0.73	0.77	0.84	0.74
GROVER_base	0.63	0.65	0.76	0.77	0.79	0.81	0.78
GROVER_large	0.59	0.60	0.71	0.71	0.73	0.75	0.72
$GROVER\_mega$	0.55	0.56	0.67	0.67	0.68	0.72	0.67
CTRL	0.79	0.83	0.99	<u>0.98</u>	<u>0.98</u>	0.99	0.98
XLM	0.62	0.69	<u>0.96</u>	<u>0.96</u>	<u>0.96</u>	0.99	<u>0.96</u>
XLNET_base	0.62	0.71	0.95	0.97	0.98	<u>0.98</u>	0.99
XLNET_large	0.49	0.70	<u>0.99</u>	<u>0.99</u>	<u>0.99</u>	1.00	0.99
FAIR_wmt19	0.54	0.57	0.74	0.75	0.78	0.74	0.76
Fair_wmt20	0.62	0.63	0.72	0.75	0.88	1.00	0.89
TRANSFO_XL	0.70	0.70	0.79	0.80	0.83	0.79	0.84
PPLM_distil	0.57	0.62	0.92	0.91	<u>0.93</u>	0.95	0.93
PPLM_gpt2	0.54	0.58	0.88	0.88	0.90	<u>0.89</u>	0.88
TuringBench (Avg F1)	0.62	0.65	0.82	0.84	<u>0.87</u>	0.88	0.86
InTheWild (Avg F1)	0.72	0.75	0.79	0.83	0.86	0.88	0.87

Table 3.1: Max. & Min. UID spans ablation study: Setting a span length of N=20 tokens maximized performance (F1 score) across large-scale datasets (N>30 leads to subsequently lower and eventually consistent performance). It can be seen that our min/max features tremendously impact performance against randomly sampled or no span features at all.

window size and found that a window size of 20 tokens per span sufficiently represented an article's UID range. We also experimented with randomly drawn and re-ordered spans and found that random features did not contribute to task performance (see Table 3.1 for ablation study results). We use the surprisal values corresponding to the highest and lowest UID scoring span as additional features and obtain fixed length UID features of length 44 for each article.

We use [57]'s implementation of UID-based scores<sup>1</sup> and use the publicly available off-the-shelf pre-trained GPT2-XL language model<sup>2</sup> to obtain conditional probabilities.

<sup>&</sup>lt;sup>1</sup>https://github.com/rycolab/revisiting-uid/tree/main

<sup>&</sup>lt;sup>2</sup>https://huggingface.co/gpt2-xl



Figure 3.4: Distribution of UID Scores of 20 authors from the TuringBench dataset grouped (dotted line) by architecture type. LMs that share architectures tend to distribute UID scores similarly.

For all our experiments, we calculate the UID features for the publically released train and test splits of all datasets of each of the 4 benchmarks as they were released by the dataset developers. We train a logistic regression model<sup>3</sup> using these features on the train splits and report performance on the test splits. We averaged performance over 3 different random seeds and set the number of maximum iterations hyperparameter to 10k after testing a set of values. We replicate all the original evaluation settings and metrics for each of the datasets (except one setting from the ArguGPT [1] dataset that required access to unreleased human evaluation data). We do this to be able to directly compare the performance of GPT-who with current state-of-the-art detection methods reported so far.

#### Datasets

To test the applicability of GPT-who across text detection tasks, we run all experiments across 4 large-scale and recent datasets that span over 15 domains and 35 recent LMs.

**TuringBench Benchmark** [79] dataset is the largest multi-class authorship attribution dataset that contains over 168k news articles generated by 19 neural text generators using 10K prompts from CNN and the Washington Post.

<sup>&</sup>lt;sup>3</sup>https://scikit-learn.org/stable/

**GPABenchmark** [80] or <u>GPT</u> Corpus for <u>A</u>cademia is a multi-domain (Computer Science (CS), Humanities and Social Sciences (HSS) and Physics (PHX)) academic articles dataset aimed at helping detection of LLM use or misuse in academic writing. It contains 150k human and 450k ChatGPT-generated articles for 3 task settings (completion, writing, and polishing).

**ArguGPT** [1] is a prompt-balanced dataset of argumentative essays containing over 4k human-written essays and 4k articles generated by 7 recent LLMs (including many variants of ChatGPT) using prompts from English datasets such as TOEFL11 [97] and WECCL [98] datasets.

"InTheWild" Deepfake Text Detection in the Wild [81] dataset is, to our knowledge, the largest text detection dataset consisting of over 447k human-written and machinegenerated texts from 10 tasks such as story generation, news article writing, and academic writing. They use 27 recent LLMs such as GPT-3.5, FLAN-T5, and LLaMA. We refer to this dataset as the "InTheWild" dataset going forward for brevity.

#### 3.3.2 Baselines & Detectors

We compare our proposed method against the following: DetectGPT <sup>4</sup> [78], GLTR<sup>5</sup> [91], an open-source implementation<sup>6</sup> of GPTZero [99], ZeroGPT [100], OpenAI's detector [101], [81]'s LongFormer-based detector<sup>7</sup> tuned for the InTheWild benchmark (we refer to this method as "ITW"), a stylometric detector<sup>8</sup> [102] and fine-tuned BERT<sup>9</sup> [103]. We are unable to report results for exhaustively all methods across all datasets due to inherent inapplicability in certain task settings. For example, most SOTA text detectors cannot be applied to the ArguGPT dataset as it only contains text written by multiple machines, while most text detectors are designed to differentiate between human-written and machine-generated texts. Beyond such limitations, we have utilized all applicable methods for 4 benchmark datasets.

<sup>&</sup>lt;sup>4</sup>https://github.com/eric-mitchell/detect-gpt

<sup>&</sup>lt;sup>5</sup>https://github.com/HendrikStrobelt/detecting-fake-text

<sup>&</sup>lt;sup>6</sup>https://github.com/BurhanUlTayyab/GPTZero

<sup>&</sup>lt;sup>7</sup>https://github.com/yafuly/DeepfakeTextDetect

<sup>&</sup>lt;sup>8</sup>https://github.com/shaoormunir/writeprints

 $<sup>^9</sup>https://huggingface.co/docs/transformers/training$ 

# 3.4 Results

#### 3.4.1 UID Signatures of Authors

Given that humans tend to optimize UID, we study if different models spread surprisal in ways that are distinguishable from each other and human-written text and if we can observe unique UID signatures of different LM families. To this end, we plot the UID score distributions of different text generators across (see Figures 3.4, 3.5(a), and 3.5(b)). We observe that, generally, the UID scores of human-written text have a higher mean and larger standard deviation than most machine-written text across writing task types, domains, and datasets. This implies that human-written text tends to be more non-uniform and diverse in comparison to machine-generated text. Hence, machines seem to be spreading information more evenly or smoothly than humans who are more likely to have fluctuations in their surprisal distributions.

Going a step further, if we compare models to other models, we see that models that belong to the same LM family by architecture tend to follow similar UID distribution. For example, in Figure 3.4, the dotted lines separate LMs by their architecture type and it can be seen, for example, that all GPT-2 based models have similar UID distributions, all Grover-based models have similarities, but these groups are distinct from each other. This indicates that UID-based features can capture differences in text generated by different LM families. To our knowledge, this is the first large-scale UID-based analysis of recent machine and human-generated text across writing tasks and domains.

We see that for most cases, humans have a higher UID (variance) score than machines, as can be seen by the higher means of their scores in the box plots. This holds when comparing human-written texts with multiple machine-generated texts over shared tasks (Figure 3.5(a)) and also when comparing their differences between tasks (Figure 3.5(b)).

#### 3.4.2 Machine Text Detection Performance

Overall, GPT-who outperforms other statistical-based detectors and is at par with transformers-based fine-tuned methods for 2 out of 4 benchmarks. For GPABenchmark (Table 3.2), across all task types and domains, GPT-who outperforms GPTZero, ZeroGPT, DetectGPT and, OpenAI's detector by over 40%. The machine-generated texts for this task are from 7 very recent and highly sophisticated LLMs (including GPT3.5, GPT3)



(a) Pairwise comparisons of human and different machine-generated texts for shared tasks: Distribution of UID Scores of 8 authors (7 models + human) from the InTheWild dataset. (m) indicates machine and (h) indicates human written texts. This is followed by the model name along the x-axis labels to indicate the different authors.



(b) Pairwise comparisons of human and different machine-generated texts for different tasks: Distribution of UID Scores of humans v.s. machines per task type. (m) indicates machine and (h) indicates human written texts. This is followed by the writing task type along the x-axis labels to indicate the different tasks.

Figure 3.5: Human and machine-text pairwise comparisons: Comparing UID Score (Variance) distributions

		Task 1	
Method	$\operatorname{CS}$	PHX	HSS
GPTZero	0.30	0.25	0.72
ZeroGPT	0.67	0.68	0.92
OpenAI Detector	0.81	0.70	0.63
DetectGPT	0.58	0.54	0.57
BERT	0.99	0.99	0.99
ITW	<u>0.98</u>	<u>0.98</u>	0.96
GPT-who	0.99	<u>0.98</u>	<u>0.98</u>
		Task 2	
Method	$\operatorname{CS}$	PHX	HSS
GPTZero	0.17	0.06	0.44
ZeroGPT	0.25	0.10	0.62
OpenAI Detector	0.64	0.24	0.27
DetectGPT	0.16	0.17	0.20
BERT	0.99	0.96	0.97
ITW	0.81	0.76	0.29
GPT-who	<u>0.84</u>	<u>0.90</u>	<u>0.80</u>
		Task 3	
Method	$\operatorname{CS}$	PHX	HSS
GPTZero	0.02	0.02	0.20
ZeroGPT	0.03	0.03	0.25
OpenAI Detector	0.06	0.04	0.06
DetectGPT	0.03	0.05	0.06
BERT	0.97	0.97	0.99
ITW	0.38	0.31	0.08
GPT-who	0.63	0.75	0.62

Table 3.2: Test Set Performance (F1 Scores) for Task 3 across different domains.

variants), making the detection of machine-generated text a much more challenging task on which GPT-who outperforms other detectors.

For **TuringBench** (Tables 3.3 and 3.4), **GPT-who** significantly outperforms GLTR by **0.32 F1** points, and at par with BERT fine-tuned for the task. The **InTheWild** dataset contains 6 testbeds with varying levels of detection difficulties, such as out-of-domain, out-of-distribution, and unseen-task test sets. We used all 6 testbeds to analyze the

Human v.	GROVER	GLTR	GPTZero	DetectGPT	GPT-who
GPT-1	0.58	0.47	0.47	0.51	1.00
$GPT-2\_small$	0.57	0.51	0.51	0.51	0.88
$GPT-2\_medium$	0.56	0.49	0.50	0.52	0.87
GPT-2_large	0.55	0.46	0.49	0.51	0.88
GPT-2_xl	0.55	0.45	0.51	0.51	0.89
GPT-2_PyTorch	0.57	0.72	0.50	0.52	0.85
GPT-3	0.57	0.35	0.47	0.52	0.84
GROVER_base	0.58	0.39	0.52	0.51	0.81
GROVER_large	0.54	0.41	0.47	0.52	0.75
$GROVER\_mega$	0.51	0.42	0.42	0.51	0.72
CTRL	0.49	0.88	0.67	0.67	0.99
XLM	0.50	0.89	0.67	0.67	0.99
$XLNET\_base$	0.58	0.75	0.51	0.67	0.98
XLNET_large	0.58	0.88	0.67	0.52	1.00
$FAIR\_wmt19$	0.56	0.56	0.56	0.51	0.74
$Fair\_wmt20$	0.58	0.49	0.50	0.51	1.00
TRANSFO_XL	0.58	0.35	0.49	0.52	0.79
PPLM_distil	0.59	0.64	0.52	0.67	0.95
$PPLM_gpt2$	0.58	0.68	0.51	0.51	0.89
Average F1	0.56	0.57	0.52	0.55	0.88

Table 3.3: Test Set Performance (F1 score) for TuringBench dataset.

performance of GPT-who in detecting machine-generated texts across increasing levels of 'wildness' and find that overall, GPT-who outperforms all other methods except the one specifically tuned to the task (ITW) across all testbeds. More importantly, GPT-who performs well even for the most challenging or 'wildest' testbed settings of unseen model and unseen domain distributions (see Table 3.5).

For the **ArguGPT** dataset (Table 3.6), we find that **GPT-who** outperforms human experts and stylometry in predicting authorship by **0.31 F1** points, but is outperformed by fine-tuned BERT. Although unable to perform as well as BERT, **GPT-who** is one of the only statistical-based detectors that can handle distinctions between machine-only texts. We were unable to evaluate other detectors as their human-generated texts were not publicly released, and they only work in human v/s machine settings.

GPT-who is a statistical-based approach that outperforms other statistical-based

Human v.	RoBERTa	BERT	ITW	Stylometry	GPT-who
GPT-1	0.98	0.95	0.92	0.99	1.00
$GPT-2\_small$	0.71	0.75	0.47	0.75	0.88
$GPT-2_medium$	0.75	0.65	0.47	0.72	0.87
GPT-2_large	0.79	0.73	0.46	0.72	0.88
GPT-2_xl	0.78	0.79	0.45	0.73	0.89
GPT-2_PyTorch	0.84	0.99	0.47	0.83	0.85
GPT-3	0.52	0.79	0.48	0.72	0.84
GROVER_base	0.99	<u>0.98</u>	0.49	0.76	0.81
GROVER_large	0.99	0.98	0.52	0.71	0.75
GROVER_mega	0.94	0.97	0.53	0.68	0.72
CTRL	1.00	1.00	0.91	0.99	0.99
XLM	0.58	1.00	0.92	0.96	0.99
XLNET_base	0.79	0.99	0.84	0.95	0.98
XLNET_large	1.00	1.00	<u>0.93</u>	1.00	1.00
$FAIR\_wmt19$	0.84	0.93	0.49	0.74	0.74
$Fair\_wmt20$	0.45	0.47	0.47	0.73	1.00
TRANSFO_XL	<u>0.96</u>	0.97	0.81	0.79	0.79
PPLM_distil	0.90	0.88	0.51	0.92	0.95
PPLM_gpt2	0.90	<u>0.89</u>	0.49	0.88	<u>0.89</u>
Average F1	0.88	0.61	0.88	0.82	0.88

Table 3.4: Test Set Performance (F1 score) for TuringBench dataset.

Setting	Testbed Type	GPTZero	GLTR	DetectGPT	BERT	ITW	GPT-who
	DS-MS	0.65	0.94	0.92	0.98	0.97	0.93
ID	CD-MS	0.63	0.84	0.6	0.98	0.97	0.88
ID	DS-CM	0.57	0.8	0.57	0.49	0.87	0.86
	CD-CM	0.57	0.74	0.57	0.49	<u>0.78</u>	0.86
OD	UM	0.58	0.65	0.6	0.84	<u>0.79</u>	0.74
OD	UD	0.57	0.72	0.57	0.68	0.8	0.77
	Average F1	0.60	0.78	0.64	0.74	0.86	<u>0.84</u>

Table 3.5: Test Set Performance (F1 score) for InTheWild dataset. ITW refers to the LongFormer-based detector trained by Li et al. (2023) specifically for this benchmark. Acronyms used in this table are as follows: ID = In-distribution, OD = Out-of-distribution, DS-MS = Domain-specific Model-specific, CD-MS = Cross-domains Model-specific, DS-CM = Domain-specific Cross-models, CD-CM = Cross-domains Cross-models, UM = Unseen Models, UD = Unseen Domains

Author	$\mathbf{Experts}^*$	Stylometry	BERT	GPT-who
text-babbage-001	0.47	0.45	<u>0.84</u>	0.85
text-curie-001	0.47	0.45	0.83	0.84
text-davinci-003	0.66	0.59	0.95	0.77
gpt-3.5-turbo	0.63	0.69	0.96	0.84
gpt2-xl	0.37	0.49	0.95	<u>0.91</u>
Average F1	0.52	0.53	0.91	0.84

Table 3.6: Test Set Performance (F1 score) for ArguGPT dataset.<sup>\*</sup> denotes results reported in [1].

approaches but is unsurprisingly outperformed by fine-tuned methods in 2 out of 4 benchmarks. In the case of statistical-based methods, it is typically very hard to come close to fine-tuned performance as such methods rely only on derived properties of the text and do not utilize the full raw text in training as is the case in the latter [104]. Despite this, GPT-who can exceed fine-tuned LM performance by 10% for 2 benchmarks.

#### 3.4.3 Running Time

We measured the time taken for the one-time training or fine-tuning and inference for 6 testbeds from the InTheWild Dataset (the largest of all our benchmarks). We compare the average running times of DetectGPT, BERT, GPT-who and a stylometric detector in Table 3.7 and find that GPT-who is the fastest as it eliminates the need for any LM fine-tuning and makes a single inference call per text sample. Other methods require either LM fine-tuning or multiple inference calls (for example, DetectGPT). This computational load is greater than a single forward inference pass through one LM (GPT2) followed by logistic regression which is what GPT-who requires.

Method	<b>One-Time Training</b>	Inference
DetectGPT	>10 hours	$60  \sec$
BERT	$\sim 1.5$ hours	$2  \sec$
Stylometry	$\sim 1.5$ hours	$2 \sec$
GPT-who	$20 \min$	$0.8  \sec$

Table 3.7: Average Running time over 6 testbeds from the InTheWild dataset.

# 3.5 Conclusion

The Uniform Information Density (UID) principle posits that humans prefer to spread information evenly during language production. We examine if this UID principle can help capture differences between Large Language Models (LLMs)-generated and human-generated texts. We propose GPT-who, the first psycholinguistically-inspired domain-agnostic statistical detector. This detector employs UID-based features to model the unique statistical signature of each LLM and human author for accurate detection. We evaluate our method using 4 large-scale benchmark datasets and find that GPT-who outperforms state-of-the-art detectors (both statistical- & non-statistical) such as GLTR, GPTZero, DetectGPT, OpenAI detector, and ZeroGPT by over 20% across domains. In addition to better performance, it is computationally inexpensive and utilizes an interpretable representation of text articles. We find that GPT-who can distinguish texts generated by very sophisticated LLMs, even when the overlying text is indiscernible. UID-based measures for all datasets and code are available at https://github.com/saranya-venkatraman/gpt-who. We present the largest analysis of the UID-based representations of human and machine-generated texts (over 400k articles) to demonstrate how authors distribute information differently, and in ways that enable their detection using an off-the-shelf LM without any fine-tuning.

# 3.6 Discussion

We propose GPT-who, a statistical-based machine-generated text detector that utilizes features inspired by the Uniform Information Density (UID) principle rooted in the observation that humans tend to evenly distribute information or surprisal in language.We leverage UID-based features, including variance of surprisals and minimally/maximally uniform sub-sequences extracted from the surprisal sequence generated by an off-the-shelf LM. We demonstrate that these features are highly effective in discerning machinegenerated text from human-generated text as they capture nuances in how models and humans distribute surprisal in their texts. Our findings have implications for enhanced text authenticity assessment.

We turn to the UID principle, which states that *humans prefer to spread information* evenly in language, to automatically extract features that measure the spread and flow of information content or surprisal in texts. Our UID-based features are formulated to capture how surprisal is distributed in an article as they measure the local and global variance, mean, and most uniform and non-uniform segments of a text. This rich and succinct representation space drives the predictive capability of our proposed detector and the interpretability of its representations. Analysis of this feature space reveals that human-written text tends to be more non-uniform in comparison to machine-generated text. Hence, machines seem to be spreading information more evenly or smoothly than humans who are more likely to have fluctuations in their surprisal distributions.

However, this finding does not imply that humans are not producing uniform text. It is important to note that our work cannot provide support for or refute the UID hypothesis which comes from psycholinguistic studies such as those in Section 3.2.1. Our work shows that, given our operationalization of UID based on prior works, machine text is relatively more uniform than human-written text. While this might seem contradictory to UID theory, it does not still disprove that humans are uniform in their language production.

We conjecture that this unexpected finding is because we use GPT-2's probability distribution to calculate surprisal, which is potentially a poor approximation of the "human" probability distribution. A closer-to-human probability distribution might (or might not) show humans to be more uniform than machines, though this determination is not within the scope of this work. It is crucial to note that uniformity is relative, and while machines are more uniform under this operationalization, it would still be true that human text is uniform as per the human's probability distribution (that we do not have access to and can only approximate using some LM distribution, for example, GPT2-XL in our case). UID theory does not make any predictions on where machine-generated text might lie in the uniform to non-uniform spectrum but only indicates that humans are arranging utterances evenly as per their own language distribution.

Irrespective of its alignment with what theory suggests, we find that the UID-based features are very useful in distinguishing authors, which is the focus of this work. This is an important consideration and helps disentangle the utility of UID-inspired features from the cognitive plausibility of those feature calculations or UID approximations. Thus, this operationalization of UID does not imply that humans are "less human" or machines are "more human" as it is an approximation of a theory that states that humans are uniform as per their language distribution. It does not have any further implications for machine-generated text and is unable to predict what happens in the case of machines.

We find that UID-based features can capture differences between text generated by not only humans and models but also capture differences between multiple models and LM families. Our main contribution is a psycholinguistically-inspired domain-agnostic statistical-based machine-generated text detector, GPT-who, that:

- Outperforms statistical approaches across 4 large-scale benchmark datasets that include texts from over 35 LLMs across more than 10 domains.
- Generalizes better to out-of-distribution datasets than SOTA detectors.
- Computationally more efficient than other supervised detectors as it does not require the fine-tuning or training of any LLMs.
- Interpretable due to its psycholinguistically motivated UID-based feature space.

While our detector may not significantly outperform fine-tuned transformers-based models, it is essential to highlight its independence from fine-tuning, offering nearly comparable performance at significantly lower computational costs and remains one of the only statistical-based detectors that can operate in multi-author settings beyond the Turing Test. These findings indicate that approaches rooted in psycholinguistic theories that delineate indicators of "human-like" language use hold enormous and untapped potential in tackling the fast catapulting and ever-changing LLM landscape. This work has implications for cognitively plausible and explainable solutions to complex challenges arising from ever-growing automated text generators.

In our pursuit of a comprehensive examination of texts produced by recent large language models, we encountered limitations arising from resource constraints and the availability of publicly accessible datasets. These factors constrained our ability to encompass a more diverse array of models and tasks, including summarization and question-answering. Furthermore, our study did not delve into whether UID-based methods extend their utility beyond detecting machine-generated text to identify potential issues such as misinformation and plagiarism. We acknowledge these constraints as part of our ongoing commitment to refining and expanding our efforts in future research endeavors. It is important to note that there are inherent limitations of AI-based tools and automated machine text detectors such as in this work. Acknowledging the fallibility of these detectors, particularly in generating false positives, we note that there is still a crucial need for human oversight and discretion in the usage of such detectors in realworld settings. For example, ethical concerns surrounding over-vigilance in scrutinizing student-written text are an important consideration for striking a balance between the convenience of automated detection and the preservation of academic integrity. By advocating for responsible development and implementation, we hope to contribute to a landscape where ethical considerations guide the integration of automatic text detection systems in educational settings, safeguarding against undue reliance and promoting fairness, equity, and respect for individual expression.

# Chapter 4 Authorship Obfuscation using syntactic n-gram patterns

Authorship Attribution (AA) and Authorship Obfuscation (AO) are two opposing tasks of growing significance in privacy research. Modern AA techniques utilize an author's distinct writing style to identify the author of a given text through an AA classifier. In contrast, AO aims to alter a text in such a way that its meaning remains intact, but an AA model is unable to accurately determine its authorship. While new AO methods have been introduced to counter the privacy risks posed by advanced AA models, they often remain impractical due to slow training and obfuscation times, sometimes taking several hours. To address this, we present ALISON, a practical AO method that (1) significantly reduces training and obfuscation time, offering over 10x faster performance than current AO approaches, (2) delivers superior obfuscation results, improving by 15%over competing methods when attacking three transformer-based AA models across two benchmark datasets, (3) does not rely on direct feedback from the target AA classifier during obfuscation, and (4) incorporates unique stylometric features for explainable and interpretable obfuscation. Additionally, ALISON proves effective in preventing four advanced AA models from identifying the authorship of texts generated by ChatGPT, while making minimal changes to the original text's meaning.

# 4.1 Introduction

Writing styles tend to be consistent across texts written by the same author, while they can vary significantly between different authors. As a result, the identity of an anonymous author can often be uncovered by examining the writing style and matching it to a set of known authorship indicators, a process called Authorship Attribution (AA). In machine



Figure 4.1: An example of ALISON successfully obfuscating a text by changing its style while preserving semantics.

learning, these authorship indicators are predictive signals that distinguish one author's writing style from others, commonly referred to as stylometric features. Stylometric features encompass various types, including lexical features (e.g., word structure and frequency of specific character sequences), syntactic features (e.g., part-of-speech patterns and the use of functional words and punctuation), and content features (e.g., the semantics of words and phrases within the text). These features are crafted to enable a machine learning model to assign a text to the correct author. These engineered features, such as Writeprints [105], often include not one but several interpretable signals such as word and character bigrams, word length distributions, or special character frequencies to improve the classification accuracy.

However, recent AA techniques [106, 107] utilize complex transformer models—e.g., BERT [108], RoBERTa [109], BertAA [106], to automatically learn useful features for AA from raw text. This removes the need to rely on explicitly engineered stylometric features. While these models are more computationally expensive to train and notorious for their lack of interpretability, they significantly outperform traditional AA classifiers [106].

As Authorship Attribution (AA) techniques become increasingly accurate and efficient, they pose a growing threat when misused by malicious actors to uncover the identities behind anonymous texts. This is particularly harmful to vulnerable groups such as NGO activists, whistleblowers, and journalists. Given the strength of current state-of-the-art (SOTA) transformer-based AA models, it is crucial to develop methods that mitigate the risk of revealing the true authorship of anonymous texts. In response, we focus on
the reverse task of AA, known as Authorship Obfuscation (AO), which seeks to defeat authorship attribution classifiers by systematically altering a text. Effective AO misleads the classifier, causing it to incorrectly attribute the text's authorship from a pool of candidates. Since AA performance generally diminishes when the number of potential authors becomes large (i.e., more than 100), and adversaries can often narrow the pool of authors to a small, finite set, we focus on authorship obfuscation in this constrained scenario rather than an open-world setting. Figure 4.1 illustrates a successful example of obfuscation against a BERT-based AA model [107].

For AO to be practical, it should meet three key criteria: (1) the ability to function with minimal knowledge of the adversary, (2) rapid execution time for long-form texts (under one second, as opposed to minutes or hours), and (3) intuitive interpretability to ensure a trustworthy obfuscation process.

First, an ideal AO approach is one that can operate successfully without significant knowledge of the adversary and its perturbation results must be able to transfer across different AA models. Methods utilizing trial and error assume that the attacked AA classifier is foreknown and accessible via callable APIs. However, such assumption is often not satisfied in real life. Moreover, there are different possible variations in the training schemes and architectures of current SOTA transformer-based text classifiers that make the obfuscation task much more challenging in practice. Although these classifiers might differ from each other in their architectures, it is an increasingly important task to develop powerful and practical authorship obfuscation methods to protect the individuals who wish to publish text-based content anonymously, such as activists or whistleblowers. Because of the varied natures and training schemes adopted by transformer architectures, an optimal authorship obfuscation method must be able to operate successfully without significant knowledge of the adversary and must be able to potentially transfer success to many adversaries.

A short running time will ensure that a higher volume of texts can be obfuscated in a shorter timeframe so as not to diminish the productivity of an author. When computational complexity becomes especially prohibitive, the identity of an author may become at risk of being revealed. However, current methods [110, 111] utilize large and slow models to perform obfuscation, resulting in a sub-optimal runtime. An authorship obfuscation method must be practical for longer texts and maintain a short running time for obfuscation to make it accessible for individuals without significant computational resources but who still wish to publish in a way that preserves privacy. A short running time will ensure that a higher volume of texts can be obfuscated in a shorter timeframe so as not to diminish the productivity of an author. When computational complexity becomes especially prohibitive, the identity of an author may become at risk of being revealed. However, current methods [110,111] utilize large and slow models to perform obfuscation, resulting in a sub-optimal runtime.

Third, an ideal AO technique should be interpretable. When the reasoning behind obfuscation decisions is unclear, it is difficult to communicate to the users that the obfuscation is performed in a trustworthy and sound manner. Unfortunately, state-of-the-art (SOTA) AO methods fail to meet these practical requirements. They typically involve lengthy running times to obfuscate text in a black-box manner, relying on repeated calls to the target model. These approaches are impractical for several reasons: obtaining a black-box understanding of the attacked model is often unrealistic, the long execution times significantly hinder an author's ability to maintain anonymity efficiently, and the lack of interpretability during the obfuscation process undermines trust in these methods.

By being enabled to interpret the decisions made by the AO method, the users also gain insight into important stylometric features of specific authors and how they are interchanged for obfuscation. However, current AO methods–e.g., [110,111], lack such interpretability. This is due to the fact that they employ genetic-based search algorithms to optimize the text perturbation process, which are often very challenging to summarize with clear interpretability.

To address the aforementioned limitations of current AO methods, we propose a novel stylometry-grounded novel obfuscation method, ALISON: ( $F\underline{a}st \ Sty\underline{l}ometr\underline{i}c \ Author\underline{s}hip$ <u>Obfuscation</u>), which overcomes these challenges as follows:

- ALISON significantly reduces obfuscation runtime by over 10x while maintaining superior semantic preservation during the obfuscation process.
- ALISON consistently outperforms competing methods, achieving about 15% higher success rates in obfuscation (i.e., lowering the prediction F1-score of state-of-the-art transformer-based AA classifiers).
- ALISON also provides interpretable explanations for its obfuscation outcomes

through the use of stylometric features, offering insights into the obfuscation process.



Figure 4.2: ALISON: Our proposed obfuscation pipeline.

Performing adversarial attacks against text models is a highly studied problem. Attack strategies can be broadly classified into operating in three settings: white-box, black-box, and blind. Attacks operating in a white-box setting have complete knowledge of the defender's model, including its architecture, training data, and the ability to query the model parameters and its prediction logits. In the black-box attack setting, the model's architecture, parameters, and training data are unknown, but the attacker has the ability to query the black-box model for prediction probabilities. In the blind attack setting, the attacker has no knowledge of the defender. In this work, we will only conduct attacks in the blind attack setting to systematically evaluate the performance of each method in a consistent manner.

#### 4.1.1 White-Box Attacks

It has been demonstrated that white-box attacks are extremely effective [112, 113], demonstrating near complete degradation of adversarial accuracy while retaining almost all of the original text. These methods access the target model's gradients and compute the most optimal perturbations to most negatively degrade the target model's logits. Gradient-based attacks are able to perform so well because their complete knowledge allows them to definitively identify the parts of the text contributing most to the final prediction, and are therefore able to minimally change the text to achieve successful obfuscation.

#### 4.1.2 Black-Box Attacks

While the black-box attack setting is intuitively more difficult to attack in, SOTA blackbox attacks are able to find a nearly comparable success rate. For example, TextFooler [108] leverages the black-box ability to query logits from the adversarial classifier to find the most important words in the classifier's prediction formation, through finding the maximum confidence decrease when a word is removed. These important words are then replaced to achieve a high obfuscation success rate. Because TextFooler is a widely accepted textual attack method in the black-box setting, we repurposed it for blind use by using it to attack maintained internal classifiers different from the hidden target classifiers.

BERT-Attack [114] utilizes a similar approach to attack BERT [107] models in the black-box setting. To rank words, they measure the logit changes as words are replaced with [MASK], and utilize BERT to replace words identified as important. The concept of these attack strategies was improved upon using locality-sensitive hashing to group potential replacements for candidate words, reducing the number of queries needed to perform a successful attack while sacrificing minimal performance [115].

The blind attack setting is the most difficult to perform. However, it is the most practical because information about the target may be difficult or impossible to obtain in practice. General textual adversarial attack strategies may be used to attack AO classifiers in their respective settings, however, additional methods exist for AO specifically. We describe a few works in this setting specifically pertaining to the blind attack setting.

We focus our work on the blind AO setting, where the adversarial attack on the AA classifier operates under two key constraints: (1) the attacker cannot directly query the AA classifier, and (2) the attacker has no access to its architecture, training data, or other internal details. These restrictions make the AO task more challenging but also more realistic compared to existing threat models in the literature, which often assume access to a public API for the target AA classifier. The following section reviews relevant work within this blind AO context.

Mutant-X [110] is an automated obfuscation method that employs genetic algorithms to iteratively perform single-word substitutions, guided by the confidence degradation obtained from a black-box perspective of the target model. Although originally designed as a black-box attack, we adapted it for use in the blind attack setting, following the transferability studies outlined in its initial work [110]. Avengers Ensemble [111] seeks to enhance Mutant-X by reducing its dependence on black-box knowledge of the target classifier. It achieves this by leveraging an ensemble-based internal classifier, which improves the method's transferability across a range of adversaries, leading to better performance in the blind attack scenario. For the remainder of this paper, we refer to this method simply as Avengers.

Rule-based obfuscation utilizes a set of specific rules to match and change specific patterns of text, however, these methods have been found to be too conservative to obfuscate against most classifiers [116]. Anonymouth [117] is a popular semi-automated obfuscation technique, meaning that it requires continuous human input to perform obfuscation, and therefore its performance is dependent on the user. We utilized the recommended internal classifiers for Mutant-X, however, we did not use these internal classifiers as target models, due to the low attribution accuracy of the classical machine-learning methods.

Other popular greedy-based black-box methods in the NLP adversarial literature, such as TextFooler [108] and BERT-Attack [118], often have a high degree of dependence on the accessibility to the target AA classifier they attack. These methods make queries to the victim model per token in order to obtain a logit-based ranking of word importance. Then, top tokens may be replaced with close neighbors in precomputed embedding spaces [108] or by leveraging token representations of large language models [118]. However, these methods often demonstrate a sharp decline in performance once the attacks are transferred to different target classifiers [108]. Additionally, such methods generally lack interpretability, as model explanations are based solely on the black-box model that is being attacked instead of revealing identifying linguistic patterns.

Lastly, large generative language models, such as ChatGPT [119], have demonstrated impressive paraphrasing capability which may be suitable for AO applications. A user may obtain a stylometrically different but semantically consistent text by prepending a fixed paraphrasing prompt to query a language model. Therefore, these model explanations have limited value beyond the black-box setting.

Because of the complexity of genetic algorithms, Mutant-X and Avengers are computationally very expensive, taking a relatively long amount of time for inference, or producing obfuscated samples. Additionally, they lack interpretability, as they make word substitutions according to the learned patterns of the genetic algorithm, which are not readily apparent. For these reasons, these methods are not practical to deploy in a real-world scenario for privacy enhancement. We aim to address these limitations with ALISON, comparing ALISON with the SOTA methods TextFooler, Mutant-X and Avengers.

# 4.2 Proposed Method: ALISON

# 4.2.1 Problem Formulation

Given a text corpus  $\mathcal{X}$ , we define an AA classifier f trained on  $\mathcal{X}$ , such that for arbitrary text  $x \in \mathcal{X}$ , f(x) attributes the authorship of x. Given  $\mathcal{T}$  is a set of texts to obfuscate, our objective is to thwart f for any text  $t \in \mathcal{T}$  by transforming t into t' such that  $f(t) \neq f(t')$ . We assume that  $\mathcal{X}$  and  $\mathcal{T}$  share the same pool of potential authors and are in a similar domain–e.g., news articles, blog posts– but do not contain any identical texts.

Moreover, we also assume no access to  $\mathcal{X}$  by the adversary. However, they do have access to another non-overlapping corpus  $\mathcal{X}^*$  with a similar size containing the same pool of authors and domain with  $\mathcal{X}$ . Such assumption is reasonable in practice, especially when online social networks have made it very convenient for anyone to access text content generated by millions of people worldwide. To evaluate our approach in this setting, we split each publicly available text classification corpus into three disjoint sets,  $\mathcal{X}, \mathcal{X}^*$ , and  $\mathcal{T}$  stratified by unique authorship labels. ALISON is designed to reduce computational complexity while advancing obfuscation success and semantic preservation during obfuscation. To do this, we employ three overarching strategies. First, we train an internal, lightweight AA classifier *once* that uses intuitive linguistic properties of part-of-speech (POS) sequences to guide the obfuscation process. Second, we aim to obfuscate a phrase of multiple words at a time instead of perturbing token by token. Third, we leverage an advanced pre-trained language model (PLM) to generate the replacement token sequence that best fits the sentence context and semantics without making queries to an embedding space. We present the entire obfuscation pipeline in Figure 4.2 and discuss each step and component in the following sections.

#### 4.2.2 One-Time Stylistic Internal AA Classifier Training

Because blind attacks on AA models often rely on an internal approximation of an arbitrary adversarial classifier to choose candidate words or phrases to be replaced, tuning the internal classifier for maximal transferability to other target classifiers is integral to producing high obfuscation success rate [111]. Therefore, we augment the traditional



Figure 4.3: An example of extracting POS trigrams.

internal classifier feature space of character n-grams with POS n-grams, features we believe to be more heavily rooted in true style. We hypothesize that while writing style encompasses word and character frequencies, more generally, writing style also encompasses frequencies of individual POS tags and their collocations. Intuitively, POS and sequences of several POS tags capture writing style because they do not describe the content of the text but rather how the ideas in the text are synthesized. Generally, an author's texts should contain similar POS sequence patterns, as they represent common textual structures used to synthesize different ideas.

**Data Extraction.** We first extract the POS tags of all texts in the corpus  $\mathcal{X}^*$ . Next, we extract character and POS tag n-grams of various lengths as features for training the internal classifier. Figure 4.3 demonstrates the procedure of extracting POS n-grams from a sample sentence with  $n \leftarrow 3$ .

An n-gram is a contiguous sequence of n linguistic units (e.g., characters, words, POS tags) within a text. Given a set of sequence lengths V, for each length  $l \in V$ , we extract all character and POS level l-grams over the entire training corpus and collect the L most frequent character and POS l-grams. The normalized frequencies of these L most frequent character and POS l-grams for each length  $l \in V$  are concatenated to form the stylistic representations of the text.

We, therefore, hypothesize that POS n-grams will be a more generally transferable feature to base 0.15scation technique on, as any highly-performing attribution model must rely on stylistic features, addressing objectives 2 and 4. Additionally, by replacing phrases governed by the POS n-grams, we reduce the computational complexity by replacing phrases instead of words, addressing objective 1.

A typical neural network is often represented by  $h_{\Theta}$ , where  $\Theta$  is a list of tunable parameters or weights ( $\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$ ). In an ideal case, the feature space of a given task is embedded in  $\Theta$  after the parameters are fully optimized. Cross-entropy loss was applied to guide neural network learning by providing feedback during the training.

$$\mathcal{L}_{CE} = -\sum_{i=1}^{n} y_i * \log(p_i), \qquad (4.1)$$

where  $y_i$  is the value of the ground-truth at label *i* and  $p_i$  is the predicted probability of label *i*. The loss is minimized by adjusting  $\Theta$  to push the probability of the target label to 1. The goal of neural network training is to converge to an optimal set of weights for the model that outputs the desired result on a given task, upon which training is terminated. The weights are iteratively updated during the training from an initially randomized set.

#### 4.2.3 Model Architecture

With V = 1, 2, 3, 4 and L = 250, for a sufficiently large corpus, the size of the feature set is generally at least 1500. Therefore, we utilized a NN with all hidden layers excluding the last having 1750 neurons, and the last hidden layer having 256 neurons. The NN model consisted of 12 total layers. Additional layers marginally improved the accuracy of the model while increasing training time. We applied Dropout to each layer except the last, at a Dropout proportion of p = 0.40. We empirically determined that a high Dropout proportion decreased dependence on any one feature and increased the obfuscation success. Dropout also serves as a form of regularization, preventing model overfitting on the training set.

# 4.2.4 Internal Classifier Training

The resulting vector representations are then used to train a fully connected neural network (NN) model on the authorship attribution task. We opt for a simple NN due to its computational efficiency without much compromise on generalization. To utilize this model for prioritizing which phases or words in a sentence to perturb first, we then extract a list of features, ranked by importance, for  $\forall t \in T$  using Integrated Gradients [120], a model interpretability algorithm that assigns an importance score to each input feature by approximating the integral of the gradients with respect to the input by making a small number of calls to the gradient of the model. The method can be applied to any differentiable model and can be used to understand feature importance by extracting rules from the model.

Integrated Gradients is a widely used SOTA method for input sensitivity analysis and feature-attribution explanation. Formally, the importance of a feature i is calculated by Integrated Gradients (*Integrated\_Gradients*<sub>i</sub>(x)) as shown in Equation 4.2.

$$(x_i - x'_i) * \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha * (x_i - x'_i))}{\partial x_i} d\alpha$$
(4.2)

Where x is the input, x' is some baseline input with the absence of features (usually a null input or random noise), and  $\alpha$  is some interpolation constant for feature perturbation from the baseline. Since the calculation of this definite integral is computationally expensive and is not generally necessary, the value of the definite integral is generally approximated through a Riemann sum.

We also multiply each extracted importance by the term  $c^{length(feature)}$  for each feature's attribution, where c is a constant. During experimentation, we empirically observed that shorter POS n-grams were more abundant at the beginning of the attributionranked n-gram lists. We believe that this behavior is because of the necessarily lesser frequency of an arbitrarily longer POS n-gram in typical texts, as each longer n-gram occurrence necessarily is an occurrence of all contiguous substrings of the n-gram, i.e., shorter n-grams. Therefore, we introduced this scaling constant to artificially inflate the importance of longer POS n-grams to compensate for this behavior.

#### 4.2.5 Replacement Phrase Generation via Masked PLM

To perform obfuscation, we must be able to generate replacement phrases using existing phrases as prompts. To do this, we leverage the masked language modeling approach used by [107]. More specifically, given a sentence and the desired word tokens to be replaced, we mask the tokens to be replaced and use this modified text as input for a BERT model under a masked token prediction task. The top prediction for each masked token is used as the word's replacement. By using a SOTA language model, we aim to minimize the degree of information loss, as the language model will be able to infer much of the contents of the phrase through context but may scramble POS sequences, which hides authorship. This token-sequence masking procedure lies at the core of ALISON's speed-up, allowing a single PLM forward pass to perturb multiple tokens.

#### 4.2.6 Text Obfuscation Process: One N-Gram at a Time

To obfuscate each  $t \in T$ , we first extract the POS tags and n-gram features for t, which are used to compute importance values as described previously. Then, we iterate through the ranked feature list in descending order of importance, omitting character n-gram features (only considering POS n-gram features) and pick the top L features. We omit character n-grams because important character n-grams are generally functional words or involve punctuation, which would negatively impact fluency upon perturbation.

Next, we attempt to match each of the top L POS n-grams to the POS n-gram profile of t. For each n-gram match found, we update t through the phrase generation procedure as described previously. Lastly, we mark this phrase as changed so that it cannot be changed in subsequent steps as to prevent any specific section of text from deviating significantly from the original. Obfuscation is complete once all matches for the top LPOS n-grams are processed.

Often, phrase replacement more than twice results in the language model predicting punctuation only for each word in the phrase to be masked, which is not desirable due to punctuation carrying no semantic meaning. One unique property of ALISON is that it will modify the text even if the internal classifier believes it will be classified incorrectly. This property is desirable because ALISON will uniformly obfuscate all texts, likely decreasing adversarial classifier confidence even if a complete obfuscation is unsuccessful. This differs from logit query-based methods because they do not attempt to perform any obfuscation if their internal classifier's prediction does not match the ground truth, leading to a large proportion of  $t \in \mathcal{T}$  being completely unedited and therefore vulnerable.

# 4.3 Experimental Setup

We evaluate our obfuscation method under the blind attack setting, meaning that we have no knowledge of the adversarial classifier's training data or architecture and also have no ability to query the target AA model.

## 4.3.1 Datasets

We use *TuringBench* [121] to evaluate ALISON on machine-generated texts. TuringBench is a collection of 160K human and machine-generated texts across 20 authors, 19 of which are neural text generation models, and one of whom is human. All texts generated from a specific prompt are in a specific subset of the data (training, validation, or testing) to limit topic bias in experimentation. TuringBench is split into 70:10:20 training, validation, and test proportions, respectively. TuringBench was designed to serve as the first benchmark environment for a variety of NLP tasks, including authorship attribution, and provides a current leaderboard of authorship attribution performances<sup>1</sup>. The texts are on a variety of topics, and all contain a large number of tokens. As all texts were found to be of sufficient length (between approximately 100 and 400 tokens), we did not remove any samples from the dataset during any portion of the experiments.

We also use the *Blog Authorship Corpus* [122] to evaluate ALISON on human-written texts. The dataset consists of the aggregated blog posts of 19,320 bloggers gathered from blogger.com, of which we select only the blogs from the top-10 most frequent authors. Both datasets are publicly available. We report all AO results on the test set.

**Target Classifiers.** We use three SOTA transformer-based models as target AA classifiers to attack: BERT [107], DistilBERT [123], and RoBERTa [109]. These target classifiers were trained on the disjoint halves of the training and validation sets. BERT [107] is a ubiquitous language model utilizing a novel transformer architecture. RoBERTa [109] advances the pre-training approach that BERT uses, while DistilBERT is a lightweight version of BERT trained through knowledge distillation. These adversarial classifiers were trained on the *1st disjoint half* of the training and validation sets. They achieved around 80% testing accuracy on on TuringBench, while demonstrating varying performance on the Blog Authorship Corpus, ranging from approximately 85% (DistilBERT) to 95% (RoBERTa) testing accuracy.

**Obfuscation Baselines and Internal Classifier Training.** We utilize TextFooler, Mutant-X, Avengers, BERT-Attack, and ChatGPT as baselines to compare against our proposed AO framework ALISON. Except for ChatGPT, these methods all maintain an internal classifier for reference during obfuscation. While many of these are black-box

<sup>&</sup>lt;sup>1</sup>This leaderboard can be found here: https://turingbench.ist.psu.edu/

attack methods, we repurposed them for the blind attack setting using the internal classifier specifications given in transferability studies instead of giving them access to our SOTA target models. Our neural-network-based n-gram classifier is trained on the disjoint 2nd half of the training and validation data that was not used to train our SOTA target models using  $V = \{1, 2, 3, 4\}$ . Internal classifiers for Mutant-X and Avengers were trained as outlined by their papers [110, 111] on the same data as our internal classifier. Specifically, we trained a random forest classifier on the Writeprints feature set for the Mutant-X method, and an ensemble consisting of multiple SVM and random forest classifier models under a soft voting scheme for Avengers. We picked these internal classifier configurations because as [110] and [111] indicated that these were the most high-performing for obfuscation and attribution. TextFooler was trained with both wordbased CNN (wordCNN) [124] and word-based LSTM (wordLSTM) internal classifiers as specified in their public implementation. We additionally tested TextFooler using our n-gram-based NN model (denoted as TextFooler-POS) to provide a fair comparison and illustrate the effectiveness of our stylometry-grounded approach. BERT-Attack was trained using standard BERT [107]. ChatGPT-based obfuscation was performed by pretending a fixed paraphrasing prompt to each text and obtaining the returned machine response.

**Obfuscation and Evaluation Procedure.** We randomly select a sample of 100 texts from the test set, stratified by label, for obfuscation. For each victim model, we only retain the obfuscation samples that the model classifies correctly, resulting in approximately 80 samples for obfuscation on TuringBench, and varying from 85 to 95 samples for the Blog Authorship Corpus. After obfuscation, we measure a few key metrics. These metrics can be broadly divided into two groups: metrics measuring obfuscation success, or the success of obfuscation in evading the adversarial classifier, and metrics measuring semantic preservation, or how semantically similar the original and obfuscated texts are.

### 4.3.2 Evaluation Metrics

• Obfuscation Success. The most intuitive measure of obfuscation success is measuring the target AA model's accuracy. Because there is a potential for the label distribution to become skewed during the removal of misclassified samples, we also measure F1-Score, a more robust metric in such a setting. To analyze the obfuscation success, we also monitor the reduction in target model accuracy between the original and obfuscated texts. Because we only retain correctly classified

	TuringBench		
Method	Obfuscation Success (Lower is Better)		
ivicultura and a second and a	Accuracy↓	$\mathbf{F1} extsf{-}\mathbf{Score}\!\!\downarrow$	
	BERT		
Mutant-X	0.8987	0.8798	
Avengers	0.8354	0.8334	
TextFooler-wordCNN	0.7089	0.6797	
TextFooler-wordLSTM	0.7342	0.6935	
TextFooler-POS	0.7595	0.7011	
BERT-Attack	0.9114	0.9179	
ChatGPT	0.7089	0.6566	
ALISON	0.6962~(-1.79%)	0.6065~(-7.63%)	
	DistilBERT		
Mutant-X	0.9494	0.9464	
Avengers	0.9113	0.8515	
TextFooler-wordCNN	0.7848	0.7556	
TextFooler-wordLSTM	0.7722	0.7705	
TextFooler-POS	0.7972	0.7955	
BERT-Attack	0.8228	0.8172	
ChatGPT	0.7456	0.6474	
ALISON	0.5823~(-21.90%)	0.4925~(-23.93%)	
	RoBERTa		
Mutant-X	0.9014	0.8527	
Avengers	0.8028	0.7393	
TextFooler-wordCNN	0.6901	0.6074	
TextFooler-wordLSTM	0.7606	0.6682	
TextFooler-POS	0.7606	0.6760	
BERT-Attack	0.8451	0.8412	
ChatGPT	0.7924	0.6569	
ALISON	0.6620~(-4.07%)	0.5624 (-7.41%)	

Table 4.1: Obfuscation success across various models. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.

samples for obfuscation, the baseline accuracy and F1-Score are 1.00. A smaller post-obfuscation accuracy and F1-Score indicates a more successful attack, and therefore greater obfuscation success.

Blog Authorship Corpus				
Method	Obfuscation Success (Lower is Better)			
	Accuracy↓	$\mathbf{F1} extsf{-}\mathbf{Score}\downarrow$		
	BERT			
Mutant-X	0.9130	0.9180		
Avengers	0.9565	0.9528		
TextFooler-wordCNN	0.9348	0.9305		
TextFooler-wordLSTM	0.9565	0.9531		
TextFooler-POS	0.9348	0.9476		
BERT-Attack	0.9130	0.8914		
ChatGPT	0.9022	0.8908		
ALISON	0.8804~(-2.42%)	0.7860~(-11.76%)		
	DistilBERT			
Mutant-X	0.9048	0.9128		
Avengers	0.9405	0.9435		
TextFooler-wordCNN	0.8810	0.8570		
TextFooler-wordLSTM	0.8810	0.8425		
TextFooler-POS	0.8810	0.8591		
BERT-Attack	0.9048	0.8784		
ChatGPT	0.9762	0.9712		
ALISON	0.7738~(-12.17%)	0.7189~(-14.67%)		
	RoBERTa			
Mutant-X	0.9895	0.9886		
Avengers	1.00	1.00		
TextFooler-wordCNN	0.3579	0.3397		
TextFooler-wordLSTM	0.3684	0.3394		
TextFooler-POS	0.3369	0.3295		
BERT-Attack	0.9053	0.8737		
ChatGPT	0.5684	0.5939		
ALISON	0.3053~(-9.38%)	0.2912~(-11.62%)		

Table 4.2: Obfuscation success across various models. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.

• **Running Time.** First, we recorded the running time of each algorithm, as an obfuscation method that requires a prohibitive amount of resources or computation time may not be scalable to real world AO scenarios. We split this time measurement into two phases, the time associated with one-time training of internal classifiers,

TuringBench					
	Semantic Preservation (Higher is Better)				
Method	METEOR↑	USE Cosine Similarity <sup>↑</sup>	$\mathbf{BERTScore}^{\uparrow}$		
		BERT			
Mutant-X	0.8381	0.9159	0.9366		
Avengers	0.8333	0.9030	0.9320		
TextFooler-wordCNN	0.8667	0.9614	0.9386		
TextFooler-wordLSTM	0.8813	0.9671	0.9430		
TextFooler-POS	0.8650	0.9635	0.9382		
BERT-Attack	0.8388	0.8701	0.9526		
ChatGPT	0.8373	0.9113	0.9490		
ALISON	0.8505 (-3.49%)	0.9682~(0.11%)	0.9583~(0.60%)		
	Dis	stilBERT			
Mutant-X	0.8450	0.9192	0.9406		
Avengers	0.8341	0.9048	0.9320		
TextFooler-wordCNN	0.8641	0.9609	0.9413		
TextFooler-wordLSTM	0.8819	0.9677	0.9447		
TextFooler-POS	0.8675	0.9657	0.9391		
BERT-Attack	0.8434	0.8737	0.9538		
ChatGPT	0.8428	0.9142	0.9494		
ALISON	0.8538 (-3.19%)	0.9685~(0.08%)	0.9588~(0.52%)		
RoBERTa					
Mutant-X	0.8182	0.9062	0.9306		
Avengers	0.8157	0.8967	0.9248		
TextFooler-wordCNN	0.8621	0.9618	0.9386		
TextFooler-wordLSTM	0.8814	0.9686	0.9446		
TextFooler-POS	0.8623	0.9624	0.9402		
BERT-Attack	0.8279	0.8603	0.9484		
ChatGPT	0.8268	0.9057	0.9436		
ALISON	0.8554 (-2.95%)	0.9701~(0.15%)	0.9595~(1.17%)		

Table 4.3: Obfuscation success across various models. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.

and the time associated with the average inference time of the retained samples.

• Semantic Preservation. We also measure metrics of semantic preservation or semantic similarity between the original and obfuscated texts. Metrics indicating

Blog Authorship Corpus					
Semantic Preservation (Higher is Better)					
Method	<b>METEOR</b> ↑	USE Cosine Similarity <sup>↑</sup>	$\mathbf{BERTScore}^{\uparrow}$		
		BERT			
Mutant-X	0.8325	0.8514	0.9237		
Avengers	0.8894	0.9028	0.9316		
TextFooler-wordCNN	0.8854	0.9472	0.9356		
TextFooler-wordLSTM	0.8811	0.9439	0.9382		
TextFooler-POS	0.8838	0.9453	0.9321		
BERT-Attack	0.9007	0.9221	0.9202		
ChatGPT	0.6720	0.8827	0.9368		
ALISON	0.8296 (-7.89%)	0.9551~(0.83%)	0.9386~(0.04%)		
	Dis	stilBERT			
Mutant-X	0.8209	0.8497	0.9135		
Avengers	0.8826	0.9044	0.9305		
TextFooler-wordCNN	0.8839	0.9465	0.9356		
TextFooler-wordLSTM	0.8786	0.9427	0.9382		
TextFooler-POS	0.8832	0.9442	0.9349		
BERT-Attack	0.9026	0.9245	0.9205		
ChatGPT	0.6524	0.8820	0.9347		
ALISON	$0.8431 \ (-6.59\%)$	0.9595~(1.37%)	0.9387~(0.05%)		
RoBERTa					
Mutant-X	0.8285	0.8514	0.9232		
Avengers	0.8886	0.9036	0.9305		
TextFooler-wordCNN	0.8857	0.9475	0.9354		
TextFooler-wordLSTM	0.8782	0.9437	0.9381		
TextFooler-POS	0.8823	0.9454	0.9322		
BERT-Attack	0.9007	0.9221	0.9202		
ChatGPT	0.6720	0.8827	0.9368		
ALISON	0.8296 (-7.89%)	0.9551~(0.83%)	0.9386~(0.04%)		

Table 4.4: Obfuscation success across various models. Best performance is shown in boldface. The percentage (%) indicates the performance gain of ALISON compared to the 2nd best competition if positive (or drop if negative) per each metric.

higher semantic preservation are favorable, as they indicate that there was a limited degree of information loss and that the perturbations to the text would not significantly impair a reader's understanding of the original text. These metrics include (1) *METEOR Score:* METEOR score is a standard for measuring the similarity between two texts in a natural language setting. It is grounded in the measure of alignments of word unigrams among texts; (2) USE Cosine Similarity: The Universal Sentence Encoder (USE) [125] is a text embedding model that is frequently adopted to accurately capture the semantics of a sentence. We utilize cosine-similarity to determine the degree of similarity between generated embeddings; (3) BERTScore: BERTScore [28] is another metric of semantic similarity that utilizes BERT's pretrained contextual embeddings. BERTScore is calculated by maximizing pairwise embedding similarities for the tokens of an original and its obfuscated text. All scores lie in [0, 1], and higher scores denote greater semantic similarity.

• Fluency. Lastly, we measure the perplexity of obfuscated texts to ensure that the obfuscation process does not diminish the human readability of obfuscated texts. The perplexity is calculated as the negative log-likelihood of LLaMA2-7B [83] over obfuscated texts.

#### **Evaluation Metrics - Obfuscation Success**

The most intuitive measure of obfuscation success is measuring target model accuracy. To adapt this definition of F1 score for multinomial classification tasks, we simply calculate an F1 score for each class, treating the current class label as positive, and all other classes as negative. We then average the value of all twenty F1 scores, without class weights, to report an overall F1 score. This calculation gives a more robust measure of the actual performance of a classifier, with less bias from potentially imbalanced classes. To analyze the obfuscation success of an obfuscation method, we specifically want to monitor the reduction in F1 between the original and obfuscated texts.

To analyze obfuscation success, we specifically want to monitor the reduction in target model accuracy between the original and obfuscated texts. Because we only retain correctly classified samples for obfuscation, the baseline accuracy and F1 score are 1.00. A smaller post-obfuscation accuracy and F1 score indicates a more successful attack, and therefore greater obfuscation success, desirable in a more optimal obfuscation technique.

#### **Evaluation Metrics - Semantic Preservation**

We also measure metrics of semantic preservation or semantic similarity between the original and obfuscated texts. Metrics indicating higher semantic preservation are favorable, as they indicate that there was a limited degree of information loss and that the perturbations to the text would not significantly impair a reader's understanding of the original text.

- METEOR Score :<u>Metric for Evaluation of Translation with Explicit OR</u>dering  $(METEOR \in [0, 1])^2$  score is a standard for measuring the similarity between two texts in a natural language setting. It is grounded in the measure of alignments of word unigrams among texts. The higher the score, the higher degree is the semantic similarity between an original and its obfuscated text.
- USE Cosine Similarity∈[0, 1]. The Universal Sentence Encoder (USE) [125] is a text embedding model that is frequently adopted in NLP literatures to accurately capture the semantic of a sentence. We use USE with cosine-similarity measure to calculate the semantic preservation between an original and its obfuscated text.
- *BERTScore*. BERTScore [28] is another metric of semantic similarity that utilizes the pretrained contextual embeddings produced by a BERT model. BERTScore is calculated by maximizing pairwise embedding similarities for the tokens of an original and its obfuscated text.

We present more details on methods underlying the calculation of these evaluation measures as follows:

**METEOR Score**: Given reference and hypothesis texts, METEOR constructs an alignment, or word matching, between the two texts that first attempts to match each unigram in the hypothesis with 0 or 1 unigram(s) in the reference. If multiple alignments are possible, METEOR chooses the alignment with the minimum number of mapping crossings. A unigram is mapped to another unigram based on exact word equality, stemming, and synonyms based on WordNet. Once an alignment is created, the matched content and function words in each alignment are used to calculate unigram precision, defined as the proportion of unigrams mapped in the generated text over the total number of unigrams in the reference text. This precision and recall are then used to calculate the METEOR score, which is based on the harmonic mean of these two values (similar to the F1 score), and a penalty factor accounting for relative unordering

<sup>&</sup>lt;sup>2</sup>https://github.com/salaniz/pycocoevalcap

between adjacent chunks between the reference and hypothesis texts.

**USE Cosine Similarity**: The Universal Sentence Encoder (USE) [125] is a text embedding model that addresses some of the fundamental limitations of previous encoding models, such as Word2vec [126], which cannot handle text beyond a single word. USE can accurately capture the semantic and natural language meanings of a sentence, including context beyond single words. There are two variants of USE, one based on a transformer encoder, and one based on a deep averaging network. We utilized the transformer encoder version of USE due to its greater power when compared to the deep averaging network variant. After extracting the embeddings from the reference and obfuscated texts, we calculate the similarity between the two texts by calculating the cosine similarity between the two embeddings. Cosine similarity is defined as follows:

$$Cosine \ Similarity = \frac{R \cdot O}{\|R\| \|O\|}$$
(4.3)

where R is the reference embedding vector, and O is the obfuscated embedding vector.

**BERTScore**: BERTScore [28] utilizes the pretrained contextual embeddings produced by BERT models. Because BERT models can infer the semantic meanings, including context, of sentences for use in various natural language tasks, it intuitively follows that the embeddings that BERT models create are powerful for natural language representation. BERTScore is calculated by maximizing pairwise embedding similarities for the tokens of two samples.

# 4.4 Results

# 4.4.1 Obfuscation Success

The experimental results on both datasets from our main obfuscation experiment are summarized by Tables 4.1, 4.2, 4.3, and 4.4. In the table, we denote the metric indicating the most favorable attack in bold (the metric with the lowest magnitude for obfuscation success metrics, and the metric with the highest magnitude for semantic preservation metrics) across each adversarial trial. Additionally, for the rows containing results for ALISON, we show the percentage change of each metric from the method that was the highest performing, excluding ALISON. Therefore a lower percentage (higher degradation

of adversarial accuracy / F1-Score) is more desirable for obfuscation success metrics, while a higher percentage (less semantic degradation) is favorable for semantic preservation metrics.

On TuringBench, we see that ALISON is consistently the best performer in terms of attack success. ALISON consistently degrades adversarial accuracy more than other methods, demonstrating improvement as high as 21.90%. Additionally, F1-Score even more pronounced degradation, with improvement as high as 23.93%.

On the Blog Authorship Corpus, results shown in Table 4.2 indicate that ALISON is consistently the best performer in terms of F1-Score and accuracy. In terms of AccuraF1-Score degradation, we see a consistent improvement when compared to other baseline methods.

## 4.4.2 Ablation of Interpretability-Based Replacement

We observe that ALISON outperforms TextFooler-POS in all trials. This demonstrates the value of ALISON's sequence replacement schema and interpretability-centric approach when compared to traditional token-by-token perturbation methods.

### 4.4.3 Computational Complexity

Running time results are summarized by Table 4.5. The One-Time Training stage encompasses all operations associated with data feature extraction and one-time training, while Inference corresponds to per-text running time.

The results indicate that ALISON outperforms all baselines both in terms of one-time training and obfuscation runtime. ALISON's total time for both one-time training and obfuscation of 100 samples indicates at least a 10x speed-up on TuringBench and at least an 18x speed-up on the Blog Authorship Corpus. ALISON is additionally at least 10x faster on TuringBench and 20x faster on the Blog Authorship Corpus with respect to one-time training and at least 10x faster during obfuscation on both datasets.

Fig. 4.4 shows the dependence of average obfuscation running time on the total length of the text. ALISON demonstrates a lesser running time increase for longer texts

Method	One-Time Training Inference		
TuringBench			
Mutant-X	4 hrs	3 min	
Avengers	6  hrs	$5 \min$	
TextFooler-wordCNN	2 hrs	$8  \mathrm{sec}$	
TextFooler-wordLSTM	1 2 hrs	$7  \mathrm{sec}$	
BERT-Attack	6  hrs	$8  \mathrm{sec}$	
ALISON	$12 \min$	0.8 sec	
Blog Authorship Corpus			
Mutant-X	8 min	10 min	
Avengers	$24 \min$	$14 \min$	
TextFooler-wordCNN	2 hrs	$11  \mathrm{sec}$	
TextFooler-wordLSTM	1  2 hrs	$9  \mathrm{sec}$	
BERT-Attack	6  hrs	$9  \mathrm{sec}$	
ALISON	$6 \min$	1.0 sec	

Table 4.5: Statistics of the one-time training runtime and the average inference time per one sample for all methods.

when compared to other methods.



Figure 4.4: Average time taken per obfuscation by different methods for increasing average length of articles.

## 4.4.4 Semantic Preservation

Across both datasets, as seen in Tables 4.3 and 4.4, ALISON consistently outperforms in semantic preservation when evaluated with USE cosine similarity, the most robust measure of semantic preservation we measured, and BERTScore. However, we observe that ALISON consistently performs the worst in terms of METEOR score on both datasets; however, we believe that this result can largely be attributed to the inherent flaws of the METEOR score, as it is generally less correlated with human judgments when compared to USE cosine similarity, which is a stronger standard for semantic similarity analysis.

## 4.4.5 Overall Text Quality

Method	TuringBench	Blog
Mutant-X	65.12	29.55
Avengers	64.51	23.12
TextFooler-wordCNN	57.69	17.96
TextFooler-wordLSTM	52.89	19.28
TextFooler-POS	56.23	18.34
ALISON	20.82	12.11

Table 4.6 demonstrates that ALISON demonstrates the best perplexity across both datasets, indicating the highest readability across all AO methods.

Table 4.6: Perplexity of post-obfuscation texts measured using LLaMA2-7B (lower is better).

#### Author Label Bias

First, we analyze the distribution of author frequencies before and after obfuscation to identify potential obfuscation bias towards an author or set of authors on both datasets. To do this, we calculate the normalized entropy of author labels over obfuscated samples.

To do this, we first generate a confusion matrix between the pre-and post-obfuscation label predictions. Each element  $C_{i,j}$  in the confusion matrix C is the number of occurrences where text was predicted to have label i pre-obfuscation was predicted to have label jpost-obfuscation. Then we calculate the normalized entropy of author labels along each row. This is the entropy of the obfuscation process with regard to the author label, for

Method	Obfuscation Success		
Wiethou	$\overline{\text{Accuracy}\downarrow}$	$\overline{\mathbf{F1}}$ -Score $\downarrow$	
GPT Output Detector - Base	0.5000	0.3670	
GPT Output Detector - Large	0.5682	0.3623	
GPTZero	0.6170	0.5323	
DetectGPT	0.5729	0.4984	

Table 4.7: Obfuscation Success across four machine text detection models.

Method	Semantic		
Method	$\overline{\mathbf{METEOR}\uparrow\mathbf{USE}}$	Cosine Similarity	$\uparrow \mathbf{BERTScore} \uparrow$
GPT Output Detector - Base	0.6966	0.8754	0.8941
GPT Output Detector - Large	0.6948	0.8734	0.9017
GPTZero	0.6897	0.8717	0.8936
DetectGPT	0.7478	0.9030	0.9134

Table 4.8: Semantic Preservation across four machine text detection models.

each starting label. For n authors, the row-wise entropy of author i is:

$$Entropy_{i} = -\sum_{i=0}^{n} \frac{ln(C_{i,j}) * C_{i,j}}{ln(n)}$$
(4.4)

A high entropy value indicates that the uncertainty associated with a variable is high (difficult to predict the exact outcome or set of outcomes of the variable), while a low entropy value indicates that the uncertainty associated with a variable is low (a certain outcome or small set of outcomes is very likely).

Because of the varied attack successes of different methods, we do not consider the raw entropy values but instead, consider the proportion of the total label entropy each author contributes. The distribution of these label entropy proportions should be as uniform as possible so that each author label transforms in an unpredictable way. A non-uniform entropy distribution across authors indicates that the obfuscation of a small pool of authors' texts contributes significantly to the overall attack success. This indicates a bias during obfuscation in regard to the transformation of author labels, a bias that can potentially be exploited by the attacked model. If the post-obfuscation prediction label were predictable based on the pre-obfuscation prediction label, an adversary would be



Figure 4.5: Distribution of author-wise contributions to label entropy post-obfuscation.

able to gain significant information about the authorship of a text based on the predicted author post-obfuscation. This bias is further not desirable since the authorship pool may vary from various obfuscation settings.

We present the individual author entropy contributions over all authors for all methods in Figure 4.5. It is visually apparent that the distribution of author entropy contributions is significantly more uniform for ALISON when compared to other methods. This indicates significantly less predictability and label bias during obfuscation when compared to other methods. There are very few labels with a small or nonexistent contribution to overall entropy, which are labels that could be trivially reverse-engineered by the targeted model, unlike the entropy distributions of other methods. We aim to maximize the entropy of the obfuscation process with respect to each author label. So, for each author i, we aim to maximize  $Entropy_i$ . A low value of  $Entropy_i$  also indicates that the pre-obfuscation predicted label i obfuscates predictably into a small set of post-obfuscation predicted labels. This indicates that the obfuscation method may not be successful without these labels to obfuscate into.

We see that the distribution of author label entropy is visually significantly more even for ALISON when compared to other methods. Additionally, ALISON demonstrates a total row-wise entropy of 3.3981, while Mutant-X demonstrates a total entropy of 0.8259



Figure 4.6: Normalized Author Label Entropy on Blog for Avengers



Figure 4.7: Normalized Author Label Entropy on Blog for ALISON

and Avengers demonstrates a total entropy of 0.7281. This indicates that ALISON is drastically less predictable with regards to authorship label obfuscation bias.

#### Interpretability

Because ALISON relies on explicitly determined criteria for obfuscation, it can explain obfuscation decisions using quantified token importances. Interpretability is generated by extracting the POS n-grams in a text and using Integrated Gradients to generate the importance of each POS n-gram, which is scaled as described previously. Top POS n-gram features may then be mapped to specific token sequences in the original text.

#### Parameter Analysis



Figure 4.8: Effect of varying L on obfuscation success and semantic preservation

Both Mutant-X and Avengers provide a default parameterization that has been tuned for authorship obfuscation on standard corpora. While this parameterization should be optimal, we present a sensitivity analysis on two key parameters shared by both Mutant-X and Avengers to demonstrate the changes in both attack success and semantic preservation. The first parameter we vary is the top-K value, or the number of top documents to select, based on the fitness function, for the next iteration of the genetic algorithm. The default value of this is 5, so we present the results for a topK value of 10. Next, we vary the number of allowed neighbors in the semantic-specific Word2Vec embedding. The number of allowed neighbors determines the number of sentiment-specific candidates tested as potential replacements in the obfuscation process. We increase the number of allowed neighbors from the default value of 5 to 10 and demonstrate the effects on attack success and semantic preservation. Because these operations drastically increase the size of the search space Mutant-X and Avengers must explore, increasing the value of these parameters also increase the running time.

In the following, we explore the effects of varying two key parameters on metrics of obfuscation using TuringBench. We first explore the effects of varying the value of L. We record the aforementioned metrics for values of L ranging from 1 to 250. The results of this experiment are summarized by Fig. 4.8.

This figure indicates that increasing the value of L increases the change rate and decreases all other metrics (both metrics of obfuscation success and semantic preservation). This result is intuitive, as increasing the number of sequences to change will, of course, increase the change rate and will therefore be more successful in fooling the adversarial classifier, and will also cause a greater difference between the original and obfuscated texts, decreasing semantic preservation.

We also explore the effect of varying the value of c from 1.0 to 1.6. The results of this experiment are summarized in Fig. 4.9.



Figure 4.9: Effect of varying c on obfuscation success and semantic preservation

From this, we observe that the value of c does not significantly impact metrics of semantic preservation, as the adversarial METEOR, USE Cosine Similarity, and BERTScore do not significantly change across values of c. However, metrics of obfuscation success decrease, then increase across  $c \in [1.0, 1.6]$ , where the optimal value of c is between 1.3 and 1.5. This supports the idea that artificially scaling importance based on length allows **ALISON** to choose sequences of optimal length to mask.

#### Limitations of METEOR Score

While ALISON generally outperforms other baselines, we demonstrate consistent underperformance when measuring semantic preservation in terms of METEOR score. We believe that this behavior is due to the inherent bias of the METEOR score against ALISON's style of perturbation. First, we observe that the METEOR score highly depends on exact spacing. During the process of encoding and decoding associated with obtaining masked-phrase substitutions, erroneous spaces are introduced into the text. While these spaces do not impact information loss significantly, they do significantly impact the

Text	Original	Obfuscated
1	I got back my first draft of my memo.	i had finished my first draft of the novel .
2	I was getting ready for work and picked out a <b>black</b> cap sleeved <b>shirt</b> that has a kind of mock wrap around neckline.	I was getting ready for work and picked out a <b>white</b> cap sleeved <b>dress</b> that has a kind of mock wrap around neckline .
3	<b>So</b> if you're noticing something looks different, you're right.	if you 're noticing something looks dif- ferent , you 're right .
4	I slept terribly last night, not sure how much sleep I got, but it didn't feel like much.	last night , not sure how much sleep I got , but it did n't feel like much .
5	I know I've parodied this song before, but the syllables fit so <b>well</b> , and there are no better choices I can think of	I know I 've parodied this song before , but the syllables fit so <b>perfectly</b> and there are no better choices I can think of
6	What's the one memory of your child- hood that makes you <b>laugh everytime</b> <b>you</b> look back?	What 's the one memory of your child- hood that makes you smile whenever look back ?
7	I got four bags of free medicine: an- tibiotics, nasal spray, and pills for my congestion.	I got four bags of free medicine : antibiotics , nasal spray , and two pills .
8	Unless you <b>are</b> Justin Timberlake.	Unless you <b>is</b> Justin Timberlake .
9	I took a look at my paycheck last night, and to my (unpleasant) surprise, <b>they</b> <b>shorted</b> me 8 hours	I took a look at my paycheck last night , and to my ( unpleasant ) surprise <b>it</b> <b>had given</b> me 8 hours .
10	I didn't have time for breakfast though, so I wound up eating one of those 'Lunchables' at work.	<b>i wanted to go</b> for breakfast though , so I wound up eating one of those 'Lunchables ' at work .

Table 4.9: Sample sentences from the Blog Authorship Corpus pre- and post- obfuscation

METEOR score. Without any phrase substitutions, the spacing changes made by BERT and the reconstruction process degrade the METEOR score to 0.93 on TuringBench and 0.90 on the Blog Authorship Corpus. However, the USE Cosine Similarity and BERTScore are minimally affected, maintaining values of around 0.99. An example of a text exhibiting near-perfect semantic preservation (as measured qualitatively, by USE Cosine Similarity, and by BERTScore) but with a low METEOR score is the following:

• Original: That is what it is. Cold and dreary. A Soda without fizz. Boogers.

• Obfuscated: i know what it is .Cold and dreary .A Soda without fizz .Boogers.

While the USE Cosine Similarity is 0.9409 and BERTScore is 0.9616, the METEOR score is 0.5152. The patterns exhibited in this example, leading to a low METEOR score, are repeated throughout many examples in our obfuscation method.

However, other baselines do not exhibit these patterns, with semantic preservation scores generally agreeing more. This is because single-word substitutions often leave spacing patterns and the relative ordering of words generally unchanged. The following, a sample taken from Avengers, illustrates this:

- Original: Went to an information session for people who might be interested in helping to teach the First Year Lawyering program next year... it's an organization called the Board of Student Advisors. The meeting wasn't funny enough to give me anything funny to write.
- *Obfuscated*: Went to an info meeting for children who might become interested in helping to teach the First Year Lawyering initiative next month... it's An watchdog called the Board of Student Advisors. The press wasn't funny enough to get me nothing humorous to publish.

This sample demonstrates a METEOR score of 0.7673, a USE Cosine Similarity of 0.8117, and a BERTScore of 0.9452.

## 4.4.6 Efficiency: Time Complexity

We also aim to provide time complexity-related factors to explain the expensive running time associated with Mutant-X and Avengers. First, genetic algorithms are inherently chaotic, utilizing a large population of individuals to optimize a fitness function. Maintaining this population and performing the operations required to mutate and progress is expensive. This sacrifices computational time for a more complete search to converge to a better optimum for the fitness function. ALISON addresses this issue by greedily replacing POS n-grams instead of performing a more complete search for faster obfuscation. The fitness function Mutant-X and Avengers aim to optimize is also complex, incorporating both METEOR score and obfuscation success. Additionally, Mutant-X and Avengers perform single-word substitutions, which take a greater number of changes to perform successful obfuscation. We reduce the complexity by replacing entire word sequences at a time.

## 4.4.7 Qualitative Analysis

We analyze the types of changes made by sampling some of the sentences pre- and post-obfuscation on TuringBench and the Blog Authorship Corpus. Table 4.9 shows a sample of ten sentences from ten distinct texts in the corpus. Changed tokens are denoted in boldface. The changes generally follow a few distinct patterns:

- Synonym Substitutions: Often, a word in the original text is replaced by a synonym in the obfuscated texts. For example, memo ⇔ novel, perfectly ⇔ well, smile ⇔ laugh, etc. These most likely result from a short POS-tag sequence being identified as important, leading to the BERT model being able to relatively accurately determine the meaning of the masked words from context, leading to little information loss.
- Contextually Acceptable Substitutions: These substitutions involve the substitution of a word or phrase with a semantically unequivalent phrase that seems plausibly correct in context. For example, white  $\iff$  black, shirt  $\iff$  dress, shorted  $\iff$ given, etc. These most likely arise from a phrase being masked that cannot be reliably determined from the surrounding context. This leads to the BERT model making an incorrect inference regarding the semantic meaning of the masked phrase. While this increases the degree of information loss between the original and obfuscated texts, it does not significantly affect the readability of the obfuscated text.
- Deletions: Sometimes, nonessential words and phrases are completely deleted after reconstruction with BERT. These deletions often occur in introductory phrases that contribute little semantic meaning. Although rarely happen, these deletions may slightly contribute to information loss or decrease readability.
- Equivalent Substitutions Resulting in Solecism: Sometimes, words or short phrases will be replaced by nearly identical substitutes semantically and functionally that result in a solecism. This results in no information loss, however, the solecism may result in decreased readability. These substitutions may indicate inherent limitations of BERT for Masked Language Modeling, or inherent grammatical errors in the data BERT was trained on. Additionally, the uncapitalized "i" is also common.

#### Obfuscating ChatGPT Texts: A Use Case

Powerful language models, including GPT-3 [127] and GPT-4 [128], have demonstrated impressive ability to comprehend and synthesize language across contexts. The impressive performance of ChatGPT [128], a conversational language model, has led to its ubiquitous use in the workplace and classroom. Though ChatGPT can assist humans with everyday tasks, its potentially dishonest applications (e.g. construing ChatGPT's output as human-written text in academic settings) make the identification of ChatGPT-written texts an important problem with extensive commercial and academic study [78,129–131]. The commercial value of ChatGPT detection further motivates an AO technique that is computationally efficient.

**Problem Formulation.** The real-world task of discriminating between ChatGPT and human-written texts is an increasingly relevant AA task that motivates the study of the corresponding AO task. We select four well-known machine-text generators, each demonstrating > 95% discrimination accuracy, to study under adversarial perturbation: GPTZero [129], DetectGPT [78], and both the Base and Large GPT Output detectors [130] released by OpenAI.

**Methodology.** We used news article headlines from TuringBench to query the OpenAI Completions API. A single request was made for each unique headline, which consisted of a fixed generation prompt prepended to the headline. The corresponding human-written texts in the TuringBench corpus provided negative examples to introduce into the corpus, generating a set of evenly distributed negative and positive examples. The experimental setup described previously was then repeated.

Main Obfuscation Trial Result. Table 4.7 shows metrics of Obfuscation Success and Table 4.8 shows Semantic Preservation against adversarial classifiers. ALISON demonstrates degradation of adversarial accuracy to *at most* 0.617 and adversarial F1-Score to *at most* 0.5323. In addition, ALISON consistently maintains a high degree of semantic similarity between original and obfuscation texts, maintaining *at least* 0.8717 USE Cosine Similarity and 0.8936 BERTScore. ChatGPT text detectors become negligibly useful at such adversarial performance, as the adversarial accuracy is close to the trivial accuracy of 0.50 in the binary classification setting.

**Entropy Result.** We observe an entropy of 0.56 associated with the human class and an entropy of 0.44 associated with the ChatGPT class. Because the distribution of authorship label entropy is not significantly skewed toward any class, ALISON does not demonstrate a significant degree of bias during the obfuscation process in transferring attributions from any specific class.

# 4.5 Discussion

We present a new authorship obfuscation technique, ALISON, based on the replacement of revealing stylistic sequences. ALISON greedily replaces text sequences matching POS n-grams identified to be important by interpreting a lightweight neural network trained to perform authorship attribution using mixed n-grams. We use ALISON to attack three SOTA transformer-based attribution classifiers and demonstrate an improvement in obfuscation success and semantic preservation when compared to *seven* diverse baselines. We demonstrate that ALISON's intuitive and simple but effective nature demonstrates a drastic improvement in computational complexity compared to baseline methods.

While authorship obfuscation enables freedom of speech for various previously described individuals including whistleblowers and journalists, it also potentially permits malicious groups to stay hidden. We acknowledge such ethical concerns but stress the need to study and design systems that can protect and enhance the freedom of speech of the public.

# Chapter 5 Collaborative Generation: Collab-Story a multi-LLM story benchmark

The rise of unifying frameworks that enable seamless interoperability of Large Language Models (LLMs) has made LLM-LLM collaboration for open-ended tasks a possibility. Despite this, there have not been efforts to explore such collaborative writing. We take the next step beyond human-LLM collaboration to explore this multi-LLM scenario by generating the first exclusively LLM-generated collaborative stories dataset called CollabStory. We focus on single-author (N = 1) to multi-author (up to N = 5) scenarios, where multiple LLMs co-author stories. We generate over 32k stories using opensource instruction-tuned LLMs. Further, we take inspiration from the PAN tasks [132] that have set the standard for human-human multi-author writing tasks and analysis. We extend their authorship-related tasks for multi-LLM settings and present baselines for LLM-LLM collaboration. We find that current baselines are not able to handle this emerging scenario. Thus, CollabStory is a resource that could help propel an understanding as well as the development of techniques to discern the use of multiple LLMs. This is crucial to study in the context of writing tasks since LLM-LLM collaboration could potentially overwhelm ongoing challenges related to plagiarism detection, credit assignment, maintaining academic integrity in educational settings, and addressing copyright infringement concerns. We make our dataset and code available at https://github.com/saranya-venkatraman/multi llm story writing.



Figure 5.1: CollabStory contains over 32k creative stories written collaboratively by up to 5 LLMs. Each story segment is generated by a single author, that then passes the narrative baton to the next, completing the storyline part by part in a sequential manner.

# 5.1 Introduction

Generative Large Language Models (LLMs) are being used more widely and becoming ubiquitous in real-world scenarios. There is particular interest in understanding the use of such LLMs in various writing tasks as writing assistants or collaborators in machine-in-the-loop settings [2–9]. So far though, this has only been explored in the case where a human is present. However given the rise of unifying frameworks that bring together and make LLMs from different sources interoperable, such as vLLM<sup>1</sup>, LangChain<sup>2</sup>, and HuggingFace<sup>3</sup>, the prospect of LLMs seamlessly collaborating and even handing off tasks to one another without external routing algorithms is on the horizon. This is particularly immediately possible with open-source models that are already being used by over 100K users per month (according to the number of downloads reported by HuggingFace). Despite the ease of interoperability of such LLMs, so far, automated writing assistants have been used only in collaboration with human authors or with a single LLM. Therefore, this study explores collaborative creative story-writing scenarios

<sup>&</sup>lt;sup>1</sup>https://docs.vllm.ai/en/stable/

<sup>&</sup>lt;sup>2</sup>https://www.langchain.com/langchain

<sup>&</sup>lt;sup>3</sup>https://huggingface.co/

Dataset	# Stories #	4 Authors	Avg # Words	M-M Collaboration	Available
STORIUM [133]	5,743	30,119	$\sim 19 \mathrm{k}$ tokens	Н-Н 🗡	$\checkmark$
CoAuthor [8]	830	58	418	H-M 🗡	$\checkmark$
StoryWars [134]	$40,\!135$	$9,\!494$	367	Н-Н 🗡	×
CollabStory [Ours]	$32,\!503$	5	725	M-M 🗸	$\checkmark$

involving multiple LLMs, i.e. LLM-LLM collaboration.

Table 5.1: Comparison of CollabStory with other existing collaborative creative story datasets. Here, "M-M"  $\rightarrow$  "Machine-Machine", "H-H"  $\rightarrow$  "Human-Human", where "H"  $\rightarrow$  "Human" and "M"  $\rightarrow$  Machine. Ours is the largest dataset to present multi-LLM or machine-machine collaborative generation.

Collaborative creative story writing entails multiple authors contributing separate segments to form a coherent storyline (see Figure 5.1 for our dataset schema). Although individual LLMs excel at generating story plots, collaborative writing presents unique hurdles. Models must seamlessly continue the existing storylines generated so far by other models, even if they do not align perfectly with their own language distribution. The rise of multi-agent Artificial Intelligence (AI) underscores the potential for combining the expertise of agents specialized in various tasks. While previous mixture-of-experts scenarios focused on agents proficient in task-oriented settings [135–139], the emergence of LLMs conversing for continuous generative tasks in open domains is noteworthy. Imagine the possibilities when multiple LLMs collaborate; one LLM can generate compelling stories, but what if we put them together?

In this study, we attempt to address this question through a collaborative creative story-writing scenario involving multiple open-source LLMs. This is a crucial setting to study in the context of writing tasks since LLM-LLM collaboration could potentially overwhelm ongoing challenges related to plagiarism detection, credit assignment, maintaining academic integrity in educational settings, and addressing copyright infringement concerns.

We focus on single-author (N = 1) to multi-author (up to N = 5) scenarios, where multiple LLMs co-author creative stories. This exploration is novel, as previous studies have primarily focused on human-LLM collaboration. Towards this goal, we generate the first multi-LLM collaborative story dataset called **CollabStory using** open-source LLMs. We select 5 frequently used LLMs (with number of downloads on HuggingFace for May 2024 provided in parenthesis): Meta's Llama (> 540k downloads, [140]), Mistral.ai's **Mistral** (> 1000k downloads, [141]), Google's **Gemma** (> 180k downloads, [142]), AllenAI's **Olmo** (> 26k downloads, [143]) and Microsoft's **Orca** (> 22k downloads, [144]) to replicate a scenario in which commonly used LLMs from different organizations are being used in conjunction towards a single task. We demonstrate how one such dataset can be developed and the considerations involved in building an iterative Multi-LLM story-writer. We take inspiration from the PAN tasks [132] that have set the standard for multi-author writing tasks and analysis for human-human collaboration for over 15 years. We replicate their task settings and present baselines of different authorship-related tasks such as authorship verification and attribution for LLM-LLM collaboration and demonstrate that current baselines are challenged by this emerging scenario. **CollabStory** is the first resource that could help propel an understanding as well as the development of new techniques to discern the use of multiple LLMs in text.

Our work is motivated by the implications of Multi-LLM settings for different stakeholders (LLM developers, end-users) and considerations (such as credit assignment, legality of usage) arising in the generative AI landscape. As one example, a malicious actor might assemble texts from different LLMs together in one document to evade current detectors and successfully spread misinformation. Our discussion will further elaborate on the tasks our dataset enables and why it is crucial to develop methods to tackle the incoming challenges of machine-machine collaboration.

# 5.2 Related Work

LLMs as Collaborative Writers. LLMs are being increasingly used as writing assistants or to paraphrase, edit or enhance human-written written texts in machinein-the-loop settings [3,5,6,9]. GhostWriter [2] and Wordcraft [7] are tools that enable users to co-write stories using instructions [7]. [4] use "writing modes" as a control signal to better align the machine during co-writing with humans. CoAuthor positions GPT3.5 as a writing collaborator for over 50 human participants to co-write creative and argumentative stories [8]. Related Datasets Despite such emerging tools, only a handful have developed datasets that can be leveraged to understand collaborative story writing. One such resource is the STORIUM dataset [133] that contains over 5k creative stories written and obtained from human-human collaboration. In terms of human-machine co-writing, CoAuthor [8] and CoPoet [31] remain one of the few publically available datasets of human-machine collaborative creative story and poem
# Number of Authors	<pre># Words per Author / # Total Words</pre>	# Author Order	# Prompts per	# Stories	
1	900 / 900	4	1800	7200	
2	450 / 900	12	600	7200	
3	300 / 900	15	480	7200	
4	225 / 900	15	480	7200	
5	180 / 900	15	480	7200	
Authors		HuggingFa	ace distribution of L	LMs used	
Gemma		$google/gemma-1.1-7b-it^8$			
Llama		$meta-llama/Llama-2-13b-chat-hf^9$			
	Mistral	mistralai/Mistral-7B-Instruct-v0.2 $^{10}$			
Orca microsoft/Orca-2-13b <sup>11</sup>			$b^{11}$		
	Olmo	allenai/OLMo-7B-Instruct $^{12}$			

Table 5.2: Summary of Data Collection Statistics

writing, respectively. Beyond creative writing, [145] developed the first machine-human academic essay dataset as a means to study boundary detection for academic settings. A comparison of CollabStory with existing datasets is provided in Table 5.1.

## 5.3 Methodology

## 5.3.1 CollabStory: Dataset Creation

We generate a dataset of creative stories using five open-source instruction-tuned LLMs: Llama2 [140], Olmo [143], Gemma [142], Mistral [141] and Orca [144] (model details are provided in Table 5.2). The main focus of our data generation is to simulate a scenario where LLMs from different sources (organizations) collaboratively work on a storyline, handing off control of the story from one LLM to the next. The stories in our dataset vary in the number of authors/LLMs involved, from being written entirely by a single LLM to written collaboratively by between 2 to up to all 5 LLMs. In this document, we refer to each of the LLMs as "authors". For cases where we refer to the human author, we specifically mention "human" author/writer. We generate our dataset by prompting various LLMs using creative writing prompts from an existing dataset called the **Writing** 

<sup>&</sup>lt;sup>8</sup>https://huggingface.co/google/gemma-1.1-7b-it

<sup>&</sup>lt;sup>9</sup>https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

<sup>&</sup>lt;sup>10</sup>https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

<sup>&</sup>lt;sup>11</sup>https://huggingface.co/microsoft/Orca-2-13b

<sup>&</sup>lt;sup>12</sup>https://huggingface.co/allenai/OLMo-7B-Instruct

**Prompts (WP)** Dataset. The Writing Prompts Dataset was collected by [146] using Reddit's r/WritingPrompts/ forum that contains premises or prompts for stories. The WP dataset consists of a cleaned subset of story prompts and corresponding humanwritten stories using filtration criteria such as removal of stories that are bot-generated, less than 30 words long, contain profanity, general announcements, and so on. We used the test split<sup>4</sup> of this dataset as the source of prompts for LLM generated stories. We also filter out prompts that do not have at least one corresponding human-written story that is at least 800 words long. We do this to ensure that the prompt itself does not preclude longer storylines. We chose 800 words as a criteria as a means to include stories that are slightly longer than the average of the dataset. The average length (number of words) of articles in the test set is 675.75 words. Out of 15138 total prompt-story pairs, this left us with 4623 data points. For each prompt, we divide the total goal article length (800-900 words) by the number of authors (N) to calculate the length of each part or story chunk to be written by each author, such that the writing load is distributed roughly uniformly amongst the LLM authors. We also generate different permutations of LLM authorship order such that every author can contribute to random parts of the story and we ensure that our dataset does not have any spurious correlations between LLM/author and story sections such as the beginning, or ending. For each value of the numbers of authors i.e.  $N \in \{1, 2, 3, 4, 5\}$ , we generate all possible permutations of author orders. For example, for N = 3, two examples of author order permutations could be:

> $Olmo \rightarrow Mistral \rightarrow Llama$ Gemma  $\rightarrow Llama \rightarrow Mistral$

From all such possible permutations, we sample the minimum of either total possible orders or 15 as the number of author orders. For each author order, we then generate stories using each of the prompts from a unique set of prompts per N. Our goal number of stories for each N was set to 7200 stories. A summary of the words written by each author, author order permutations, and prompts per author, as well as the pool of 5 authors and their corresponding model checkpoints used for generating all story parts is shown in Table 5.2.

<sup>&</sup>lt;sup>4</sup>https://www.kaggle.com/code/ratthachat/writingprompts-combine-one-line-data-for-gpt2/ input?select=writingPrompts

## 5.3.2 LLM prompting

For each value of N, we used different prompts to generate story parts sequentially, as detailed in Table 5.3. Utilizing the vLLM library<sup>5</sup>, we accessed and generated text from various LLMs. Initially, we conducted a pilot study to refine our prompts by generating and reviewing 100 articles. For the "Beginning" prompt, the first LLM used only the original r/WritingPrompts/ input. For subsequent parts, we found that longer input prompts reduced story length, so we used Falcon.ai summarizer<sup>6</sup> to condense the story so far into under 80 words, allowing LLMs to generate longer sequences. We also included the last sentence of the story so far for smooth continuity. Prompts for different sections only varied in their instructions to "begin", "continue", or "conclude" the story. Additionally, we added an instruction to prevent LLMs from generating extraneous instructions.

Prompt	Templates
--------	-----------

Beginning Prompt

You are a creative story writer. Write a story that starts with the prompt {starting prompt} in around  $\{n\}$  words. Do not add any instructions. Start the story as follows:

Middle Prompt

Write {n} words to continue this storyline: {summary of story so far}. Continue from this sentence: {last sentence from previous part}

Ending Prompt

Write {n} words to conclude this storyline: {summary of story so far}. Do not add any instructions. Continue from this sentence: {last sentence from previous part}

Table 5.3: Prompt templates for different parts of the story.  $\{n\}$  here denotes the number of target words for each author.

For for all  $N \ge 2$ , we provided the summary of the story so far as an input in the

<sup>&</sup>lt;sup>5</sup>https://docs.vllm.ai/en/stable/

<sup>&</sup>lt;sup>6</sup>https://huggingface.co/Falconsai

prompt. To make sure that our story parts had smooth continuity, we also used the last sentences of the story so far as input. This made sure that the generating LLM has access to the last sentence in addition to the overall storyline to continue the story as seamlessly as possible. This second input is denoted as "last sentence from previous part" in Table 5.3. Other than this, our prompt for the three types of story sections only differed in the instruction of writing either the "beginning", "continue", or "conclude" the storyline so far. We also had to add an instruction to stop the LLMs from generating any additional instructions as from our pilot study, we found that some LLMs (Orca and Llama) would often first generate a rephrasing or more detailed version of our instruction before generating the actual story content.

## 5.3.3 Dataset Cleaning

For each prompt, we gave each LLM 20 maximum attempts to re-generate that particular story part if it fell 15 or more words shorter than the goal length in the previous iteration. Despite this, we had instances of very short story parts that would have made the average article length too short or led to a very skewed representation of one LLM v/s the rest. Thus, we discarded such stories. Additionally, we were able to notice two formatting peculiarities for Gemma and LLama. Particularly, Gemma's story parts often began with a short title for the section it was to generate surrounded by "###" for example "###" The return of the Jedi ###". Llama on the other hand was appending a "The end" whenever it was its turn to write the ending part of a story. We removed all cases of these two substrings using regular expressions search and deletion as a means to unify the flow of the story across all LLMs and to make sure particular LLMs weren't identifiable only due to such formatting details. We also removed all extra spaces from the stories and any repetitions of the instructions in rare cases.

#### Examples of story parts written by Gemma followed by other authors

We inspected 100 randomly sampled stories and inspected the parts written by Gemma in search of visibly discernible features or peculiarities that might have explained the ease of its detection as compared to other LLMs. To the naked eye, this text seemed to be similar to all the other parts of the story since we subjected all the story parts to the same filtration process. We suspect that other factors might be in play here such as the author order or tone differences, and leave this exploration to future attempts (see Tables 5.12, 5.13 and 5.14 for examples of story samples).

## Collaborative Story Example: Story parts written by different authors

#### Part 1

Years ago, in the depths of a dark and desolate forest, I made a promise to a witch. Desperate for her help to save my beloved firstborn son, I agreed to offer her my firstborn in return for her healing magic. The old crone's cackling laughter still echoed in my ears as I made that fateful vow. But time passed, and my wife bore us three more children. With each birth, I was filled with relief and gratitude, knowing that our family had been spared the tragedy that had once threatened to tear us apart. Yet no matter how hard I tried, I couldn't bring myself to fulfill my promise to the witch. Now, years have passed, and the witch has grown tired of waiting. Her messages, delivered by raven messenger, are becoming more and more urgent. The ground beneath my feet seems to shake with her anger, and I can't help but fear for the consequences of my broken vow. Desperate for a solution, I turn to the only person who might be able to help me. But will she be able to save me from the wrath of the vengeful witch?

#### Part 2

Her pale, ethereal face held a cruel smile, her voice like the rustling of dry leaves in a desolate forest. "Your son's life hangs in the balance, dear one," she hissed, her voice dripping with malice. "Yet, you have failed to fulfill your part of the bargain." Panic surged through me as I realized the witch was right. I had never presented her with a child. The burden of my desperation had become a heavier chain, tethering me to an unspoken debt. Desperation fueled the flames of my fear, as I knew that if I failed, not only would my son succumb to his fate, but the wrath of the vengeful witch would descend upon me. Time seemed to warp, stretching into an agonizing eternity as I considered my options. Could I find a way to appease her, to escape the consequences of my foolish decision?

#### Part 3

I paced back and forth in my chambers, my mind racing with thoughts of the witch and the debt I owed her. I had been so desperate to save my son that I had agreed to give her my firstborn child in exchange for her healing magic. Now, years had passed, and the witch had grown tired of waiting. She had come to me, demanding that I fulfill my end of the bargain. I knew that I could not give her my son. He was all I had left of my beloved husband, and I would rather die than lose him. But I also knew that I could not defy the witch. She had the power to take my life and the lives of my children with just a wave of her hand. I spent hours pacing, trying to think of a way to appease her. But no matter what I did, I could not come up with a solution. I was trapped, and I knew it. The witch had me in her grasp, and she would not let me go. Finally, in desperation, I decided to seek out the help of a wise old woman who lived in the village.

## Part 4

A few days ago, I came to her, my heart heavy with grief and my soul wounded. She, in turn, looked at me with a kind and gentle expression. Then, with a wave of her hand, she began her spell. I watched as the darkness in my heart began to fade, and my love for my children grew stronger than ever. She whispered words of power and love, and the bond between my wife and I was restored. From that day forward, we were a family once more, united by the magic of the wise old woman.

#### Part 5

The wise old woman had always been a renowned healer in our village. Her knowledge of herbs and potions was unmatched, and many came to her seeking relief from various ailments. I was one of those who sought her help when my firstborn son fell gravely ill. His fever raged, and his tiny body was wracked with pain. Desperate for her help to save my beloved firstborn son, I agreed to give her my firstborn in exchange for her healing magic. I returned to my wife, heartbroken and ashamed, to tell her of the bargain I had made. But time passed, and my wife bore us three more children. With each birth, I was filled with relief and gratitude, knowing that our family had been spared the tragedy of losing a child. The wise old woman had honored her part of the bargain, and my son grew strong and healthy under her care. He became a skilled healer in his own right, following in the footsteps of the wise old woman who had saved his life.

Table 5.4: Example of a 5-part LLM story from **CollabStory** dataset for the prompt: "Years ago, you promised your firstborn to a witch. Since then, despite your best efforts, you can't seem to give him away. The witch is starting to get pretty mad."

## 5.3.4 Post-processing and filtering

Though we used instruction-tuned LLMs, they do not follow instructions perfectly. Though our goal number of words per story was 800-900 words, we used the upper limit to calculate the number of words each LLM should generate. From our pilot study, we found that most LLMs were undershooting their target number of words in the instruction. We also filtered out all stories in which at least one part was under 50 words long. We also removed all extra spaces from the stories and any repetitions of the instructions in rare cases. We also filtered for some additional types of noises detailed in Section 5.3.3. After this filtration, we were left with the following number of stories per  $N \in [1, 5]$ : 7164, 7070, 6093, 6955, 5221 for a total of 32, 503 stories. An example of one such story from our dataset can be read in detail in Table 5.4.

N=1 (H)	N=1 (M)	N=2	N=3	N=4	N=5
		# V	Vords		
1352.26	725.03	1090.67	1154.44	1091.99	995.42
$\pm 425.11$	$\pm 288.32$	$\pm 207.43$	$\pm 112.24$	$\pm 85.67$	$\pm 74.20$
		# Ser	ntences		
84.23	41.90	60.49	64.38	59.29	53.86
$\pm 34.85$	$\pm 61.94$	$\pm 14.49$	$\pm 11.57$	$\pm 11.24$	$\pm 9.41$
	:	# Words (AV	G) per sentenc	ce	
17.82	18.27	18.42	18.30	18.81	18.88
$\pm 28.18$	$\pm 3.92$	$\pm 2.85$	$\pm 2.45$	$\pm 2.47$	$\pm 2.51$
		Vocabula	ry richness		
$0.34 \pm 0.05$	$0.40 \pm 0.09$	0.36	0.36	0.37	$0.39 \pm 0.03$
		$\pm 0.05^{\hat{*}}$	$\pm 0.03^{\hat{*}}$	$\pm 0.03^{\hat{*}}$	
		% of st	opwords		
31.26	37.00	37.77	37.39	37.71	37.41
$\pm 4.76$	$\pm 4.77$	$\pm 3.11$	$\pm 2.67$	$\pm 2.39$	$\pm 2.39$
		Read	ability		
80.28	75.16	75.34	75.25	74.37	74.80
$\pm 33.26$	$\pm 9.83$	$\pm 8.13$	$\pm 7.80$	$\pm 8.00$	$\pm 8.14$
		Ent	ropy		
38.49	26.03	38.42	40.30	37.97	34.69
$\pm 12.25$	$\pm 13.10$	$\pm 7.93$	$\pm 4.75$	$\pm 4.07$	$\pm 3.58$
		Cohe	erence		
$0.38 \pm 0.05$	$0.46 \pm 0.06$	$0.46 \pm 0.06$	$0.46 \pm 0.04$	$0.46 \pm 0.04$	$0.46 \pm 0.04$

Table 5.5: Comparison of descriptive features for articles with number of authors  $N \in [1, 5]$ , as compared with human-written single-author stories for the same prompts. Here, "H"  $\rightarrow$  "Human"; "M"  $\rightarrow$  Machine. \* represents statistical significance (p<0.01) compared to human-written stories.

## 5.4 Dataset Analysis

We compare the LLM-generated single and multi-authored texts with the single-authored human written stories to study the relative quality of LLM generated stories as the number of authors present increases. To do this, we report the average and standard

Author	K=1	K=2	K=3	K=4	K=5
		# V	Vords		
Gemma	172.97	157.17	124.51	129.76	133.75
	$\pm 16.47$	$\pm 36.22$	$\pm 46.13$	$\pm 47.16$	$\pm 46.90$
Llama	172.51	170.88	173.35	174.76	172.23
	$\pm 19.91$	$\pm 13.28$	$\pm 15.29$	$\pm 16.18$	$\pm 19.33$
Mistral	177.25	182.24	178.09	178.82	178.15
	$\pm 12.61$	$\pm 12.88$	$\pm 22.71$	$\pm 19.47$	$\pm 22.69$
Olmo	168.01	197.91	194.89	192.64	191.41
	$\pm 8.69$	$\pm 18.93$	$\pm 23.64$	$\pm 26.60$	$\pm 30.28$
Orca	174.45	175.61	178.11	178.01	177.89
	$\pm 22.20$	$\pm 10.42$	$\pm 15.62$	$\pm 14.37$	$\pm 16.57$
		Lexical	Diversity		
Gemma	$0.67 \pm 0.04$	$0.67 \pm 0.05$	$0.70 \pm 0.07$	$0.69 \pm 0.07$	$0.68 \pm 0.06$
Llama	$0.60 \pm 0.05$	$0.61 \pm 0.05$	$0.59 \pm 0.05$	$0.58 \pm 0.05$	$0.58 \pm 0.06$
mistral	$0.64 \pm 0.04$	$0.62 \pm 0.04$	$0.61 \pm 0.05$	$0.61 \pm 0.04$	$0.61 \pm 0.05$
Olmo	$0.62 \pm 0.05$	$0.57 \pm 0.06$	$0.58 \pm 0.06$	$0.58 \pm 0.06$	$0.57 \pm 0.07$
Orca	$0.62 \pm 0.05$	$0.62 \pm 0.04$	$0.61 \pm 0.04$	$0.61 \pm 0.05$	$0.60 \pm 0.05$
		Read	ability		
Gemma	75.95	77.13	75.13	74.11	73.28
	$\pm 8.53$	$\pm 9.70$	$\pm 12.29$	$\pm 11.81$	$\pm 12.96$
Llama	83.11	82.61	82.75	80.88	80.13
	$\pm 8.35$	$\pm 8.50$	$\pm 9.00$	$\pm 10.10$	$\pm 9.19$
Mistral	81.04	83.99	81.59	82.02	79.91
	$\pm 8.58$	$\pm 8.40$	$\pm 10.07$	$\pm 9.09$	$\pm 9.44$
Olmo	80.78	83.31	81.41	80.55	80.45
	$\pm 9.01$	$\pm 9.87$	$\pm 9.81$	$\pm 10.73$	$\pm 10.20$
Orca	83.08	82.51	80.86	79.95	79.51
	$\pm 8.54$	$\pm 8.45$	$\pm 9.65$	$\pm 9.97$	$\pm 9.80$
		Cohe	erence		
Gemma	$0.49 \pm 0.07$	$0.47 \pm 0.08$	$0.47 \pm 0.08$	$0.47 \pm 0.08$	$0.48 \pm 0.08$
Llama	$0.44 \pm 0.08$	$0.47 \pm 0.08$	$0.47 \pm 0.08$	$0.47 \pm 0.08$	$0.49 \pm 0.08$
mistral	$0.46 \pm 0.07$	$0.44 \pm 0.07$	$0.45 \pm 0.07$	$0.45 \pm 0.07$	$0.46 \pm 0.07$
Olmo	$0.47 \pm 0.08$	$0.43 \pm 0.07$	$0.45 \pm 0.08$	$0.45 \pm 0.08$	$0.45 \pm 0.08$
Orca	$0.44 \pm 0.07$	$0.45 \pm 0.08$	$0.45 \pm 0.08$	$0.46 \pm 0.08$	$0.47 \pm 0.07$

Table 5.6: Descriptive Statistics or Features for stories generated by different authors for different parts of the stories. Here, "K" represents the part of the story written, i.e. K=1 corresponds to the first part of the story, K=2 referees to the second part, and so on.



Figure 5.2: N on the X-axis denotes the number of authors, and N=1(H) and N=1(M) correspond to the human-written and machine-generated single-authored texts, respectively. All other texts (N >=2) are multi-LLM generated. Y-axis shows the values of the measure shown in each subplot as mentioned in the headings. For all measures, we show the average and standard deviation for N going from 1 to 5. For all measures except vocabulary richness (3rd column, 1st row), increasing the number of authors (N) does not lead to statistically significant deviations from the human text distribution.

deviation of different measures such as the average number of (i) words, (ii) sentences, (iii) vocabulary richness, (iv) percentage of stopwords, (v) readability scores, (vi) entropy, and (vii) coherence scores using the TextDescriptives Library<sup>7</sup> for different numbers of authors (N) in Figure 5.2. We measure (vii) **creativity** using OCSAI (Open Creativity Scoring with Artificial Intelligence) [?] that provides text creativity scores using an LLM (GPT-4) fine-tuned on human annotations of creativity. From Figure 5.2, we see that across most measures, there isn't a significant deviation or decline as the number of authors is increased, i.e. collaboration or introducing multiple LLMs in the generation setting does not disturb the quality of the stories generated. In fact, the only statistically significant difference between LLM-coauthored and human-written stories in vocabulary richness measured by type-token-ratio (TTR) for  $N \in [2, 3, 4]$ . For all other measures, our dataset's machine-generated stories follow similar distributions to human-written stories. Automated measures of readability and coherence utilized the human-generated stories as the reference text, and even for these measures, we see that collaboratively generated text scores do not deviate from the reference score distribution. This analysis indicates that LLMs are able to collaboratively generate stories without any significant changes in quality through sequential prompting.

We report the average and standard deviation of number of words, sentences, vocabu-

<sup>&</sup>lt;sup>7</sup>https://github.com/HLasse/TextDescriptives

Negative Sample Source	Correct Wins	Neither Wins	Incorrect Wins
Different Story	75%	25%	0%
Same Story	35%	55.33%	9.67%

Table 5.7: Continuity evaluation of pairs of story parts using GPT-40 using different negative sample sources (either from within the same story or not). This table presents the percentage of story parts for which either only the correct part was evaluated as having continuity, or the negative sample (incorrect) or both.

lary richness, readability scores, and coherence scores using TextDescriptives Library<sup>8</sup> for all parts of the story, as well as for different numbers of authors (N) in Table 5.5. There is a statistically significant difference between LLM-coauthored and human-written stories in vocabulary richness measured by type-token-ratio (TTR) for  $N \in [2, 3, 4]$ . For all other measures, our dataset's machine-generated stories follow similar distributions to human-written stories. We thus ensure the quality of our dataset using automated measures of readability and coherence by using human-generated stories as the reference text. Detailed statistics for each LLM as well as each part of the story are provided in Table 5.6.

## 5.4.1 Story Continuity

An important aspect of collaborative storywriting is the notion of continuity. Particularly, we wanted to measure if the different story parts generated sequentially by different LLM authors followed a logical and cohesive plot. To evaluate this, we used a prompt-based evaluation using GPT-40 [82] to discern if two consecutive story parts continue a story well or not. To do this, we develop two types of pairs of story parts: one pair containing the "Correct" or the positive sample which points to the actual next part in the story, or "Incorrect" which is the negative sample that is either a random story part drawn from a different story or from the same one (but not the true next one, any other part). In other words, for the negative samples taken from within the same story, we ensure to not use the correct consecutive part as a negative sample. An example of this setup is illustrated in Figure 5.3, where given a story part, there is an actual continuation ("Correct") and a random negative sample ("Incorrect") story part drawn from either

<sup>&</sup>lt;sup>8</sup>https://github.com/HLasse/TextDescriptives



Figure 5.3: To evaluate the continuity of story parts, we sample two potential continuations to a story part, one correct and one incorrect. In this figure, we see an example of a story part on the left and two candidates to be evaluated for continuity. In green are some terms highlighted to show continuity in topics, characters, and plot lines, and in red are highlighted sections of the text that suggest discontinuity. In this example, our evaluation method deemed both these candidates as unfit continuations. For the correct or actual continuation, this was due to a plot repetition, while the negative sample is discontinuous due to a topic shift.

the same story or a different one. We then use the following prompt structure to obtain continuity evaluations for both candidate next parts and then compare if the actual continuation was the one with a better continuation evaluation. We provide GPT-40 a pair of texts at a time (out of a total of 600 pairs) and ask it the following question: "Does Part 2 serve as a good continuation of Part 1 in terms of logical flow, coherence, and consistency? Please respond with Yes or No." GPT-40 generated the binary 'yes/no" responses and an explanation. We then compare the percentage of story parts for which either the correct answer got a "Yes" and the incorrect one got a "No" response (Correct Wins), if both candidate continuations were evaluated the same (Neither Wins), or if the Incorrect answer got a "Yes" and the Correct one got a "No". We present the results in Table 5.7. We see that when the negative story parts are sampled from different stories than the one that the first part belongs to, then there are 0 cases for which the incorrect part wins. This is an expected result, as story parts that belong to different stories would have different topics, plot lines, characters, and so on. Hence, detecting this discontinuity should not be hard, as is the case. An example of a case where Neither Wins and both candidate story parts get an evaluation as being discontinuous is provided in Figure 5.3. As illustrated by this example, it is crucial to note that our prompt encourages the model to evaluate the logical or common-sense flow of the stories. Hence, topic overlap alone was insufficient to mark two parts as continuous. In the harder setting, when the negative sample is also a story part from within the same story, we see that there is a marked increase in both the Neither Wins and Incorrect Wins scenarios (see Table 5.7) where Incorrect Wins goes from 0 to 9.67% and Neither Wins has an over 30% increase. This is expected since many of the story parts were not the next immediate part but a few sequences apart. They all follow the same story and share some logical flow despite the distance in their occurrences in the story. For example, the fourth part of story might get a positive continuity evaluation, just as the second, third, and fifth would if compared with the first part of the story. Over 85% of Neither Wins story pairs had a continuous evaluation for both candidates (142 out of 166). Thus, out-of-the-box LLMs are able to follow plot lines and logic even when continuing each others' partially written parts at any point in the story.

## 5.5 Authorship Analysis: Extending PAN tasks for multi-LLM scenario

Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection, known as PAN tasks [132], have presented a persistent challenge, establishing benchmarks for analyzing multi-authored text among humans for more than 15 years. We extend the most common and repeated authorship-related tasks from the PAN multi-human-author task suite to the multi-LLM scenario. We then fine-tune and report performance using the following 5 baseline methods: Multinomial Naive Bayes (**MNB**) [147], Support Vector Machine (**SVM**) [148], **BERT** [103], **ALBERT** [149], and **RoBERTa** [65].

## 5.5.1 Task 1: Is a story written by multiple authors or not?

We randomly sample articles from the single-LLM authored stories i.e. N = 1 as the negative class v/s articles from the multi-authored settings where  $N \in [2, 3, 4, 5]$  as the positive class. We sample from the single-LLM stories to keep the class distribution equal, based on the number of articles for each N. From Table 5.8, we see that for all methods, the performance at N = 5 is higher than for N = 2, gradually increasing with the value of N. Stories that have a higher number of authors are more distinct

raon	r. manu		01 1100	
Method	N=2	N=3	N=4	N=5
MNB	0.83	0.86	<u>0.86</u>	0.87
SVM	0.78	0.8	0.83	0.82
BERT	0.85	0.85	0.83	0.87
ALBERT	0.83	<u>0.89</u>	0.9	<u>0.92</u>
RoBERTa	0.88	0.92	0.9	0.93
AVG	0.83	0.86	0.86	0.88

Task 1: Multi-author or not

Table 5.8: Performance is shown as F1-scores. AVG denotes average F1-score for each N. For each  $N \ge 2$  we evaluate the classifiers on their ability to distinguish the stories from those written with N = 1. Best performing method is in **bold** and second highest <u>underlined</u> for each N.

from single-authored ones. We conjecture that introducing more authors in the article might lead to more variations in the text, making stories with N = 5 authors most easily distinguishable from stories without any such variations i.e. N = 1.

## 5.5.2 Task 2: How many authors have written a story?

The second task is to predict the number of authors involved in generating a story. For the CollabStory dataset, this means that class labels  $\in [1, 5]$ . From Table 5.9 we see that the task of predicting exactly how many authors have co-written a story is easiest for N = 1 in conjunction with findings from Task 1 that showed that multi-authored text can be more easily distinguished from single-authored text. Thus, here too it seems to be easiest to separate the single-authored texts from  $N \ge 2$ . However, for multi-authored stories, only BERT and RoBERTa perform better than other baselines (>0.72 F1), especially for  $N \in [4, 5]$ . Overall, the performance across this task is low.

#### 5.5.3 Task 3: Authorship Verification

This is a pair-wise sentence classification task where the goal is to predict if two adjacent sentences are written by the same author or not. For this task, we used all the sentences at LLM-LLM boundaries, that is the last sentence of part i and the first sentence of part i + 1. The negative class data samples were sampled as random pairs of consecutive sentences within each story part.

1001 2.	I ICulcu	i unibei	or mut	1015 (11	)
Method	N=1	N=2	N=3	N=4	N=5
MNB	0.72	0.50	0.49	0.48	0.51
SVM	0.68	0.43	0.52	0.40	0.54
BERT	0.79	0.70	0.64	0.75	0.81
ALBERT	0.70	0.57	0.55	0.53	0.65
RoBERTa	<u>0.76</u>	<u>0.68</u>	<u>0.63</u>	<u>0.72</u>	0.74
AVG	0.73	0.58	0.57	0.58	0.65

Task 2: Predict Number of Authors (N)

Table 5.9: All scores are F1-scores. AVG denotes average F1-score for each N. This task is particularly challenging with only the single-authored stories (N = 1) being correctly classified. For all multi-authored texts, BERT and RoBERTa perform better than others. Best performing method is in **bold** and second highest <u>underlined</u> for each N.

Task 9.	numons	mp ver	meauor	1
Method	N=2	N=3	N=4	N=5
MNB	0.65	0.64	0.60	0.63
SVM	0.63	0.62	0.62	0.63
BERT	0.73	0.73	0.73	0.71
ALBERT	0.92	<u>0.89</u>	0.89	<u>0.89</u>
RoBERTa	<u>0.91</u>	0.91	0.90	0.89
AVG	0.76	0.75	0.74	0.75

Task 3: Authorship Verification

Table 5.10: All scores are F1-scores and AVG denotes average F-1 scores for each N for the task of detecting authorship boundaries between sentence pairs. We see that generally, performance is slightly higher for N = 2 across all classifiers. Best performing method is in **bold** and second highest <u>underlined</u> for each N.

From Table 5.10, we see that transformers-based fine-tuned methods perform well at this task. We also note that detecting sentence authorship boundaries seems to be slightly easier for the 2-author case than for  $N \geq 3$ .

## 5.5.4 Task 4: Authorship Attribution

Authorship Attribution involves predicting exactly who the author of a text article is. In the case of multi-LLM text, we design this task such that each data sample is homogeneous or each part is written by a single author and the classifier's task is to identify its author. From Table 5.11, we see that most of the authors seem hard to identify irrespective of the value of N, except for Gemma. We were expecting attribution

			N=1			
Method	Orca	Olmo	Llama	Mistral	Gemma	AVG
MNB	_	0.70	0.71	0.64	0.99	0.76
SVM	-	0.61	0.68	0.58	0.97	0.71
BERT	-	0.70	0.71	0.64	0.99	0.76
ALBERT	-	0.78	0.73	0.70	0.99	0.80
RoBERTa	-	<u>0.73</u>	0.70	0.68	0.99	<u>0.78</u>
			N=2			
Method	Orca	Olmo	Llama	Mistral	Gemma	AVG
MNB	0.49	0.51	0.52	0.51	0.92	0.62
SVM	0.51	0.55	0.54	0.59	0.79	0.62
BERT	0.54	0.54	0.63	0.58	0.95	<u>0.68</u>
ALBERT	0.56	0.58	0.63	0.59	0.96	0.69
RoBERTa	0.49	0.62	<u>0.60</u>	0.56	0.94	0.68
			N=3			
Method	Orca	Olmo	Lama	Mistral	Gemma	AVG
MNB	-	0.60	0.67	0.63	0.94	0.71
SVM	-	0.57	0.65	0.57	0.82	0.65
BERT	-	0.58	0.69	0.67	0.95	0.72
ALBERT	-	0.64	0.71	0.68	0.95	0.75
RoBERTa	-	0.71	0.71	0.67	0.96	0.76
			N=4			
Method	Orca	Olmo	Llama	Mistral	Gemma	AVG
MNB	-	0.58	0.65	0.63	<u>0.91</u>	0.69
SVM	-	0.58	0.67	0.59	0.80	0.66
BERT	-	0.59	0.70	0.68	0.93	0.73
ALBERT	-	0.66	0.73	0.70	0.93	0.75
RoBERTa	-	0.66	0.68	0.64	0.93	<u>0.73</u>
N=5						
Method	Orca	Olmo	Llama	Mistral	Gemma	AVG
MNB	0.54	0.54	0.56	0.53	0.86	0.61
SVM	0.56	<u>0.61</u>	0.60	0.54	0.79	0.62
BERT	0.60	0.57	<u>0.62</u>	0.54	0.93	0.65
ALBERT	0.58	0.55	0.65	0.62	<u>0.92</u>	<u>0.66</u>
RoBERTa	0.56	0.69	0.61	0.58	0.92	0.67

Table 5.11: F1-scores for identifying the author of story parts across articles written by different numbers of authors. The 5 columns show each of the labels or authors. AVG denotes average F1-scores across all authors. Best performing method is in **bold** and second highest <u>underlined</u>. 107

to be easier the fewer the number of authors since the length of the parts contributed by each author would be longer. But there does not seem to be any such correlation in our dataset i.e. length contributed by each author does not correlate with their detection. To further explore why Gemma was easily identifiable, we manually inspected a random sample of 100 articles for discernible features or peculiarities in story parts written by Gemma v/s all other authors. We provide examples of story parts generated by Gemma and other LLMs in the Tables 5.12, 5.13 and 5.14). We found that the text generated by Gemma is not noticeably different from that of other authors. We leave a deeper analysis of potential factors, such as training data-induced biases, story part, or author order for future work.

## 5.6 Conclusion

We present CollabStory, the first exclusively LLM-LLM or machine-machine collaborative story dataset, and demonstrate the tasks it enables. We find that LLMs are able to collectively generate creative stories at par with human-written stories via sequential prompting. Using this dataset, we demonstrate which multi-LLM authorship tasks are most challenging. Recent developments have significantly advanced LLM-assisted writing, sparking widespread discussions about the nature of authorship. Beyond using LLMs for paraphrasing, editing, and enhancing text, there exists an extreme scenario where text is generated entirely by multiple LLMs. Our work addresses this extreme case, raising several nuanced authorship concerns: Who should be considered the true creative source in such a situation? Should all LLMs involved be credited? Or should the human developers designing the prompts be acknowledged as the primary authors? Moreover, should the LLM that contributed the most—whether in terms of word count, narrative depth, or plot twists—be granted greater ownership? We will soon need "Catch As Catch Can" methods to not only find all points where authorship changes within an article but also simultaneously attribute each independent segment to the specific LLM author. As more and more LLMs are becoming easier to access, malicious actors could combine texts from different LLMs to evade automated and in-built misinformation flaggers, or students might circumvent credibility checks by having different LLMs write different sections of an academic article. Thus, CollabStory has been developed as a resource with long-form stories written by multiple LLMs to support the development and expansion of tasks and methods that can help address incoming challenges brought by LLM-LLM interactions.

Gemma	Not Gemma
The chase has begun, and adrenaline floods his veins. The blinding head- lights pierce through the night, illu- minating the frantic dance of tires on asphalt. The adrenaline-fueled chase weaves through the labyrinth of city streets, sirens screaming like tor- mented spirits. The driver, hands white on the wheel, desperately maneuvers, adrenaline masking the fear gnawing at his conscience. Every corner, every al- leyway, feels like a potential trap, each sudden movement a potential assault. The police officer, unwavering in his pur- suit, his seasoned instincts guiding him through the chaos. He maneuvers his cruiser with precision, anticipating the driver's next move. His heart pounds in his chest, adrenaline battling with the growing sense of urgency. He knows the city like the back of his hand, weaving through traffic, his eyes glued to the fleeing vehicle. As the chase intensifies, the driver's desperation escalates. He risks collisions, weaving through pedes- trians, pushing the limits of his vehicle's performance. The officer remains calm, focused, his training taking over. He knows he must bring this chase to an end, but he also understands the risks involved.	Swallowing his fear, he decides to take the next exit and lead his pursuers through the winding, narrow streets of the old town. The streets are alive with the sound of market vendors calling out their wares, children laughing, and the faint sounds of a distant band playing music. The narrow, cobblestone alleys are lined with ancient buildings, each one more beautiful than the last. He knows that this labyrinth of streets will give him an advantage, but it also in- creases the risk of colliding with a pedes- trian or another car. Despite the danger, he continues to navigate the maze of alleys, turning sharply at each intersec- tion. He can hear the revving of engines and the screeching of tires behind him, but he refuses to look back. He knows that every twist and turn could be his last, but he is determined to outsmart his pursuers and escape. As he rounds the final corner, he sees the safety of the forest beyond the old town walls. He grits his teeth, puts the pedal to the metal, and bursts through the gates, leaving the chaos of the chase behind him.

Table 5.12: Example 1: As can be seen from these two consecutive story parts, the text written by Gemma is not necessarily different from that of other authors (in the second column). Thus, from our manual inspection, we were unable to attribute any noticeable features to the higher authorship attribution performance for Gemma.

Table 5.13: Example 2: As can be seen from these two consecutive story parts, the text written by Gemma is not necessarily different from that of other authors (in the second column). Thus, from our manual inspection, we were unable to attribute any noticeable features to the higher authorship attribution performance for Gemma.

## 5.7 Discussion

Recent developments have significantly advanced LLM-assisted writing, sparking widespread discussions about the nature of authorship. Beyond using LLMs for paraphrasing, editing, and enhancing text, there exists an extreme scenario where text is generated entirely by multiple LLMs. Our work addresses this extreme case, raising several nuanced authorship

Gemma	Not Gemma
In the unspoken words hanging be- tween us, Sarah and I understand that the journey to healing will be arduous. There will be moments of doubt, days when the weight feels unbearable, when the waves of sadness threaten to pull us under. But we are not in this to- gether. We are stronger than our indi- vidual struggles. Together, we will learn to navigate the storms, to embrace the sunshine, to rediscover who we are be- yond the baggage we once carried.	Sarah's smile brightens my day, and I know that this new chapter in our lives is going to be filled with adven- ture, growth, and most importantly, love. We'll stumble, we'll trip, but we'll always have each other to pick us back up and move forward, stronger than be- fore. I take Sarah's hand and squeeze it tightly.
Her words seemed to hang in the air like acrid smoke, but I knew better. The voice, the chilling certainty in its de- livery, had belonged to something far more sinister. I glanced at the others, their faces pale, their eyes wide with disbelief. "There's no way this is real," one whispered, her voice trembling. But the undeniable presence of death in our living room told us otherwise. The fig- ure shifted, its skeletal hands brushing the floor with an unholy whisper. "Your time has come," it hissed, its voice like nails on a chalkboard. Panic surged through me, my limbs frozen in terror. The others stumbled around, scram- bling for escape, but their movements were clumsy, their screams muted by fear. In that moment, I realized that the only way out was through the front door, directly into the path of the crea- ture that had come for me. With a surge of adrenaline, I broke free from the group sprinting towards the exit	As I reached the door, it swung open to reveal Tia, the figure from my night- mares, standing there. But instead of fear, I felt relief. She had a warm smile on her face, and her eyes were full of compassion. "You have been chosen to bear the mark of the guardian," she ex- plained, holding out the hourglass. "I have protected this town for generations, and now it's your turn to carry on the tradition." I hesitated for a moment, but then I accepted the hourglass with a sense of pride and purpose. Together, Tia and I faced the darkness that lurked outside, ready to protect our town and its people from the evil that sought to harm them. From that day on, I be- came the guardian of my town, using the powers granted to me by Tia to keep the peace and protect those around me. The people of the town were grateful, and I knew that I had found a new pur- pose in life.

Table 5.14: Example 3: As can be seen from these two consecutive story parts, the text written by Gemma is not necessarily different from that of other authors (in the second column). Thus, from our manual inspection, we were unable to attribute any noticeable features to the higher authorship attribution performance for Gemma.

PAN Task equiv- alent	Task Description: Multi- LLM scenario	Real-world Implications
Predict multi- author or not	To determine if a text in- cludes content from multiple LLMs or not	Credit Assignment and In- tellectual Property (IP) reg- ulation
Predict number of authors	To predict the number of LLMs involved in writing an article	Keeping track of LLM-LLM agent interactions in grow- ing open-source market
Author Verifica- tion	To detect when authorship switches between LLMs	To detect perjury, misinfor- mation injection, falsifying editing in news articles, and text obfuscation
Authorship At- tribution	Predicting who wrote each text segment?	Plagiarism detection
Style Change De- tection and Attri- bution	Finding all positions in the text where authorship changes and who wrote each segment	Classroom settings: Aca- demic Integrity, detecting use of multiple open-source and free-to-use LLMs to sur- pass detection methods

Table 5.15: Real-world implications of the tasks involved in understanding LLM-LLM collaboration for writing tasks

concerns: Who should be considered the true creative source in such a situation? Should all LLMs involved be credited? Or should the human developers designing the prompts be acknowledged as the primary authors? Moreover, should the LLM that contributed the most—whether in terms of word count, narrative depth, or plot twists—be granted greater ownership? There have been recent works addressing this question of ownership. For example, [150] and [151] found that humans felt a sense of higher ownership when they wrote longer prompts and when the AI assistant-generated text length was shorter, respectively. The question of authorship when text is repeatedly paraphrased using LLMs has also been deeply considered by [152], who find that the notion of authorship is task-dependent and cannot be generalized.

These questions have profound implications for various stakeholders in the burgeoning socio-technical system of generative AI. Our research introduces authorship-related tasks using CollabStory, which can help address these concerns by accurately discerning the

usage of multiple LLMs in texts. Our extension of PAN-inspired authorship tasks is closely linked to real-world implications, as follows:

#### Task 1: Predict multi-author or not

In the rapidly expanding and fiercely competitive market for LLMs, the ownership of content and the ability to prove the origins of creative work are becoming increasingly crucial. As the market evolves, closed-source LLMs are implementing stricter regulations and demanding credit assignment under various distribution licensing norms. In this context, the capacity to demonstrate that a text incorporates generated output from multiple LLMs is essential. This capability can effectively prevent any single stakeholder or developer from erroneously claiming exclusive rights to the content, thereby bolstering the defense against wrongful intellectual property (IP) claims.

#### Task 2: Predict number of authors

Predicting the exact number of LLMs involved in the writing process can help keep track of the frequency and extent to which LLMs are used collaboratively, as more and more models enter the open-source market. This is essential to understand whether such usage improves task performance or introduces inefficiencies beyond a certain threshold. Understanding the optimal number of LLMs or the degree to which LLMs can leverage each other's strengths in writing tasks is vital. It ensures effective collaboration without unnecessary complexity, maximizing the benefits of combined model capabilities while avoiding overkill and collaboration for its own sake.

#### Task 3: Author Verification

With LLMs increasingly paraphrasing and editing each other's texts, it becomes crucial to identify which spans were generated by different LLMs. Consider a scenario where a news article is paraphrased by one LLM and subsequently edited by another, with the latter introducing fallacies or misinformation. In such cases, discerning the contributions of each LLM is essential for identifying malicious LLM agents or the infiltration of critical content, such as media and news articles. This capability has significant applications, including detecting perjury and combating the adversarial obfuscation of text, thus maintaining the integrity and reliability of information.

#### Task 4: Authorship Attribution

Identifying the exact LLMs responsible for authoring a text is crucial for detecting and addressing plagiarism. This is particularly important in academic settings, where students might use closed-source LLMs without complying with content ownership and usage declaration regulations. This is possible also in cases where content from one LLM is being posed as that from another to claim higher ability or quality. An example of such a situation might be in a bid to motivate financial investors hoping to monetize and utilize LLMs for specific domains (such as medical applications, educational tools, and creative content generation).

In this work, we present CollabStory, the first exclusively LLM-LLM or machinemachine collaborative story dataset, and demonstrate the tasks it enables. We elaborate on why multi-LLM authorship tasks are crucial to study by discussing their real-world implications (summarized in Table 5.15). We will soon need "Catch As Catch Can" methods to not only find all points where authorship changes within an article (style change detection) but also simultaneously attribute each independent segment to the specific LLM author (attribution). As more and more LLMs are becoming easier to access, malicious actors could combine texts from different LLMs to evade automated and in-built misinformation flaggers, or students might circumvent credibility checks by having different LLMs write different sections of an academic article. Thus, CollabStory was developed as a resource with long-form stories written by multiple LLMs to support the development and expansion of tasks and methods that can help address incoming challenges brought by LLM-LLM interactions.

This work demonstrates one way of collecting a collaborative multi-LLM dataset. However, several variants are possible. Of course, as the LLM space is ever evolving, newer LLMs (e.g. Llama 3) became available as we were already collecting this dataset. Another aspect is that our dataset was collected in a uniform manner such that all LLMs contributed somewhat equal portions of text to a story. The next step would be to train a routing algorithm or a randomizer that could generate non-uniform collaborative texts. Our current analysis is unable to account for this setting and we leave this for future work. Additionally, the iterative generation process is resource-intensive and not easily scalable. We also acknowledge that LLM tasks beyond story writing are essential for a deeper understanding of how LLMs collaborate in open-ended generation tasks.

Using LLMs for creative story writing could relay some of the biases and harmful stereotypes present in the LLMs original training data since all our LLMs are trained on data from the internet. This is an important consideration before or during the dissemination of any such generated texts or stories. Transparency of the source of generated articles is important to avoid deception or wrongful content attribution. With creative writing tasks, it is also important to address any impact on creative professionals and guidelines to ensure that LLMs help enhance rather than undermine human creativity. We study LLM story-writing as a means to better prepare for a future of LLM-generated creative texts that might be misused in classroom settings, to manipulate public opinion on social media forums, and also to protect human writers against plagiarism amongst many other potential non-ethical usages.

## Chapter 6 Conclusion

Authorship is of growing importance and becoming a more and more urgent topic of interest as newer LLMs surpass previous abilities and quality of text generation in every iteration. In this thesis, we delve into the nuances of authorship-related research questions that can help inform methods to tackle some of the challenges of detection and obfuscation as well as resource-development to support future work to address emerging LLM- or agent-like collaborative scenarios. The key findings of our work are as follows:

# 6.1 Human and machine-generated texts have differences in surprisal distributions

When we compared the human and machine-written texts using their surprisal distributions i.e. UID scores, we found that under most decoding settings, machine text is far more uniform than human-generated texts (Chapter 2). Contrary to our initial expectations, we also found that non-uniformity in machine-generated responses is a more desirable characteristic than uniformity. Despite the lack of correlation between UID scores and surprisal with human judgments overall, we observed that UID scores become relevant when controlled for surprisal, indicating that uniformity might not be the optimal goal for regularizing decoding algorithms [96]. This implies that focusing on non-uniform information density could address the "likelihood trap" problem, wherein models produce lower quality text when sampling from the extremes of their likelihood space [68]. Therefore, we propose that decoding algorithms should be adjusted to mimic the information density patterns found in human-generated texts to improve response quality across various likelihood ranges.

# 6.2 Surprisal as a means to detect machine-generated text

To tackle the challenge of detecting machine-generated text, we introduced GPT-who, a statistical-based detector inspired by the Uniform Information Density (UID) principle. This principle posits that humans distribute information or surprisal evenly in language. Our detector leverages UID-based features, including the variance of surprisals and uniformity metrics derived from the surprisal sequences generated by a standard language model. In Chapter 3, we demonstrate that these features prove highly effective in distinguishing machine-generated text from human-generated text by capturing the subtleties in information distribution. This approach offers valuable insights for enhancing text authenticity assessments.

Our analysis shows that human-written text generally exhibits more non-uniformity compared to machine-generated text, which tends to distribute information more smoothly. This observation aligns with the operationalization of UID based on our current methods, though it does not directly support or refute the UID hypothesis rooted in psycholinguistic studies. The unexpected findings may arise from using GPT-2's probability distribution, which might not accurately represent human probability distributions. Thus, while our operationalization shows machines as more uniform, it does not necessarily contradict UID theory but highlights the complexity of measuring uniformity in texts.

This finding underscores the utility of UID-based features in distinguishing between texts produced by humans and various language models. GPT-who is this a psycholinguistically-inspired, domain-agnostic text detector that (1) outperforms statistical approaches across four large-scale benchmark datasets from over 35 LLMs and multiple domains, (2) generalizes better to out-of-distribution datasets compared to stateof-the-art detectors, (3) is computationally efficient, avoiding the need for fine-tuning or training additional language models and, (4) provides interpretability through its psycholinguistically motivated UID-based feature space [153].

Although our detector may not surpass fine-tuned transformer models in performance, it offers nearly comparable results at lower computational costs and is one of the few statistical-based detectors capable of operating effectively in multi-author settings. This demonstrates the potential of psycholinguistically grounded approaches in addressing challenges posed by advanced automated text generators.

While we demonstrate the effectiveness of UID-based approaches to evaluate and detect machine-generated texts, the application of UID-based methods to broader issues such as misinformation and plagiarism remains unexplored, and we anticipate that this feature space could also provide much insights for such tasks in future studies.

Furthermore, the inherent limitations of AI-based text detectors, including the risk of false positives, highlight the need for human oversight in real-world applications. Ethical considerations, particularly in academic settings, are crucial to balance automated detection with maintaining academic integrity. We advocate for the responsible development and implementation of such systems to ensure fairness, equity, and respect for individual expression, promoting a more balanced integration of automated text detection technologies.

## 6.3 Syntactic patterns can help obfuscate authorship

We presented ALISON in Chapter 4, an innovative technique for authorship obfuscation that focuses on the replacement of revealing stylistic sequences. Utilizing a greedy algorithm, ALISON replaces text sequences that match POS n-grams identified as significant through a lightweight neural network trained to perform authorship attribution with mixed n-grams. We apply ALISON to three state-of-the-art transformer-based attribution classifiers and showcase its effectiveness by achieving superior obfuscation success and semantic preservation compared to seven diverse baseline methods. The simplicity and effectiveness of ALISON also result in a marked improvement in computational complexity over existing methods [154].

While authorship obfuscation techniques like ALISON offer crucial benefits by safeguarding the freedom of speech for individuals such as whistleblowers and journalists, they also raise ethical concerns regarding their potential misuse by malicious groups seeking to conceal their identities. We recognize these ethical implications and emphasize the importance of continued research and system design to balance the protection of free speech with mitigating risks associated with malicious applications. Our work highlights the need for robust, ethically sound solutions that enhance public discourse while addressing the challenges posed by the potential misuse of obfuscation technologies.

## 6.4 Collaborative writing through LLM prompting

Recent advancements in LLM-assisted writing have prompted significant discussions regarding authorship, particularly in extreme scenarios where multiple LLMs collaboratively generate text. This raises complex questions about the true source of creative work: Should all LLMs involved be credited, or should the human developers who designed the prompts be acknowledged? Additionally, should the LLM that contributed the most—whether by word count, narrative depth, or plot twists—receive greater ownership? Previous studies, such as [150] and [151], have explored related questions, revealing that human ownership perception varies with prompt length and AI-generated text length. Similarly, [152] found that authorship is task-dependent and cannot be generalized.

Our work introduces a novel approach to addressing these authorship concerns through CollabStory, which facilitates the study of multi-LLM authorship tasks (Chapter 5). This is the first exclusively LLM-LLM collaborative story dataset that used simple sequential prompting to get up to 5 different LLMs to collaboratively write creative stories. This dataset is designed to support the development of methods that address challenges arising from multi-LLM interactions [155]. As the landscape of LLMs continues to evolve, it is crucial to develop methods to track and attribute text contributions accurately, especially as newer models and collaborative approaches emerge. We present several critical tasks that highlight the importance of understanding and managing authorship in the evolving landscape of generative AI.

Our current dataset includes uniformly contributed text segments from multiple LLMs, but future work should explore non-uniform collaborative texts and address scalability issues. Furthermore, the iterative generation process employed is resource-intensive, highlighting the need for more efficient methods. While our focus is on story writing, extending these techniques to other creative and open-ended tasks is essential for a comprehensive understanding of LLM collaboration.

Lastly, the use of LLMs for creative writing raises concerns about potential biases and harmful stereotypes inherent in training data. Ensuring transparency and ethical considerations in the dissemination of generated texts is crucial to avoid deception and protect human creativity. This thesis aims to prepare for future challenges posed by LLM-generated texts and to safeguard against unethical uses, ensuring that LLMs enhance rather than undermine human creativity and integrity in various applications.

## 6.5 Open Challenges

The following are some open challenges that require future efforts to improve machinegenerated text evaluation, authorship analysis, attribution, obfuscation, and text generation at large:

## 6.5.1 Datasets that showcase LLM abilities

As LLMs become better at tasks and expand to newer tasks (e.g., code generation, reasoning, and qualitative evaluation of texts), it is crucial to develop datasets that are representative of their varied behaviors with diverse examples of LLM-generated responses across task types. Current benchmarks, including those introduced in this dissertation, provide data for only one type of generation task (for example, dialog response and creative story generation). There is a need for comprehensive and aggregated datasets that include different tasks to help the research community understand and evaluate LLMs better and holistically, and to keep track of their growing impact temporally as newer tasks emerge.

## 6.5.2 Evaluation Measures and Human Alignment

Evaluation measures also need to keep up with the sophistication of LLM outputs. For example, better metrics must help evaluate the common-sense rationale, truthfulness, and biases of generated texts. Moreover, as LLMs generate longer-form texts and work collaboratively, we also need to develop better metrics for assessing the continuity, flow and cohesiveness of such emerging collaborative texts. There also needs to be continued work to measure the human perception of generated texts as a means to evaluate and better align LLM generation to human preferences.

## 6.5.3 Detection and Obfuscation of multi-lingual texts

Most LLMs are now multilingual, and a vast body of literature on multilingual and multimodal texts (such as image-text caption pairs) is also emerging [156–158]. Despite this, most methods for detection and obfuscation are still limited to the mono-lingual setting, with skewed efforts focused on English language texts. Thus, current authorship attribution and obfuscation methods need to be adapted to handle this scenario and be able to assess the veracity of the sources of data extending to multiple languages and modalities.

## 6.5.4 Need for interpretable methods

There is a need for interpretable methods that can help support humans in understanding and detecting if a text is not truly written by an author claiming it (i.e., if a text has been obfuscated or wrongfully claimed), in assisting them in detecting deepfake texts, and as a means to warn the more extensive online audience of potentially harmful or ill-meaning authors in online forums. Such methods should also be integrated and deployed on online forums where deepfake texts are commonly found, especially in digital spaces that concern critical domains such as political discourse, news media, and scientific knowledge dissemination.

## 6.6 Ethical Considerations

This dissertation includes materials and examples of texts generated by large language models (LLMs). While these models produce coherent and contextually relevant outputs, their generated content may not always be factually accurate or truthful. As such, it is crucial to evaluate the information presented critically, cross-check it against reliable sources, and verify its authenticity before considering broader dissemination. The inherent limitations of LLMs, including potential biases, inaccuracies, and the risk of generating misleading information, should be considered during analysis to ensure responsible usage and interpretation. In addition, all the methods and datasets developed as part of this dissertation were limited to the English language, and the findings cannot be generalized to other languages.

## Bibliography

- LIU, Y., Z. ZHANG, W. ZHANG, S. YUE, X. ZHAO, X. CHENG, Y. ZHANG, and H. HU (2023) "ArguGPT: evaluating, understanding and identifying argumentative essays generated by GPT models," arXiv preprint arXiv:2304.07666.
- [2] YEH, C., G. RAMOS, R. NG, A. HUNTINGTON, and R. BANKS (2024) "Ghost-Writer: Augmenting Collaborative Human-AI Writing Experiences Through Personalization and Agency," arXiv preprint arXiv:2402.08855.
- [3] KIM, J., S. SUH, L. B. CHILTON, and H. XIA (2023) "Metaphorian: Leveraging Large Language Models to Support Extended Metaphor Creation for Science Writing," in *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, pp. 115–135.
- [4] ZHONG, W., J. NARADOWSKY, H. TAKAMURA, I. KOBAYASHI, and Y. MIYAO (2023) "Fiction-Writing Mode: An Effective Control for Human-Machine Collaborative Writing," in *Proceedings of the 17th Conference of the European Chapter of* the Association for Computational Linguistics, pp. 1752–1765.
- [5] SINGH, N., G. BERNAL, D. SAVCHENKO, and E. L. GLASSMAN (2023) "Where to hide a stolen elephant: Leaps in creative writing with multimodal machine intelligence," ACM Transactions on Computer-Human Interaction, 30(5), pp. 1–57.
- [6] YANG, D., Y. ZHOU, Z. ZHANG, T. J.-J. LI, and L. RAY (2022) "AI as an Active Writer: Interaction strategies with generated text in human-AI collaborative fiction writing," in *Joint International Conference on Intelligent User Interfaces* Workshops: APEx-UI, HAI-GEN, HEALTHI, HUMANIZE, TEXSS, SOCIALIZE (IUI-WS 2022), CEUR-WS Team, pp. 56–65.
- [7] YUAN, A., A. COENEN, E. REIF, and D. IPPOLITO (2022) "Wordcraft: story writing with large language models," in 27th International Conference on Intelligent User Interfaces, pp. 841–852.
- [8] LEE, M., P. LIANG, and Q. YANG (2022) "Coauthor: Designing a humanai collaborative writing dataset for exploring language model capabilities," in *Proceedings of the 2022 CHI conference on human factors in computing systems*, pp. 1–19.

- [9] CLARK, E., A. S. ROSS, C. TAN, Y. JI, and N. A. SMITH (2018) "Creative writing with a machine in the loop: Case studies on slogans and stories," in 23rd International Conference on Intelligent User Interfaces, pp. 329–340.
- [10] DUGAN, L., D. IPPOLITO, A. KIRUBARAJAN, S. SHI, and C. CALLISON-BURCH (2023) "Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 12763–12771.
- [11] CLARK, E., T. AUGUST, S. SERRANO, N. HADUONG, S. GURURANGAN, and N. A. SMITH (2021) "All That's 'Human'Is Not Gold: Evaluating Human Evaluation of Generated Text," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7282–7296.
- [12] IPPOLITO, D., D. DUCKWORTH, C. CALLISON-BURCH, and D. ECK (2020) "Automatic Detection of Generated Text is Easiest when Humans are Fooled," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1808–1822.
- [13] DUGAN, L., D. IPPOLITO, A. KIRUBARAJAN, and C. CALLISON-BURCH (2020) "RoFT: A Tool for Evaluating Human Detection of Machine-Generated Text," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 189–196.
- [14] IRTIZA TRIPTO, N., S. VENKATRAMAN, D. MACKO, R. MORO, I. SRBA, A. UCHENDU, T. LE, and D. LEE (2023) "A Ship of Theseus: Curious Cases of Paraphrasing in LLM-Generated Texts," arXiv e-prints, pp. arXiv-2311.
- [15] CELIKYILMAZ, A., E. CLARK, and J. GAO (2020) "Evaluation of text generation: A survey," arXiv preprint arXiv:2006.14799.
- [16] GEHRMANN, S., E. CLARK, and T. SELLAM (2023) "Repairing the Cracked Foundation:: A Survey of Obstacles in Evaluation Practices for Generated Text," *Journal of Artificial Intelligence Research*, 77.
- [17] GAO, R., H. E. MERZDORF, S. ANWAR, M. C. HIPWELL, and A. SRINIVASA (2024) "Automatic assessment of text-based responses in post-secondary education: A systematic review," *Computers and Education: Artificial Intelligence*, p. 100206.
- [18] CHOI, J. S. and S. A. CROSSLEY (2022) "Advances in readability research: A new readability Web app for English," in 2022 International Conference on Advanced Learning Technologies (ICALT), IEEE, pp. 1–5.
- [19] WADUD, M. A. H. and M. R. H. RAKIB (2021) "Text coherence analysis based on misspelling oblivious word embeddings and deep neural network," *International Journal of Advanced Computer Science and Applications*, 12(1).

- [20] JARVIS, S. (2013) "Defining and measuring lexical diversity," Vocabulary Knowledge: Human ratings and automated measures, 47, p. 13.
- [21] COLLA, D., M. DELSANTO, M. AGOSTO, B. VITIELLO, and D. P. RADICIONI (2022) "Semantic coherence markers: The contribution of perplexity metrics," *Artificial Intelligence in Medicine*, **134**, p. 102393.
- [22] KANEKO, M., Y. SAKAIZAWA, and M. KOMACHI (2017) "Grammatical error detection using error-and grammaticality-specific word embeddings," in *Proceedings* of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 40–48.
- [23] AMIDEI, J., P. PIWEK, and A. WILLIS (2018) "Rethinking the agreement in human evaluation tasks," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3318–3329.
- [24] VAN DER LEE, C., A. GATT, E. VAN MILTENBURG, and E. KRAHMER (2021) "Human evaluation of automatically generated text: Current trends and best practice guidelines," *Computer Speech & Language*, 67, p. 101151.
- [25] HOWCROFT, D. M., A. BELZ, M. CLINCIU, D. GKATZIA, S. A. HASAN, S. MA-HAMOOD, S. MILLE, E. VAN MILTENBURG, S. SANTHANAM, and V. RIESER (2020) "Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions," in 13th International Conference on Natural Language Generation 2020, Association for Computational Linguistics, pp. 169–182.
- [26] ISOZAKI, H., T. HIRAO, K. DUH, K. SUDOH, and H. TSUKADA (2010) "Automatic evaluation of translation quality for distant language pairs," in *Proceedings of the* 2010 conference on empirical methods in natural language processing, pp. 944–952.
- [27] BANERJEE, S. and A. LAVIE (2005) "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the* acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp. 65–72.
- [28] ZHANG, T., V. KISHORE, F. WU, K. Q. WEINBERGER, and Y. ARTZI (2019) "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*.
- [29] UCHENDU, A., V. MIKHAILOV, J. LEE, S. VENKATRAMAN, T. SHAVRINA, and E. ARTEMOVA (2022) "Tutorial on artificial text detection," in 15th International Conference on Natural Language Generation (INLG): Tutorial.
- [30] UCHENDU, A., S. VENKATRAMAN, T. LE, and D. LEE (2024) "Catch Me If You GPT: Tutorial on Deepfake Texts," in *Proceedings of the 2024 Conference of the* North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts), pp. 1–7.

- [31] CHAKRABARTY, T., V. PADMAKUMAR, and H. HE (2022) "Help me write a Poem
   Instruction Tuning as a Vehicle for Collaborative Poetry Writing," in *Proceedings*of the 2022 Conference on Empirical Methods in Natural Language Processing
  (Y. Goldberg, Z. Kozareva, and Y. Zhang, eds.), Association for Computational
  Linguistics, Abu Dhabi, United Arab Emirates, pp. 6848–6863.
  URL https://aclanthology.org/2022.emnlp-main.460
- [32] OPENAI (2023), "ChatGPT: Optimizing Language Models for Dialogue," https: //openai.com/blog/chatgpt, accessed: 2024-09-29.
- [33] FRANK, A. F. and T. F. JAEGER (2008) "Speaking rationally: Uniform information density as an optimal strategy for language production," in *Proceedings of the annual meeting of the cognitive science society*, vol. 30.
- [34] XU, Y. and D. REITTER (2018) "Information density converges in dialogue: Towards an information-theoretic model," *Cognition*, (170), pp. 147–163.
- [35] JAEGER, T. F. (2010) "Redundancy and reduction: Speakers manage syntactic information density," *Cognitive psychology*, **61**(1), pp. 23–62.
- [36] MAHOWALD, K., E. FEDORENKO, S. T. PIANTADOSI, and E. GIBSON (2013) "Info/information theory: Speakers choose shorter words in predictive contexts," *Cognition*, **126**(2), pp. 313–318.
- [37] TILY, H. and S. PIANTADOSI (2009) "Refer efficiently: Use less informative expressions for more predictable meanings," *Proceedings of the Workshop on the Production of Referring Expressions: Bridging the gap between Computational and Empirical approaches to Reference.*
- [38] SHANNON, C. E. (1948) "A mathematical theory of communication," The Bell System Technical Journal, 27(3), pp. 379–423.
- [39] XU, Y. and D. REITTER (2016) "Entropy converges between dialogue participants: Explanations from an information-theoretic perspective," Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1, pp. 537–546.
- [40] JAEGER, T. F. and R. P. LEVY (2007) "Speakers optimize information density through syntactic reduction," in Advances in Neural Information Processing Systems, pp. 849–856.
- [41] MANIN, D. (2006) "Experiments on predictability of word in context and information rate in natural language," arXiv preprint cs/0612136.
- [42] PIANTADOSI, S. T., H. TILY, and E. GIBSON (2011) "Word lengths are optimized for efficient communication," *Proceedings of the National Academy of Sciences*, 108(9), pp. 3526–3529.

- [43] FENK, A. and G. FENK (1980) "Constancy in Short-term Memory-constancy in Linguistic Information Flow," *Journal for Experimental and Applied Psychology*, 27(3), pp. 400–414.
- [44] MEISTER, C., T. PIMENTEL, G. WIHER, and R. COTTERELL (2022) "Typical decoding for natural language generation," *arXiv preprint arXiv:2202.00666*.
- [45] WEI, J., C. MEISTER, and R. COTTERELL (2021) "A Cognitive Regularizer for Language Modeling," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 5191–5202.
- [46] MEISTER, C., R. COTTERELL, and T. VIEIRA (2020) "If beam search is the answer, what was the question?" in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, pp. 2173-2185. URL https://www.aclweb.org/anthology/2020.emnlp-main.170
- [47] LEVY, R. and T. F. JAEGER (2007) "Speakers optimize information density through syntactic reduction," Advances in neural information processing systems, 19, p. 849.
- [48] HALE, J. (2003) "The information conveyed by words in sentences," Journal of Psycholinguistic Research, 32(2), pp. 101–123.
- [49] AYLETT, M. and A. TURK (2004) "The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech," *Language and speech*, **47**(1), pp. 31–56.
- [50] HOLTZMAN, A., J. BUYS, L. DU, M. FORBES, and Y. CHOI (2020) "The Curious Case of Neural Text Degeneration," in *International Conference on Learning Representations*. URL https://openreview.net/forum?id=rygGQyrFvH
- [51] ZHANG, H., D. DUCKWORTH, D. IPPOLITO, and A. NEELAKANTAN (2021) "Trading Off Diversity and Quality in Natural Language Generation," in *Proceedings* of the Workshop on Human Evaluation of NLP Systems (HumEval), Association for Computational Linguistics, Online, pp. 25–33. URL https://www.aclweb.org/anthology/2021.humeval-1.3
- [52] NADEEM, M., T. HE, K. CHO, and J. GLASS (2020) "A Systematic Characterization of Sampling Algorithms for Open-ended Language Generation," in Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Suzhou, China, pp. 334–346. URL https://www.aclweb.org/anthology/2020.aacl-main.36

- [53] GOLOVANOV, S., R. KURBANOV, S. NIKOLENKO, K. TRUSKOVSKYI, A. TSELOUSOV, and T. WOLF (2019) "Large-Scale Transfer Learning for Natural Language Generation," in *Proceedings of the 57th Annual Meeting of the Association* for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, pp. 6053–6058. URL https://www.aclweb.org/anthology/P19-1608
- [54] OLUWATOBI, O. and E. MUELLER (2020) "DLGNet: A Transformer-based Model for Dialogue Response Generation," in *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, Association for Computational Linguistics, Online, pp. 54–62. URL https://www.aclweb.org/anthology/2020.nlp4convai-1.7
- [55] FAN, A., M. LEWIS, and Y. DAUPHIN (2018) "Hierarchical Neural Story Generation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Melbourne, Australia, pp. 889–898. URL https://www.aclweb.org/anthology/P18-1082
- [56] LI, J., M. GALLEY, C. BROCKETT, J. GAO, and B. DOLAN (2016) "A Diversity-Promoting Objective Function for Neural Conversation Models," in *Proc. of NAACL-HLT*.
- [57] MEISTER, C., T. PIMENTEL, P. HALLER, L. JÄGER, R. COTTERELL, and R. LEVY (2021) "Revisiting the Uniform Information Density Hypothesis," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 963–980.
- [58] RADFORD, A., J. WU, R. CHILD, D. LUAN, D. AMODEI, and I. SUTSKEVER (2019) "Language models are unsupervised multitask learners," *OpenAI blog*, 1(8), p. 9.
- [59] ZHANG, S., E. DINAN, J. URBANEK, A. SZLAM, D. KIELA, and J. WESTON (2018) "Personalizing Dialogue Agents: I have a dog, do you have pets too?" in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, pp. 2204–2213. URL https://www.aclweb.org/anthology/P18-1205
- [60] DINAN, E., V. LOGACHEVA, V. MALYKH, A. MILLER, K. SHUSTER, J. UR-BANEK, D. KIELA, A. SZLAM, I. SERBAN, R. LOWE, ET AL. (2020) "The second conversational intelligence challenge (convai2)," in *The NeurIPS'18 Competition*, Springer, pp. 187–208.
- [61] JAIN, A., V. SINGH, S. RANJAN, R. RAJKUMAR, and S. AGARWAL (2018) "Uniform Information Density Effects on Syntactic Choice in Hindi," in *Proceedings of*
the Workshop on Linguistic Complexity and Natural Language Processing, Association for Computational Linguistics, Santa Fe, New-Mexico, pp. 38-48. URL https://www.aclweb.org/anthology/W18-4605

- [62] PAPINENI, K., S. ROUKOS, T. WARD, and W.-J. ZHU (2002) "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318.
- [63] POPOVIĆ, M. (2015) "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Association for Computational Linguistics, Lisbon, Portugal, pp. 392–395. URL https://www.aclweb.org/anthology/W15-3049
- [64] SELLAM, T., D. DAS, and A. PARIKH (2020) "BLEURT: Learning Robust Metrics for Text Generation," in *Proceedings of the 58th Annual Meeting of the Association* for Computational Linguistics, Association for Computational Linguistics, Online, pp. 7881-7892.
   URL https://www.aclweb.org/anthology/2020.acl-main.704
- [65] LIU, Y., M. OTT, N. GOYAL, J. DU, M. JOSHI, D. CHEN, O. LEVY, M. LEWIS, L. ZETTLEMOYER, and V. STOYANOV (2019) "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692.
- [66] REIMERS, N. and I. GUREVYCH (2019) "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics. URL https://arxiv.org/abs/1908.10084
- [67] POST, M. (2018) "A Call for Clarity in Reporting BLEU Scores," in Proceedings of the Third Conference on Machine Translation: Research Papers, Association for Computational Linguistics, Brussels, Belgium, pp. 186–191. URL https://www.aclweb.org/anthology/W18-6319
- [68] ZHANG, H., D. DUCKWORTH, D. IPPOLITO, and A. NEELAKANTAN (2021) "Trading Off Diversity and Quality in Natural Language Generation," in *Proceedings* of the Workshop on Human Evaluation of NLP Systems (HumEval), Association for Computational Linguistics, Online, pp. 25–33. URL https://www.aclweb.org/anthology/2021.humeval-1.3
- [69] ZELLERS, R., A. HOLTZMAN, H. RASHKIN, Y. BISK, A. FARHADI, F. ROESNER, and Y. CHOI (2019) "Defending against neural fake news," *Advances in Neural Information Processing Systems*, **32**.
- [70] UCHENDU, A., T. LE, K. SHU, and D. LEE (2020) "Authorship attribution for neural text generation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8384–8395.

- [71] CARLINI, N., F. TRAMER, E. WALLACE, M. JAGIELSKI, A. HERBERT-VOSS, K. LEE, A. ROBERTS, T. B. BROWN, D. SONG, U. ERLINGSSON, ET AL. (2021)
  "Extracting Training Data from Large Language Models." in USENIX Security Symposium, vol. 6.
- [72] LEE, J., T. LE, J. CHEN, and D. LEE (2023) "Do language models plagiarize?" in Proceedings of the ACM Web Conference 2023, pp. 3637–3647.
- [73] DESHPANDE, A., V. MURAHARI, T. RAJPUROHIT, A. KALYAN, and K. NARASIMHAN (2023) "Toxicity in ChatGPT: Analyzing Persona-assigned Language Models," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1236–1270.
- [74] JI, Z., N. LEE, R. FRIESKE, T. YU, D. SU, Y. XU, E. ISHII, Y. J. BANG, A. MADOTTO, and P. FUNG (2023) "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, 55(12), pp. 1–38.
- [75] SHEVLANE, T., S. FARQUHAR, B. GARFINKEL, M. PHUONG, J. WHITTLESTONE, J. LEUNG, D. KOKOTAJLO, N. MARCHAL, M. ANDERLJUNG, N. KOLT, ET AL. (2023) "Model evaluation for extreme risks," arXiv preprint arXiv:2305.15324.
- [76] PU, J., Z. SARWAR, S. M. ABDULLAH, A. REHMAN, Y. KIM, P. BHATTACHARYA, M. JAVED, and B. VISWANATH (2022) "Deepfake Text Detection: Limitations and Opportunities," arXiv preprint arXiv:2210.09421.
- [77] UCHENDU, A., T. LE, and D. LEE (2023) "Attribution and obfuscation of neural text authorship: A data mining perspective," ACM SIGKDD Explorations Newsletter, 25(1), pp. 1–18.
- [78] MITCHELL, E., Y. LEE, A. KHAZATSKY, C. D. MANNING, and C. FINN (2023) "Detectgpt: Zero-shot machine-generated text detection using probability curvature," arXiv preprint arXiv:2301.11305.
- [79] UCHENDU, A., Z. MA, T. LE, R. ZHANG, and D. LEE (2021) "TURINGBENCH: A Benchmark Environment for Turing Test in the Age of Neural Text Generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2001–2016.
- [80] LIU, Z., Z. YAO, F. LI, and B. LUO (2023) "Check Me If You Can: Detecting ChatGPT-Generated Academic Writing using CheckGPT," arXiv preprint arXiv:2306.05524.
- [81] LI, Y., Q. LI, L. CUI, W. BI, L. WANG, L. YANG, S. SHI, and Y. ZHANG (2023) "Deepfake Text Detection in the Wild," *arXiv preprint arXiv:2305.13242*.
- [82] OPENAI (2023) "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, https://arxiv.org/abs/2303.08774. URL https://arxiv.org/abs/2303.08774

- [83] TOUVRON, H., T. LAVRIL, G. IZACARD, X. MARTINET, M.-A. LACHAUX, T. LACROIX, B. ROZIÈRE, N. GOYAL, E. HAMBRO, F. AZHAR, ET AL. (2023) "Llama: Open and efficient foundation language models," arXiv preprint arXiv:2302.13971.
- [84] PENEDO, G., Q. MALARTIC, D. HESSLOW, R. COJOCARU, A. CAPPELLI, H. ALOBEIDLI, B. PANNIER, E. ALMAZROUEI, and J. LAUNAY (2023) "The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only," arXiv preprint arXiv:2306.01116, 2306.01116. URL https://arxiv.org/abs/2306.01116
- [85] SADASIVAN, V. S., A. KUMAR, S. BALASUBRAMANIAN, W. WANG, and S. FEIZI (2023) "Can AI-generated text be reliably detected?" arXiv preprint arXiv:2303.11156.
- [86] CHAKRABORTY, S., A. S. BEDI, S. ZHU, B. AN, D. MANOCHA, and F. HUANG (2023) "On the possibilities of ai-generated text detection," *arXiv preprint arXiv:2304.04736*.
- [87] ZHAO, W. X., K. ZHOU, J. LI, T. TANG, X. WANG, Y. HOU, Y. MIN, B. ZHANG, J. ZHANG, Z. DONG, ET AL. (2023) "A survey of large language models," arXiv preprint arXiv:2303.18223.
- [88] ZHONG, W., D. TANG, Z. XU, R. WANG, N. DUAN, M. ZHOU, J. WANG, and J. YIN (2020) "Neural Deepfake Detection with Factual Structure of Text," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2461–2470.
- [89] KUSHNAREVA, L., D. CHERNIAVSKII, V. MIKHAILOV, E. ARTEMOVA, S. BARAN-NIKOV, A. BERNSTEIN, I. PIONTKOVSKAYA, D. PIONTKOVSKI, and E. BURNAEV (2021) "Artificial Text Detection via Examining the Topology of Attention Maps," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 635–649.
- [90] LIU, X., Z. ZHANG, Y. WANG, Y. LAN, and C. SHEN (2022) "CoCo: Coherence-Enhanced Machine-Generated Text Detection Under Data Limitation With Contrastive Learning," arXiv preprint arXiv:2212.10341.
- [91] GEHRMANN, S., H. STROBELT, and A. M. RUSH (2019) "GLTR: Statistical Detection and Visualization of Generated Text," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 111–116.
- [92] GALLÉ, M., J. ROZEN, G. KRUSZEWSKI, and H. ELSAHAR (2021) "Unsupervised and distributional detection of machine-generated text," arXiv preprint arXiv:2111.02878.

- [93] HE, X., X. SHEN, Z. CHEN, M. BACKES, and Y. ZHANG (2023) "MGT-Bench: Benchmarking Machine-Generated Text Detection," arXiv preprint arXiv:2303.14822.
- [94] SU, J., T. Y. ZHUO, D. WANG, and P. NAKOV (2023) "DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection of Machine-Generated Text," arXiv preprint arXiv:2306.05540.
- [95] YANG, X., L. PAN, X. ZHAO, H. CHEN, L. PETZOLD, W. Y. WANG, and W. CHENG (2023) "A Survey on Detection of LLMs-Generated Content," arXiv preprint arXiv:2310.15654.
- [96] VENKATRAMAN, S., H. HE, and D. REITTER (2023) "How do decoding algorithms distribute information in dialogue responses?" in *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 923–932.
- [97] BLANCHARD, D., J. TETREAULT, D. HIGGINS, A. CAHILL, and M. CHODOROW (2013) "TOEFL11: A corpus of non-native English," *ETS Research Report Series*, 2013(2), pp. i–15.
- [98] WEN, Q., L. WANG, and M. LIANG (2005) "Spoken and written English corpus of Chinese learners," *Foreign Language Teaching and Research Press.*
- [99] TIAN, E. and A. CUI (2023) "GPTZero: Towards detection of AI-generated text using zero-shot and supervised methods." https://gptzero.me.
- [100] ZEROGPT (2023) "AI Text Detector," https://www.zerogpt.com.
- [101] SOLAIMAN, I., M. BRUNDAGE, J. CLARK, A. ASKELL, A. HERBERT-VOSS, J. WU, A. RADFORD, G. KRUEGER, J. W. KIM, S. KREPS, ET AL. (2019) "Release Strategies and the Social Impacts of Language Models," arXiv preprint arXiv:1908.09203.
- [102] ABBASI, A. and H. CHEN (2008) "Writeprints: A Stylometric Approach to Identity-Level Identification and Similarity Detection in Cyberspace," ACM Transactions on Information Systems (TOIS), 26(2), pp. 1–29.
- [103] KENTON, J. D. M.-W. C. and L. K. TOUTANOVA (2019) "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings* of NAACL-HLT, pp. 4171–4186.
- [104] JAWAHAR, G., M. ABDUL-MAGEED, and V. LAKS LAKSHMANAN (2020) "Automatic Detection of Machine Generated Text: A Critical Survey," in *Proceedings of* the 28th International Conference on Computational Linguistics, pp. 2296–2309.
- [105] ABBASI, A. and H.-C. CHEN (2008) "Writeprints: A Stylometric Approach to Identity-level Identification and Similarity Detection in Cyberspace," ACM Transactions on Information Systems, 26, pp. 1–29.

- [106] FABIEN, M., E. VILLATORO-TELLO, P. MOTLICEK, and S. PARIDA (2020) "BertAA : BERT fine-tuning for Authorship Attribution," in *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, NLP Association of India (NLPAI), Indian Institute of Technology Patna, Patna, India, pp. 127–137. URL https://aclanthology.org/2020.icon-main.16
- [107] DEVLIN, J., M.-W. CHANG, K. LEE, and K. TOUTANOVA (2019) "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 4171–4186.
- [108] JIN, D., Z. JIN, J. T. ZHOU, and P. SZOLOVITS (2020) "Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence* (AAAI).
- [109] LIU, Y., M. OTT, N. GOYAL, J. DU, M. JOSHI, D. CHEN, O. LEVY, M. LEWIS, L. ZETTLEMOYER, and V. STOYANOV (2019) "RoBERTa: A Robustly Optimized BERT Pretraining Approach," ArXiv, abs/1907.11692.
- [110] MAHMOOD, A., F. AHMAD, Z. SHAFIQ, P. SRINIVASAN, and F. ZAFFAR (2019) "A Girl Has No Name: Automated Authorship Obfuscation using Mutant-X," *Proceedings on Privacy Enhancing Technologies*, **2019**, pp. 54–71.
- [111] HAROON, M., M. F. ZAFFAR, P. SRINIVASAN, and Z. SHAFIQ (2021) "Avengers Ensemble! Improving Transferability of Authorship Obfuscation," ArXiv, abs/2109.07028.
- [112] GUO, C., A. SABLAYROLLES, H. JÉGOU, and D. KIELA (2021), "Gradient-based Adversarial Attacks against Text Transformers," . URL https://arxiv.org/abs/2104.13733
- [113] XU, J. and Q. DU (2020) "TextTricker: Loss-based and gradient-based adversarial attacks on text classification models," *Eng. Appl. Artif. Intell.*, **92**, p. 103641.
- [114] LI, L., R. MA, Q. GUO, X. XUE, and X. QIU (2020), "BERT-ATTACK: Adversarial Attack Against BERT Using BERT," . URL https://arxiv.org/abs/2004.09984
- [115] MAHESHWARY, R., S. MAHESHWARY, and V. PUDI (2021) "A Strong Baseline for Query Efficient Attacks in a Black Box Setting," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp. 8396–8409.

URL https://aclanthology.org/2021.emnlp-main.661

- [116] POTTHAST, M., F. SCHREMMER, M. HAGEN, and B. STEIN (2018) "Overview of the Author Obfuscation Task at PAN 2018: A New Approach to Measuring Safety," in Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018 (L. Cappellato, N. Ferro, J. Nie, and L. Soulier, eds.), vol. 2125 of CEUR Workshop Proceedings, CEUR-WS.org. URL http://ceur-ws.org/Vol-2125/invited\_paper\_16.pdf
- [117] MCDONALD, A. W. E., J. ULMAN, M. BARROWCLIFT, and R. GREENSTADT (2013) "Anonymouth Revamped : Getting Closer to Stylometric Anonymity,".
- [118] LI, L., R. MA, Q. GUO, X. XUE, and X. QIU (2020) "BERT-ATTACK: Adversarial Attack Against BERT Using BERT," in *Conf. on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), pp. 6193–6202.
- [119] OUYANG, L., J. WU, X. JIANG, D. ALMEIDA, C. L. WAINWRIGHT, P. MISHKIN, C. ZHANG, S. AGARWAL, K. SLAMA, A. RAY, J. SCHULMAN, J. HILTON, F. KELTON, L. MILLER, M. SIMENS, A. ASKELL, P. WELINDER, P. CHRISTIANO, J. LEIKE, and R. LOWE (2022), "Training language models to follow instructions with human feedback," 2203.02155.
- [120] SUNDARARAJAN, M., A. TALY, and Q. YAN (2017) "Axiomatic Attribution for Deep Networks," in Proceedings of the 34th International Conference on Machine Learning.
- [121] UCHENDU, A., Z. MA, T. LE, R. ZHANG, and D. LEE (2021) "TuringBench: A Benchmark Environment for Turing Test in the Age of Neural Text Generation," in Conf. on Empirical Methods in Natural Language Processing (EMNLP).
- [122] SCHLER, J., M. KOPPEL, S. ARGAMON, and J. W. PENNEBAKER (2006) "Effects of age and gender on blogging." in AAAI spring symposium: Computational approaches to analyzing weblogs, vol. 6, pp. 199–205.
- [123] SANH, V., L. DEBUT, J. CHAUMOND, and T. WOLF (2019) "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108.
- [124] KIM, Y. (2014) "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882.
- [125] CER, D., Y. YANG, S.-Y. KONG, N. HUA, N. LIMTIACO, R. ST. JOHN, N. CONSTANT, M. GUAJARDO-CESPEDES, S. YUAN, C. TAR, B. STROPE, and R. KURZWEIL (2018) "Universal Sentence Encoder for English," in *Conference* on Empirical Methods in Natural Language Processing.
- [126] MIKOLOV, T., K. CHEN, G. CORRADO, and J. DEAN (2013), "Efficient Estimation of Word Representations in Vector Space," . URL https://arxiv.org/abs/1301.3781

- [127] BROWN, T. B., B. MANN, N. RYDER, M. SUBBIAH, J. KAPLAN, P. DHARI-WAL, A. NEELAKANTAN, P. SHYAM, G. SASTRY, A. ASKELL, S. AGARWAL, A. HERBERT-VOSS, G. KRUEGER, T. HENIGHAN, R. CHILD, A. RAMESH, D. M. ZIEGLER, J. WU, C. WINTER, C. HESSE, M. CHEN, E. SIGLER, M. LITWIN, S. GRAY, B. CHESS, J. CLARK, C. BERNER, S. MCCANDLISH, A. RADFORD, I. SUTSKEVER, and D. AMODEI (2020), "Language Models are Few-Shot Learners," 2005.14165.
- [128] OPENAI (2023), "GPT-4 Technical Report," 2303.08774.
- [129] TIAN, E. (2022), "GPTZero," https://gptzero.me/, accessed: 2023.
- [130] SOLAIMAN, I., M. BRUNDAGE, J. CLARK, A. ASKELL, A. HERBERT-VOSS, J. WU, A. RADFORD, G. KRUEGER, J. W. KIM, S. KREPS, M. MCCAIN, A. NEWHOUSE, J. BLAZAKIS, K. MCGUFFIE, and J. WANG (2019), "Release Strategies and the Social Impacts of Language Models," 1908.09203.
- [131] WANG, Z., T. LE, and D. LEE (2023) "UPTON: Preventing Authorship Leakage from Public Text Release via Data Poisoning," in *Findings of Conf. on Empirical Methods in Natural Language Processing (EMNLP-Findings).*
- [132] BEVENDORFF, J., I. BORREGO-OBRADOR, M. CHINEA-RÍOS, M. FRANCO-SALVADOR, M. FRÖBE, A. HEINI, K. KREDENS, M. MAYERL, P. PĘZIK, M. POTTHAST, ET AL. (2023) "Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection: Condensed Lab Overview," in International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, pp. 459–481.
- [133] AKOURY, N., S. WANG, J. WHITING, S. HOOD, N. PENG, and M. IYYER (2020) "STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (B. Webber, T. Cohn, Y. He, and Y. Liu, eds.), Association for Computational Linguistics, Online, pp. 6470–6484. URL https://aclanthology.org/2020.emnlp-main.525
- [134] DU, Y. and L. CHILTON (2023) "StoryWars: A Dataset and Instruction Tuning Baselines for Collaborative Story Understanding and Generation," in *Proceedings of* the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3044–3062.
- [135] ZHANG, Y., S. YANG, C. BAI, F. WU, X. LI, X. LI, and Z. WANG (2024) "Towards Efficient LLM Grounding for Embodied Multi-Agent Collaboration," arXiv preprint arXiv:2405.14314.

- [136] PAN, B., J. LU, K. WANG, L. ZHENG, Z. WEN, Y. FENG, M. ZHU, and W. CHEN (2024) "AgentCoord: Visually Exploring Coordination Strategy for LLM-based Multi-Agent Collaboration," arXiv preprint arXiv:2404.11943.
- [137] LIU, Z., Y. ZHANG, P. LI, Y. LIU, and D. YANG (2023) "Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization," arXiv preprint arXiv:2310.02170.
- [138] LI, Y., Y. ZHANG, and L. SUN (2023) "Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents," arXiv preprint arXiv:2310.06500.
- [139] LI, H., Y. CHONG, S. STEPPUTTIS, J. CAMPBELL, D. HUGHES, C. LEWIS, and K. SYCARA (2023) "Theory of Mind for Multi-Agent Collaboration via Large Language Models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (H. Bouamor, J. Pino, and K. Bali, eds.), Association for Computational Linguistics, Singapore, pp. 180–192. URL https://aclanthology.org/2023.emnlp-main.13
- [140] TOUVRON, H., L. MARTIN, K. STONE, P. ALBERT, A. ALMAHAIRI, Y. BABAEI, N. BASHLYKOV, S. BATRA, P. BHARGAVA, S. BHOSALE, ET AL. (2023) "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288.
- [141] JIANG, A. Q., A. SABLAYROLLES, A. MENSCH, C. BAMFORD, D. S. CHAPLOT, D. D. L. CASAS, F. BRESSAND, G. LENGYEL, G. LAMPLE, L. SAULNIER, ET AL. (2023) "Mistral 7B," arXiv preprint arXiv:2310.06825.
- [142] TEAM, G., T. MESNARD, C. HARDIN, R. DADASHI, S. BHUPATIRAJU, S. PATHAK, L. SIFRE, M. RIVIÈRE, M. S. KALE, J. LOVE, ET AL. (2024) "Gemma: Open models based on gemini research and technology," arXiv preprint arXiv:2403.08295.
- [143] GROENEVELD, D., I. BELTAGY, P. WALSH, A. BHAGIA, R. KINNEY, O. TAFJORD, A. H. JHA, H. IVISON, I. MAGNUSSON, Y. WANG, ET AL. (2024) "Olmo: Accelerating the science of language models," arXiv preprint arXiv:2402.00838.
- [144] MUKHERJEE, S., A. MITRA, G. JAWAHAR, S. AGARWAL, H. PALANGI, and A. AWADALLAH (2023) "Orca: Progressive learning from complex explanation traces of gpt-4," arXiv preprint arXiv:2306.02707.
- [145] ZENG, Z., L. SHA, Y. LI, K. YANG, D. GAŠEVIĆ, and G. CHEN (2024) "Towards Automatic Boundary Detection for Human-AI Collaborative Hybrid Essay in Education," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 22502–22510.

- [146] FAN, A., M. LEWIS, and Y. DAUPHIN (2018) "Hierarchical Neural Story Generation," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 889–898.
- [147] LOSADA, D. E. and L. AZZOPARDI (2008) "Assessing multivariate Bernoulli models for information retrieval," ACM Transactions on Information Systems (TOIS), 26(3), pp. 1–46.
- [148] VAPNIK, V. (1998) "Statistical Learning Theory,".
- [149] LAN, Z., M. CHEN, S. GOODMAN, K. GIMPEL, P. SHARMA, and R. SORI-CUT (2019) "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," in *International Conference on Learning Representations*.
- [150] JOSHI, N. and D. VOGEL (2024) "Writing with AI Lowers Psychological Ownership, but Longer Prompts Can Help," *arXiv preprint arXiv:2404.03108*.
- [151] DHILLON, P. S., S. MOLAEI, J. LI, M. GOLUB, S. ZHENG, and L. P. ROBERT (2024) "Shaping Human-AI Collaboration: Varied Scaffolding Levels in Co-writing with Language Models," in *Proceedings of the CHI Conference on Human Factors* in Computing Systems, pp. 1–18.
- [152] TRIPTO, N. I., S. VENKATRAMAN, D. MACKO, R. MORO, I. SRBA, A. UCHENDU, T. LE, and D. LEE (2023) "A Ship of Theseus: Curious Cases of Paraphrasing in LLM-Generated Texts," arXiv preprint arXiv:2311.08374.
- [153] VENKATRAMAN, S., A. UCHENDU, and D. LEE (2024) "GPT-who: An Information Density-based Machine-Generated Text Detector," in *Findings of the Association* for Computational Linguistics: NAACL 2024, pp. 103–115.
- [154] XING, E., S. VENKATRAMAN, T. LE, and D. LEE (2024) "ALISON: Fast and Effective Stylometric Authorship Obfuscation," in *Proceedings of the AAAI Conference* on Artificial Intelligence, vol. 38, pp. 19315–19322.
- [155] VENKATRAMAN, S., N. I. TRIPTO, and D. LEE (2024), "CollabStory: Multi-LLM Collaborative Story Generation and Authorship Analysis," 2406.12665. URL https://arxiv.org/abs/2406.12665
- [156] MACKO, D., R. MORO, A. UCHENDU, J. LUCAS, M. YAMASHITA, M. PIKULIAK, I. SRBA, T. LE, D. LEE, J. SIMKO, ET AL. (2023) "MULTITUDE: Large-Scale Multilingual Machine-Generated Text Detection Benchmark," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9960–9987.
- [157] YANG, Z., R. DABRE, H. TANAKA, and N. OKAZAKI (2024) "Scicap+: A knowledge augmented dataset to study the challenges of scientific figure captioning," *Journal of Natural Language Processing*, **31**(3), pp. 1140–1165.

[158] SIDOROV, O., R. HU, M. ROHRBACH, and A. SINGH (2020) "Textcaps: a dataset for image captioning with reading comprehension," in *Computer Vision–ECCV* 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, Springer, pp. 742–758.

#### Vita

#### Saranya Venkatraman

## Education

The Pennsylvania State University	Aug 2018 - Dec 2024
PhD in Information Science and Technology (IST)	
GB Pant Engineering College, New Delhi, India	Aug 2012 - July 2016
BTech in Computer Science & Engineering	

# **Publications**

[1] **Saranya Venkatraman**, Nafis Irtiza Tripto, and Dongwon Lee."CollabStory: Multi-LLM Collaborative Story Generation and Authorship Analysis" arXiv pre-print 2024.

[2] **Saranya Venkatraman**, Adaku Uchendu, and Dongwon Lee."GPT-who: An Information Density-based Machine-Generated Text Detector." NAACL Findings 2024.

[3] Nafis Irtiza Tripto, **Saranya Venkatraman**, Dominik Macko, Robert Moro, Ivan Srba, Adaku Uchendu, Thai Le, and Dongwon Lee. "A Ship of Theseus: Curious Cases of Paraphrasing in LLM-Generated Texts." ACL 2024.

[4] Eric Xing, **Saranya Venkatraman**, Thai Le, and Dongwon Lee."ALISON: Fast and Effective Stylometric Authorship Obfuscation." AAAI 2024.

[5] **Saranya Venkatraman**, He He, and David Reitter. "How do decoding algorithms distribute information in dialogue responses?" EACL Findings 2023.

[6] Saranya Venkatraman, Prasenjit Mitra, Sherice N. Clarke, Andrea Gomoll, Zaynab Gates, Sushil S., Tarang Tripathi, and Amy Ogan. "ClassInSight: Automating Analysis of Classroom Discussions to Support Teacher Noticing and Reflection on Dialogic Pedagogy." EARLI 2021.

### Work Experience

Google LLC, New York, NY	May-Aug 2020
Research Intern - Google Assistant	
Samsung Research America (SRA), Mountain View, CA	May-Aug 2018
Intern - Artificial Intelligence Center	
Cadence Design Systems, Inc., San Jose, CA	May-Aug 2017
Machine Learning Intern - R&D Center	