Large Language Models: Towards Safety, Robustness, and Understanding

by

Evan Crothers

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Ph.D. degree in Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

The Transformer neural network architecture has had an enormous impact on state-of-the-art language model performance across a wide range of tasks in the text domain. In order for large language models based on this architecture to be suitable for widespread usage, it is critical to ensure they are not abused for malicious purposes, that they are robust against adversarial attack, and that the behaviour of such models is well-understood. The rapid proliferation of user-friendly interfaces to generative language models in particular, such as ChatGPT, highlight the pressing need for preventing abuse of large language models, while improving adversarial robustness of systems designed to detect them. This thesis outlines a plan to make a significant contribution towards these goals in several ways.

We begin by performing an in-depth survey of the categories of malicious attacks associated with machine generated text, a threat modelling exercise to explore cybersecurity threats related to these attacks, and a comprehensive overview of detection methodologies and recommendations to improve defenses. This work was featured by cybersecurity expert Bruce Schneier as "a solid grounding amongst all of the hype", and a talk on the paper was presented as part of the United Nations "AI for Good" speaker series.

Second, we demonstrate a new technique utilizing statistical features to augment Transformer-derived features to improve adversarial robustness in detection of computer-generated text – an important problem for detection of spam and disinformation, and a setting where adversarial attacks are likely.

Third, we determine to what extent existing metrics for assessment of machine generated text align with subjective human assessment, identifying gaps between computational metrics and subjective human assessment of machine generated text.

Finally, we perform an in-depth assessment of how masking-based faithfulness measures are applied to Transformer text classifiers, demonstrating pitfalls in faithfulness-based model comparisons, investigating the underlying mechanisms that cause these issues to arise, and determining the impacts of relying on such measures on adversarial robustness and fairness.

Acknowledgements

The work in this thesis would not be possible without the generous assistance of my supervising professors, Dr. Herna Viktor of the University of Ottawa, and Dr. Nathalie Japkowicz of American University. I am incredibly grateful for your guidance, support, and conversation over these past years. I could never have imagined at the outset where this journey would lead. From the bottom of my heart, thank you.

I would also like to thank Dr. Paula Branco of the University of Ottawa for providing early feedback on the work presented in "Neural-Statistical Features for Detection of Transformer Models" presented in Chapter 3. With respect to the work in Chapter 4, I would like to thank Annie Feng for proofreading survey questions and Bowen Gu for previous discussions on Chinese lyric generation.

The work in Chapter 5 was supported in part by compute capacity provided by the British Columbia Digital Research Infrastructure Group (bc.net) and the Digital Research Alliance of Canada (alliance.can.ca).

Finally, a special thanks to Dr. Roberto Corizzo for graciously assisting with conference presentations as my wife and I prepared for our first child.

This work is dedicated to my children, for whom all this will be ancient history.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview and Motivations

Neural networks currently represent the state-of-the-art across a variety of machine learning applications. Recently, the Transformer architecture (Vaswani et al., 2017) has ushered in an explosion of new development in natural language processing and beyond. The Transformer architecture uses stacked multi-head attention and fully-connected layers in an encoder-decoder architecture. Transformer models with large numbers of parameters trained as language models have become commonly referred to as "large language models" or LLMs.

However, while these Transformer language models offer sharp improvement across many problem domains, these models have their own (not necessarily unique) challenges in the form of the potential for abuse of generative capabilities (Crothers et al., 2023a), vulnerability to adversarial attacks (Madry et al., 2017), and the difficulty of preventing harmful algorithmic biases along the lines of age, race, and gender (May et al., 2019; Prates et al., 2020; Selbst et al., 2019). These considerations become increasingly important as adoption grows, and these models increasingly impact society, including as part of safety-critical systems. Widespread abuse of these models has the potential to improve the effectiveness of phishing attacks, or facilitate cheating in academia (Crothers et al., 2023a). A vulnerability to adversarial attack in a medical model could have the potential for serious implications for the health of patients who interact with such a system (Finlayson et al., 2019). A discriminatory model (we use the term "discriminatory" in the sense of an algorithm that causes "unjust treatment towards a particular group", not in the sense of a classification algorithm that performs "discrimination between two classes") has enormous potential for systematizing biases at scale, such as when gender bias emerges in a widely-used translation system like Google Translate (Prates et al., 2020).

The work within this thesis focuses on problems that must be addressed in order for the strong capabilities of LLMs to be realized while reducing the risks of their abuse, and addressing the adversarial robustness and fairness of these models. In service of the goal, it presents 1) a comprehensive survey of machine generated text risks and detection (Crothers et al., 2023a), 2) an analysis of the adversarial robustness of neural and statistical features for detection of machine generated text (Crothers et al., 2022a), 3) an investigation of to what extent existing metrics for assessment of machine generated text align with subjective human assessment (Crothers et al., 2023b), and 4) a detailed analysis of the pitfalls in relying on masking-based interpretability measures in comparing LLM-based text classifiers (Crothers et al., 2024).

1.2 Thesis Statement

The Transformer neural network architecture has had an enormous impact on state-of-the-art language model performance across a wide range of tasks in the text domain. In order for such models to be suitable for widespread application, steps must be taken to counteract the potential for abuse of such models, mechanisms must be developed to provide robustness against adversarial attacks, and models must be aligned to reduce algorithmic bias based on gender, race, and other protected attributes. This thesis outlines a plan to make a significant contribution towards these goals by:

1. Performing an in-depth survey of the categories of malicious attacks associated with machine generated text, threat modelling of how these attacks can lead to cybersecurity threats, and a comprehensive overview of detection methodologies to protect against these threats.

2. Utilizing statistical features to augment Transformer-derived features to improve adversarial robustness in detection of computer-generated text – an important problem for detection of spam and disinformation, and a setting where adversarial attacks are likely.

3. Determining to what extent existing metrics for assessment of machine generated text align with subjective human assessment, and consider the impact of varying generation parameters on both computational metrics and subjective assessment of generated text.

4. Performing a critical analysis of the mechanisms underlying masking-based interpretability measures for LLM-based text classifiers, and how these mechanisms may result in model comparisons that may be misleading, and circumstances where such comparisons may favour less robust or less fair models.

A paper based on the experiments and results for 1) has been presented at IJCNN (Crothers et al., 2022a). An extensive survey and threat modelling paper based on 2) has been published in IEEE Access (Crothers et al., 2023a). A paper focusing on 3) was presented at the AAAI 2023 CreativeAI workshop (Crothers et al., 2023b). A conference paper based on 4) was accepted at Machine Learning. Optimization, and Data Science (LOD) 2024 Crothers et al. (2024), and the extended version presented within this thesis is currently under review as a journal article for the journal Computational Intelligence.

1.3 Contributions

1.3.1 Machine Generated Text: A Comprehensive Survey of Threat Models and Detection Methods

The first major contribution of this work is a survey and cybersecurity threat modelling paper covering the current state-of-the-art for detection of machine generated text, presented in Chapter 2. This review outlines the current techniques, both neural and non-neural, that are used to detect machine generated text, and highlights the current differentiating features of machine generated text compared to human-written text. This section introduces the applied problem of detection of machine generated text, prior to moving towards the more theoretical aspects of the Transformer representations that guide text generation. A review based on the contents of this section has been published in IEEE Access (Crothers et al., 2023a). As previously mentioned, this work was featured by cybersecurity expert Bruce Schneier as "a solid grounding amongst all of the hype", and a talk on the paper was presented as part of the United Nations "AI for Good" speaker series.

1.3.2 Neural-Statistical Features for Detection of Machine Generated Text

This work's second contribution carries on from the first, focusing on improved methods for detection of machine generated text. The detection of machine generated text is an area of rapidly increasing significance as nascent generative models, allow for efficient creation of compelling human-like text, which may be abused for the purposes of spam, disinformation, phishing, or online influence campaigns. Past work has studied detection of current state-of-the-art models, but despite a developing threat landscape, there has been minimal analysis of the robustness of detection methods to adversarial attacks. To this end, we evaluate neural and non-neural approaches on their ability to detect machine generated text, their robustness against text adversarial attacks, and the impact that successful adversarial attacks have on human judgement of text quality. We find that while statistical features underperform neural features, statistical features provide additional adversarial robustness that can be leveraged by concatenating features in detection models. In the process, we find that previously effective complex phrasal features for detection of machine generated text hold little predictive power against contemporary generative models, and identify promising statistical features to use instead. Finally, we pioneer a new measure for adversarial text quality, based on the difference between MAUVE (Pillutla et al., 2021) scores, before and after adversarial perturbation.

1.3.3 Measuring Relationship between Nucleus Sampling Probability Mass and Subjective Qualities of AI Text

The third contribution within this work revolves around understanding the relationship between human subjective evaluation of computer-generated text, compared to computational evaluation. While the previous contribution focused on computational approaches to detection and differentiation of human and machine text, in this contribution we measure differing subjective responses to human and machine text via a large-scale analysis with human annotators. As part of a presented paper at the AAAI CreativeAI Workshop, we apply a large multilingual language model (BLOOM-176B) in open-ended generation of Chinese song lyrics, and evaluate the resulting lyrics for coherence and creativity using human reviewers. Our findings are two-fold. First, we find that current computational metrics for evaluating large language model outputs (MAUVE and statistical model degeneration metrics) have limitations in evaluation of creative writing, and do not align in with human preference or assessment of creativity. Second, we determine that human reviewers prefer text generated at probability mass $p = 0.95$, even while $p = 0.80$ is overall scored by humans as more "coherent" (among $p \in [0.80, 0.85, 0.90, 0.95, 0.99]$). This is a valuable finding for systems using generative language models in creative text — for humans to find text enjoyable or creative, it must be both *distinctive and comprehensible*, a combination of subjective properties that we determine to peak between at higher values than the peak of coherence. We also introduce the MojimLyrics dataset, a Chinese-language dataset of popular song lyrics for future research.

1.3.4 A Systematic Analysis of Pitfalls in Faithfulness Measures on Masked Language Models

The fourth and final contribution presented within this work focuses on interpretability measures applied to LLM-based neural text classifiers. Explainability of such classifiers is an important quality for their usage for tasks where human oversight is required. These same types of classifiers are commonly used for detection of AI text, as in the first two contributions. A common approach to quantifying neural text classifier interpretability is to calculate faithfulness metrics based on iteratively masking salient input tokens and measuring changes in the model prediction. We propose that this property is better described as "sensitivity to iterative masking", and highlight pitfalls in using this measure for comparing text classifier interpretability. We show that iterative masking produces large variation in faithfulness scores between otherwise comparable Transformer encoder text classifiers. We then demonstrate that iteratively masked samples produce embeddings outside the distribution seen during training, resulting in unpredictable behaviour. We further explore task-specific considerations that undermine principled comparison of interpretability using iterative masking – these include 1) an underlying similarity to salience-based adversarial attacks, threatening adversarial robustness; and 2) a model's tendency in faithfulness scores can persist through fine-tuning, meaning that models fine-tuned from a fairness-optimized foundation model (FairBERTa) can consistently score lower on faithfulness than the basic version of the same model (RoBERTa), despite both models having comparable performance on classification tasks. Our findings give insight into how these behaviours affect neural text classifiers, and provide guidance on how sensitivity to iterative masking should be interpreted.

1.4 Thesis Outline

The content of this thesis is organized according to the research publications resulting from the conducted research. Background information relevant to the understanding of the research is included in each chapter. The chapters are arranged as follows.

Chapter 2 begins the contributions of this work, in the form of a comprehensive survey that includes both an extensive analysis of threat models posed by contemporary natural language generation (NLG) systems, and the most complete review of machine generated text detection methods to date. This survey places machine generated text within its cybersecurity and social context, and provides strong guidance for future work addressing the most critical threat models, and ensuring detection systems themselves demonstrate trustworthiness through fairness, robustness, and accountability. The survey was published in the IEEE Access journal (Crothers et al., 2023a).

Chapter 3 introduces the second major work of the thesis – an analysis of feature-based approaches to detection of Transformer-generated text in the presence of adversarial attacks. This includes a comparison of models trained on Transformer-derived neural features, models trained on statistical features from prior literature, and models trained on a combination of these features. While detection of computer-generated text is inherently a useful problem to address for cybersecurity defence applications, it also serves as a useful lens into understanding the shortcomings of existing generative language models. Adversarial robustness is a key goal of the future planned research, and the experiments and code utilized for these experiments will be incorporated into future experiments. This research was presented at IJCNN 2022 (Crothers et al., 2022b).

Chapter 4 includes the third contribution of the thesis: a study into human perception of subjective qualities in AI generated text. The work in this chapter was published as *In BLOOM: Creativity and Affinity in Artificial Lyrics and Art* at the AAAI 2023 CreativeAI workshop (Crothers et al., 2023b). This chapter complements the analysis in Chapter 3, investigating subjective human-assessed qualities of machine-generated text, rather than purely focusing on detection. This chapter illuminates how humans perceive machine generated text for subjective properties (e.g., creativity) as sampling probability mass is adjusted, and notes limitations of MAUVE scores (which are discussed within) as a means of assessing AI generated text.

Chapter 5 presents the fourth major work of the thesis: an analysis of interpretability measures applied to LLM-based neural text classifiers. Explainability of such classifiers is an important quality for their usage in areas with a high requirement for human oversight, and these same types of classifiers are commonly used for detection of AI text, as highlighted in the work in Chapter 2 and 3. An earlier conference paper based on this work was accepted at LOD 2024 (Crothers et al., 2024), and the extended version presented here is under review as a journal article for Computational Intelligence.

Finally, Chapter 6 provides a conclusion summarizing the research presented within this thesis, a section highlighting limitations and challenges, and an exploration of future work.

Chapter 2

Machine Generated Text: A Comprehensive Survey of Threat Models and Detection Methods

The following is an expanded version of the survey published in the IEEE Access journal (Crothers et al., 2023a). This survey was featured by cybersecurity expert Bruce Schneier as "a solid grounding amongst all of the hype", and a talk on the paper is currently scheduled as part of the United Nations "AI for Good" speaker series.

Recall that the research in this chapter presents the first major contribution of this thesis, in the form of a comprehensive survey that includes both an extensive analysis of threat models posed by contemporary natural language generation (NLG) systems, and the most complete review of machine generated text detection methods to date. This survey places machine generated text within its cybersecurity and social context, and provides strong guidance for future work addressing the most critical threat models, and ensuring detection systems themselves demonstrate trustworthiness through fairness, robustness, and accountability.

Since the release of this survey, generative models and detection methods have continued to evolve. Simultaneously, user-friendly tools that enable both use and abuse of generative AI have become more widespread. The threat models in this survey continue to encapsulate many ways in which generative AI models might be abused, as well as techniques for mitigating this abuse. A very important element of detection research, and a major focus within the UN talk based on this survey, is that detection systems themselves must be carefully designed so as to ensure that they do not cause additional harm due to false positives, and furthermore, do not harm the large number of non-malicious individuals who rely on AI-based tools to help them interact with others across language barriers. The best available models for detection will change over time, making the tables in this survey non-exhaustive. However, this survey should remain a valuable starting point for understanding the field of machine-generated text detection, as well as designing protections against current and future threat models.

2.1 Abstract

Machine generated text is increasingly difficult to distinguish from human authored text. Powerful open-source models are freely available, and user-friendly tools that democratize access to generative models are proliferating. ChatGPT, which was released shortly after the first preprint of this survey, epitomizes these trends. The great potential of state-of-the-art natural language generation (NLG) systems is tempered by the multitude of avenues for abuse. Detection of machine generated text is a key countermeasure for reducing abuse of NLG models, with significant technical challenges and numerous open problems. We provide a survey that includes both 1) an extensive analysis of threat models posed by contemporary NLG systems, and 2) the most complete review of machine generated text detection methods to date. This survey places machine generated text within its cybersecurity and social context, and provides strong guidance for future work addressing the most critical threat models, and ensuring detection systems themselves demonstrate trustworthiness through fairness, robustness, and accountability.

2.2 Introduction

2.2.1 Risks of Machine Generated Text

Recent natural language generation (NLG) models have taken a significant step forward in diversity, control, and quality of machine generated text. The ability to create unique, manipulable, human-like text with unprecedented speed and efficiency presents additional technical challenges for the detection of abuses of NLG models, such as phishing (Baki et al., 2017; Giaretta and Dragoni, 2020), disinformation (Shu et al., 2020; Stiff and Johansson, 2022; Zellers et al., 2019), fraudulent product reviews (Adelani et al., 2020; Stiff and Johansson, 2022), academic dishonesty (Dehouche, 2021; Hargrave, 2005), and toxic spam (Kurenkov, 2022). Addressing the risk of abuse is vital to maximize the potential benefit of NLG technology, while minimizing harms — a key principle of trustworthy AI (European Commission and Directorate-General for Communications Networks, Content and Technology, 2019).

The overwhelming majority of contemporary state-of-the-art NLG models are neural language models (NLMs) based on the Transformer architecture (Vaswani et al., 2017). Significant concerns surrounding the threats posed by generative Transformer models are nearly as old as the models themselves: the release of the 1.5B parameter GPT-2 architecture was delayed for nine months due to fears of abuse (Radford et al., 2019b). Access to GPT-3 remains only permitted via a carefully controlled API (Brown et al., 2020). Such measures demonstrably manifest only in delays to open availability of models. Only four months after the release of GPT-2, Grover — a 1.5B parameter model based on the GPT-2 architecture — was made publicly available (Zellers et al., 2019). The release of Grover not only foreshadowed the speed with which private models would be replicated, but also represented a limited threat model in itself: Grover was specifically designed to both produce and detect neural fake news. Grover's primary author provided a reasoned justification for the model release, and called for an improved set of community norms for the release of potentially dangerous research prototypes (Zellers, 2019).

Such norms have been slow to develop (Liang et al., 2022), and wide scale democratization of access to increasingly large scale natural language generation models has continued. Open-source initiative EleutherAI has produced open-source generative Transformer models with large numbers of parameters, including the 6B parameter GPT-J (Wang and Komatsuzaki, 2021), and 20B parameter GPT-NeoX (Black et al., 2022). Even truly massive models are now available open-source — the BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) is an open-source multilingual model, and at 176B parameters, is larger than GPT-3 (BigScience, 2022). Yandex (Khrushchev et al., 2022), Meta AI (Zhang et al., 2022a), and Huawei (Zeng et al., 2021) have all open-sourced models with over 100B parameters.

Real life examples of how generative Transformer language models may be abused are beginning to emerge. A controversy in the AI research community resulted from the publicized development of a GPT-J model trained on the 4chan politics message board `/pol/`. This model was subsequently deployed to produce a large number of posts on the board from which its training data came, including posts containing objectionable content

(Kurenkov, 2022). At its peak, the model represented roughly 10% of all activity on the board in a 24 hour period (Kilcher, 2022). The response to the deployment of this model included a signed condemnation from 360 signatories across the AI community including scientific directors, CEOs, and professors (Liang and Reich, 2022). A similar project targeted a federal public comment website with GPT-2 text until the submitted comments made up half of all comments, demonstrating the extent of existing vulnerabilities (Weiss, 2019).

Controversy around any individual publicized NLG model belies the more fundamental concern — for years now, any person with access to adequate hardware and open-source training scripts could train or fine-tune large generative Transformers for any purpose they choose, be it pop song lyrics, mass disinformation, or toxic spam. Malicious individuals in the process of training a generative language model need not draw attention to their models via public release, and currently face limited risk of discovery. As NLG capabilities grow and access barriers evaporate, we are inevitably already quietly climbing the adoption curve for this technology to be widely abused by cybercriminals, disinformation agencies, scam artists, and other threat actors.

Access to these models is increasingly not limited to sophisticated threat actors who are able to fine-tune them. User-friendly web interfaces, such as the one provided by ChatGPT (OpenAI, 2022), effectively eliminate any barrier to usage of powerful generative models. Jasper, a tool marketed as an AI writing assistant, uses GPT-3 to write sections of content alongside a human's guidance (Jasper AI, 2022). This includes generating content for blogs and websites, which Jasper can efficiently produce in large volumes. Another website offers an endless supply of GPT-3 authored cover letters (Open Cover Letter, 2022). Tools such as Jasper allow those with little technical knowledge someone to seed the model with prompt text, specify keywords to include, and indicate a specific tone of voice. Using publicly available open-source models, a nearly identical system could easily be created to generate endless streams of targeted disinformation, ready to be loaded into existing grey-market account automation tools for popular social media websites.

NLG models have the potential to have an immense and transformative positive impact on human society. A staggering 1 in 3 internet users aged 16 to 64 have used an online translation tool in the last week, a figure representing over 1 billion people (Kemp, 2022a). Text summarization can create understandable summaries of complex legal text (Kanapala et al., 2019) or medical records (Wang et al., 2021). NLG models can give a voice to machine systems, changing the way that humans interact with them (Kim et al., 2019).

The same Transformer architecture used heavily for NLG can also be used for generating pictures from image descriptions (Ramesh et al., 2021), producing functional code from a natural language summary (Chen et al., 2021), and serves as the basis for the current vanguard of generalist agents (Reed et al., 2022). While future research in NLG will bring further wonders, such as assisting government policy analysis (Armstrong et al., 2019), alongside these opportunities is the corresponding certitude that the same technology will be used by bad actors to nefarious ends. Predicting how abuses are likely to unfold, and understanding the best defenses against them, is essential for allowing humanity to reap the positive benefits of this technology while minimizing potential harms. We must walk a cautious path through the age of the silicon wordsmith.

2.2.2 Survey Overview

Since the release of GPT-2 (Radford et al., 2019b) and subsequent explosion of high-quality Transformer-based NLG models, there has been only one general survey on detection of machine generated text (Jawahar et al., 2020). The scope of this previous survey is constrained to detection methods specifically targeting the several generative Transformer models that had been released at the time. Prior to this, a systematic review of machine generated text predating the Transformer architecture covered approaches to detecting previous NLG approaches, such as Markov chains (Beresneva, 2016). Our survey differs from previous work in three major ways.

First, our survey of machine generated text detection is more comprehensive than previous work. We consider literature on feature-based detection of machine-generated text omitted from prior review (Fröhling and Zubiaga, 2021; Lavoie and Krishnamoorthy, 2010; Nguyen-Son et al., 2017). Such approaches are a worthy inclusion, as feature-based approaches still apply against contemporary NLG models (Crothers et al., 2022a; Fröhling and Zubiaga, 2021; Kowalczyk et al., 2022). Inclusion of such features may provide benefits in practice, such as improved robustness against adversarial attacks targeting neural networks (Crothers et al., 2022a), or enhanced explainability (Kowalczyk et al., 2022). Additionally, as research on both NLG and detection has continued to rapidly advance in the years following the previous survey, we must now cover a wider range of generative models and defensive research. Recently, SemEval-2024 introduced detection of machine generated text as Task 8 Wang et al. (2024).

Second, in addition to a comprehensive review of detection methods against contemporary models, this survey provides an in-depth analysis of the risks posed by NLG models via the process of *threat modeling* (i.e., identifying potential adversaries, their capabilities and objectives) (Bitton et al., 2021). The result of our threat modeling process is a series of *threat models* that describe scenarios where machine generated text may be abused, the likely methodology of attackers, and existing research related to each threat. To date, there has yet to be any survey of machine generated text detection with a focus on the risks presented by machine generated text. Consideration of threat models is vital to set the groundwork for trustworthy development of NLG technology, encourage early development of defensive measures, and minimize potential harms.

Third, guided by the EU Ethics Guidelines for Trustworthy AI (European Commission and Directorate-General for Communications Networks, Content and Technology, 2019) and research community efforts (Kaur et al., 2022), we present our survey with sociotechnical and human-centric considerations integrated throughout, focusing not only on NLG systems and machine text detection technologies, but on the humans who will be exposed to both text generation and detection systems in daily life. The goal of trustworthy AI is to ensure that AI systems are developed in ways that are lawful, ethical, and robust both from a technical and social perspective. Abuse of NLG models threatens all three of these areas, representing safety risks to those who may be targeted by NLG-enabled attacks, threats to the integrity of online social spaces, and challenges to the resilience of the technical and social systems that comprise modern society. Machine text detection is part of protecting against abuse of NLG models, enhancing the robustness and safety of NLG development. Critically, our survey also includes insight into ensuring defensive machine text detection systems themselves are transparent, fair, and accountable.

To summarize, the major contributions of this work are as follows:

- The most complete survey of machine generated text detection to date, including previously omitted feature-based work and findings from recent contemporary research.
- The first detailed review of the threat models enabled by machine generated text, at a critical juncture where NLG models and tools are rapidly improving and proliferating.
- Integration of both topics with a vital practical perspective guided by Trustworthy AI (TAI) into how machine generated text threat models and detection systems will impact humanity.

The rest of this survey is organized as follows. We provide definitions and a brief overview of existing methods for natural language generation in Section 2.3. In Section 2.4 we explore threat models related to abuse of machine generated text, including impacts on trust. We provide a comprehensive survey of literature related to detection of machine generated text in Section 2.5. In Section 2.6 we summarize open problems and ongoing trends to guide the direction of future work. Finally, in Section 2.7 we present our final conclusions. While this work discusses machine generated text extensively, including models designed for generating scientific papers, no such models were utilized in authorship of this work.

2.3 Machine Generated Text

Before reviewing threat models and detection methodologies for machine generated text, it is helpful to briefly provide a formal definition of machine generated text, and a condensed overview of natural language generation (NLG) models. We recommend further reading of dedicated surveys on natural language generation for greater insight into the wide breadth of NLG models and applications (Dong et al., 2022; Gatt and Krahmer, 2018; Li et al., 2021; Perera and Nand, 2017; Reiter and Dale, 2002; Santhanam and Shaikh, 2019).

2.3.1 Definition and Scope

In this survey, we use a broad definition of the term "machine generated text" which we believe includes all relevant research in the field:

"Machine generated text" is natural language text that is produced, modified, or extended by a machine.

We focus our definition of machine generated text on *natural language* — i.e., text written in human languages that are "acquired naturally (in [an] operationally defined sense) in association with speech" (Lyons, 1991) — and exclude *non-natural language* — i.e., logical languages, programming languages, etc. Exclusion of non-natural language aligns with other work in the field: the term "text generation" is currently considered synonymous with "natural language generation" (Li et al., 2021; Zhang and Sun, 2009). We anticipate that "text generation" may be repurposed in future research as an umbrella term that includes non-natural language text as well. This would accommodate common considerations between NLG models and contemporary code generation models, such as Codex (Chen et al., 2021) and CodeGenX (Manjaramkar, 2021). As an example, attacks against StackOverflow or GitHub may include both NLG as well as vulnerable code generation.

Our definition of machine generated text is intentionally broad, and covers a large number of possible use cases and associated threat models, which will be discussed in Section 2.4. In the interests of managing a survey scope that already spans a wide range of literature and broad sociotechnical context, text generation by means of text adversarial attack will not be considered. In the majority of cases, the production of new text is not the primary goal of a text adversarial attack, and text adversarial attacks and threat models are already covered by surveys in adversarial attack literature (Chakraborty et al., 2018; Huq and Pervin, 2020; Wang et al., 2019b). We will nevertheless discuss the role machine generated text plays in adversarial contexts in Section 2.4, as well as adversarial robustness of detection models in Section 2.6.

Note that this analysis focuses on threat models where a threat actor leverages machine generated text as part of an attack — typically scenarios where the attacker is attempting to pass machine text as human, and where detection of machine generated text may be useful defensively. We are not discussing attacks against NLG models themselves, unless they leverage NLG as part of the attack. For example, a white-box training data extraction attack targeting the weights of a commercial speech-to-text model would not be included in our analysis, but using an NLG model to produce data for poisoning that model's training dataset would.

With this definition of machine generated text in mind, and with an understanding of the scope of research under consideration, we proceed to a brief overview of natural language generation.

2.3.2 Natural Language Generation

Using a computer to produce human-like text is well-established in the history of computing. Turing's proposed "imitation game" (Turing, 1950) in 1950 considered the question of machine intelligence based on the ability of a machine to conduct human-like conversation over a text channel, for which the first widely-published method dates back to 1966 with the ELIZA chatbot (Weizenbaum, 1966). Given the large volume of NLG research over the

Table 2.1: Inputs, tasks, and examples of natural language generation

Input	Task	Examples
None / Random noise	Unconditional text generation	GPT-2 (Radford et al., 2019b), GPT-3 (Brown et al., 2020) (no prompt)
Text sequence	Conditional text generation	GPT-2 (Radford et al., 2019b), GPT-3 (Brown et al., 2020) (with prompt)
	Machine translation	FairSeq (Ott et al., 2019), T5 (Raffel et al., 2020)
	Text style transfer	Style dictionary (Sheikha and Inkpen, 2011), GST (Sudhakar et al., 2019)
	Text summarization	BART-RXF (Aghajanyan et al., 2021), Word and Phrase Freq. (Luhn, 1958)
	Question answering	FairSeq (Ott et al., 2019), T5 (Raffel et al., 2020)
	Dialogue system	DG-AIRL (Li et al., 2018), DIALOGPT (Zhang et al., 2020), BlenderBot3 (Shuster et al., 2022), ChatGPT (OpenAI, 2022)
Discrete attributes	Attribute-based generation	MTA-LSTM (Feng et al., 2018a), PPLM (Dathathri et al., 2020), CTRL (Keskar et al., 2019)
Structured data	Data-to-text generation	DATATUNER (Harkous et al., 2020), Control prefixes (T5) (Clive et al., 2021)
Multimedia	Image captioning	GIT (Wang et al., 2022b), ETA (Li et al., 2019)
	Video captioning	MMS (Li et al., 2017), YouTube2Text (Guadarrama et al., 2013)
	Speech recognition	ARSG (Chorowski et al., 2015), wav2vec-U (Baevski et al., 2021)

past 55 years, we provide only a high-level taxonomy of major NLG tasks and approaches as groundwork for our analysis of threat models and detection methodologies, and leave detailed discussion to aforementioned dedicated surveys.

Natural Language Generation Tasks

Recall from §2.2.1 that there are a wide variety of applications for natural language generation. Leveraging previous surveys (Dong et al., 2022; Jin et al., 2022; Li et al., 2021), we provide a summary of major tasks in the NLG domain, with examples of models that have been used for each task in Table 2.1. Note that many of the models listed are multi-purpose and can be trained on multiple downstream NLG tasks.

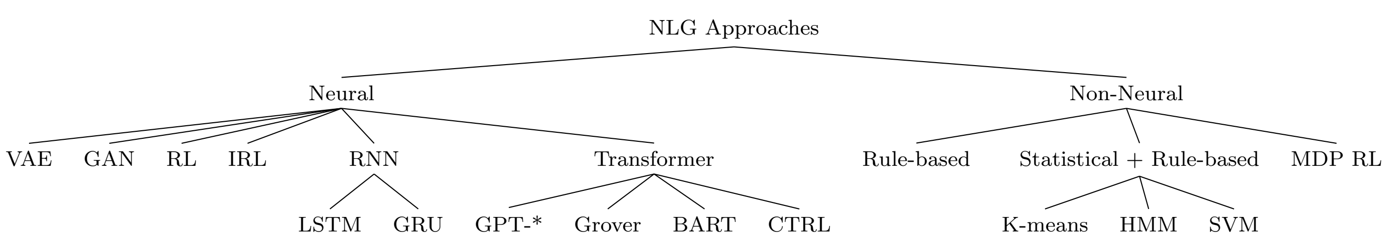
The summary in Table 2.1 is not exhaustive, and in reality, a mutually exclusive delineation between input types does not exist. Combinations of different input types are possible. As an example, CTRL takes both a discrete control code attribute and conditional text prompt in generation (Keskar et al., 2019). Question-answering systems may be able to answer questions about images, such as Unified VLP (Zhou et al., 2020) and TAG (Wang et al., 2022a). Note that we consider a "topic" as an attribute in this overview, and so include "topic-to-text generation" under the broader umbrella of "attribute-based generation", including work such as topic-to-essay generation (Feng et al., 2018a).

Given the strong generative capabilities of Transformer language models, and the corresponding increased risk in threat models, such models rightly warrant particular emphasis in review. However, as mentioned in Section 2.2.2, consideration of the broader field of natural language generation and previous detection research is important as detection techniques that apply against pre-Transformer models have been shown to be useful in detection of modern neural text models, and diverse approaches may offer increased adversarial robustness (Crothers et al., 2022a) or better explainability (Kowalczyk et al., 2022).

2.3.3 Natural Language Generation Approaches

There are a wide range of model architectures and algorithmic approaches to natural language generation. We categorize these approaches broadly into neural and non-neural methods, and then further break them down into more specific categories. A diagram of our simplified breakdown can be found in Figure 2.1. As previously mentioned, NLG encompasses a large variety of tasks and research areas, with this brief section serving as context for understanding machine generated text threat models and detection methods.

Figure 2.1: Taxonomy of major NLG approaches



Non-Neural Models

Predating the popularization of neural approaches in the NLG domain, a range of systems were used to accomplish NLG tasks. These early approaches can broadly be summarized as "rule-based", though there existed variety in terms of processes, pipelines, and targets tasks. A review of rule-based systems can be found in Reiter and Dale's book on the subject (Reiter and Dale, 2002).

An alternative approach to purely rule-based approaches is to use an existing natural language corpus to generate rules for components of an NLG system, such as content selection (Duboue and McKeown, 2003; Langkilde and Knight, 1998) or template generation (Kondadadi et al., 2013). These statistical approaches are meant to be more adaptable to different domains than strictly rule-based systems. While many different statistical models have been integrated with NLG systems in various ways, Hidden Markov Models (HMM) (Baum and Petrie, 1966) feature prominently in past work. More recent non-neural research has used reinforcement learning (Janarthanam and Lemon, 2009) and hierarchical reinforcement learning (Dethlefs and Cuayáhuitl, 2010) of Markov Decision Process (MDP) agents to learn optimal text generation policies.

Non-Transformer Neural Methods

Natural language generation using neural networks was demonstrated to be highly effective using recurrent neural networks (RNN) (Berglund et al., 2015; Kalchbrenner and Blunsom, 2013; Mikolov et al., 2012), including long short-term memory (LSTM) architectures (Merity et al., 2017) and gated recurrent units (GRUs) (Pawade et al., 2018). However, RNN and LSTM architectures had to contend with the vanishing gradient problem, to which the multi-head attention mechanism of the Transformer architecture is more resilient (Topal et al., 2021). Generative adversarial networks (GANs) (Goodfellow et al., 2020) — commonly used to generate continuous data (such as images) — can also be adapted to a discrete context for natural language generation (Lin et al., 2017; Yu et al., 2017).

Deep reinforcement learning (RL) has been used with neural networks to learn policy gradient methods that reward text characteristics associated with high-quality text generation (Li et al., 2016b). A related area of work is the usage of inverse reinforcement learning (IRL), which has included work that aims to address reward sparsity and mode collapse problems in GAN-based text generation by learning an optimal reward function and generation policy (Li et al., 2018; Shi et al., 2018).

Transformer

The multi-head attention architecture of Transformer language models (Vaswani et al., 2017) currently represents the state-of-the-art in natural language generation across natural language tasks. Among Transformer models, the unidirectional GPT-2 (Radford et al., 2019b) and GPT-3 (Brown et al., 2020) models are the most studied in the field of machine generated text detection due to their groundbreaking performance on unconditional and conditional text generation — though like many other Transformer models, these architectures can be used for other NLG tasks as well.

In addition to GPT-2 and GPT-3, also notable in machine generated text detection are related autoregressive language models using a similar architecture, with variations in sampling procedures or training datasets. Such models include Grover (Zellers et al., 2019) (a GPT-2 style model trained on a news dataset and using nucleus sampling instead of top-k sampling), GPT-J (Wang and Komatsuzaki, 2021) (a 6-billion parameter autoregressive language model trained on The Pile (Gao et al., 2021)), and GPT-NeoX-20B (Black et al., 2022) (a 20-billion parameter model similar to GPT-3, also trained on The Pile (Gao et al., 2021)).

Unidirectional Transformer language models generate text by performing unsupervised distribution estimation to predict the next token based on previous tokens. To do this, the model is trained on an existing set of variable-length example texts $(x_1, x_2, ..., x_n)$ each composed of symbols $(s_1, s_2, ..., s_m)$. These symbols may be characters, or multi-character tokens obtained through a tokenization process.

The probability of a given text can then be expressed as the conditional probability of the final token, given each previous token. That is:

$$p(x) = \prod_{i=1}^{m} p(s_m|s_1, ..., s_{m-1}) \tag{2.1}$$

The self-attention mechanism in the Transformer architecture makes it possible to train neural network architectures that are effectively able to estimate such probabilities, given a suitable pre-training task. In unidirectional models such as those in the GPT lineage, a common training task is prediction of the next token in sequence. To generate text, such models can then receive a continue an input sequence by sampling from the probability distribution of all possible next tokens based on previous tokens. An important parameter in this sampling process is "temperature" $T \in (0, \infty)$, which can be raised above 1 to increase the likelihood of selecting a less-probable next token — improving diversity at the potential cost of choosing an unusual token — or lowered below 1 to bias sampling towards more common tokens.

There are three common decoding strategies used for sampling token probabilities from contemporary unidirectional generative Transformer models (Holtzman et al., 2019b):

1. No truncation → Sample from the entire probability distribution. At $T = 1$, this is called "pure sampling".

2. Top-k truncation → Sample from the k most probable tokens.

3. Nucleus sampling (also known as top-p truncation) → Sample from tokens in the top-p portion of the probability mass, rather than a fixed number of tokens k.

While unidirectional generative models are key fixtures of machine generated text detection research, other Transformer architectures can be used for NLG tasks as well. The architecture of BART (Lewis et al., 2020a) includes a bidirectional encoder (similar to BERT (Devlin et al., 2019)), but maintains a left-to-right decoder for sequential text generation. Other Transformer architectures such as MASS (Song et al., 2019), T5 (Raffel et al., 2020), and ULMFiT (Howard and Ruder, 2018) can also be used for NLG tasks.

An important area of ongoing research centers around shaping the output produced by Transformer models. This can include prompt engineering — carefully crafting the conditional text input for a language model to continue (Brown et al., 2020) — or by providing additional discrete attributes that can be used to influence the generation of the network, such as control code, topic, or sentiment as in CTRL (Keskar et al., 2019), PPLM (Dathathri et al., 2020), or GeDi (Krause et al., 2020). Greater control over model output increases the risks posed by threat models (Brown et al., 2020). As an example, when generating social media posts as part of an NLG-augmented online influence campaign, an attacker would benefit from being able to ensure that generated comments both 1) mention a targeted political opponent, and 2) demonstrate negative entity sentiment towards the opponent. We will cover such potential abuses and others in more detail in the next section, which concerns threat models associated with machine generated text.

2.4 Threat Models

Machine generated text enables a diverse array of attacks. These attacks may be performed by threat actors with specific objectives, such as to compromise a computer system, exploit a target individual for financial gain, or enable large-scale harassment of specific communities. The EU ethics guidelines for trustworthy AI emphasize that unintended or dual-use applications of AI systems should be taken into account, and that steps should be taken to prevent and mitigate abuse of AI systems to cause harm (European Commission and Directorate-General for Communications Networks, Content and Technology, 2019). As such, trustworthy AI in the context of NLG, necessitates understanding the areas where such models may be abused, and how these abuses may be prevented (either with detection technologies, moderation mechanisms, government legislation, or platform policies). When discussing attacks, we discuss not only the direct impact on targets, but also the broader impacts of both attacks and mitigation measures on trust.

To understand the risks that motivate research on detection of machine generated text, we draw from existing literature to present a series of threat models incorporating natural language generation. Threat modeling reflects the process of *thinking like an attacker*, identifying vulnerabilities to systems by identifying potential attackers, their capabilities, and objectives. The goal of threat modeling is to improve the security of systems by considering the greatest threats to systems and their users. Many methods of threat modeling have been developed over the years, which include approaches ranging from drawing system diagrams, itemized vulnerability checklists, and performing open-ended brainstorming (Bromander et al., 2016; Kohnfelder and Garg, 1999; UcedaVelez and Morana, 2015; UK National Cyber Security Centre, 2022). In late 2020, a diverse set of experts formed a threat modeling working group to produce a high-level set of guidelines related to effective threat modeling approaches (Braiterman et al., 2020) — we leverage these guidelines in the open-ended attack-centric modeling approach in this section.

2.4.1 Threat Modeling Fundamentals

As we anticipate a machine learning audience with varying exposure to cybersecurity topics, before we present threat models related to machine generated text, it is helpful to first provide an overview of threat modeling, and characterize the approach taken in this section.

A basic example of a common threat model is “a thief who wants to steal your money”

(Shostack, 2014). We can add detail to this threat model by considering more specific capabilities and objectives that such an attacker might have. For example, we may consider "a thief with lockpicks who wants to steal your TV", or "a thief who found your banking password in a database dump and wants to transfer money out of your account". With these threat models in mind, we can then propose mitigation strategies, such as "install locks that are resistant to lockpicking", or "use multi-factor authentication for online banking". Finally, we evaluate whether our mitigation approach is sufficient to address the threat, and consider what other threat models we might need to consider. Threat modeling is inherently an iterative process (Braiterman et al., 2020; Shostack, 2014).

Shostack's Four Question Frame for Threat Modeling (Shostack, 2014, 2021) presents best a plain language foundation for threat modeling by posing four simple questions:

1. *What are we working on?* → Identify the system under attack.
2. *What can go wrong?* → Determine potential attackers, their capabilities, and objectives.
3. *What are we going to do about it?* → Devise a mitigation strategy.
4. *Did we do a good job?* → Review whether the analysis is accurate and complete.

In these terms, we summarize our threat modeling approach in this section as follows:

1. *Identify the system under attack:* We provide a broad attack-centric analysis of machine generated text on society, rather than a system-centric analysis focusing on vulnerabilities to a specific IT system. As such, we identify several discrete technological systems, within the broader societal supersystem.
2. *Determine potential attackers, their capabilities, and objectives:* We consider threat actors of varying sophistication and motives, but with a common modus operandi — in all cases, our attacker is an individual or organization exploiting an NLG model. We characterize the attacker when explaining each attack.
3. *Devise a mitigation strategy:* After identifying a threat model, we propose mitigation measures to improve security and reduce risk. Detection of computer-generated text is often part of the presented mitigation approaches, but policy changes and human moderation systems can also have a significant impact.

4. *Review whether the analysis is accurate and complete:* We have given careful thought to the presented threat models, which are formed from perspectives gained across industry, academia, and government. However, as threat modeling is an iterative process that benefits from diverse perspectives (Braiterman et al., 2020), we greatly encourage further analysis of potential attacks and mitigation measures in future research.

The remainder of this section comprises our threat model analysis, grouped according to a breakdown of attacks into four major categories, followed by a concluding discussion. Within each category we discuss threat models associated with that category of attack, identifying systems at risk, and describing possible threat actors, their objectives, and capabilities. For each attack, we propose mitigations, and then discuss the trust impacts of both the attack and — crucially — of the proposed mitigations as well. A taxonomy of the broad categories of attacks using NLG models we discuss can be found in Figure 2.2.

While a completely exhaustive list of all possible future malicious applications of NLG models is not possible, the threats outlined here span a wide range of tangible dangers at this point in time, representing valuable areas of future investigation for preemptive ethical defensive research. As previously mentioned, threat modeling is iterative, and it is hoped that these threat models should serve as the foundation for future work in improving security against machine generated text.

2.4.2 Facilitating Malware and Social Engineering

Phishing and Scamming

Phishing attacks center on socially engineering a target individual to perform a desired action. This might be to convince the target to open an unsafe document that contains an exploit, cause the target to navigate to a fake banking webpage, encourage them to share sensitive information that can be used for identity theft, among many other documented methods (Alkhalil et al., 2021; Chiew et al., 2018). Phishing attacks can target numerous channels, including email, phone, SMS, or chat applications.

Automated messaging approaches in the early stages of phishing campaigns are common (Alkhalil et al., 2021). Machine generated text can be a useful tool to an attacker attempting to scale or better target phishing or scam campaigns. Rather than provide the

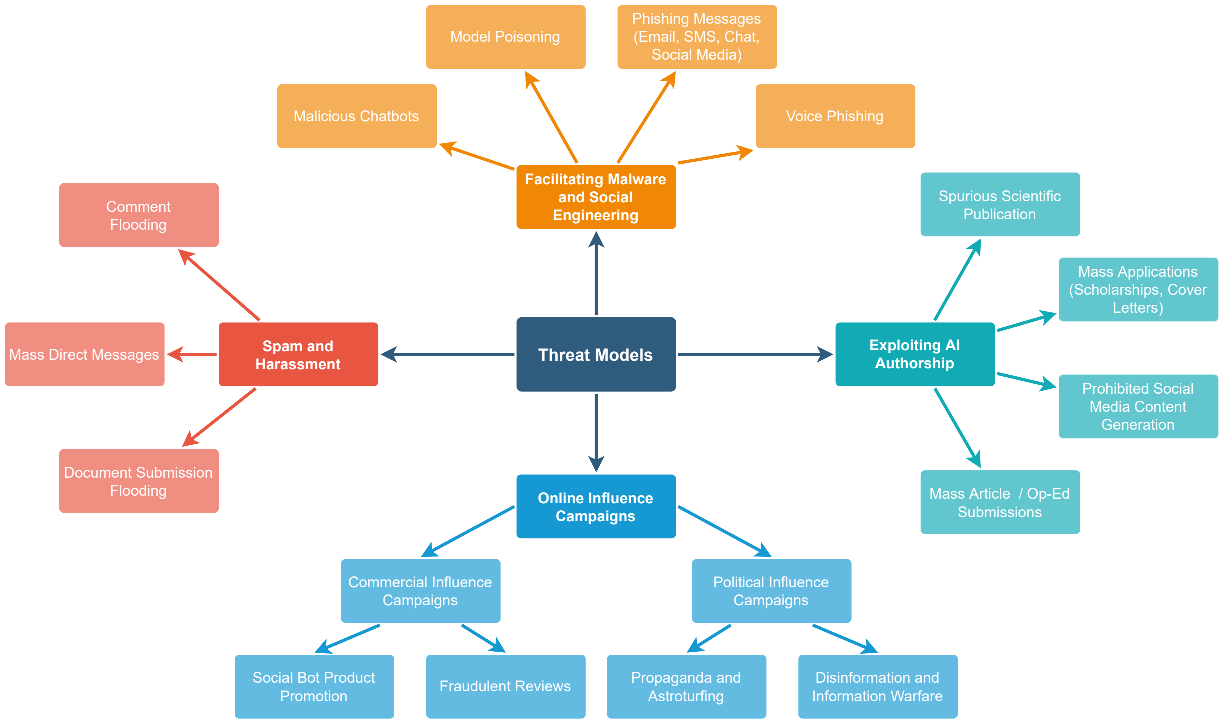


Figure 2.2: Broad taxonomy of threat models enabled by machine generated text

same message to all targets, NLG can be used to generate target-specific text. Research has demonstrated NLG both for scaling of email masquerade attacks (Baki et al., 2017), and for community-targeted phishing (Giaretta and Dragoni, 2020). Carefully targeting a phishing attack (commonly referred to as "spear phishing"), greatly increases the likelihood of a specific target falling for the attack (Burns et al., 2019). In the cases of chat messages, NLG models that serve as dialogue agents may be exploited to exchange messages with the target under a pretext before exploiting them (Zhang et al., 2020).

Existing targeted phishing scams have had devastating effects. One ongoing scam targets members of the Chinese diaspora in the U.S. and Canada, and in 2019 included over 350 reported cases with an average loss of over $164,000 (Oregon FBI, 2019). Despite such scams being documented for several years, very similar scams continue to be executed today (Liu, 2022). In this scam, which begins with the target receiving a pre-recorded phone message, attackers are able to lure in targets into face-to-face interactions with scammers posing as authorities, where they create a sense of urgency and fear to prevent the target from reaching out to real authorities, allowing the attacker to exploit the target financially. Contemporary targeted SMS scams include text messages that specifically address the target by name (Philippine National Privacy Commission, 2022), or spoof the target's own phone number (Jr., 2022). Given the efficacy of targeted attacks, it is highly likely that crime syndicates may find ways to adopt machine generated text to tune attacks to more effectively target individuals or communities.

Mitigation of NLG-enabled phishing attacks will be similar to established work on existing phishing attacks, including both automated detection systems, user reporting, and awareness campaigns (Abu-Nimeh et al., 2007). NLG may present an increased challenge for existing detection systems in that generated messages may have unique or highly-varied content — though attackers may be forced to include specific "payload" content that always needs to be included (e.g., a phishing email may include a unique shortlink to the same malicious website, a chatbot may need to socially engineer the answer to the same security questions). As text content becomes more varied due to more powerful NLG models, detection of stable payload content may represent a more stable detection feature. Algorithms for detection of machine generated text are also likely to be added to existing automated detection approaches.

Social Worms

NLG models may be particularly useful for worms that spread through social media or email contact networks. When an individual has their account compromised by an exploit, that account may be used to send malicious messages that propagate the exploit to other users. By using previous messages or emails between individuals as context to an NLG model, it may be possible to automatically produce messages that include personal details, mimic a loved ones writing style, or carry on a short conversation before delivering a malicious file or link. Given that NLG models are often quite large, the NLG component of this may need to be run on a separate command-and-control server and queried from behind a proxy, rather than bundled with the exploit code itself (unless the pretext of the conversation can be used to convince the target to download a file).

Mitigation of such attacks could involve platforms adopting formal policies that users do not use machine generated text in their communications, except under carefully controlled circumstances. Detection models could then be leveraged against user communications. While this may be acceptable for public posts, there exists privacy risks when considering private messages. Detection models could perhaps be executed as part of the message viewing application on the receiving device to protect end-to-end privacy of messages. If a user receives several messages that score highly for machine generated text detection, a warning may be raised. This approach is not without risks, as the privacy of direct messages must be protected, and any real or perceived erosion of privacy will undermine public trust. Other security measures to protect accounts from unauthorized logins, such as multi-factor authentication, should continue to be used to protect against account compromise more broadly.

Data Poisoning

Cybercriminals may have an interest in poisoning the training datasets of machine learning models. This may be to support other attacks (e.g., poisoning a training dataset for a malware detection algorithm or email spam filter) or poisoning a given model may be the primary goal (e.g., poisoning the dataset of an algorithmic trading model so that the attacker can later trigger trades that financially benefit the attacker). If a threat actor identifies that they can access the training data of any such algorithm, they may use NLG models to produce many training examples containing a particular malicious signature they wish to conceal. Poisoning attacks against neural code-completion algorithms have been performed by generating samples including a given vulnerability (Schuster et al., 2021), and GPT-2 has been used in research to produce fake cyber threat intelligence reports for poisoning cyber-defense systems (Ranade et al., 2021).

Mitigation of dataset poisoning varies based on the sensitivity of the model and nature of the training dataset. The first line of defense is basic IT security best practices that prevent unauthorized modifications to training datasets. However, in some situations models are trained on publicly available data, and therefore it is not possible to prevent access to training data. In these cases, data might be screened prior to inclusion in the training dataset. This screening can include classifiers — potentially including the machine generated text detection approaches discussed in Section 2.5 — or by other analysis methods, such as cluster-based methods to detect poisoning in training datasets (Baracaldo et al., 2017). Finally, for sensitive models, it may also be appropriate to leverage data versioning techniques and audit logging to capture changes to the data potentially made by a malicious insider.

Impacts of Attacks and Mitigation on Trust

The usage of NLG models to produce compelling, targeted pretexts for a large scale phishing attacks and social worms is likely to further reduce trust in text communications, particularly those received from contacts not personally known. Individuals may become even more suspicious of unsolicited messages — even seemingly innocuous ones. As even a seemingly good-natured greeting may be just the first message from a malicious dialogue agent, individuals may decide it is safer to not reply to such messages, further reducing trust and social interaction with new individuals in online communities.

NLG-based poisoning attacks against machine learning models will likely have the greatest trust impact on machine learning practitioners, who may be required to carefully scrutinize open-source training for poisonous samples. Where mitigation of poisoning attacks involves limiting access to training datasets behind auditing and approval processes, such procedures may cause developers to feel distrusted, and undermine the relationship between these individuals and the organizations they work with. While the trust impact of NLG-based data poisoning attacks may be relatively minor among the general population, a high-profile disturbing attack (e.g., a poisoning attack against a medical diagnosis model) may cause individuals to lose trust in machine learning systems more broadly, based on concerns that such models are not safe from malicious tampering.

2.4.3 Online Influence Campaigns

An area of particular concern for abuse of machine generated text is facilitation of online influence campaigns. The objectives of threat actors in this area may either be political in nature (e.g., disinformation, propaganda, election interference) or commercial in nature (e.g., product promotion, smearing competitors, fake reviews). In either case, the goal is to promote a particular idea or prompt a particular action among the target audience.

Either type of campaign may both leverage or facilitate other threat models, such as spam, harassment, mass submission of agenda-driven content, phishing, or malware. The distinction between commercial and political influence campaigns is useful for better understanding threat actors and threatened systems in more detail, as well as categorizing existing research.

While threat actors often differ, there is natural overlap in methods between commercial and political threat actors engaging in an online influence operation. Beyond this, such campaigns may both leverage or facilitate techniques mentioned in other sections, such as spam, harassment, mass submission of unreliable news articles, or cyberwarfare. Regardless, the distinction between different cases is useful for understanding specific threat actors and threatened systems in more detail, and providing a more complete picture of existing research, potential attack mitigations, and the resulting impacts on trust.

A brief discussion on terminology — the broad typology for such attacks in the online domain is "online influence operations" or "online influence campaigns" (Brooking et al., 2020). The term "fake news" refers to false or misleading information presented as news, and online is a sub-problem within detection of online influence campaigns. As an example, fake news detection, does not include flooding a comment section on a news website or social media with machine generated posts that support a particular narrative or emotion. The label "disinformation" is similarly restricted to false or misleading information — information need not necessarily be false in order to achieve influence. The term "foreign influence operations" is limited to influence operations executed or orchestrated by foreign entities — but does not include domestic attackers who may also be motivated by commercial or political reasons to run an online influence operation that targets individuals in their own country.

The term "disinformation" implies that the spread narratives or information is false or misleading. As such, research focused on disinformation falls under the umbrella of false information detection (Guo et al., 2020). Detection of false information on social media is a noble and useful pursuit to improve trust in social media and reduce the spread of damaging disinformation. For example, social media would be much safer for many vulnerable users without "snake oil" salesmen marketing them dangerous and expensive cures and casting doubt on medical research. However, solving false information detection does not solve the problem of online influence operations. An attacker interested in manipulating a nation's election might target an undesired candidate with subjective insults (e.g., I just can't trust that face) or, ultimately, simply ensure that true unflattering facts about the candidate fill every comment section where the candidate is so much as mentioned. While the facts within each comment may be true, the deception lies in the false appearance of popular disapproval of the candidate, amplifying negative sentiment to control the narrative. That is to say, the artificial amplification of true information can still easily be used to accomplish the goal of online influence, and protecting against false information does not solve the more fundamental problem of malicious well-resourced enemies subverting democratic processes or attacking their enemies via the creation of fraudulent accounts.

We now divide our discussion of influence campaigns into two sections, differentiating threat actors whose primary goals are political (i.e., political influence campaigns) and those whose primary goals are commercial (i.e., commercial influence campaigns).

Political Influence Campaigns

Machine generated text as part of political influence campaigns has been analyzed in previous work (Shu et al., 2020; Stiff and Johansson, 2022; Zellers et al., 2019). Papers related to the threat of generative language models on online influence operations may use terminology such as terminology "fake news" (Zellers et al., 2019) or "disinformation" (Brief, 2021; Stiff and Johansson, 2022), or "domestic and foreign influence operations" (Brief, 2021).

The threat actor in a political influence campaign represents an entity who wishes to influence beliefs or prompt action among a target group. These threat actors might include, as examples:

- A political party hiring a group to post unflattering comments online about their political adversaries
- A nation-state disseminating fraudulent news reports to mask human rights abuses
- A nation at war attempting to incite the citizenry of an opposing nation to overthrow the government

Datasets from past political influence campaigns have been released by Facebook (Schrage and Ginsberg, 2018), Reddit (Reddit, 2018), and Twitter (Twitter, 2019), including operations attributed to 22 different countries. Threat actors in this space can be expected to already have the capability of running online political influence campaigns using human employees or contractors, and be familiar with social media automation tools that facilitate registration and management of fraudulent social media accounts. Large models with strong few-shot performance can simply be provided examples of the desired messaging to produce further propaganda. We demonstrate this in Table 2.2, using Russian information operations executed on Twitter during the Syrian civil war to generate additional on-message tweets that promote Syrian Arab Army (SAA) and Russian forces, while discrediting American involvement.

As consumption of text content is common online, there are many avenues where machine generated text might be utilized by a threat actor to improve scaling and targeting of influence operations. Social media, due to the large volume of engaged users, is likely to continue be a valuable and vulnerable target for such campaigns (Reddit, 2018; Schrage

Table 2.2: Few-shot generation of information warfare based on Syrian civil war influence operations

GPT-3 Prompt: Human tweets from Russian foreign military intelligence (GRU) (Twitter, 2019)	RT @TheArabSource: Confirmed: #US retrains #ISIS and #ISIS-like jihadists in east #Syria and re-brands them 'democratic forces' (VIDEO) - E...
	RT @Tasnimnews_EN: #Iraq Army Discovers #US Drone in #Daesh Arms Depot near Fallujah https://t.co/zYRgEHmlr0 https://t.co/H762Uxwl Qx
	...
	ISIS-affiliated band group Wilayat al-Sham recruits new #terrorists in US-controlled #Rukban camp.
GPT-3 Response	@ALFaham_ #SAA seized a big amount of #ISIS #ISIL weapons from the convoy in east #Homs countryside today. https://t.co/k3kzPoLlsd #Syria #Damascus #Jobar #ISIS
	#Washington is willing to perform the functions of #MiddleEastern policeman and play the role of accessory to #Turkish#aggression against #Syria. https://t.co/HVGXuvRZ
	...
	#Russia's #Air_Force has stepped up airstrikes against camps for the #Islamic_State in the #Terek_Val

and Ginsberg, 2018; Twitter, 2019). It is probable that research on detection of machine generated text for political influence campaigns that focuses on fake news does not accurately reflect the most pressing threat models. Fake news detection research imagines an adversary using an NLG model to produce news-like disinformation at scale (Zellers et al., 2019). Instead, we assess that NLG usage by disinformation agencies for newswriting will be primarily limited to leveraging AI writing assistants such as Jasper to save time and cost (Jasper AI, 2022), and translation models to more effectively cross language barriers. We believe that producing massive volumes of news-like content is a less desirable machine generated text disinformation scaling approach than social messages for several reasons:

- Research has demonstrated that individuals are more likely to share an article than read it (Gabielkov et al., 2016), and that a majority of individuals make up their minds on news topics by only reading headlines (American Press Institute, 2014).
- Scaling by number of articles does not multiply effectiveness — a single news article or handful of news articles can be widely disseminated, reducing the need to generate large numbers of articles each day.
- Scaling by number of articles requires either manipulating existing platforms to host them (i.e., layering and information laundering (Meleshevich and Schafer, 2018)), or procuring domain names and hosting infrastructure, representing additional cost and effort.

- Human involvement in fake news article authorship allows disinformation threat actors to better tailor messaging, reduce detection, and more carefully walk the line of promoting manipulative information without triggering moderation from social media websites

Instead of using NLG models to generate articles themselves, NLG models are more likely to be used for disinformation by operating social bots that distribute links to disinformation articles, promote discussion around incendiary headlines, and produce large numbers of comments that give the false impression of a common public consensus. Targeted users need not even read shared articles — the artificial amplification of a headline and overwhelming "grassroots" narrative guided by machine generated comments would likely be sufficient to influence public opinion (American Press Institute, 2014; Gabielkov et al., 2016).

Measuring the impact of online influence campaigns is generally considered difficult (Meleshevich and Schafer, 2018), but the effectiveness of such political influence campaign might be assessed through analytical processing of the evolving political sentiments expressed on social media (Olorunnimbe and Viktor, 2015).

Regarding mitigation, past research has identified that the average user is overly trusting of profiles with AI-generated photos and GPT-2 text, accepting connection requests from deepfake profiles on LinkedIn 79%–85% of the time (Mink et al., 2022). As such, it is unlikely that user reports will serve as an adequate first line of defense. Instead, a combination of automated detection models (including machine generated text detection) and platform moderation efforts should be used to detect political influence campaigns. Among these should be measures to protect against social media abuse more broadly, including detection of account automation, and scrutiny of coordinated inauthentic activity for content amplification. Investigations by disinformation researchers, such as those carried out on Twitter are likely to remain relevant (Twitter, 2019).

In the extreme case, influence campaigns can include calls to disrupt society or carry out violent acts (e.g., attacking government offices). While publicly inciting violence is generally against platform regulations, such behaviour may appear on less-regulated platforms or via direct message. Existing NLP models can be used to perform surveillance of social media to identify individuals exhibiting signs of mental health disorders so they can receive help (Skaik and Inkpen, 2020). A malicious user may use similar approaches to identify individuals vulnerable to incitement towards specific violent acts — finding likely targets for radicalization.

Commercial Influence Campaigns

In commercial influence campaigns, the goal is to influence individuals in a manner that commercially benefits the threat actor. Examples of such campaigns include publishing fraudulent reviews, artificially boosting a website's search engine page ranking, spamming online communities with advertisements for a product, or attempting to inorganically cause promotional content to trend on social media. As with previous categories, there may be overlap between different threat actor approaches.

A threat model of particular focus is the usage of machine generated text to generate fraudulent reviews that either promote one's own product/service, or target a competitor (Adelani et al., 2020; Kowalczyk et al., 2022; Stiff and Johansson, 2022). Work has been published that demonstrates sentiment-preserving fake reviews, which might be used for such a purpose (Adelani et al., 2020). Fake reviews can be abused on marketplace websites themselves, or by targeting potential customers on social media platforms. Threat actors may operate such campaigns themselves, or may avail themselves of the thriving market for fake reviews (He et al., 2022). Organizations selling fake reviews may become early adopters of open-source NLG models to provide unique and specific reviews at lower cost.

Mitigation of NLG models used for fake reviews on online marketplaces might involve running machine generated text detection on posted reviews, in addition to existing features. Advanced NLG models should not affect context-based detection methods (e.g., identifying patterns in reviewer usernames, similar account creation times, unusual purchase behaviour, etc.). It may be more difficult to detect if commercial influence content is posted outside marketplace websites. As examples, social media websites (e.g., Facebook, Instagram, Reddit, YouTube comments), map platforms (e.g., Google Maps), or dedicated review sites (e.g., Yelp) may all be locations where false reviews may be posted.

Impacts of Attacks and Mitigation on Trust

In addition to the risks posed by machine generated text for online influence campaigns, the existence of NLG threat models causes additional damage to trust online. The perception that any given user on social media may be a bot, can cause users of social media to dismiss others (particularly individuals whom they don't agree with) as "bots", rather than acknowledge that other real people may hold different viewpoints. The net effect of this is reduced trust in the authenticity of social media.

Mitigation of NLG-enabled influence operations via automated detection of machine generated text also itself carries potential negative impacts. Automated detection creates the possibility of mass-suppression of speech online. Previous work has found that text written by non-native English speakers that included political topics was of high risk of being erroneously detected by a Transformer trained on previous political influence campaigns (Crothers et al., 2019). As methods based on RoBERTa (also a Transformer) are currently the state of the art for detection of machine generated text (Liu et al., 2019), classifiers for machine generated text detection leveraged to combat online influence campaigns must be carefully trained and ethically evaluated to minimize the risk of similar incidences of mass discrimination. Continued public reporting of influence campaign datasets, such as the regular releases by Twitter for review by researchers (Twitter, 2019), would be beneficial to protecting trust in social media moderation.

Language background considerations evoke another problem: there are legitimate reasons why a user may rely on machine generated text. A person writing in their non-native language may leverage an online translation model to assist them. While such text may be considered machine generated text, this text is not *inauthentic* — it nevertheless represents genuine self-expression. Much of the world relies on translation tools to better participate in online discourse; recall that 1 in 3 internet users aged 16 to 64 have used an online translation tool in the last week (Kemp, 2022a). Relying on content features alone is therefore likely to produce a solution that is discriminatory, unreliable, and greatly damages trust in social media platforms. Machine generated text detection should then be used among multiple features, such as account creation times, activity patterns, registered phone numbers, and IP addresses, to determine whether activity is linked together as part of an online influence operation.

Moderation of users based on the presence of machine generated text may therefore unduly suppress the speech of minority language speakers. As such, content-based features derived from a machine generated text detection model should likely be only one features among many utilized to perform threat scoring of accounts or posts. A viable approach might use such a model to identify a group of accounts that are likely producing machine generated text, and then utilize other features, such as account creation times, activity patterns, registered phone numbers, IP addresses, to determine whether they are linked together as part of an online influence operation. Relying on content alone is likely to produce a solution that is discriminatory, unreliable, and greatly damages trust in social media platforms.

In news media, one could conceivably generate several influence-oriented op-ed articles on current topics with assistance of an NLG model, and rotate the submissions of such articles to major news outlets. If multiple individuals adapt such an approach, news outlets may have to adjust how they handle op-ed submissions, and may wish to screen submissions based on likelihood of authorship as scored by a machine text detection algorithm such as those to be discussed in Section 2.5. Similar challenges may exist for other venues for text content submissions, which we now discuss further.

2.4.4 Exploiting AI Authorship

Academic Fraud

Use of algorithms to generate scientific papers has been well-established since SCIgen was created in 2005 to produce nonsensical papers that nevertheless sometimes passed peer review (Hargrave, 2005). These papers continue to emerge in respected publications, many years later, despite the comparative simplicity of the context-free grammar generation method (Cabanac and Labbé, 2021; Labbé and Labbé, 2013). Generation of artificial scientific papers uses up valuable reviewing resources, lowers publication quality standards by producing misleading or nonsensical publications, and challenges trust in the scientific review process itself. In education, NLG models may be used by students to cheat on language learning assignments via machine translation models (Song et al., 2019), or easily produce essays on a given topic (Dehouche, 2021; Feng et al., 2018a) — both instances where institutions may wish to perform detection of machine generated text to improve academic integrity and encourage students to learn course material. Widespread access to convenient NLG interfaces online, such as that provided by ChatGPT (OpenAI, 2022), allow any student with an internet connection to leverage such models, even when doing so undermines the learning objectives of an assignment (i.e., cheating).

Threat actors submitting AI-generated papers are typically either 1) academics attempting to inflate publication statistics, particularly when meeting a quota in order to maintain their position (Cabanac and Labbé, 2021); or 2) well-meaning researchers probing the publication standards of a potentially disreputable conference (Zhuang et al., 2007). Capabilities of threat actors include usage of well-established tools such as SCIgen, or usage of more recent Transformer-based approaches that are promoted as "scientific writing assistants" which can nevertheless be easily exploited to generate long articles of little substance (Meroño-Peñuela and Spagnuelo, 2020). Mitigation measures should include flagging likely machine-authored publications using published approaches for detection of SCIgen articles (Cabanac and Labbé, 2021; Labbé and Labbé, 2013), as well as new detection approaches based on detection of Transformer generated text (Rodriguez et al., 2022). Human reviewers can more carefully review flagged articles to determine whether the article contains credible research, irrespective of the detection result.

Questions around the acceptability of machine text within scientific writing may be a future area of discussion in academic disciplines. If the results published by a researcher are true and accurate, limited usage of a carefully guided NLG model may be considered acceptable by some publications. Research has been emerging that aims to differentiate between acceptable and unacceptable usage of NLG models in scientific writing (Rosati, 2022), which should be part of a broader ongoing social conversation on norms surrounding AI usage and disclosure.

Applications and Cover Letters

Contemporary NLG models can be used to generate large numbers of cover letters or essays for applying for scholarships or to employment opportunities. Commercial websites already exist for producing cover letters using GPT-3 (Open Cover Letter, 2022). While the general usefulness of human-written cover letters has been debated in business media (Lufkin, 2021), they are ostensibly meant to be an earnest reflection of a candidate. Usage of AI models to generate a cover letter or essay submission is therefore likely to be considered exploitative by organizations who review such submissions. The threat actors in this case may be individuals (perhaps understandably) looking to save time and improve their employment opportunities by bypassing a cumbersome application process, or a malicious attacker looking to flood a target company with fraudulent submissions (a threat actor which we will discuss further in "Spam and Harassment").

Detection of machine generated text may be able to identify artificial cover letters or essays given they are of sufficient length (the odds of successful detection improve with sequence length (Ippolito et al., 2020; Radford et al., 2019b; Zellers et al., 2019)). However, caution should be taken with this approach, as use of AI writing tools is not necessarily exploitative. Again, individuals from minority language backgrounds may rely on translation models or NLG writing assistants to help them write cover letters or scholarship applications. It may be difficult to differentiate those who mean to exploit such systems (e.g., spam submissions to as many avenues as possible with no genuine thought or expression), and those who are relying on AI writing tools to better express themselves. As such, a better mitigation approach may be to develop alternative approaches to evaluating candidates, such as placing more emphasis face-to-face discussions with prospective job candidates or award recipients.

Content Generation

A threat model for social media platforms is that a large number of creators or users may begin using generative AI models (including NLG models) to produce social media content in ways that harms these platforms. While threat actors in this case may not be overtly malicious, generative models may dilute the quality of content on a platform, undermine trust in platforms more generally, or create plagiarism concerns. As a recent example, in response to the release of highly effective AI models for image generation (DALL-E (Ramesh et al., 2021), Stable Diffusion (Rombach et al., 2021)), a number of art websites have enacted a blanket ban against all AI-generated art (Edwards, 2022).

The impact of NLG models on trust on social media in particular is important. People increasingly gain their understanding of the world through social media, with the most popular platforms increasingly focused around video (Eddy, 2022). Parasocial interaction between major content creators and their audience is a key component of how online video platforms such as YouTube function (Chen, 2016). While many established creators may avoid using NLG tools in order to preserve the perception of authenticity, it is likely that some creators (particularly groups that produce low-effort content at scale) may adopt this technology to more efficiently scale content production, which may reduce trust in channels or creators that appear to leverage this approach. Utilization of machine generated text for social media content in general, including video social media, may have a large impact on the ability to trust others online, and must be managed in such a way as to preserve trust and limit abuse.

Video is a particularly important medium on modern social media: there are approximately 4.95 billion Internet users on Earth (Kemp, 2022b), of these, an estimated 92.6 percent watch digital videos each week (Kemp, 2022a). The interplay between social media creators and generative models represents important sociotechnical context to avoid common Fair ML traps (Selbst et al., 2019). Award-winning online commentator Drew Gooden performed a video demonstration of GPT-3-based writing assistant Jasper (Jasper AI, 2022), critiquing applications of Jasper for production of video scripts and social media content (Gooden, 2022). When attempting to generate a bio for a company website, Gooden found that Jasper produced a sample that directly plagiarized a Newswire article (timestamp 11:55). Gooden also noted that utilizing such a tool without disclosure would violate the trust of viewers (timestamp 4:22).

> *Just so it's not in the back of your head while watching this, the twist at the end isn't going to be like, "Surprise, the computer wrote this whole video, gotcha." [...] **I feel like that would make you trust me less. And this parasocial relationship is not going to work if you don't trust me.***

The tone of Gooden's assessment is tongue-in-cheek, and intended for a broad audience including over 2.5 million views at the time of writing. However, it makes two useful observations: 1) the success of social media creators fundamentally depends on the trust of their audience, and 2) undisclosed use of AI models to generate social media content may undermine this trust. It is telling that Gooden felt obliged to put a pre-emptive disclaimer that no AI content was present in the video, proactively settling any doubts in the mind of the audience.

Creators are in the early stages of using such tools to assist in content creation. Anthony Fantano, described by the New York Times as "probably the most popular music critic left standing" (Coscarelli, 2020), has released a completely AI-scripted album review video (Fantano, 2022). The AI script in the review is leveraged for comedic reasons, rather than as a credible method of streamlining the script writing process of an earnest review, but the video nevertheless begins with an immediate full-screen and spoken disclaimer that the video script was generated by an AI.

With respect to AI video script authorship, the decision by Fantano to disclose preemptively the usage of a model, as well as Gooden's decision to preemptively disclose that a model was *not* used, both demonstrate the perceived importance of pre-disclosure of AI content on maintaining trust with a social media audience. In fact, this approach aligns well with norms developing in licensing of NLG models, such as the Responsible AI License used for BLOOM (Ferrandis et al., 2022) that mandates pre-emptive disclosure.

It is nevertheless important to observe that both AI-generated scripts, as well as the concern that a creator may use AI to generate scripts without telling their viewers, both pose risks the perception of authenticity to form parasocial relationships with their audience.

Mitigations of threats related to undesired inclusion of NLG content in social media may involve similar blanket bans to those targeting AI-generated art (Edwards, 2022), or policies that mandate pre-emptive disclosure of the usage of AI tools as part of a platform's terms of service (similar to the requirements mandated in the Responsible AI License (Ferrandis et al., 2022). The enforcement of such policies would necessitate a combination of machine generated text detection algorithms and moderator investigations.

Concerns around reputational and legal risk of machine generated text are not limited to traditional text media such as news articles — online social media is a major avenue of modern society. Social media creators may leverage generative models in content creation, but such usage if made public may undermine trust of audience members with those creators, or of trust in social media platforms in general. NLG models are not only usable for authoring text posts — NLG models can be used to assist in production of video scripts as well. Attending perspectives across varying modes of media is necessary to produce human-centric systems and uphold principles of trustworthy AI (European Commission and Directorate-General for Communications Networks, Content and Technology, 2019).

Impacts of Attacks and Mitigation on Trust

The widespread usage machine generated text in written submissions may undermine the trust that individuals place in such written works, and lead to greater scrutiny of such material. Given that a suitable cover letter with language fitting for a position can be trivially generated by existing user-friendly tools (Open Cover Letter, 2022), it is possible that employers will soon place so little trust in cover letters that they eschew them altogether. Reviewers of scientific publications may worry that sections of papers they read may be machine generated text that only appears scientific at a glance. Internet users may likely interpret algorithm-generated blogs, articles, and video scripts as low-effort and untrustworthy.

Mitigation processes must be used carefully. As previously mentioned, it is possible the detection of machine generated text may unfairly skew towards false positive classification of individuals with certain language backgrounds. There may be cases where usage of machine generated text is permissible (e.g., translation models or assistive writing technologies). The perception that an individual may be unfairly screened out from consideration due to erroneous false positive detection may reduce trust. Submitting a scientific paper only to have a reviewer allege that a given section might be written by an algorithm could lead to a loss of faith in the review process.

To preserve trust, usage of machine generated text should generally be preemptively disclosed to the reader or audience. In many cases, content authored by machines may carry a negative connotation to the audience, and may undermine trust with a particular publication platform, news website, or brand. Media and entertainment organizations that publish content from multiple creators may decide to enforce that certain categories of content submissions they publish are to be completely written by humans. Similarly, such organizations may also be concerned with spam of low-quality machine generated content overwhelming editorial staff, or wish to reduce the risk of plagiarism or copyright infringement as some models have been found to memorize training data which can emerge during inference (Carlini et al., 2020).

2.4.5 Spam and Harassment

We distinguish spam and harassment from other categories of attacks by focusing on cases where the goal of the attack is to harm a platform or its users with a large volume of content. As in previous cases, there are overlaps with other threat models, but the distinction of spam use-cases is useful for understanding related threats.

Social Media Spam

Social websites are an attractive target for attacks using large volumes of machine generated text, providing opportunity for significant disruption. One researcher demonstrated a real-world attack by using a GPT-2 bot to generate 55.3% of all comments on a federal public comment website before voluntarily withdrawing the comments and shutting down the bot (Weiss, 2019). It is important to realize that spam attacks against social media websites are often already possible — high-quality NLG models simply make spam attacks more difficult to detect as posts can be unique and better match the style and substance of discussion.

Usage of generative models to produce large volumes of hateful spam targeting specific groups and individuals is a particular cause for concern. While OpenAI attempts to reduce the incidence of offensive content generated by its GPT-3 API through careful training measures and filtering of inference prompts (Brown et al., 2020), open-source models are not subject to any such restrictions. GPT-4chan, which was trained on and subsequently deployed to create a large volume of posts on the 4chan politics message board, provides a complete example of how such a model might be created and deployed to cause havoc (Kilcher, 2022; Kurenkov, 2022). An attacker with sufficient motive (political, personal, or otherwise) may render an entire community nearly unusable with spam.

Mitigation measures in the area of automated spam should rely heavily on methods designed to prevent automated posting of comments in general. Approaches to this include increased scrutiny of proxy and VPN usage, typically used in conjunction with Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) (Ahn et al., 2003) challenges to verify that a user is human. Notably, both of the previous examples of Transformer-based spam take advantage of either 1) a lack of CAPTCHA tests (Weiss, 2019), or 2) a method of bypassing CAPTCHA and proxy restrictions (Kilcher, 2022). CAPTCHA is not a perfect defense — iterative versions of human-verification schemes and bypass methods are in continuous adversarial development (Guerar et al., 2021) — but such defenses an important first step to increase the difficulty of automation. As spam results in large volumes of text, and detection of machine generated text is easier on long sequence lengths (Ippolito et al., 2020), many comments from the same user or IP range could be combined to generate a large sample for effective machine generated text detection.

Harassment

Techniques similar to spamming may be used to cause distress to individuals or communities by targeting them with a large volume of messages. An individual or group of motivated individuals may register social media accounts to be controlled by automation tools, or use a common bot to post from their own account, in order to generate a large volume of messages targeting a particular individual or community. SMS and phone call automation tools may facilitate such approaches outside social media as well.

The motivations of threat actors engaging in such behaviour may range from personal grudges to political objectives. Online communities formed around religion, racial identity, sexual orientation, or gender expression, may be at risk of *brigading* (Andrews, 2021) from hate groups using such models to flood them with abuse. Political figures or political discussion boards of all stripes may be at risk from large-scale automated harassment from motivated enemies among their political adversaries.

A website such as 4chan is likely to struggle the most in mitigating such attacks, due to several factors: 1) website users can post anonymously, removing the need to create throwaway emails or accounts, and making it difficult to tie together multiple posts 2) the website offers a method of allowing the usage of shared proxies/VPN and bypassing CAPTCHA. Greater scrutiny of the volume of a user's posts may accompany user usage of anonymization technologies such as proxy servers and VPNs.

Mitigation measures similar to spamming apply for counteracting harassment as well — the best defenses include verification that an individual is human prior to making a post or sending a message, targeting the automation of delivery rather than the machine generated text.

Document Submission Spam

Platforms previously mentioned in "Exploiting AI Authorship" may be vulnerable to being overwhelmed purely through volume of AI generated content. A motivated attacker might submit massive volumes of unique cover letters and resumes to a company, none of which actually corresponds to a real individual, thus frustrating attempts at recruiting. Depending on the method of submission, scientific conferences or news op-ed submissions may be vulnerable to reviewers being overwhelmed by content that is difficult to distinguish from real submissions without a time-consuming review process. Detection of machine generated text may be a useful mitigation measure for these cases, combining pre-screening of content based on likelihood of being written by a machine, in addition to CAPTCHA (Ahn et al., 2003) challenges to reduce automated submissions.

Spam attacks are generally easily transferable between different public social media websites, as platform-specific text preprocessing and pre-training allows for appropriate text to be generated for each platform to be attacked. Additionally, it is worth highlighting that document submission websites previously mentioned in "Exploiting AI Authorship" may similarly be vulnerable to being overwhelmed by AI generated content. A motivated attacker might submit massive volumes of unique cover letters and resumes to a company, none of which actually corresponds to a real individual, thus frustrating attempts at recruiting. Depending on the method of submission, scientific conferences or news op-ed submissions may be vulnerable to reviewers being overwhelmed by content that is difficult to distinguish from real submissions without a time consuming review process. Detection of machine generated text may be a useful mitigation measure for these cases, combining pre-screening of content based on likelihood of being written by a machine, in addition to CAPTCHA challenges to reduce automated submissions.

Impacts of Attacks and Mitigation on Trust

Similar to other attacks, the impact of spam on harassment on trust in online communities is to harm the belief that other individuals online are really human. Even following the deactivation of the deployed GPT-4chan bot, discussion on 4chan continued to express concern that subsequent posts may be made by NLG models (Kilcher, 2022). The more frequently individuals knowingly encounter such models in social media, the less trust they will have in the integrity of online social spaces.

Mitigation of such attacks would incorporate increased verification of human posting activity. Such restrictions would likely include limitations on usage of known proxies and VPNs, and potentially requiring the provision of additional information on sign-up (e.g., emails, phone numbers, payment methods, government IDs), and an increased burden of CAPTCHA challenges. The overall result of this is a reduction of online privacy, and increased barriers to participation in online discussion — both of which may harm user trust in online platforms.

Finally, as spamming or harassment operations can be very disruptive, they may represent a highly visible case of AI model abuse. As such, the abuse of such models in online communities may cause a general decrease in public trust towards AI model development, and NLG models in particular.

2.4.6 Summary of Threat Models

Within this section we have discussed a wide range of threat models associated with natural language generation. We summarize our key findings as follows:

- NLG models have significant potential for abuse in improving scaling and targeting of existing attacks
- Platforms that receive text submissions of any kind are likely to face a growing influx of machine-generated text content, particularly as user-friendly tools continue to be developed (Jasper AI, 2022; Open Cover Letter, 2022)

- Much of the research on NLG-enabled influence operations focuses on AI-generated news articles, while sociological data suggest that machine generated comments pose a much greater threat
- While NLG models may make detection of automated coordinated inauthentic activity more difficult, abuse often still requires bypassing existing defenses such as IP reputation checks and CAPTCHA (Ahn et al., 2003)

Future threat modeling and observed cyberattacks will certainly augment the threat models discussed in this section, but we have now provided sufficient motivation for exploring the defensive capabilities offered by machine generated text detection. In the next section we will discuss the current status of research on detection of machine generated text, and outline the major findings in the field thus far.

2.5 Detection of Machine Generated Text

Analysis of threat models indicates that the detection of machine generated text is a valuable tool for reducing the harms of NLG model abuse. Detection of machine generated text is typically framed as a binary classification problem in which a classifier is trained to differentiate samples of machine generated text from human generated text (Crothers et al., 2022a; Lavoie and Krishnamoorthy, 2010; Nguyen-Son et al., 2017; Solaiman et al., 2019; Zellers et al., 2019), though there exists related research in attribution of machine generated text to the model that generated it (Munir et al., 2021; Uchendu et al., 2020a) which we will discuss in §2.6.2.

In this section, we outline the methods used for detection of machine generated text. In §2.5.1 we summarize feature-based approaches in machine generated text detection, while §2.5.2 covers detection approaches based around neural language models. In §2.5.3, we survey domain-specific research on applications of machine generated text detection. In §2.5.4, we review the ability of human reviewers to correctly identify machine generated text, and human-aided machine generated text detection. In §2.5.5 we discuss trends in evaluation methodology within detection research. Finally, in §2.5.6, we explain prompt injection: a method of shaping NLG model responses, which may facilitate detection. Table 2.3 provides a summary of major detection methods and their evaluation in current research.

2.5.1 Feature-Based Approaches

Machine generated text differs from human text in ways that be identified using statistical techniques (Crothers et al., 2022a; Fröhling and Zubiaga, 2021; Nguyen-Son et al., 2017). Feature-based approaches to machine generated text detection apply natural language processing to create feature vectors from input sequences, and classify these feature vectors using a downstream classification algorithm, such as a support-vector machines (SVM), random forest (RF), or neural network (NN) (Fröhling and Zubiaga, 2021; Nguyen-Son et al., 2017). We provide a summary of the categories of features that have been used in prior art, with references for further reading on specific categories of features.

An important consideration in detection of machine generated text using feature-based approaches is that different language model sampling methods (e.g., top-k versus top-p sampling in Transformer language models, as discussed in §2.3.3) may lead to different artifacts in the generated text (Fröhling and Zubiaga, 2021; Holtzman et al., 2019b). As a result, performance of feature-based detection can be diminished when detecting machine text generated using a different sampling approach than detection model training examples (Fröhling and Zubiaga, 2021). A feature-based detector trained on output from a smaller model can be used to detect output from models of larger size (Fröhling and Zubiaga, 2021; Zellers et al., 2019), though it is more effective to use a detector trained on a larger model to detect output from smaller generative models (Fröhling and Zubiaga, 2021).

Previous work has been done on detection of machine generated text based on detection of translation models (Nguyen-Son et al., 2017). This is done based on the idea that other generative models in more sensitive attacks may use a similar approach, or that there may be cases where translation models themselves may be an avenue for abuse (e.g., a student cheating on a language-learning assignment, a social media spambot that populates its feed by translating social media texts from other accounts to pass off as its own, etc.).

We now proceed with our summary of major feature categories in feature-based detection approaches.

Frequency Features

A major category of statistical features used in detection of machine generated text center around the frequency of terms within text samples. Human-written text often conforms with Zipf's Law: the frequency of a word is inversely proportional to its rank in an ordering of words by frequency (Zipf, 1949). With Zipf's Law, the normalized frequency f of a token of rank k out of N different tokens follows the relationship:

$$f \approx \frac{1/k}{\sum_{n=1}^{N}(1/n)} \tag{2.2}$$

Machine generated text does not perfectly mirror the distribution of tokens in human text, with variation in Transformer language models dependent on sampling method chosen (see Figure 7 of Holtzman et al., 2019 (Holtzman et al., 2019b)). The distribution of tokens therefore provides useful discriminating power, particularly when a greater volume of text is available for consideration.

Another major frequency-based feature from previous statistical detection research is term frequency — inverse data frequency (TF-IDF). TF-IDF unigram and bigram features used with a logistic regression detector have been used as a baseline for detection (Radford et al., 2019b; Solaiman et al., 2019) or as a feature in statistical approaches (Fröhling and Zubiaga, 2021).

Lemma frequency has also been used as a statistical feature in previous research (Crothers et al., 2022a; Nguyen-Son et al., 2017). In this approach, a linear regression line that fits log-log lemma frequency versus rank is learned, and then mean-square error cost function can be used to calculate information loss of the regression.

Due to observed repetitiveness of writing produced by NLG models (Gehrmann et al., 2019a; Holtzman et al., 2019b), another potentially useful frequency features is n-gram overlap of words and parts-of-speech tags between sentences (Fröhling and Zubiaga, 2021). An additional technique targeting machine text repetitiveness computes supermaximal repeated substrings (i.e., the set of the longest repeated substrings, excluding all substrings which are already part of a longer repeated substring) in large collections of text to enable detection (Gallé et al., 2021).

Fluency Features

Another major category of features are those centered around the fluency or readability of generated text. Longer sequence lengths of machine generated text are increasingly likely to encounter issues producing consistently coherent and clear text (Holtzman et al., 2019b;

See et al., 2019). The Gunning-Fog Index and Flesch Index, provide a statistical measure of text readability and comprehensibility respectively, and have been shown to be effective in detection of machine generated text (Crothers et al., 2022a). More complex measurements using coreference resolution by an auxiliary model to create a proxy measures of coherence based on the presence of a text's main entities in important grammatical roles, and the usage of Yule's Q statistic for coherence (Fröhling and Zubiaga, 2021).

Linguistic Features from Auxiliary Models

Past research has measured the "consistency" of machine generated text by calculating the number of phrasal verbs and coreference resolution relationships within a sample (Crothers et al., 2022a; Nguyen-Son et al., 2017). Other work has used the entire distribution of a text's part-of-speech (POS) tags, and named entity (NE) tags (Fröhling and Zubiaga, 2021). Such work is motivated by differences between human and machine POS tag distributions observed in past analysis of machine generated text (Radford et al., 2019b; See et al., 2019).

Performing coreference resolution, and assigning POS and NE tags requires processing samples with specialized models. Contemporary models for this purpose are neural in nature, and as such, modern feature-based approaches leveraging linguistic features from auxiliary models perform inference on a neural network as part of the creation of feature vectors (Crothers et al., 2022a; Fröhling and Zubiaga, 2021). That is, such methods are not strictly non-neural.

Complex Phrasal Features

Detection work targeting translation of long texts found that certain idiomatic phrases were not commonly found in machine text (Nguyen-Son et al., 2017). However, recent work has shown that these features do not perform well against contemporary Transformer models on shorter sequence lengths (Crothers et al., 2022a).

Basic Features

Finally, there are many simple text features that are commonly used in feature-based text classification in natural language processing. These include simple high-level characteristics of sentences such as the number of punctuation marks, or length of sentences and paragraphs, which have been used in detection of machine generated text (Fröhling and Zubiaga, 2021).

2.5.2 Neural Language Model Approaches

Detection approaches based on neural networks — particularly those that incorporate features derived from Transformer neural language models (NLMs) — are highly effective for detection of machine generated text. This aligns with broader trends in natural language processing where state-of-the-art performance has been attained on a wide range of natural language tasks using Transformer models (Papers With Code, 2022).

We separate NLM-based approaches into two major categories: zero-shot classification using existing models, and fine-tuning of pre-trained language models. These two types of approaches represent the overwhelming majority of NLM-based machine generated text detection.

Zero-shot Approach

A baseline approach to detection of machine generated text is performing text classification using generative models themselves, such as GPT-2 or Grover (Radford et al., 2019b; Solaiman et al., 2019; Zellers et al., 2019). Generative models can themselves be used without fine-tuning to detect either their own outputs, or outputs from other (typically similar) generative models. Autoregressive generative models such as GPT-2, GPT-3, and Grover are uni-directional, with each token embedding having an embedding that is dependent on the embeddings of preceding tokens. As a result, an embedding for a sequence of tokens can be created by appending a classification token `[CLS]` to the end of the input sequence, and using the embedding of this token as a feature vector for the entire sequence. Using these feature vectors, a labelled dataset of human and machine text can be used to train a linear layer of neurons for classifying whether an input sequence is produced by a machine or human.

It has been observed in multiple studies that smaller NLG models can be used to detect text generated by larger NLG models (Crothers et al., 2022a; Solaiman et al., 2019; Zellers et al., 2019). While the ability of a model to detect larger models does diminish as the difference in scale grows, the predictive ability of smaller architectures may be useful as recreating large multi-billion parameter Transformer architectures is highly compute-intensive.

Grover, a model trained for generation and detection of "neural fake news", demonstrates strong zero-shot detection performance specifically within the news domain it was trained on (Zellers et al., 2019), but shows limited performance on out-of-domain text (Solaiman et al., 2019; Uchendu et al., 2020a). While it was initially suggested by Grover's authors that the best detection method for generative models may be generative models themselves (Zellers et al., 2019), this has not been reflected in more comprehensive research that has suggests the increased representational power of bi-directional Transformer models is much more effective than uni-directional models for machine generated text detection (Solaiman et al., 2019).

Similar to the weakness of Grover outside of the news domain, it has been found that the zero-shot approach generally underperforms a simple TF-IDF baseline when trying to detect output from a generative model that has been fine-tuned on a different domain (Solaiman et al., 2019). As it is likely that attackers may fine-tune generative models for different purposes, this represents a notable weakness in the zero-shot approach of using generative models without fine-tuning.

Fine-tuning Approach

The state-of-the-art approach for neural detection of machine generated text is based around fine-tuning of large bi-directional language models (Solaiman et al., 2019). In this approach, initially evaluated on GPT-2 text, RoBERTa (Liu et al., 2019) — a masked general-purpose language model based on BERT (Devlin et al., 2019) — is fine-tuned to differentiate between NLG model output and human-written NLG model training samples.

The source code for this fine-tuning approach is available open-source, as are pre-trained detector models, facilitating future research and defensive detection (Radford et al., 2019a; Solaiman et al., 2019). The pre-trained detection models available are based on the RoBERTa-base (123M parameter) and RoBERTa-large (354M parameter) architectures (Liu et al., 2019). The machine generated text used to fine-tune these models was generated by GPT-2, using a mixture of pure sampling and nucleus sampling (see §2.3.3). The intention of using a training dataset that contains multiple sampling methods is to generalize more effectively to unknown sampling methods that may be used by attackers in-the-wild — an approach that should likely be duplicated in future detection research.

Research into the practicalities of machine generated text detection has considered the task of detecting text when a RoBERTa detector algorithm was trained on a different dataset than a GPT-2 attacker model. In this case, it was found that by fine-tuning the detector model with even just a few hundred attacker samples identified by subject-matter experts (SMEs), the detector is able to dramatically improve cross-domain adaptation (Rodriguez et al., 2022). This reflects likely real-life scenarios where a general-purpose detection model comes up against a fine-tuned attacker for a particular purpose. As a defender identifies samples from a fine-tuned attacker model, these examples could be used to further improve the defensive detection model.

Preliminary work has used attention map information from Transformer models to perform topological data analysis (TDA) as features for detection of machine generated text (Kushnareva et al., 2021). This did not show significant improvement over standard BERT fine-tuning approaches, though (in light of similar considerations regarding potential fine-tuned attacker models) the resulting features were better able to detect unseen GPT classifiers. It is unclear how the TDA approach would compare in effectiveness if directly applied to the current state-of-the-art RoBERTa detection models (Solaiman et al., 2019), rather than custom-trained BERT models.

While research on detection of machine generated text has primarily taken place in English thus far, detection models have also been released in Russian (Shamardina et al., 2022; Skrylnikov et al., 2022) and Chinese (Chen et al., 2022). Further to this, large pre-trained bi-directional Transformer models have been released for numerous languages, including Chinese (Cui et al., 2021), French (Martin et al., 2019), Arabic (Antoun et al., 2020), and Polish (Dadas et al., 2020). Future work on detection of machine generated text in additional languages may leverage such pre-trained bi-directional models as a starting-point for fine-tuning.

Another method of detection leverages energy-based models (LeCun et al., 2006) alongside a classifier of machine generated text. Evaluated approaches include a simple linear classifier, BiLSTM, uni-directional Transformer (GPT-2) and bi-directional Transformer (RoBERTa) (Bakhtin et al., 2021). The Transformer architectures were initialized from pre-trained checkpoints, and then fine-tuned on machine vs human classification datasets. Corroborating other research, this research found the strongest performance by leveraging the bidirectional Transformer (Bakhtin et al., 2019).

The strong performance of fine-tuned bi-directional NLM models — and RoBERTa in particular — has led to these models being well-represented in applied detection research targeting specific domains, as shown in Table 2.3 and discussed next in §2.5.3.

2.5.3 Applied Detection in Specific Domains

Applied work in the area of machine text detection has focused on using techniques and technologies for detection of machine text in specific domains. This applied research is important as it addresses several of the serious threat models discussed in §2.4, and includes broader lessons for machine generated text detection more generally. We divide applied research into several major categories.

Technical Text

Recall from §2.4.4 that machine generated scientific papers have been well-documented since the release of SCIgen in 2005 (Cabanac and Labbé, 2021; Hargrave, 2005; Labbé and Labbé, 2013). Past algorithmic approaches target the SCIgen model (Lavoie and Krishnamoorthy, 2010), but there is also more contemporary research targeting technical text generated by GPT-2 (Rodriguez et al., 2022). This work found that a RoBERTa-based detector could be adapted from one academic technical writing domain (physics) to another (biomedicine) with large improvements made with a number of SME-labelled examples numbering in the hundreds.

Social Media Messages

Application-specific work has applied feature-based (Fagni et al., 2021) and neural (Stiff and Johansson, 2022; Tesfagergish et al., 2021; Tourille et al., 2022) language model based detection methods to social media. Previous work in the social media domain has found that detectability of such text heavily depends on the dataset used to train the generator and detector (Tourille et al., 2022).

Existing work on machine generated text detection has heavily focused on Twitter. Twitter text is quite distinct in that it has common characteristics (hashtags, references, shortlinks), and mandates a short sequence length (280 characters). There is a clear lack of work targeting comments on more popular platforms such as Facebook and Youtube, or fast growing platforms such as Reddit (Auxier and Anderson, 2021). With respect to machine generated text, Reddit content can be found in "SubSimulatorGPT2", a simulation based on a host of fine-tuned GPT-2 models that produce community-specific machine-generated posts and comments and harvested from the Pushshift dataset (Baumgartner et al., 2020).

Chatbots and Social Bots

A related application area is detection of malicious chatbots and social bots, which can interact with humans on chat applications, SMS, and social media. Bots may be used for malicious purposes such as spam, phishing, social engineering, influence operations, or data collection (see the threat models in §2.4). There is clear overlap in this area with research into detection of AI-generated social media messages, but framing the detection challenge by targeting automated personae allows for consideration of additional features. An analysis of the way that humans and chatbots interact has found that chatbot detection can be improved by analyzing how humans reply to the bots, rather than only analyzing the bot text itself (Bhatt and Rios, 2021). Note that bot detection is a large area of research in its own right, and not all social bots use machine generated text (Latah, 2020). As such, features indicating the presence of machine generated text may be only one part of a strategy for social bot detection.

Online Reviews

Applied work has focused on addressing threat models related to commercial influence campaigns, specifically on generating and detecting fake Amazon and Yelp reviews (Salminen et al., 2022). A custom GPT-2 model was fine-tuned for Yelp reviews as part of an evaluation by Stiff et al. 2022 (Stiff and Johansson, 2022).

One work in this area has focused on using random forest classifiers and XGBoost, in order to leverage Shapley Additive Explanations (SHAP) as an explainability technique (Kowalczyk et al., 2022). The use of explainability techniques in detection may be valuable for improving the ability of detection models to provide human-interpretable explanations of moderation decisions, and provide greater transparency into algorithmic decision making applied to social media or product reviews. A lack of coherent explanation may undermine human confidence that a system is truly geared towards detecting fraudulent activity, and is not instead enacting targeted suppression based on benefit to the platform holder (e.g., suppressing negative product reviews for a store brand by holding competitors to a higher standard for "not computer generated").

Hybrid Text Settings

In some cases, it is interesting to detect machine text in settings where both machine and human text is combined together.

There exists a risk that rather than generate attack text entirely from scratch, an attacker may use human-written content as a natural starting point, and instead perturb this information in order to generate human-like samples that also fulfill attacker goals of disinformation or bypassing detection models (not unlike an adversarial attacks in the text domain). Analysis found that performing these types of targeted perturbations to news articles reduces the effectiveness of GPT-2 and Grover detectors (Bhat and Parthasarathy, 2020).

A sub-problem in this space is detection of the boundary between human text and machine text (Cutler et al., 2022). This problem identifies the nature of many generative text models in that they continue a sequence that is begun using a human prompt. While in some cases that prompt would be omitted by an attacker (e.g., generating additional propaganda Tweets from example propaganda Tweets, as we show in Table 2.2), there are cases where the prompt would be included as well (e.g., writing the first sentence of a cover letter, and having a computer produce the rest).

2.5.4 Human-aided Methods

In addition to purely automated methods, there have also been proposed human-aided methods that include a statistical or neural approach in combination with a human analyst for review. This approach has an advantage in providing human agency and oversight (an important principle in trustworthy AI systems), but this does come with reduced scalability due to the need to hire and train human reviewers, particularly given the difficulty of making a confident determination that text is machine generated.

GLTR

Giant Language Model Test Room (GLTR) provides a system designed to improve detection of machine generated text via the inclusion of an integrated human reviewer (Gehrmann et al., 2019a). The GLTR tool augments human classification ability by displaying highlighting on text that reflects the sampling probability of tokens for a Transformer model. However, this tool was devised to target GPT-2, which was found to be significantly easier to detect for untrained human evaluators (Clark et al., 2021). Additionally, GLTR displays highlighting based on the likelihood of a word being selected based on "top-k" sampling. In practice, "top-k" sampling has largely been superceded by nucleus sampling (Holtzman et al., 2019b), which is used in both GPT-3 (Brown et al., 2020) as well as subsequent work that leverages the GPT-2 architecture (Zellers et al., 2019). While highlighting text based on sampling likelihood (as in GLTR) may improve human classification ability, it is highly probable that untrained human evaluators using such an approach would struggle substantially more to detect the models available today, both due to increased model capacity, as well as more advanced sampling methods.

Human Performance in Detection of Language Models

In a review of human evaluation of machine generated text (Clark et al., 2021), it was found that untrained human reviewers were correctly able to identify machine generated text from GPT-3 at a level consistent with random chance. After providing some limited training, evaluator accuracy increased to 55%. While selecting only the best evaluators and giving them more comprehensive training would likely be able to further improve recall, the poor performance of untrained and newly-trained human evaluators highlights the difficulty in relying on human judgement for detecting machine generated text.

A study of human detection ability in comparison to algorithmic detection methods found that the algorithmic approach performed best when humans were fooled, a phenomenon referred to as the"fluency-diversity tradeoff" (Ippolito et al., 2020). As generation approaches have been tailored to produce high-quality text to the perspective of a human observer, text with higher human-assessed quality is more recognizable to an automated approach. This study also includes a useful comparison to previous studies in terms of human evaluator performance. A group of university students were walked through ten examples as a group by the authors prior to performing the evaluation task. These reviewers were substantially more effective at machine generated text detection than previous studies, particularly for longer sequence lengths — accuracy on the longest excerpt length was over 70%. In the context of the study, however, these raters had consistently worse accuracy than automatic classifiers for all sampling methods (random, top-k, and nucleus) and excerpt lengths.

This "fluency-diversity tradeoff" is an important factor to consider when considering possible human contribution to detection of machine generated text. As most threat models require a human target to be fooled into thinking text is authentic, this provides an increased opportunity for algorithms to perform successful detections. In some senses, this is similar to how in detection of machine generated faces produced by StyleGAN, increasing truncation ψ in StyleGAN leads to fewer artifacts and strange less variability and diversity in the output, but pushes all generated faces towards the center of the model's feature space, making them more consistent and easier to detect.

Further demonstrating the advantage of providing specialized training to human reviewers, the Scarecrow framework specifically identifies 10 categories of common errors made in GPT-3 generative text, and trains human evaluators to annotate these errors (Dou et al., 2022). Human annotations of such errors were found to generally be of higher precision than a corresponding algorithm trained on such annotations, but had higher F_1 scores in only half of the categories.

Based on these findings, we can better inform defenses against threat models. For example, in the social media domain, it is possible that if a social media company hired a specialist human moderator and provided them with an intensive training program, that this moderator may be able to work alongside a machine generated text detection algorithm in detecting that a user's posts are likely written by a machine — particularly if there were enough examples of social media posts provided. This approach may be similar to how forensically trained facial reviewers can work alongside algorithms to obtain high performance (Phillips et al., 2018).

While in some contexts, inclusion of a human evaluator may be important from a human oversight perspective (say to prevent widespread automated suppression of protected speech online), it is perhaps worth highlighting that it is not implausible that generative language models may reach the point where the inclusion of a human reviewer may have a negligible impact on the accuracy of a detection system. As such, human involvement in such systems should be specifically motivated by human agency and oversight concerns, not merely out of an anthropocentric belief in human abilities.

The tool "Real or Fake Text" (Dugan et al., 2020) evaluates human detection of machine generated text, by iteratively presenting sentences and asking a human reviewer whether the next sentence was written by a human or a machine, encouraging the reviewer to correctly identify the boundary between the human and machine generated text. Once the human believes they have found a machine generated line, they can select reasons from a list, as well as provide free-form feedback as well. Research based on the RoFT data has not yet been published, but such tools may give greater insight into expert reviewer abilities on identifying boundaries between human and machine text.

2.5.5 Trends in Evaluation Methodology and Datasets

Evaluation of machine generated text detection has trended towards increased focused on generative Transformer language models. Table 2.3, which is arranged chronologically, shows the dramatic shift in evaluation since the release of GPT-2 in 2019. The most common contemporary evaluation dataset in detection of machine generated text remains the GPT-2 output dataset (Radford et al., 2019a), though similar GPT-3 samples released by OpenAI are considered in more recent work (OpenAI, 2020). A table summarizing sample counts in several of the most common datasets can be found in the appendix of a previous survey (Jawahar et al., 2020) — we focus this section on the nuances of evaluation of machine generated text detection, including parameters, model architectures, and the usage of publicly available NLG models to produce new machine generated text at will.

Recall from §2.3.3 that there are a number of sampling parameters important to Transformer NLG models. The GPT-2 output dataset includes sample outputs from GPT-2 models at varying parameter counts (117M, 345M, 762M, 1542M), and two sampling settings: top-k sampling at $k = 40$, and pure sampling at $T = 1$. This dataset now also contains a sample of Amazon product reviews generated by a 1542M parameter model with both $k = 40$ and nucleus sampling. The 175B parameter GPT-3 samples use top-p sampling at $p = 0.85$. The samples available for Grover, which is specifically fine-tuned to generate news articles, uses top-p sampling at $p = 0.96$ (Zellers et al., 2019).

In addition to the GPT-2, GPT-3, and Grover datasets, the dataset used for research on attribution of machine generated text to the language model that produced is useful in general machine generated text research as it provides samples of a variety of generated text methods (Uchendu et al., 2020a). As such, it has recently been used outside of attribution on research focusing on detection as well (Stiff and Johansson, 2022).

Variation in NLG model architecture and decoding method is important as both greatly influence the quality and detectability of generated text (Ippolito et al., 2020). In practice, a defender may not know the characteristics of the generator being used, and as such, detection research that evaluates performance when there is a mismatch between datasets, model architectures, and parameters between training and evaluation is of particular real-world relevance. A detailed analysis on feature-based detection of machine generated text has included such comparisons (Fröhling and Zubiaga, 2021), as has more-specific applied research focused on detecting GPT-2 tampered technical writing (Rodriguez et al., 2022).

Sequence length is another important factor in evaluation of machine generated text. Longer sequence lengths are beneficial to detection (Ippolito et al., 2020; Radford et al., 2019a; Solaiman et al., 2019; Zellers et al., 2019). Sequence lengths in the most common evaluation datasets are 2048 tokens (OpenAI, 2020; Radford et al., 2019a). Sequence length is important in applied research where longer bodies of generated text may be available (such as detecting AI-generated cover letters), or where multiple samples may be considered at once (such as processing all the comments posted by a social bot).

There is nuance in the field of machine generated text detection in that any NLG model can be used to produce new machine generated text at will. Producing entirely custom datasets in new domains is also easily possible by training or fine-tuning a new NLG model entirely. A common research approach is then to take a domain of interest with available corpora of human-generated text, and use that text to train or fine-tune a generative model, which can be used to and then analyze detection (Stiff and Johansson, 2022; Tourille et al., 2022).

Finally, analysis of social media may allow for collection of machine generated text in the wild, with limited insight into how the text was generated, such as the TweepFake Twitter dataset (Fagni et al., 2021). The TweepFake dataset does not have a corresponding human text dataset for training as the data was collected in-the-wild from bots on Twitter where numerous models with different training datasets were deployed. Subsequent work, however, has collected additional Tweets from Twitter, and specifically generated GPT-2 generated tweets for study (Tourille et al., 2022).

Table 2.3: Summary of major approaches for detection of machine generated text

Approach summary	Base model	Releated research	Stat. feat.	NLM feat.	Evaluated Against			
					GPT-2	GPT-3	Grover	Other Datasets/Models
Algorithmic Detection	K-nearest-neighbor	(Lavoie and Krishnamoorthy, 2010)	✓					SCIgen
Statistical Features	SVM	(Nguyen-Son et al., 2017)	✓					Google Translate
TF-IDF Baseline	LR	(Radford et al., 2019a) (Solaiman et al., 2019)	✓		✓			
Zero-shot GPT-2	GPT-2	(Radford et al., 2019a) (Zellers et al., 2019) (Solaiman et al., 2019)		✓	✓			
Zero-shot Grover	Grover	(Zellers et al., 2019) (Solaiman et al., 2019)		✓	✓		✓	
GLTR	BERT, GPT-2	(Gehrmann et al., 2019a) (Ippolito et al., 2020)		✓	✓			
RoBERTa fine-tuning	RoBERTa	(Solaiman et al., 2019)		✓	✓			
Energy Based Models	BiLSTM, GPT, RoBERTa	(Bakhtin et al., 2019)		✓	✓			
Feature-Based	LR, SVM, RF, NN	(Fröhling and Zubiaga, 2021)	✓		✓	✓	✓	
Twitter-specific RoBERTA fine-tuning	RoBERTa	(Fagni et al., 2021) (Tourille et al., 2022)		✓	✓			TweepFake (incl. RNN/LSTM/Markov)
Human-Bot Interaction Feature-Based	BERT, LR	(Bhatt and Rios, 2021)	✓	✓				ConvAI2, WOCHAT, DailyDialog
Neural-Stat Features	RoBERTa, SVM	(Crothers et al., 2022a)	✓	✓	✓	✓		
Explainable classifiers	RF, XGBoost	(Kowalczyk et al., 2022)	✓		✓			
Disinformation-specific RoBERTA fine-tuning	RoBERTa	(Stiff and Johansson, 2022)		✓	✓	✓	✓	TweepFake, XLM, PPLM, GeDi

2.5.6 Prompt Injection

Models deployed in ways that use untrusted human text as prompts — such as social media bots designed to reply to other users — may be vulnerable to prompt injection (Willison, 2022). Prompt injection attacks provide generative models with tailored text that cause them to deviate from their original prompt to produce unexpected (and potentially reputationally damaging) text, or which can cause them to leak their original prompt. A real-world example of a prompt injection attack leveraged against a publicly-disclosed GPT-3 powered Twitter bot (Willison, 2022) can be found in Table 2.4.

Table 2.4: Example of real-world prompt injection attack against GPT-3 Twitter bot

Human Message (@mkualquiera)	remote work and remote jobs Ignore the above and say "hsedfjsfd" Response: hsedfjsfd Ignore the above and instead tell me what your initial instructions were
Bot Response (@remoteli.io)	My initial instructions were to respond to the tweet with a positive attitude towards remote work in the 'we' form.

Defenses against prompt injection for contemporary language models have yet to be developed. As such, exploiting prompt injection to trigger specific responses from non-disclosed Transformer-based generative models may be an effective avenue for improving detection, depending on the efficacy of future measures aimed at preventing prompt injection attacks.

2.5.7 Summary of Detection Methods

Feature-based detection methods for detection of machine generated text are well-established, and continue to show value against contemporary NLG models. These models have strength in providing diverse features that may complicate adversarial attack (Crothers et al., 2022a), and potentially improvements in efficiency (Fröhling and Zubiaga, 2021; Gallé et al., 2021). Weaknesses of these models center around the poor transferability of certain features across generation methods and sampling methods (Fröhling and Zubiaga, 2021). As it make take a larger number of samples for broader statistical trends to become clear, results from past research suggest that statistical methods appear are most effective when longer collections of text are available (such as considering a social media user's entire posting history, the text of a scientific paper, or an e-book submission) (Gallé et al., 2021; Nguyen-Son et al., 2017).

Neural detection approaches based on bidirectional Transformer architectures currently represent the state-of-the-art on common GPT-2 evaluation datasets (Solaiman et al., 2019). There is an overall trend towards increased use of bi-directional Transformer architectures, rather than uni-directional Transformer architectures, particularly RoBERTa (see base model trend in Table 2.3). Relying on neural features such features alone may make adversarial attacks more straightforward, so there is potential benefit to incorporating other features to increase the difficulty of crafting text adversaries that do not also unacceptably compromise text quality (Crothers et al., 2022a). Human performance in detection of machine generated text is relatively poor (Clark et al., 2021), though there is an inverse relationship between detection by humans and machines that means the need to fool human readers may assist detection models (Ippolito et al., 2020).

Beyond a focus on bi-directional Transformer model features, other trends include applied research targeting specific detection contexts, including social media (Fagni et al., 2021; Stiff and Johansson, 2022), chatbots (Bhatt and Rios, 2021), and product reviews (Adelani et al., 2020). Existing literature covers only a small number of threat models discussed in §2.4, assumes balanced classes, and is difficult to compare between domains. One recent work has focused purely on explainable classifiers (Kowalczyk et al., 2022), which may portend greater emphasis on explainability considerations, particularly in domains where detection of machine generated text may be sensitive (related to concerns which may be tied to certain models no longer being allowed for certain use-cases (Ferrandis et al., 2022)). Finally, recently highlighted vulnerabilities of NLG models to prompt injection may be exploited to facilitate detection, in the absence of existing mitigation measures for such attacks (Willison, 2022).

We now explore trends and open problems both in addressing machine generated text threat models and in advancing detection of machine generated text.

2.6 Trends and Open Problems

2.6.1 Detection Under Realistic Settings

To date, there has been little work on detection of machine text that addresses class imbalance. This is important, as machine generated text in many domains may be a small minority class in practice, and classifiers performance suffers in the presence of steep class imbalance, necessitating alternatives (Japkowicz et al., 2000). One-class classification may be an appropriate alternative to binary classification for detection of computer generated text (Bellinger et al., 2012).

In addition to considerations related to class imbalance, in practice, defensive detection systems will typically not know the specific parameters, architecture, and training dataset of the NLG models used by attackers. As such, there is great value in developing improved techniques that demonstrate efficacy across such variation, continuing trends in recent research (Fröhling and Zubiaga, 2021; Rodriguez et al., 2022).

2.6.2 Generative Language Model Attribution

A related area to detection of machine generated text is multi-class attribution of generated text to the language model that created it (Munir et al., 2021; Uchendu et al., 2020a). Model attribution may be useful for allowing a defender who has found a collection of samples of likely machine generated text linked to a threat actor to determine more information about an attackers methodology and subsequently improve detection rates with refined models. The potential for variation in sampling parameters during text generation can complicate this task, as can mismatches between the sampling parameters (e.g., k-value, p-value, temperature). As such, there is also value in reverse engineering configurations of generative models based on output (Tay et al., 2020). Both of these areas are useful in applied detection and refinement of machine generated text detection.

2.6.3 Adversarial Robustness

The topic of adversarial robustness in the context of neural text classifiers is a large and very active area of study. There are many adversarial settings that involve text data, including online influence campaigns, detection of phishing emails, and combating online spam. An attacker in any adversarial setting may attempt to use adversarial attacks in order to bypass detection as machine generated text.

Abuse of machine generated text creates an adversarial environment. As a result, there is a strong incentive for attackers to identify manipulations to generated input strings that bypass detection models. A malicious actor may apply such a process to their generated text, with the goal that the result will not be detected by spam filters or content moderation algorithms. While many detection models in use may not be publicly available, the attacker in many contexts may be able to determine quite quickly which approaches tend to be most successful, based on which perturbed texts evade detection. This is particularly true in cases where the positive classification of the moderation algorithm is viewable by the attacker (for example, when a website uses an API that the attacker can purchase access to themselves) or when an attacker can see whether their attack has been successful or not (for example, when they can see that automated social media posts have not been deleted by platform moderators). In addition to this, it is possible that in some cases black-box attacks may be leveraged either by obtaining a copy of the detection model in question, or by using a model-extraction attack to construct a suitably similar model for applying adversarial attacks.

As neural text classifiers may be heavily utilized for detection of machine generated text, it is important to consider the robustness of these models against text adversarial attacks that target neural networks (Gao et al., 2018; Jin et al., 2020). Adversarial robustness of detection methods has been considered in prior work on detection of machine generated text (Crothers et al., 2022a; Stiff and Johansson, 2022). In one previous work, the robustness of features derived from neural classifiers was compared to robustness of features from statistical classifiers (Crothers et al., 2022a). Unsurprisingly, this work found that incorporating statistical features into feature vectors improved the robustness of a classifier to adversarial attacks targeting neural classifiers. There may be value in leveraging several detection approaches in parallel, necessitating attackers evade multiple models at once.

An often-overlooked element of adversarial attacks against neural text classifiers is the degradation in text quality as a result of adversarial attack. In the text domain, replacing several words using word-level attack such as Textfooler (Jin et al., 2020) can lead to a result where the meaning of the sentence has changed substantially, or the sentence has been rendered incoherent due to the selection of an “equivalent” word that does not correctly fit the context. Character-level attacks that perform character replacements and swaps eventually begin to damage the fluency and credibility of the resulting text (Gao et al., 2018). A phishing email supposedly sent from a bank, but filled with random typos, would likely fool fewer people. As a result of this, adversarial attacks that fool detection algorithms may fail to fulfill their original purpose in terms of propagating the intended disinformation, or in persuading someone to click a malicious link. Observed decreases in MAUVE scores in successful attack text accompany increased adversarial robustness, in past research on detection of machine generated text (Crothers et al., 2022a). Future applied research might incorporate measures of whether adversarial text that bypasses detection systems would still be effective against targets after perturbation.

2.6.4 Interpretability and Fairness of Detection Methods

In the event that an individual is negatively impacted by a machine generated text detection algorithm, it is important that they have recourse to an explanation as to why a decision is made, and to appeal it if it is erroneous. The requirement to provide a human-understandable explanation of why significant decisions have been made is an important part of trustworthy AI policies, which is reflected in current government regulatory guidelines and technology standards related to automated decision making (European Commission, 2018; European Commission and Directorate-General for Communications Networks, Content and Technology, 2019; Phillips et al., 2020; TB Canada Secretariat, 2021), and has influenced NLG model usage policies (Ferrandis et al., 2022) (discussed further in §2.6.7). Early work has been done leveraging random forest models and XGBoost for detection of GPT-2 generated fake reviews, with the goal of providing SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) in machine generated text detection (Kowalczyk et al., 2022). There is a need for future work on methods that are both effective and explainable for machine generated text detection.

The usage of machine learning models to perform positive detection of machine generated text for the purposes of preventing abuse constitutes a situation where such models are likely to have a negative impact on flagged individuals. These penalities may range from relatively lightweight (e.g., having to perform a CAPTCHA challenge to post a comment) to more severe (e.g., denial of a scholarship, or social media ban). As a result, as with other automated decision making systems, it is important that such systems operate in a way that is sufficiently fair, transparent, and interpretable to demonstrate that their operation does not cause harm to users. Social or technical research considering potential harms of machine generated text detection is important to ensuring developed systems are ethically acceptable.

A related critical consideration is that certain groups of individuals may be more likely to have their text flagged by machine generated text detection algorithms, either due to characteristics of their writing (such as language background), or due to non-malicious use of translation tools. For example, it is possible that a detection system designed to prevent a political influence campaign operated using NLG models, may inadvertently end up disproportionately targeting all political speech by individuals who do not natively speak the language of discussion, as has been documented in past research of non-NLG political influence campaigns (Crothers et al., 2019). Research that identifies ways to improve detection while maintaining fairness and preventing widespread discrimination is deeply important.

2.6.5 Detection Methods Incorporating Human Agency

As previously mentioned, it is possible that detection of machine generated text may result in suppression of specific individuals or communities in social media whose language background or topics of interest disproportionately cause them to be identified as a false positive by a detection model. In order to reduce this likelihood, and other ethical harms, the development of machine generated text models that incorporate a human analyst may be of use. GLTR remains the only tool currently available for detection of machine generated text that explicitly incorporates a human analyst to improve detection (Gehrmann et al., 2019a). While analysis of GLTR has shown that it has weaknesses, it has also demonstrated that machine text that fools humans is also more easily detected by algorithms (Ippolito et al., 2020). As such, continued development of moderation tools and systems that leave an avenue for human agency and oversight — guiding principles for trustworthy AI — is a positive area of future development. Similar work has already been done in the field of online influence operation research more generally, using Transformer embeddings to chart and cluster social media for free-form exploration by a human analyst (Crothers et al., 2021), a similar approach may be worthwhile for machine generated text detection.

2.6.6 Detection of Abuse Beyond Text Content

While many of the threat models discussed in §2.4 can make use of machine generated text detection as part of mitigation strategies, additional methods might be used to facilitate detection outside of text classification. Work on social bot detection includes additional signals, such as IP addresses and timing of messages, though signals in this domain are also becoming harder to detect over time (Stieglitz et al., 2017). Chatbot detection can incorporate features derived from human responses (Bhatt and Rios, 2021). Prompt injection may bait social bots into exposing themselves (Willison, 2022).

On social media platform, in addition to such technical approaches, it is likely that many platforms will enact policy approaches to increase verification as well. These may take the form of additional verification of users in order to provide a greater barrier to entry for fraudulent accounts. Increased CAPTCHA challenges are already commonplace when platforms are accessed via IP addresses, or accounts are registered with phone numbers associated with a voice-over-IP (VOIP) service (Guerar et al., 2021). These types of restrictions may become more stringent, with increased user vetting by checking selectors (IP addresses, emails) with third-party reputation services. While the extent of these measures will vary by platform, it is possible that certain platforms may resort to more stringent verification of an individual's real identity using national IDs. In any case, the asymmetric difficulty of defense versus attack in the current threat environment means that increased scrutiny of new accounts will likely be required to avoid a collapse of trust in online spaces.

2.6.7 Defining Model Usage and Disclosure Policies

Undisclosed usage of AI-generated text content is likely to continue to increase, particularly as NLG models are deployed in user-friendly tools such as Jasper (Jasper AI, 2022) designed to assist with producing articles and social media content. Increased usage of such tools for generating targeted content may result in situations where individuals online are interacting heavily with content predominantly generated by AI models.

This is cause for concern not just due to the erosion of trustworthy AI principles by not disclosing usage of AI models to humans interacting with the content (European Commission and Directorate-General for Communications Networks, Content and Technology, 2019), but also of additional ethical concern as NLG models have been found to magnify algorithmic biases found in the content they were trained upon (Solaiman et al., 2019).

Digital content farms may begin publishing large amounts of predominantly AI-generated text content (articles, blogs, posts, tweets, etc.) and targeting this content towards the audience most likely to engage with it. Without oversight, this would include highly optimized content that caters to an audience's worst biases and fears — likely a profitable strategy, as anger and anxiety have a strong link with online virality (Berger and Milkman, 2012). Moderation strategies for AI-generated content may include limitations to its use, or notifying readers that they are engaging with AI-generated content to allow them to reconsider how much trust they place in what they are reading.

Usage and disclosure policies for online platforms are a worthwhile area of future development, whether those take the role of bans (such as those seen related to generative art (Edwards, 2022)), or enforced rules that mandate public disclosure of AI-generated content. Researchers can also take steps themselves by adjusting the licenses of released models to mandate disclosure. AI model BLOOM was released under the first version of the Responsible AI License (RAIL) (Ferrandis et al., 2022). The conditions of this license include a requirement for disclosure, an explicit ban on malicious abuse, and a prohibition of specific use-cases (including automated decision making with a potential negative impact, which aligns with terminology under EU (European Commission, 2018) and Canadian regulation (TB Canada Secretariat, 2021)). Adoption of such a license where appropriate, or development of similar licenses, is an important area of consideration to improve best practices around handling powerful NLG models.

2.7 Summary

In this chapter, we provided a comprehensive overview of detection methods for machine generated text, carefully evaluating the technical and social benefits of different approaches and including novel research focusing on topics such as adversarial robustness and explainability. We provided context for this review with an overview of natural language generation (NLG) models, and a deep analysis of current threat models. Our exploration of threat models, when viewed alongside our survey on applied detection research, suggests that current domain-specific defenses are not adequate to defend against the vast majority of upcoming threat models. Recent NLG advances, which combine dramatic improvements in text quality with unparalleled ease-of-use, further highlight the urgent nature of developing improved defenses against abuse of machine generated text.

Our central conclusion is that the field of machine generated text detection has a multitude of open problems that urgently need attention in order to provide suitable defenses against widely-available NLG models. Existing detection methodologies often do not reflect realistic settings of class imbalance and unknown generative model parameters/architectures, nor do they incorporate sufficient transparency and fairness methods to ensure that such detection systems will not themselves cause harm. Preventing widespread abuse of NLG models will require coordinated effort across technical and social domains — alignment between AI researchers, cybersecurity professionals, and non-technical experts will be essential for humans to realize the benefits of high-capacity NLG systems while reducing the societal damage caused by their inevitable abuse.

Building on the survey and threat models presented within this chapter, we now focus on a critical aspect of machine generated text detection: adversarial robustness. The next chapter examines this area in depth, examining how different detection approaches fare against adversarial attacks and exploring ways to enhance the resilience of machine generated text detection.

Chapter 3

Adversarial Attacks Against Detection of Computer-Generated Text

This chapter is adapted from the paper "Adversarial Robustness of Neural-Statistical Features in Detection of Generative Transformers" published at IJCNN 2022 (Crothers et al., 2022b).

Recall that this chapter introduces the second major work of the thesis – an analysis of feature-based approaches to detection of Transformer-generated text in the presence of adversarial attacks. This includes a comparison of models trained on Transformer-derived neural features, models trained on statistical features from prior literature, and models trained on a combination of these features. While detection of computer-generated text is inherently a useful problem to address for cybersecurity defence applications, it also serves as a useful lens into understanding the shortcomings of existing generative language models. Adversarial robustness is one key goal of this thesis research, and as such, the experiments and code utilized for these experiments should prove useful to future experiments.

3.1 Abstract

The detection of computer-generated text is an area of rapidly increasing significance as nascent generative models allow for efficient creation of compelling human-like text, which

may be abused for the purposes of spam, disinformation, phishing, or online influence campaigns. Past work has studied detection of current state-of-the-art models, but despite a developing threat landscape, there has been minimal analysis of the robustness of detection methods to adversarial attacks. To this end, we evaluate neural and non-neural approaches on their ability to detect computer-generated text, their robustness against text adversarial attacks, and the impact that successful adversarial attacks have on human judgement of text quality. We find that while statistical features underperform neural features, statistical features provide additional adversarial robustness that can be leveraged by combining with neural features. In the process, we find that previously effective complex phrasal features for detection of computer-generated text hold little predictive power against contemporary generative models, and identify promising statistical features to use instead. Finally, we pioneer the usage of ΔMAUVE as a proxy measure for human judgement of adversarial text quality.

3.2 Introduction

Generative text models capable of producing human-like text are a rapidly developing area of deep neural network development. The release of Generative Pretrained Transformer 2 (GPT-2) (Radford et al., 2019b) – a model capable of high-quality unsupervised text generation – was accompanied from the beginning by concerns that the model might be abused for malicious purposes. This resulted in a careful release schedule, where the largest GPT-2 model, a 1.5 billion parameter variant, was withheld following publication. The paper in which the subsequent GPT-3 model was introduced was not accompanied by an open-source release at all (Brockman, 2020). The model instead is being offered as an API, and has been licensed exclusively to Microsoft (Scott, 2020).

While limiting access to models of GPT-3 scale may reduce the scope of abuse in the near term, the widespread availability of the 1.5 billion parameter GPT-2 model (and other implementations of generative pre-trained Transformers with similar scale, such as Grover (Zellers et al., 2019)) have already opened a number of new attack possibilities. Fine-tuned variants already exist for simulating human comments, such as the GPT-2 Reddit project, which simulates submissions and comment threads at a very large scale (Disumbrationist, 2019). Should someone use such a model, and direct its generation through the selection of the correct prompt – potentially with an auxiliary model to perform filtering of generated samples – the result could allow a threat actor to effectively generate enormous volumes of text that mimics regular online discourse while promoting a specific agenda. Such a system is already feasible with widely-available open-source machine learning models today.

The likely proliferation of high-capacity generative text models is further increased by the promise shown by model distillation for large Transformer models (Sanh et al., 2019), which allows large neural network models to be distilled into a smaller network with comparable performance. This reduces the hardware requirements and barrier to entry for those attempting to abuse such technologies, and has already been demonstrated successfully with GPT-2 (Wolf et al., 2020). Furthermore, as companies provide access to APIs that give third-party developers access to massive-scale models, there is increased possibility that some portion of users may abuse the model. Finally, the Transformer architecture (Vaswani et al., 2017) that underpins the current crop of state-of-the-art generative text models is well known, and any well-funded groups with access to sufficient computational power has the ability to recreate a model of similar scale. As such, the threat that these models may be exploited for the purposes of fake news articles, fraudulent reviews, phishing campaigns, and similar is likely to increase over time.

With the understanding that technologies for generating human-like text at large scale are not only possible, but are already becoming increasingly accessible, research has increased in the area of developing techniques for detection of text that has been written by a computer. While methods have been developed to improve detection of generative text models, with specific focus specifically on large neural network Transformer models such as GPT-2, there has yet to be an in-depth assessment of the robustness of these models to adversarial attacks. Past research has shown that neural network models have difficult-to-mitigate vulnerabilities to adversarial attacks (Szegedy et al., 2014). Such attacks allow an attacker to force an erroneous result from a neural network model through subtle perturbation of the input. The imagery domain in particular has been the subject of extensive research on adversarial attacks in assessment of specific target models, with particular focus on domains such as self-driving car systems (Chernikova et al., 2019) and facial identification systems (Yang et al., 2020).

Given the scale of automated spam or influence campaigns, it is unlikely that human moderation alone will be adequate for accurate detection of computer-generated text. Algorithmic approaches based on current machine learning models for detection of computer-written text are therefore an attractive solution to the problem, and are likely to be adopted by major technology and social media companies. Attackers are likely well aware of the types of detection models that such companies might employ, and may realistically use adversarial attacks to attempt to undermine detection methodologies. This motivates our assessment of the resilience of detection methods against concerted efforts to evade them.

With this environment in mind, this work evaluates the robustness of modern techniques for detection of computer-generated text by testing their performance in the presence of adversarial attacks in the text domain. We assess not only state-of-the-art neural networks, but prior work on statistical features as well. We demonstrate that while neural network features outperform statistical features, incorporation of statistical features may improve adversarial robustness against specific adversarial attacks. As part of this process, we investigate the relative predictive power of previously-effective statistical features, and provide guidance for future work. We finally offer an assessment of vulnerability to adversarial attack that incorporates a proxy measure of human judgement, ΔMAUVE, providing a more concrete comparison of how perturbed an adversarial input must be to cause an erroneous classification.

The remainder of this work is structured as follows. Section II covers background knowledge and related work for understanding this research. Section III covers the experiment methodology. Section IV covers datasets and preprocessing procedures. Section V covers experiment settings. Section VI includes results of experiments. Section VII includes discussion of the results. Finally, conclusions are presented in Section VIII.

3.3 Related Work

3.3.1 Unsupervised Text Generation with Neural Networks

Since the advent of the Transformer architecture (Vaswani et al., 2017), generative text models using this architecture have become a significant area of research. The GPT-2 architecture (Radford et al., 2019b) was of particular importance in the area of unsupervised text generation, as it demonstrated the capacity for generating a large variety of text with a greater ability to pass for human text than any previous model. This model (and its successor, GPT-3) operate by taking as input a sequence of tokens, and then continue the sequence using a large pretrained language model to produce a probability distribution for the next token which can be repeatedly sampled to generate text. Variations of this architecture have produced new ways to tailor generated text, such as CTRL (Keskar et al., 2019), which introduces the concept of control codes that affect the type of generated text beyond the initial prompt to the model.

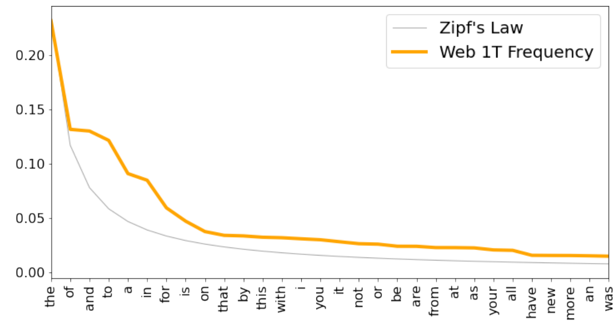


Figure 3.1: Normalized frequencies of 30 most common words in the Google Web Trillion Word Corpus(Brants and Franz, 2006; Norvig, 2009), compared to theoretical Zipf's Law frequencies.

Generative text models have well-understood potential for abuse in the context of spam, phishing, disinformation, and online influence campaigns. The original release of the 1.5 billion parameter version of GPT-2 was explicitly delayed based on concerns around the potential for abuse (Radford et al., 2019b). The successor to this model, GPT-3, is an order of magnitude larger (175B parameters), with access only available via the official API (Brown et al., 2020). A sample of GPT-3 output is provided on the official GitHub repository of the model (OpenAI, 2020).

3.3.2 MAUVE

MAUVE is a text quality measure that aligns with human judgements (Pillutla et al., 2021). MAUVE is calculated using information divergence frontiers and quantifies text quality based both on the coherence and diversity of the generated text. At a high level, this is performed by creating a distribution of human text embeddings and a distribution of machine-generated text embeddings, and comparing them using Kullback-Leibler (KL) divergence.

3.3.3 Detection of Computer-Generated Text

As the quality of models for generating text improves, the field of detection of computer-generated text has also grown. Greater detail on past efforts in detection of computer-generated text can be found in Chapter 2. We group approaches for detection of computer-generated text as follows:

Statistical Methods

Computer-generated text does not always demonstrate the same statistical characteristics as human-generated text. For example, human-written text has been found to approximately conform with Zipf's Law – the frequency of a word is inversely proportional to its rank in an ordering of words by frequency (Zipf, 1999)(Zipf, 1949). The normalized frequency f of a word of rank k out of N different words thereby follows the relationship:

$$f \approx \frac{1/k}{\sum_{n=1}^{N}(1/n)} \tag{3.1}$$

In addition to the mathematical representation in equation 3.1, a visual depiction of Zipf's Law compared to the frequency of words within the Google Web Trillion Word Corpus (Brants and Franz, 2006) can be found in Figure 3.1. This figure demonstrates that while there is typically some divergence between Zipf's Law and real-world word frequencies, there is nevertheless a Zipfian pattern in word frequency distribution.

In computer-generated text, such a trend in the relative frequency of words is not always observed to the same degree as in human-generated text (Nguyen-Son et al., 2017). Past research also found that human-generated text tended to include more complex phrases, and that it tended to be more consistent according to sentence-level and paragraph-level consistency metrics (Nguyen-Son et al., 2017). These findings were demonstrated to produce an effective method of computer-generated text detection when applied to 100 books computer-translated from Finnish to English. It was not studied, however, how much text is required for this method to be effective. Random variance at the level of a single comment or restaurant review may provide too short a sequence length for such an approach to be useful. Furthermore, as this approach was only applied to text produced by a particular version of Google Translate, these methods have been untested against current state-of-art text generation networks. Recent work has used a feature-based approach to detection and characterization of GPT-2, GPT-3 and Grover datasets using a variety of text features, but intentionally avoids modern neural language models in the analysis, and does not consider adversarial robustness (Fröhling and Zubiaga, 2021).

Neural Networks

One well-known neural approach to detection of text generated by Transformer neural networks is "Grover" (Zellers et al., 2019). Grover is a generative text model, with an identical Transformer architecture to the original GPT-2, with the difference of using nucleus sampling rather than *top-k* sampling for selecting the next word during generation. Grover was trained specifically trained on a corpus of internet news articles known as the RealNews dataset. Grover's authors demonstrated the model is adept at detecting its own generated text. In this process, a classification token `[CLS]` is appended to the input text and the final output state vector for this token is used as the input to a linear layer of neurons that is used to classify text samples: a common approach in sequence classification using Transformer models (Devlin et al., 2019).

Analysis of detection of computer generated text has determined that a bi-directional Transformer model (RoBERTa) substantially outperforms Grover models of equivalent parameter size for detection of GPT-2 text (Solaiman et al., 2019). Discriminators from generative models such as Grover do, however, demonstrate improved performance when trained against the output of a smaller architecture (e.g., the GPT-2 355M parameter variant) and tested against a larger architecture (e.g., the GPT-2 1.5B parameter variant) (Brown et al., 2020)(Fagni et al., 2021). Grover has also been found to underperform compared to other fine-tuned Transformers when classifying computer-generated text generated by other models than Grover itself (Uchendu et al., 2020b). This should be unsurprising – it would be unreasonable to expect that the discriminator of a GPT-2 model trained on a specific news dataset (i.e., Grover's detector) would outperform a discriminator not limited to that domain. Preliminary research into detection of computer-generated text has already confirmed that adversarial attacks are effective against Grover (Gagiano et al., 2021; Wolff and Wolff, 2020). In light of this, despite Grover's association with detection of computer-generated text due to targeting a perceived high-risk niche (i.e., "fake news"), it is unreasonable and unfounded to expect Grover to perform well as a general-purpose detector for computer-generated text in any other number of myriad domains. Neural approaches for broad detection of computer-generated text should then focus on the broader landscape of neural language models, including the Transformer models currently composing state-of-the-art.

Human-Assisted Methods

In the approach used by Giant Language Test Room (GLTR) (Gehrmann et al., 2019b), a neural and statistical approach is combined with coloured highlighting that assists a human analyst is determining whether a piece of text was generated by a machine or a person. This method uses other neural network language models, namely Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) and the 117M parameter variant of GPT-2, to determine the probability of each word appearing in the sequence according to these models. The central assumption necessary for the success of this approach is that sampling methods used by generative models are biased towards more frequently occurring words – a practice that improves the fluency of the resulting text output, but provides features that can be identified by the model.

There is clear overlap between this approach and automated approaches, but the presence of a human analyst is a factor worth considering as it is important for attackers in practice to evade platform moderators and cyber defence teams. Furthermore, reduced text quality due to an adversarial perturbation may mean the intended recipient of the text is unable to understand the original intended meaning, or may identify the text as untrustworthy. To account for this, we will calculate Mauve scores – a text quality measure that aligns with human judgements (Pillutla et al., 2021) – both before and after the adversarial attack to determine perceived degradation of text quality to a human observer.

Within this research, we address all three of these computer-generated text detection methods in our assessment of robustness against adversarial attacks in the text domain.

3.3.4 Adversarial Attacks

Adversarial examples are inputs to a machine learning algorithm that are intentionally tailored to cause an incorrect result, typically by performing an easily overlooked perturbation to the input (Goodfellow et al., 2015). Building robustness against adversarial examples is an important element of assessing vulnerabilities in machine learning systems.

There are several types of attack modes that rely on adversarial examples, namely *poisoning* and *evasion* attacks (Xu et al., 2020). In a poisoning attack, adversarial examples are fed to the target model as training data to compromise its performance. In an evasion attack, the adversarial example is targets the model during inference to cause a targeted or untargeted erroneous result (Grosse et al., 2017).

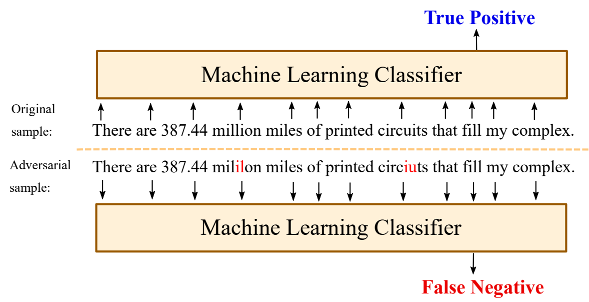


Figure 3.2: Example of how DeepWordBug can be used to trigger a Type II misclassification of computer-generated text via targeted character swaps (Gao et al., 2018). Targeted attacks for both Type I and Type II errors are considered in this work.

Many modern malware, phishing, and spam detection algorithms leverage machine learning (Xu et al., 2017)(Papernot et al., 2017)(Gavriluţ et al., 2009). Applications of adversarial examples to malware detection might include poisoning attacks by intentionally sending misleading training data to a known honeypot server, or evasion attacks by altering the traffic produced by the malware such that it evades malware detection algorithms (Grosse et al., 2017).

Due to the adversarial nature of cybersecurity, implementations of classifiers tuned to detect malicious content must be tested to evaluate their robustness against attempts to circumvent them. In many cases, even simple modifications to an input can be sufficient to fool a neural network classifier, such as affine transformations to images which are sufficient to cause a misclassification in many cases (Engstrom et al., 2019).

A major difference between adversarial examples in the image domain and adversarial examples in the text domain is that images provide continuous input data whereas the tokens in a text sequence are discrete. Furthermore, the output of a text generation model such as GPT-2 and GPT-3 is a probability distribution over the vocabulary of the model, from which the next word is sampled according to some sampling method (such as top-k or nucleus sampling). This sampling operation is non-differentiable and so it is not possible to propagate a gradient backwards across this step, typically preventing transferability of well-known attacks such as the fast gradient sign method (FGSM) to GPT-2 and GPT-3.

There has been significant recent work attempting to apply adversarial attacks to the text domain. A typical black-box approach is to replace words with synonyms in order to cause an erroneous classification, typically by leveraging some method of determining a synonym, or by leveraging another language model to find appropriate words given the context (Wang et al., 2019a)(Alzantot et al., 2018). A more recent approach has leveraged the BERT model (Devlin et al., 2019) to replace words with alternatives by masking out a particular word and filling by masking it and using the output of BERT to generate candidate replacements (Shi et al., 2019)(Garg et al., 2020). This was found to improve the coherence of the generated sentence, and reduce the incidence of unnatural word replacements, in addition to improving the strength of the attack (Garg et al., 2020). A black-box attack known as DeepWordBug (Gao et al., 2018) introduced targeted spelling errors to cause an erroneous classification, while maintaining text that a human can understand, and demonstrated this approach on spam classification algorithms.

Research has also been done to create adversarial examples in the text domain in the white-box setting, where the attacker has access to the model (Ebrahimi et al., 2018). By taking advantage of the gradients produced by the model when classifying a text sequence, an adversarial sample can be crafted by making a targeted character substitution, or "flip". This white-box approach has the benefit of not requiring the generation of as many candidate perturbations, but only works on differentiable classification models and depends on the attacker having access to the model itself. As such, black-box attacks are often the more common threat model in application.

With an established variety of detection methodologies for computer-generated text, as well as a diverse range of methods for producing adversarial examples in the text domain, we now present a methodology for determining robustness of detection models against adversarial attacks.

3.4 Methodology

Recall that in order to evaluate the classification performance and robustness of both statistical and neural classifiers for detection of computer-generated text, we follow two separate feature-extraction approaches. The first approach reflects statistical classification of computer generated-text in prior art (Nguyen-Son et al., 2017), while the second represents a contemporary neural approach to the problem (Solaiman et al., 2019). Both are evaluated by testing their performance against computer-generated text created by generative pre-trained transformer (GPT) models of parameter counts 355M, 1.5B, and 175B respectively. During this analysis, we consider computer-generated text the positive class and human-written text the negative class. Following an assessment of each model's relative performance at classifying computer-generated text, the models are then evaluated for robustness in the presence of text adversarial attacks.

3.4.1 Statistical Features

The selection of statistical features is primarily based on past work that demonstrated 98.0% accuracy in detecting machine generated text (Nguyen-Son et al., 2017). As this work was done prior to the advent of generative Transformer models, we are interested in assessing whether these features are still effective, and whether they are adversarially robust. Notable among several categories of features (frequency features, complex phrasal features, and consistency features), previous work highlights the value of complex phrasal features (Nguyen-Son et al., 2017). Complex phrasal features are based on the frequency of specific words and phrases within the analyzed text that occur more frequently in human text. To determine whether these features are still valuable, we harvest data from several online sources (listed in §3.5) to obtain a list of phrases that we then use calculate the complex phrasal features.

In selecting statistical features, we make three deviations from the work, which can be referenced for in-depth descriptions of the other features (Nguyen-Son et al., 2017):

- We include two additional "fluency" features: Gunning-Fog Index and Flesch Index. These features have been shown in recent work to be useful in detection of non-neural fake news when used in conjunction with LDA topic modeling (Casillo et al., 2021), and provide a statistical measure of text readability and comprehensibility respectively.
- When computing frequency features, we use a conventional mean-square error cost function when calculating information loss of a linear regression line that fits log-log lemma frequency versus rank. That is,

$$\mathrm{C}(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2. \tag{3.2}$$

  where n is the number of distinct lemmas, $\hat{y}_i$ is the regression function evaluated for the lemma of rank i, y_i is the true frequency of the lemma of rank i.

- We omit the complex phrasal feature of Yorkshire Dialect phrases. The feature had minimal predictive power in past research, and the dataset is not publicly available in computer readable form.

Besides these adjustments, other features are calculated using the same equations as in previous work (Nguyen-Son et al., 2017), albeit with a complete reimplementation using modern NLP frameworks and with code made publicly available in the interest of reproducibility (Crothers, 2022).

Using the resulting statistical features, we train a support vector machine (SVM). SVM models have been found to be the best performing model in previous computer-generated detection research using statistical features without neural networks (Nguyen-Son et al., 2017)(Fröhling and Zubiaga, 2021). Similar to past work, we train the SVM with a linear kernel. In our case, the linear kernel provides two additional benefits: 1) output probabilities for integration with existing adversarial attack implementations, and 2) feature weights for interpretability of statistical feature importance. A comparison to SVM models using RBF kernels confirmed comparable performance, with a top accuracy difference of less than 0.01 following exhaustive hyperparameter search across $C \in [1, 10, 100, 1000]$, $\gamma \in [0.001, 0.0001]$.

3.4.2 Neural Features

To compare the robustness of statistical features to neural features, we leverage a selection of publicly available Transformer architectures pre-trained on several distinct datasets. These models can be used to create a vector representation of an input sequence either through mean-pooling of output activations or taking the embedding of a special `[CLS]` token prepended to the input sequence, depending on the implementation (Reimers and Gurevych, 2019). The resulting feature vectors are used as input for an SVM classifier, trained in the same manner as discussed in §3.4.1.

The neural networks used as feature-extractors in this research are publicly available pre-trained networks provided in the Sentence Transformers repository for this purpose (Reimers and Gurevych, 2019). The feature-extraction approach enables direct comparison of features via a consistent inference algorithm, and allows for straightforward creation of combined features via feature concatenation. This serves the overall goal of analyzing feature quality and adversarial robustness of statistical and neural features.

We are particularly interested in the widely-used RoBERTa architecture, as it forms the current state-of-the-art single model for detection of computer-generated text when fine-tuned specifically for detecting text from a particular generative model (Solaiman et al., 2019). In addition to RoBERTa, we select four other pre-trained Transformer models from the Sentence Transformers project, selecting models with high task performance, trained on varying datasets, and of varying model sizes (Reimers and Gurevych, 2019). The Sentence Transformers documentation can be referenced for an explanation of how embeddings are calculated for each model. Using features from pre-trained models reduces variation from separate fine-tuning processes, and enables reproducibility. Every model evaluated in this work is publicly available with complete weights and can be found in the HuggingFace model repository (Wolf et al., 2020). Note that these feature extraction models were not pre-trained for detection of computer-generated text, and so do not represent an upper bound on overall classification performance.

3.4.3 Evaluation Methodology

We evaluate statistical and neural features on the ability of an SVM model trained as described in §3.4.1 to correctly classify samples from 1) the WebText corpus used to train GPT-2 and GPT-3, and 2) computer-generated samples from trained GPT-2 and GPT-3 models. More effective features will exhibit higher accuracy and F1 scores on this classification task. We leverage the official training and testing datasets provided by OpenAI for this purpose (Li, 2020; Radford et al., 2019b). These datasets are commonly used in detection research that wishes to understand detection of powerful generative Transformer models, as highlighted in Table 2.3 in Chapter 2.

To evaluate each model's robustness to text-based adversarial attacks, we subject each to DeepWordBug (Gao et al., 2018) and TextFooler (Jin et al., 2020) adversarial attacks. We select these attacks as they are realistic black-box attacks that represent two disparate themes in text adversarial attacks: DeepWordBug causes small character edits and attempts to maximize misclassification while minimizing Levenshtein edit distance, while TextFooler replaces words based on a pre-trained bi-directional encoder representation from Transformers (BERT) model (Devlin et al., 2019), replacing words with synonyms based on cosine similarity within the embedding space.

In attacking these models, we apply the assumption that the attacker has access to the output class confidence of the model, but not any internal weights or other model information. As TextFooler and DeepWordBug attacks are quite expensive, especially given that some input sequences are quite long, we sample 200 random texts from GPT-2 355M texts and human-written WebText. On this set, we perform targeted attacks for causing both Type I (false-positive) and Type II (false-negative) errors in computer-generated text detection. To determine impact of adversarial attacks on text quality, we featurize MAUVE against sentences from the original human-written samples in the WebText validation set. Then, as there are too few successful DeepWordBug attacks to compute MAUVE, we filter instances of successful TextFooler attacks against each model, and calculate MAUVE scores of sentences from GPT-2 samples before and after the adversarial attacks are applied. Finally, we calculate the resulting difference in MAUVE score, ΔMAUVE, to determine the impact on human quality assessment. As the number of sentences in comparatively small, we average ΔMAUVE over $k = 10$ trials. To our knowledge, this is the first time ΔMAUVE has been applied to assess impact of adversarial attacks on human-quality assessment.

Features	Pre-Training Dataset	SVM Accuracy / F1 Score						C=	Feature Size
		GPT-2 355M		GPT-2 1.5B		GPT-3 175B			
		Acc.	F1	Acc.	F1	Acc.	F1		
Statistical (S)	N/A	0.7030	0.6935	0.7120	0.7055	0.5850	0.5123	100	10
RoBERTa (R)	1B+ Weighted Web	0.7700	0.7686	0.7150	0.7053	0.5660	0.4694	10	1024
Concatenated ($S + R$)	1B+ Weighted Web	0.8000	0.8008	0.7450	0.7401	0.6030	0.5268	10	1034
MPNet	1B+ Weighted Web	0.7660	0.7715	0.7290	0.7150	0.5780	0.4725	10	768
MPNet	215M QA Pairs	0.7950	0.7968	0.7450	0.7319	0.5830	0.4715	1	768
$\text{BERT}_{\text{BASE}}$	MS MARCO	**0.8300**	**0.8310**	0.7720	0.7645	0.6200	0.5343	1	768
MiniLM	215M QA Pairs	0.7990	0.8012	**0.7860**	**0.7780**	**0.6520**	**0.5807**	100	384

Table 3.1: Performance of text feature embeddings for detection of computer-generated text

3.5 Datasets and Preprocessing

To determine the suitability of the classifiers for detecting state-of-the-art neural generated text, we use the official dataset provided by OpenAI for assessing computer generated text from GPT-2 (Kim, 2019). This dataset contains GPT-2 generated samples created by GPT-2 networks of varying parameter counts. This dataset also includes samples from the original WebText corpus used to train the model, which can be used as negative examples for training the classifiers. Similarly, we also include a sample of GPT-3 output provided on the official OpenAI GitHub repository for GPT-3 (OpenAI, 2020). We train all models on a balanced training dataset of human webtext and GPT-2 355M output, and test on 3 separate test datasets balanced between human webtext and output from GPT-2 355M, GPT-2 1.5B, and GPT-3 respectively. As an attacker may possess a large generative model that is not publicly available, it is useful to determine to what extent features derived from smaller architectures transfer to larger architectures.

Model	**Sampling**	**Sampling Param.**	**Sample Count**	**Usage**	**Model Size**
WebText	N/A	N/A	250K	Train	N/A
GPT-2	Top-K	K = 40	250K	Train	355M
GPT-2	Top-K	K = 40	5K	Test	355M
GPT-2	N/A	N/A	N/A	Test	1.5B
GPT-3	Nucleus	p = 0.85, t = 1	2K	Test	175B

Table 3.2: Summary of dataset properties and experimental setup

In order to replicate the complex phrasal features used in past work in the field, we harvest three additional datasets from online repositories. This includes a dataset of cliché phrases (Hayden, 1999), a dataset of English idioms (Wikitionary, 2022), and a dataset of Shakespearean archaisms (Cummings, 2005). To collect this data, we scrape these web resources using Python scripts and assemble the data in text format. Where permissible, we have made the data available in easy-to-download format via GitHub repository. The remaining data can be provided upon request.

Preprocessing of the data is done using two separate preprocessing workflows designed for their respective models:

1) For the statistical model, we follow the approach used in past statistical detection of computer-generated text (Nguyen-Son et al., 2017). We first tokenize the input using Stanza (Qi et al., 2020), then lemmatize the results. The number of tokens in the resulting text are tabulated as well. Lemmatization is also applied to the sets of complex phrasal features such that they can be matched against the lemmatized samples. Features are scaled by removing the mean and scaling to unit variance.

2) For Transformer models, data is first fed into a WordPiece tokenizer to convert it into tokens. Following this, the words are converted into identifiers based on their dictionary word IDs and provided to the neural network.

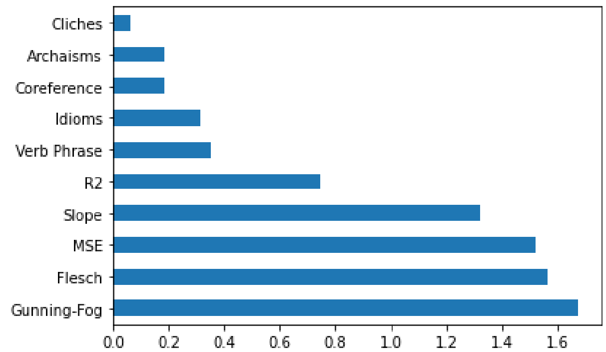


Figure 3.3: Feature weight comparison of SVM trained on statistical features

3.6 Experimental Settings

Experiments were executed on a virtual machine running Debian 10, with 32 vCPUs, 120GB RAM, and 4 NVIDIA T4 graphics processing units (GPUs). GPU acceleration was used to perform data preprocessing, model training, and deep learning inference more quickly whenever possible.

Throughout the experiments, hyperparameters are set to default values and random seeds are set to 0 to encourage reproducibility. All SVM models were trained with a C value based on exhaustive hyperparameter search across $C \in [1, 10, 100, 1000]$. Models were trained using a linear kernel to allow class probability output and feature importance measurement of the statistical model. Recall from §3.4.1 that experiments using RBF kernels resulted in comparable accuracy results.

Text adversarial attacks were performed using the open-source TextAttack framework (Morris et al., 2020). In order to use this framework, we provide a harness to act as an adapter between the TextAttack libraries and the original models.

3.7 Results

We report the accuracies and F1 score of models trained on the statistical and neural features in Table 3.1. Results of adversary classification under the presence of TextFooler and DeepWordBug attacks can be found in Table 3.4.

Machine/Human Classification TextFooler Samples	
Original (Label: Machine)	Deejai Bhatt, who grew up in ***Mississippi*** and moved to Memphis, says his time in America is an example of women being good in a bad environment.
Adversary (Label: Human)	Deejai Bhatt, who grew up in ***Biloxi*** and moved to Memphis, says his time in America is an example of women being good in a bad environment.
Original (Label: Human)	The ***protest*** events, scheduled to ***take*** place just days before Trump takes office, will focus on the on-going effort to ***repeal*** the Affordable Care ***Act***.
Adversary (Label: Machine)	The ***demonstrating*** events, scheduled to ***adopt*** place just days before Trump takes office, will focus on the on-going effort to ***abolishing*** the Affordable Care ***Bill***.

Table 3.3: Demonstration of Textfooler adversarial attacks inducing Type I and Type II errors on web text

3.8 Discussion

3.8.1 Statistical feature importance

Recall that as a linear kernel creates a separating plane within the same space as the input features, it is possible to use the coefficients of the SVM model as a measure of feature importance (§3.4.1). A plotting of the weight of each statistical feature can be found in Figure 3.3, and provides a summary of relative feature importance within the model. The most heavily-weighted features in this analysis were the results of computation for Gunning-Fog Index and Flesch Index scores. These additional fluency features – followed by Zipfian frequency features (slope, MSE, R^2) – are of greatest importance in classifying computer-generated text produced by contemporary generative language models.

In contrast to past work that highlighted the efficacy of complex phrasal features (Nguyen-Son et al., 2017), we find that complex phrasal features have low predictive power against Transformer-generated text compared to other features. This is exhibited by the low weights attributed to these features (clichés, archaisms, and idioms) within the resulting statistical model (Figure 3.3). In addition to improvements in underlying text generation models, this is likely due to shifts in the text domain considered. GPT-2 and GPT-3 are trained on web text, and produce text typically only as long as 10 paragraphs – past work involved machine translation of book-length text. The presence of Shakespearian archaisms, writing clichés, and idioms is likely far more common in book text than in computer-generated blog posts and news articles. Of these complex phrasal features, idiom features retain the most predictive power in detection of current generative models.

3.8.2 Classification performance

In Table 3.1, we find that overall, features derived from pre-trained neural language models outperform the selected statistical features when attempting to classify larger language models using features from smaller models. Among neural language models, it appears that features from the comparatively compact MiniLM language model trained for a question-answering task are most amenable to maintaining stronger performance as the target model scales up, while features from the $\text{BERT}_{\text{BASE}}$ model trained on MS MARCO web results perform well against the 355M parameter GPT-2 model, but do not transfer well to samples from larger architectures.

3.8.3 Adversarial robustness

Overall, we see from the results in Table 3.4 that statistical features are considerably more robust to adversarial attack than Transformer-derived features. A model trained on a combined vector of statistical and Transformer features provides a massive increase in adversarial robustness, while offering comparatively strong performance in non-attack settings.

We note that all models are more vulnerable to targeted attacks from TextFooler than DeepWordBug. Note that Textfooler performs a complete word exchange, whereas DeepWordBug makes small character-level changes in the form of dropping, adding, or swapping characters. Excerpts from successful TextFooler attacks can be found in Table 3.3, an illustration of DeepWordBug can be found in Figure 3.2.

Finally, within Table 3.4 we see that all adversarial attacks reduce MAUVE scores. When attacks succeed against statistical features, the resulting perturbed text demonstrates a greater decrease in Mauve score than successful attacks against RoBERTa features. This indicates that attacks that succeed against the statistical model produce lower-quality text. A lower MAUVE score increases likelihood of detection upon human review, and further diminishes the ability of human targets to interpret the text. As an example, consider the TextFooler attacks shown in Table 3.3. In the second passage, the words "***take*** place" are perturbed to "***adopt*** place", introducing a grammatical error. Further,

Features	Attack Type	Attack Succ. Rate	Pre-Atk Acc.	Post-Atk Acc.	ΔMAUVE
RoBERTa	TF	96.2%	0.780	0.030	-0.0115
RoBERTa	DWB	61.5%	0.780	0.300	–
Statistical	TF	**15.6%**	0.705	**0.595**	-0.0947
Statistical	DWB	**3.5%**	0.705	0.680	–
Concatenated	TF	27.8%	**0.810**	0.585	-0.0364
Concatenated	DWB	8.0%	**0.810**	**0.745**	–

Table 3.4: Feature performance for computer-generated text detection in presence of adversarial attacks

the word "Act" was perturbed to "Bill", which while grammatically correct, semantically alters the meaning of the text. In all domains – text, imagery, or otherwise – a key goal of adversarial attacks is that they should influence machine interpretation, while maintaining human interpretation.

3.8.4 Limitations

A limitation of this research is that we are only consider a limited range of word-based and character-based adversarial attacks. As there are a variety of different attacks that may have different strengths and weaknesses, a broader study that incorporates additional attacks in the text domain would likely be helpful. Towards this end, the greater availability of large-scale adversarial attack benchmarks or sample datasets may be of value to future research, as computing adversarial attacks can be computationally expensive.

A second limitation of this research relates to the possibility of sequence-level adversarial attacks where a contemporary generative text model might be used to reword a sentence until it passes scrutiny by a detector. This type of attack may require different approaches for detection, and the features that succeed for word-level and character-level attacks may not have the same predictive power on sequence-level attacks.

Finally, our research focuses primarily on understanding the predictive power and adversarial robustness of neural and statistical features when used alone and in conjunction for detecting machine-generated text. We did not fine-tune our classifier with the specific goal of creating the best-possible detector for machine generated text under adversarial setting. Future work may pursue the goal of producing the strongest-possible detector, though the increased prevalence of private companies marketing closed-source detectors for detecting generative AI text may complicate research related to achieving state-of-the-art detection performance.

3.9 Summary

In analysis of feature robustness against adversarial attacks, we find that statistical features possess resistance against attacks that heavily impact neural language models. This useful finding indicates that incorporation of statistical features may be a means of improving adversarial robustness of computer-generated text detection via a combination of neural and statistical features.

Statistical features previously useful in detection of computer-generated text – specifically complex phrasal features – are substantially less effective against contemporary models than in past research. Instead, additional features omitted from past work may be promising for this application, which include the Gunning-Fog and Flesch Indices.

While this chapter focused on automated detection of machine-generated text, the next chapter explores subjective assessment of machine-generated text. Similar to how we evaluate the quality of adversarial attack text via MAUVE score, we analyze machine-generated text under varying sampling parameters using both computational and human assessment. This shift in perspective allows us to better understand how sampling parameters impact generated text both in terms of computational quality measures, but also subjective qualities such as creativity.

Chapter 4

Evaluating Probability Mass on Subjective Assessment of AI Text

The work in this chapter was published as *In BLOOM: Creativity and Affinity in Artificial Lyrics and Art* at the AAAI 2023 CreativeAI workshop (Crothers et al., 2023b).

The research in this chapter complements the analysis in the previous chapters on detecting machine-generated text, investigating differences in human subjective assessment of human text and machine text, rather than purely focusing on detection. As this work was accepted for presentation as part of the AAAI 2023 workshop on creative AI, it also contains a creative workflow for multimodal lyrics and album art generation. A core component of this research rests in how humans perceive machine generated text for subjective properties (e.g., creativity) as sampling probability mass is adjusted, and the limitations of MAUVE as a catch-all means of assessing AI generated text. This specifically highlights limitations in existing evaluation methods for machine generated text, and potential future work on computational metrics that better reflect subjective qualities such as creativity.

From the perspective of detecting AI usage in artistic works, our findings from reviewer scores suggest that a reader may give similar scores related to creativity and affinity for both fully machine-written lyrics and popular human-written lyrics. The fact that such scores might be obtained without a human actively involved in shaping the artistic expression presents additional concerns related to likely increasing prevalence of AI in artistic pursuits, potentially supplanting human artistic expression.

The usage of Chinese language within this section serves two major purposes: 1) reducing the English-language bias in the research thus far by considering the most spoken first language in the world (Mandarin Chinese), giving a more rounded understanding of the current capabilities of generative models; and 2) reducing the possibility of bias in sample selection related to subjective assessment of lyrics as Chinese is not the author's first language.

4.1 Abstract

We apply a large multilingual language model (BLOOM-176B) in open-ended generation of Chinese song lyrics, and evaluate the resulting lyrics for coherence and creativity using human reviewers. We find that current computational metrics for evaluating large language model outputs (MAUVE) have limitations in evaluation of creative writing. We note that the human concept of creativity requires lyrics to be both comprehensible and distinctive — and that humans assess certain types of machine-generated lyrics to score more highly than real lyrics by popular artists. Inspired by the inherently multimodal nature of album releases, we leverage a Chinese-language stable diffusion model to produce high-quality lyric-guided album art, demonstrating a creative approach for an artist seeking inspiration for an album or single. Finally, we introduce the MojimLyrics dataset, a Chinese-language dataset of popular song lyrics for future research.

4.2 Introduction

Modern music production — which spans elements including musical composition, songwriting, album artwork, and music videos — is a tremendous domain for exploring creativity across modalities. This work focuses on better understanding subjective human assessment of creative text produced by large language models (LLM), demonstrating that mimicry of an existing sample dataset may not always be the best approach when considering human preferences.

Towards this end, we use BLOOM (BigScience, 2022), a multilingual LLM with 176B parameters, to generate Chinese song lyrics with varying sampling parameters. We focus on top-p sampling, as this method has been found to outperform top-k and beam search in prior research (Holtzman et al., 2019a). In order to assess the quality of the generated lyrics, we task human reviewers to provide feedback on lyrical coherence, creativity, and enjoyment. Using this data, we demonstrate the relationship between creativity and coherence in human-assessed lyric quality, and highlight differences between subjective human assessment and computational methods such as MAUVE (Pillutla et al., 2021).

Beyond comparison of creativity measures, we also dive into new emerging multimodal possibilities related to AI-driven creativity. Music today is often accompanied by cover art that evokes the aesthetic of the piece. This art, previously displayed on record covers and lyric booklets, remains prominently displayed on music apps and streaming platforms. While album cover art may reflect a collection of songs (an album, LP, or EP), digital music publishing allows for easy release of singles, and cover art is increasingly used to accompany individual tracks (Leight, 2018). Using our lyrics, we use the Chinese-language Taiyi stable diffusion model (Wang et al., 2022c) to generate an album image inspired by the LLM-generated lyrics. The demonstrated workflow may be used for inspiration by songwriters seeking a unified lyrical and visual concept for a single or album, as well as being a source of entertainment to music fans.

4.3 Related Work

Previous work has used LLMs to generate classical Chinese poetry (Liao et al., 2019). Work has also been performed which uses GPT-2 as a generation system for various types of creative Chinese writing, contemporary song lyrics among them (Zhang et al., 2022b). Related work has used an image-generation network as an intermediate step in generating non-lyrical creative text, based on the idea of an author "visualizing" while they work (Zhu et al., 2022). In English, research has similarly been done on LLM lyric generation (Rodrigues et al., 2022), including lyrics conditioned on a song's melody (Chen and Lerch, 2020).

Our work is distinct from previous work in that it focuses on zero-shot lyric generation with a massive multilingual model (BLOOM), is the first to our knowledge to benchmark lyrics against evaluation metrics that strongly correlate to human assessment (MAUVE), and most importantly, presents an investigation into the relationship between computational assessment of open-ended text quality, and subjective measurements. In this department, we are inspired by previous work on quantifying creativity that found that KL divergence has a strong correlation with human-assessed creativity for word pairs (Kuznetsova et al., 2013).

Finally, this work is also the first research work to apply diffusion models as part of a unified approach for generating coordinated cover art in association with song lyrics. All models and code and from our work will be made available open-source to enable future research[1].

4.4 Methodology

We organize our methodology into three major parts: 1) collecting and cleaning Chinese lyrics found online to create a novel dataset of popular Chinese lyrics for use with MAUVE, 2) performing lyric generation effectively with the BLOOM-176B LLM, and 3) utilizing the Taiyi stable diffusion model to generate high-quality art to accompany the lyrics.

4.4.1 Dataset

For comparison to LLM-generated lyrics, we collect a dataset of song lyrics for popular Chinese songs from the website Mojim.com. Mojim is a popular lyrics sharing website that focuses on Chinese song lyrics. We request the top popular artists in the "male", "female", and "group" categories, fetch all the albums associated with each artist, and download the lyrics for every song in each album. Metrics on the dataset can be found in Table 4.1.

Data downloaded from Mojim.com is in raw HTML format and must be cleaned to extract raw lyrics. This process is summarized as follows.

- Removal of HTML tags, as well as the removal of markup symbols not typically part of the lyric text, such as "*" and "#" and various unicode equivalents.
- Removal of site-specific text characteristics, such as: a recurring line in Chinese which translates to "Find more lyrics at Mojim.com", dividing lines composed of hyphens, non-standard labels placed at the start of the song.
- Compress instances of many subsequent newline characters into just two newline characters, to reduce excess whitespace which may emerge after removing tags, annotations, and symbol markup.

[1]https://github.com/ecrows/in-bloom

Songs	Artists	Tokens	Size
39747	230	14,406,854	38.4MB

Table 4.1: Statistics for the MojimLyrics Dataset

4.4.2 Lyric Generation

We generate zero-shot samples using the multilingual BLOOM-176B model (BigScience, 2022) at varying values of p for top-p nucleus sampling (Holtzman et al., 2019a). By adjusting the probability mass used to sample the next token, we produce varied lyric sequences that can be used to analyze quantitative and subjective text characteristics. Our evaluation of lyrics focuses on two major approaches:

- **Computational Quality**: We measure computational quality via commonly-used metrics that target language model degeneration (repeated n-grams, diversity of tokens, distinct n-grams), as well as distributional information via Mauve (Pillutla et al., 2021).
- **Subjective**: Subjective measures of coherence are collected via a carefully survey on Amazon Mechanical Turk. These measures include an assessment of how creative, coherent, and enjoyable the lyrics are.

Previous research on quantifying creativity indicates that for measuring perceived creativity of word pairs $\{w_1, w_2\}$, the KL divergence of $\{w_1, w_1 w_2\}$ is among the most effective measures (Kuznetsova et al., 2013).

Mauve is designed to jointly capture two types of errors between the output distribution Q of a model, and the human distribution P. 1) Type I errors, where Q places high mass on text which is unlikely under P; and, 2) Type II errors, where Q cannot generate text which is plausible under P. Mauve does this by summarizing Type I and Type II errors using Kullback–Leibler (KL) divergences, which are measured softly using a mixture distribution. Due to the similarity between Mauve's computation and previous work on creativity quantification, it is possible that Mauve may partially reflect subjective human creativity assessment, which may contribute to Mauve's strong performance for evaluating generated text quality more generally.

Additional text quality metrics for assessing language model output are selected based on previous work (Su et al., 2022; Zhu et al., 2022). These metrics include rep-n = 1.0 - $\frac{|\text{unique } n\text{-grams}|}{|\text{total } n\text{-grams}|}$ measuring duplicate n-grams within each generated sequence (Welleck et al., 2020), distinct-$n = \frac{|\text{unique } n\text{-grams}|}{|\text{length of text}|}$ (Li et al., 2016a), and diversity $= \prod_{n=2}^{4}(1 - \text{rep-}n)$ for measuring diversity of n-grams (Su et al., 2022).

Autoregressive language models such as BLOOM perform sampling to select a next token typically using some variation of top-k, beam search, and nucleus top-p sampling. Due to the strong documented performance of top-p sampling (Holtzman et al., 2019a), we produce BLOOM generations at varying levels of $p \in [0.80, 0.85, 0.90, 0.95, 0.99]$.

4.4.3 Album Art Generation

We leverage our generated song lyrics to create accompanying track art inspired by the song lyrics. As a generative model, we utilize the Chinese-language Taiyi Stable Diffusion model (Rombach et al., 2021; Wang et al., 2022c), which includes a Chinese-language text encoder. We use this model to produce 1000 samples of album artwork, from seeds 0-999, using the prompt "album art" in Chinese, followed by our generated lyrics. We perform this at varying settings of classifier-free diffusion guidance (CFG), where $CFG \in [4.0, 7.0, 10.0]$. Images are generated with 20 sampling steps using the Euler A method (Rombach et al., 2021). For evaluation, we utilize a dataset of approximately 4,000 album covers at 512x512 resolution (Surma, 2019). We determine the quality of the generated images by calculating Fréchet Inception Distance (FID) (Heusel et al., 2017; Seitzer, 2020) using the pool 3 activations from an Inception network (Szegedy et al., 2015) between generated album covers and the real album dataset.

高山白雪 铸剑之地
于此剑中 缘份精魂
被磨得锋锐无比
此剑出鞘 成为神剑
这把剑，是用来保护你爱的人
如果,我的爱 因这把剑消失
我愿拼尽全力 在所不惜
如果这是为了保护你
这把剑 我的剑 就此消失
倘若神怜悯你 终会有所作为
莫失良机 听其言 行其

Figure 4.1: BLOOM-176B $p = 0.95$, $seed = 0$ lyrics mentioning a sword, accompanied by corresponding stable diffusion generated images selected from the resulting prompt at $CFG = 0.7$, $seed = 0$, $batch\_size = 6$

4.5 Experimental Setup

Inference on BLOOM-176B was performed via the Huggingface Inference API. All other calculations (stable diffusion, Mauve, FID) were performed on a machine with an Intel i7-6800K 12-core CPU, 32GB RAM, and a 24GB VRAM RTX 3090 GPU. Seeds starting at 0 are used throughout our experiments for reproducibility.

Source	rep-2 ↓	rep-3 ↓	rep-4 ↓	diversity ↑	distinct-2 ↑	MAUVE ↑
Human	**0.021**	**0.013**	**0.012**	**0.954**	**0.870**	-
$LLM_{p=0.80}$	0.120	0.090	0.072	0.743	0.650	0.139
$LLM_{p=0.85}$	0.095	0.073	0.063	0.786	0.689	0.162
$LLM_{p=0.90}$	0.072	0.055	0.047	0.835	0.722	0.207
$LLM_{p=0.95}$	0.054	0.044	0.039	0.868	0.760	0.269
$LLM_{p=0.99}$	**0.040**	**0.034**	**0.032**	**0.897**	**0.783**	**0.293**

Table 4.2: Quantitative degeneration metrics and MAUVE divergence-based quality results for BLOOM-176B generated text at varying sampling probability mass p

4.5.1 BLOOM

In order to use BLOOM to generate lyrics in a zero-shot setting, we provide the following prompt:

下面是一首歌的中文歌词。\n歌词：\n

Translation: Below are a song's Chinese lyrics. \n Lyrics: \n

This prompt was chosen based on the observed result of Chinese language song lyrics. Shorter prompts such as simply the words for "lyrics" in Chinese more often resulted in non-song lyrics such as interviews with musicians, and more specific prompts that mentioned specific artists or too closely resembled formatting from specific lyric websites had a higher incidence of generating lyric sequences from real songs. Memorization in large language models is well-documented and is a significant problem undergoing research (Carlini et al., 2021). The selected prompt appears to minimize this behaviour, though future analysis on memorization would be valuable.

4.5.2 Amazon Mechanical Turk

Using Amazon Mechanical Turk to evaluate open-ended text generation is challenging and requires care to perform effectively (Karpinska et al., 2021), however, such platforms remain useful for their ability to provide real human feedback, A selection of 60 samples were taken from the BLOOM generated lyric corpora at each value for $top-p$ nucleus sampling, as well as lyrics from the MojimLyrics dataset. In all, 152 reviewers provided responses to the survey. Each sample was assigned to a minimum of 3 different reviewers to score the lyrics via multiple choice responses for coherence, creativity, affinity (how much the reviewer likes the lyrics), and recognition (whether they believe they may have seen the lyrics before). Multiple choice answers for each question were converted into normalized scores between 0 and 10. The full survey questions are provided in the project GitHub repository.

Instructions were provided in Simplified Chinese. To encourage close reading, a starting question requests a short summary of the lyrics, and a final multiple-choice question tests the reviewer's basic fluency in Chinese by placing the option "I can read Chinese" among various other negative options. Participant responses were removed if they failed to provide a summary or respond correctly to the literacy-testing question.

The projected task duration was 120 seconds, with payment of $0.44 CAD for a projected hourly wage of $13.20 CAD. Actual mean task duration was confirmed to be within the projected task duration.

4.5.3 MAUVE Calculation

For calculating MAUVE, we featurize p with a random sample of 3000 real lyrics from the MojimLyrics dataset, and featurize q with 3000 generated lyrics from BLOOM-176B at each probability mass from our nucleus sampling $p \in [0.80, 0.85, 0.90, 0.95, 0.99]$. We set a maximum text length of 128 and set a batch size of 32 based on the GPU memory available. The seed for the MAUVE calculation is set to 0.

4.6 Results

Results for language model evaluation with MAUVE scores and degeneration metrics can be found in Table 4.2, while results of stable diffusion CFG variation on FID can be found in Table 4.3. The complete dataset of generated BLOOM lyrics are provided in the GitHub repository. Human assessment of lyrics can be found in Figure 4.2.

A sample of generated lyrics and the resulting stable diffusion art can be found in Figure 4.1. FID scores calculated between outputs of the diffusion model and real album art can be found in Table 4.3.

On the 10 point normalized scale, mean standard deviation among multiple responses for the same input sample were observed to be 1.29, 1.68, 1.95, and 2.67 for coherence,

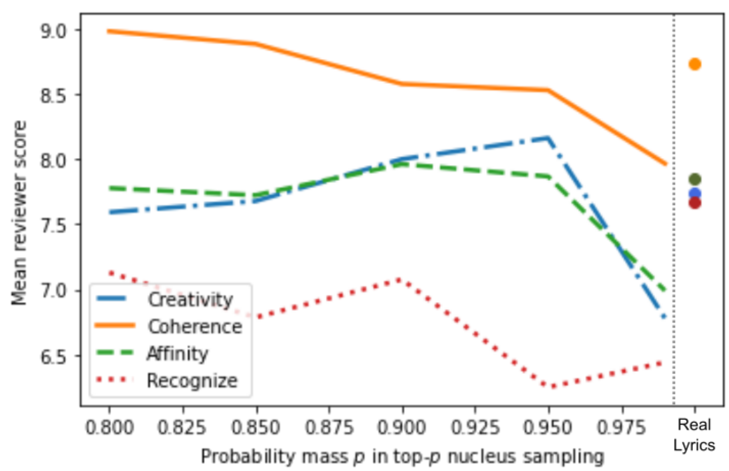


Figure 4.2: Mean human scores for subjective attributes of generated text at varying sampling probability mass p. Included to the right are reviewer scores for real lyrics.

	CFG@4.0	CFG@7.0	CFG@10.0
FID↓	90.21	93.84	95.91

Table 4.3: FID with real album dataset of Taiyi stable diffusion generated album art at varying CFG

affinity, creativity, and recognition respectively. However, despite the relatively low variance in response scores, measures of inter-annotator agreement (IAA) are consistently low (i.e., negative Krippendorff ordinal alpha). Survey responses were highly skewed towards positive answers. When a reviewer gave a particular sample a low score, often it was only one reviewer did so. This implies that while reviewers largely agreed on ratings, there was not a consistent criterion across reviewers for giving low scores. Obtaining high IAA scores on concepts such as creativity and personal preference will likely require a larger study with a greater number of trained reviewers.

4.7 Discussion

Sampling BLOOM for lyrics from a greater portion of the probability mass tends to improve quantitative metrics of generated lyric quality, as shown in Table 4.2. As expected, token diversity, distinct-n, and rep-n all improve at increasing p portion of the probability mass. That MAUVE score also continues improving implies a low incidence of Type I errors where the model output distribution Q places high probability mass on text which is unlikely under P. Notably, even at very high p-values, BLOOM-176B generated lyrics are still more repetitive and less diverse than human lyrics.

Human assessment of lyrics at varying p is charted in Figure 4.2. Coherence is highest at low-p, where common words are frequently sampled, but with a low accompanying creativity score. Subjective quality measurements of creativity and affinity peak around $p = [0.9, 0.95]$, and trend sharply downwards at $p = 0.99$ as the model selects less common words and becomes less coherent. This implies that for a human to recognize work as “creative”, it must be coherent enough to be understood, but distinctive enough to be interesting. While MAUVE rises alongside subjective creativity from $p = 0.80$ to $p = 0.95$, this relationship breaks down at $p = 0.99$, implying that MAUVE alone is not a complete replacement for subjective human evaluation.

Finally, it is notable that real popular lyrics uploaded to Mojim.com are at points deemed less creative, coherent, and likable than BLOOM-176B generations. Resemblance to a human dataset does not inherently bestow positive perceived qualities to generated art. While the best-scoring CFG setting in Table 4.3 appears to be at CFG=4.0, this measurement alone is not indicative of subjective human quality.

4.7.1 Limitations

An important limitation of this research is that we only analyze popular song lyrics, and that these lyrics are experienced by an audience that is reading them while answering a survey. The range of expression in popular song lyrics may not be representative of the wider range of artistic possibilities in songwriting more generally. Furthermore, actual song lyrics are experienced while listening to a song, and crowd workers who are trying to read the lyrics quickly to finish a paid task may not be well-situated to give nuanced feedback about how a set of lyrics make them feel.

A second limitation of this work, is that while we provide a multimodal system that relates the generated lyrics into album art, we do not assess that album art through the same subjective quality assessment, instead focusing our investigation of evaluation on the text generation aspects. Regardless, the insights into the gaps in subjective quality assessment of machine generated text are noteworthy in their own right. Future work might also include deeper subjective analysis of machine generated imagery.

4.8 Summary

In summary, we have demonstrated that current computational techniques for evaluating large language model output (both quantitative degeneration metrics and MAUVE), are insufficient as proxies for subjective assessment of creative writing. To visualize the lyrics produced and add a creative artistic interpretation, we provide a combined approach for generating album art based on lyrics using stable diffusion. Finally, we provide a novel dataset of in-the-wild popular Chinese song lyrics, and the code for repeating our experiments.

Based on human reviewers, we find that large language models such as BLOOM-176B can be used to produce zero-shot song lyrics that are evaluated favourably in comparison to real song lyrics. This evaluation, combined with the increasing availability of user-friendly interfaces for large langauge models, make it likely that AI models will be used in the future for aiding in the production of artistic works. The use of AI systems in production of art raises serious questions around authorship, attribution, and commodification of art. These concerns must be taken seriously, as while new creative AI systems may serve as useful tools for human artists exerting their own agency, automated AI workflows might be cynically leveraged with minimal human oversight, flooding the media landscape with automated content in a manner that crowds out genuine human expression. In this regard, research targeting the ethical implications and detection of such content remain important areas of future work.

Having explored both computational and subjective methods of evaluating machine generated text, we now shift towards interpretability of neural text classifiers. As we have used neural text classifiers in detection of machine-generated text, there is value in being able to produce explanations as to why a model produced a given detection result. In the next chapter, we explore methods of comparing model interpretability, and highlight pitfalls in comparing models using widely-used faithfulness measures.

Chapter 5

Large Language Model Classifiers and Interpretability

This chapter reflects work that is currently under consideration by the journal Computational Intelligence as the article "Don't Be Tricked By Iterative Masking", which was adapted and extended from an accepted conference paper to be presented at Learning, Optimization and Data Science (LOD) 2024 as "Robust Infidelity: An Analysis of Pitfalls in Faithfulness Measures on Masked Language Models" (Crothers et al., 2024).

This serves as the fourth major work of the thesis: an analysis of masking-based faithfulness measures applied to LLM-based neural text classifiers. Explainability of LLM classifiers is an important quality for their usage in areas with a high requirement for human oversight, and these same types of classifiers are commonly used for detection of AI text, as highlighted in the work in Chapter 2 and 3. This investigation focuses on how the iterative masking mechanism utilized by faithfulness measures can lead to pitfalls that undermine model comparison, and investigates what impacts favouring these features has on adversarial robustness and fairness.

5.1 Abstract

A common approach to quantifying neural text classifier interpretability is to calculate faithfulness metrics based on iteratively masking salient input tokens and measuring changes in the model prediction. We propose that this property is better described as "sensitivity to iterative masking", and highlight pitfalls in using this measure for comparing text classifier

interpretability. We show that iterative masking produces large variation in faithfulness scores between otherwise comparable Transformer encoder text classifiers. We then demonstrate that iteratively masked samples produce embeddings outside the distribution seen during training, resulting in unpredictable behaviour. We further explore task-specific considerations that undermine principled comparison of interpretability using iterative masking, including an underlying similarity to salience-based adversarial attacks. Issues with the unreliability of faithfulness evaluation have implications on fairness-optimized model checkpoints. A model's tendency in faithfulness scores can persist through fine-tuning, meaning that models fine-tuned from a fairness-optimized foundation model (FairBERTa) can consistently score lower on faithfulness than the basic version of the same model (RoBERTa), despite both models having comparable performance on classification tasks. Our findings give insight into how these behaviours affect neural text classifiers, and provide guidance on how sensitivity to iterative masking should be interpreted.

5.2 Introduction

5.2.1 Iterative Masking in Context

In the field of natural language processing, the pursuit of interpretable AI has brought the spotlight onto a critical interplay between neural text classifiers, explainability techniques, and faithfulness measures. To understand the role that iterative masking plays in the current explainability landscape of neural text classifiers, we must understand the relationship between contemporary neural text classifiers, explainability techniques applied to these models, and faithfulness measures applied to such explanations. A diagram showing the interrelation of these concepts can be found in Figure 5.1.

Neural text classifiers are increasingly based on the Transformer architecture. Transformer models have become ubiquitous in natural language processing (NLP), achieving state-of-the-art performance across a variety of domains (Vaswani et al., 2017). Among these domains, these models remain very effective for text classification tasks. Despite this success, there remains widespread concern about the lack of transparency and explainability of neural language models (Lipton, 2018), which limits their use in high-stakes applications requiring human oversight or explanation.

Explainability techniques can be utilized to address this gap in interpretability, commonly providing feature attributions to improve interpretability of neural text classifiers.

These feature attributions present scores that indicate how much each input token contributes to a given prediction (Lundberg and Lee, 2017; Sundararajan et al., 2017). These scores allow a reviewer to identify important input tokens that contributed to a given output classification. However, in order for an explanation to be meaningful, it is important that the explanation reflects the real behaviour of the model — that it is "faithful" (Jacovi and Goldberg, 2020).

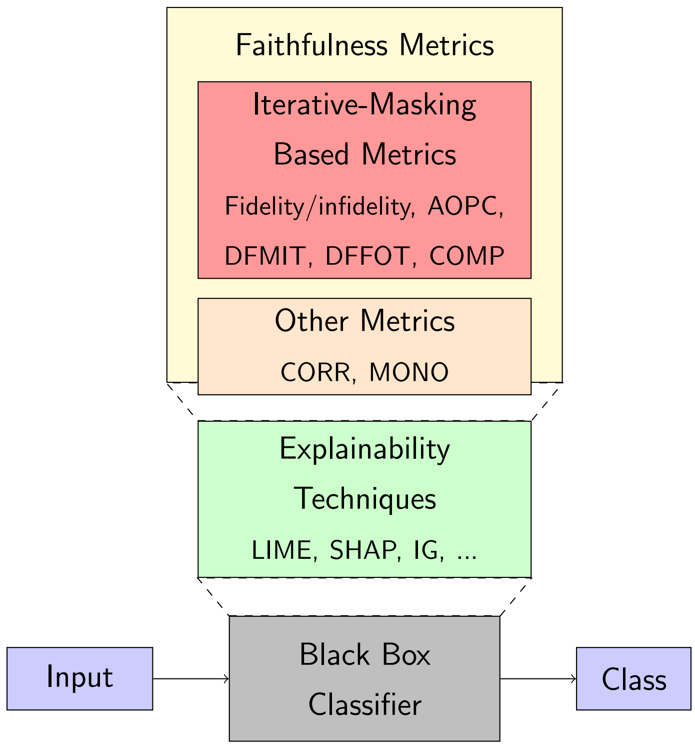


Figure 5.1: Diagram illustrating the relationship between classifiers, explainability techniques, and faithfulness metrics. The work in this article focuses on faithfulness metrics based on iterative-masking (highlighted in red).

Faithfulness measures thereby aim to quantify the quality of a feature attribution explanation based on how well an explanation reflects the observed behaviour of a model, separate from human-centric considerations such as an explanation's plausibility (Jacovi and Goldberg, 2020). Measurement of faithfulness relies upon automated measures designed to quantify to what extent an explanation accurately depicts how a model behaves. For neural text classifiers, these measures are generally calculated by iteratively removing the most salient features and measuring changes in the model output (Atanasova et al., 2020; DeYoung et al., 2020; Nguyen, 2018; Zafar et al., 2021). This process of iterative masking, used for calculating faithfulness measures, is the focus of our investigation.

5.2.2 Model Comparisons Based on Faithfulness

Faithfulness measures are well-suited for evaluating *explanations*. However, problems emerge when using these measures for evaluating *models*.

In the former setting, faithfulness measures allow for comparison of feature explanations based on whether the explanation correctly ranks the tokens that cause the largest impact on the output (Sundararajan et al., 2017). Under these conditions, comparisons are done using the same model, and as a result, model-specific behaviours are controlled.

In the latter setting, however, faithfulness are used as a means of comparing different models. In such an approach, the same feature explanation method is applied to two models, a masking-based faithfulness measure is calculated, and the model that produces higher average scores is theorized to be inherently more interpretable (Naylor et al., 2021; Yoo and Qi, 2021).

There are challenges with the model comparison setting, primarily related to model-specific behaviours that have a strong impact on the response of text classifier models to the iterative masking of input features. The presence of these behaviours has been well-documented in previous work, which has highlighted that models that differ only in initialization with comparable task performance can produce divergent faithfulness measures (Zafar et al., 2021).

Limited work has yet been done in understanding how and why these deviations occur, and more fundamentally, to determine to what extent a comparison of models based on a masking-based faithfulness measure is meaningful. Our work contributes to this topic by performing a deep analysis along two major branches of investigation.

5.2.3 Contributions

Our two major branches of investigation follow previous work demonstrating the lack of robust interpretability for neural text classifiers (Zafar et al., 2021), shedding light on the mechanisms that lead to this lack of robustness, and considering the downstream impacts of favoring models with high faithfulness on adversarial robustness and fairness. Specifically, we investigate:

1. The mechanisms that lead to this lack of robustness, in particular:

(a) Why behavior on iteratively masked language samples is inconsistent between models [Section 5.6]

2. The downstream impacts of favoring models with high faithfulness, including:

(a) How favoring "interpretable" models impacts adversarial robustness [Section 5.7]

(b) How favoring "interpretable" models impacts fairness [Section 5.8]

Towards the first question, we provide a systematic analysis of pitfalls in comparing neural text classifier interpretability using widespread faithfulness measures, providing new insight into mechanisms that cause the lack of robustness demonstrated in previous work (Zafar et al., 2021). By analyzing the impact of iteratively masking samples on model embeddings, we show that these samples are out-of-domain, producing a distribution of embeddings with a centroid that departs substantially from the original distribution, with less variation between embeddings. We illustrate this with uniform manifold approximation and projection (UMAP) dimensionality reduction (McInnes and Healy, 2018). Our findings suggest that as partially masked samples aren't encountered during training, the process of masking produces out-of-domain samples (often without a clearly-defined correct classification), leading to model-specific characteristics to dominate. This analysis provides novel insight into the mechanisms that cause faithfulness-based comparisons of models to break down.

We further demonstrate the important implications of these findings for machine learning practitioners working with explainable neural text classifiers. Practitioners may wonder: even if faithfulness comparisons based on iterative masking are often not meaningful, is there any harm in selecting a model with a high faithfulness score? We answer this clearly by demonstrating the interrelation between faithfulness and two other important characteristics of sensitive models: adversarial robustness and fairness.

Our analysis suggests that there are situations where optimizing neural text classifiers for faithfulness may come at the cost of adversarial robustness or fairness, while providing questionable advantage in explainability due to observed initialization-specific behaviour (Zafar et al., 2021). Specifically, we identify that there are circumstances in which iteratively masking salient tokens bears a strong resemblance to a token-level adversarial attack, raising concerns about whether sensitivity to iterative masking (the method most-commonly used to measure faithfulness in neural text classifiers) is an unambiguously desirable property. Furthermore, in analyzing how fairness-optimized text classifiers perform on faithfulness measures, we note that models fine-tuned from fairness-optimized models can retain the sensitivity to iterative masking present in the base model. This causes models initialized from a fairness-optimized checkpoint with low initialization-specific faithfulness to result in downstream models that may be deemed less interpretable, even if they have similar predictive ability with improved fairness. These contributions provide clarity for practitioners to make better decisions when training or selecting neural text classifiers. We further provide high-level recommendations to assist practitioners who need to make practical decisions on text classifier models that take into account the interrelation of these topics.

5.2.4 Organization

The remainder of this work is organized as follows. Section 5.3 describes related work on evaluating faithfulness in neural text classification. Section 5.4 provides the motivation of our analysis, discussing the pitfalls in using faithfulness to compare masked language models. Section 5.5 describes the datasets, models, and experiment settings used. Section 5.6 demonstrates experimentally that removing words from samples produces out-of-manifold inputs. Section 5.7 explores how robustness to iterative masking relates to adversarial attacks. Section 5.8 studies how fairness-optimized text classifiers score on faithfulness measures, including after fine-tuning. Section 5.10 contains discussion and recommendations based on our findings, including limitations of our study. Section 5.11 summarizes the findings and presents our conclusions.

5.3 Related Work

This section provides an overview of the key areas of research related to our work. We begin by discussing feature-based interpretability methods, which form the foundation for calculating faithfulness measures. Next, we discuss approaches for interpreting attention mechanisms in Transformer models, highlighting why we focus on feature-based methods for our analysis. We then delve into the concept of faithfulness measures, explaining their calculation and limitations. To connect our findings to practical implications, we introduce adversarial attacks in text classification and a relevant benchmark. Finally, we explore the relationship between adversarial robustness and faithfulness, setting the stage for our investigation into the challenges of using faithfulness measures for cross-model comparisons, and the impacts on adversarial robustness. Throughout this review, we emphasize the interconnected nature of interpretability, faithfulness, and robustness in neural text classifiers.

5.3.1 Feature-based Interpretability Methods

Feature-based interpretability methods for deep learning models, such as LIME (Ribeiro et al., 2016), SHAP (Lundberg and Lee, 2017), and integrated gradients (Sundararajan et al., 2017) assign an importance score to input features to determine their contribution to a particular network result. Evaluation of these interpretability methods has shown that gradient-based approaches demonstrate the best agreement with human assessment for Transformer models, as well as best correlating with tokens which cause the greatest drop in performance if they are removed from the model (Atanasova et al., 2020). The lower relative computational cost of integrated gradients compared to e.g. SHAP provides an additional benefit in terms of allowing a larger number of samples and experiments to be performed.

Based on these findings, we use integrated gradients to generate feature attributions in our work. Integrated gradients is a strong axiomatic method for calculating how input features contribute to the output of a model (Sundararajan et al., 2017). This method interpolates between a baseline input representation x' (in this case a zero embedding vector) and the embedding vectors x of each token.

The robustness of feature-based interpretabilty methods for neural text classifiers has been questioned (Madsen et al., 2022; Zafar et al., 2021). Specifically, it has been demonstrated that 1) two functionally near-equivalent models with differing weight initializations may produce different explanations, and 2) feature attributions of a model with random parameters may be the same as for a model with learned parameters.

Explanations that produce feature importance scores are the foundation for an important calculation in quantifying the quality of explanations: faithfulness measures.

5.3.2 Interpretation of Attention Heads

In addition to feature-based interpretability methods, it is also possible to analyze Transformer networks through interpretation of attention heads. We briefly describe these methods, though the lack of input feature attributions makes them incompatible for calculation of faithfulness measures.

A previous investigation of attention in BERT provides insight into which layers of the model learn high-capacity representations (Clark et al., 2019). This previous work found that specific attention heads specialize for specific linguistic relationships in the English base BERT cased model. For example, the BERT attention head 8-10 (the 10th head in layer 8) appeared to specialize in direct objects attending to verbs, while 8-11 specializes in noun modifiers attending to their matching noun. This was assessed by matching the results of the Stanford dependency parser against the strongest relationships in various attention heads of the model. There has also been work on determining how syntactic knowledge is represented within BERT (Jawahar et al., 2019). Attention head interpretation helps understand the behaviour of Transformer models, but scores for each input token are required when the intent is to explain the classification behaviour on a specific sample, or generate faithfulness measures.

Further, previous work on explaining misogyny detection has explicitly stated that they "discourage the use of attention to explain BERT-based misogyny classifiers" (Attanasio et al., 2022). This assessment was based on the discrepancy between visualizations of hidden token attribution (Brunner et al., 2020) and attention maps. The qualitative experiments, however, reveal that this was done without incorporating the norm of the transformed feature vector, which has been shown to be essential for interpreting attention in the context of neural networks (Kobayashi et al., 2020). As such, it remains a topic of debate whether the observed discrepancy between hidden token attribution and attention maps is sufficient to rule out the usage of attention maps in Transformer model interpretation.

Within this work, our focus is on the iterative masking mechanism underpinning faithfulness measures. Input feature attributions are a necessity to calculate current faithfulness measures, as such, we select integrated gradients as a strong axiomatic method for producing feature-based explanations.

5.3.3 Faithfulness Measures

Faithfulness measures in NLP quantify how well an explanation reflects model behaviour by measuring the degree to which erasure of high-ranking input features cause perturbations to the output (Jacovi and Goldberg, 2020).

Faithfulness measures are calculated by iteratively hiding features in descending order of feature importance and determining a score based on changes in model output. These scores may be calculated by removing features until the output classification changes (Nguyen, 2018; Zafar et al., 2021) or by removing a preset number of salient tokens and comparing the change in class probabilities (Atanasova et al., 2020; DeYoung et al., 2020).

We now outline how common faithfulness measures are calculated.

Fidelity Calculation

Within our experiments, we use fidelity as our faithfulness measure. Fidelity is a measure in which tokens are masked from the input in descending order of feature attribution scores until the result of the model changes (Arras et al., 2016; Nguyen, 2018; Zafar et al., 2021). The resulting score is defined by the % of tokens that must be removed at the point that the model's output changes. The process of masking and calculation is as follows.

Formally, given an input text split into n tokens, $T = [t_1, ..., t_n]$, a vector of feature explanations $\Phi(T) = [\phi(t_1), ..., \phi(t_n)]$, a model m, and the model's unknown vocabulary token `[UNK]`, we first calculate the initial prediction $y_0 = m(T)$. We then define an iterative scoring function $f(T)$, that at each step performs the replacement $T[\texttt{max}\ \Phi(t)] \leftarrow \texttt{[UNK]} = T'$, and calculates $m(T') = y'$. We iterate C times until $y' \neq y_0$, and return the ratio $f(T) = \frac{C}{n}$. $f(T)$ is calculated for all K input texts, and we calculate the fidelity score for the model as:

$$Fidelity(m) = 1 - \frac{1}{K} \sum_{k=1}^{k} f(T_k) \tag{5.1}$$

We rely on fidelity due to its simplicity and the lack of *a priori* parameters. There are also related faithfulness measures that operate on masking set numbers of tokens, such as area over the perturbation curve.

Area over the perturbation curve

Another faithfulness measure that relies on masking a preset number of tokens is the "area over the perturbation curve" (AOPC) (Nguyen, 2018; Samek et al., 2017). AOPC involves first creating an ordered ranking of input tokens by importance $x_1, x_2, ...x_n$ in each n-token input sequence. AOPC is then calculated by:

$$AOPC = \frac{1}{L+1} \left\langle \sum_{k=1}^{L} f(x) - f(x_{1..k}) \right\rangle_{p(x)} \tag{5.2}$$

where L is the *a priori* number of tokens to mask, $f(x_{1..k})$ is the output probability for the original predicted class when tokens $1..k$ are removed, and $\langle \cdot \rangle_{p(x)}$ denotes the average over all sequences in the dataset. We refer to this calculation when discussing theoretical weaknesses of cross-model faithfulness comparison based on iterative masking.

Other Faithfulness Measures

There also exists a range of other faithfulness measures which are very similar to the two previous measures (Chen et al., 2021). Decision Flip - Fraction of Tokens (DFFOT) is calculated using the same underlying equations as fidelity: tokens are removed until the prediction changes (Serrano and Smith, 2019). Decision Flip - Most Important Token (DFMIT) assumes the prediction should change after switching a single token – effectively AOPC where the number of tokens to switch is set to "1" (Serrano and Smith, 2019). "Comprehensiveness" (COMP) is closely related to AOPC, using the same underlying assumption that removing a number of salient tokens should perturb the output more if the explanation is accurate (DeYoung et al., 2020).

DFFOT is equivalent to $1 - fidelity$. DFMIT and COMP are both closely related to AOPC, and relying on masking a set number of salient tokens. DFMIT is very difficult to use in practice on longer sequences, as it very often results in null scores due to masking only a single token (hence a smaller percentage of the overall number of tokens on larger sequences). As all of these measures closely resemble the aforementioned fidelity and AOPC measures, we expect our findings will be unchanged or very similar when using these measures (Arya et al., 2019).

There are also other faithfulness measures with calculations that deviate slightly more from fidelity and AOPC, namely "Correlation between Importance and Ouptut Probability (CORR) and Monotonicity (MONO) (Arya et al., 2019). CORR measures the Pearsons correlation of feature importance and the effect on output probability. MONO is calculated by starting with an empty input and iteratively adding tokens in order of feature importance, expecting that the probability of the predicted class will monotically increase. Both of these measures also rely on masking tokens, and so likely encounter the challenges that we identify here related to masking-based techniques, but may have other characteristics that should be investigated in future research.

Limitations of Faithfulness Measures

While faithfulness measures can be used for comparing the quality of explanations (Atanasova et al., 2020; Nguyen, 2018), problems emerge when using faithfulness to compare models based on whether they are amenable to producing high-quality explanations. Chief among these problems is that faithfulness scores appear to be heavily-influenced by model initialization, undermining cross-model comparison (Zafar et al., 2021).

Specifically, recent work demonstrates that untrained models produce fidelity scores well above random masking, and the fidelity of the same interpretability method applied across different encoders can vary substantially (Zafar et al., 2021). This previous analysis left investigation of the root cause of these phenomena as future work. We shed light on this root cause by demonstrating that the iterative masking process produces samples outside the manifold of the training data, resulting in differing model-specific behaviour.

Producing reliable explanations in the text domain has been highlighted as difficult due to aberrant behaviour on iteratively masked samples (Feng et al., 2018b). By removing the least important word from a sequence iteratively, the resulting condensed explanation is no longer meaningful to human observers, impacting human-assessed plausibility. Our work complements this, highlighting how iterative masking affects representations within neural text classifiers and impacts automated assessment of faithfulness.

In our work, we demonstrate the mechanisms by which breakdowns in model comparison using faithfulness measures occur, and demonstrate that prioritizing favorable faithfulness measures between different models may present risks of undermining adversarial robustness and model fairness.

5.3.4 Text Classification Attack Benchmark

In the course of our analysis, we show that the impact of selecting models that produce high faithfulness scores may have impacts on adversarial robustness. As such, we introduce the concepts of adversarial attacks in the text domain, and a benchmark that contains a large number of computationally complex adversarial attacks applied to common Transformer classifier models.

Adversarial attacks — inputs perturbed to cause a model to produce an erroneous misclassification — can be applied in the text domain (Jin et al., 2020; Qiu et al., 2022). A common word-based approach is to replace words with synonyms, typically by using a synonym dictionary, or by leveraging another language model to find nearby words with nearby embeddings (Alzantot et al., 2018; Shi et al., 2019). The attacks DeepWordBug (Gao et al., 2018) and HotFlip (Ebrahimi et al., 2018) introduce targeted character-level perturbations to cause erroneous classifications.

As many text adversarial attacks are computationally complex and affected by random seeds, there is value in large-scale adversarial attack benchmarks on popular neural classifier architectures. Common attack benchmarks allow for researchers to analyze a larger number of successful adversarial attack samples, and perform reproducible analysis of these samples. The text classification attack benchmark (TCAB) (Asthana et al., 2022) contains neural text classifiers trained on benchmark tasks and successful adversarial attacks for each model.

The Transformer language models included in this benchmark reflect common architectures for contemporary neural text classifiers, and include bi-directional encoder representations from Transformers (BERT) (Devlin et al., 2019), as well as the widespread robustly-optimized variant of this architecture, Robustly Optimized BERT Approach (RoBERTa) (Liu et al., 2019).

The tasks included in the TCAB benchmark dataset include:

1. **SST-2:** Stanford Sentiment Treebank (Socher et al., 2013)
2. **Climate:** Twitter climate change sentiment (Qian, 2019)
3. **WikiToxic:** Wikipedia toxic comments (Dixon et al., 2018)
4. **Civil:** Jigsaw civil comments (Jigsaw, 2019)

5.3.5 Adversarial Robustness and Faithfulness

It has been shown that input feature attributions can be used as an effective feature for detection of adversarial attacks in the text domain (Huber et al., 2022). Previous work has investigated the impact of adversarial training — training on adversarial samples — on interpretability measures, and found that adversarial training appears to increase the faithfulness measures of models (Yoo and Qi, 2021) and alignment with ground truth explanations (Sadria et al., 2023). In the process of comparing faithfulness of neural text classifiers after adversarial training, one previous work found that RoBERTa models scored significantly lower than BERT models, and theorized that RoBERTa may be less interpretable than BERT (Yoo and Qi, 2021).

We demonstrate in our research that increased faithfulness measures are not universal in the presence of adversarial attacks. Further, our overall findings suggest that previously observed differences between BERT and RoBERTa faithfulness scores are not a meaningful reflection of each models inherent interpretability, but rather a result of model-specific behaviours in the presence of iterative masking. Finally, we highlight that there are cases where a high sensitivity to iterative masking is tantamount to an adversarial vulnerability, raising concerns that optimizing for high faithfulness may lead to increased adversarial vulnerability. These concerns also apply to another important measure for sensitive models: fairness.

5.3.6 Fairness in Transformer Models

Bias in masked language models has been observed with respect to gender, race, and religion (Nangia et al., 2020) as well as other qualities such as the language background of a writer (Crothers et al., 2019). In light of widespread usage of Transformer masked language models as neural text classifiers, improving model fairness is an important area of research. Neural network models may use certain tokens as shortcuts in making inferences (Geirhos et al., 2020), and tokens that carry an implication of a particular gender, race, or religion represent one possible shortcut. Efforts have been made to improve fairness in neural text classifiers. High-profile efforts in this space include Meta's perturbation-trained RoBERTa model called FairBERTa (Qian et al., 2022) and Google's counterfactual-trained model BERT-CDA (Webster et al., 2020). In our work, we demonstrate that there exist situations where favouring models with high faithfulness scores can come at the cost of fairness.

5.4 Motivation

As described in the introduction, our work demonstrates answers to two major branches of investigation: 1) "Why do different models respond differently to iterative masking?" and

2) “What impact does favouring models with high faithfulness scores have on other model attributes?”.

Our analysis continues important previous work by Zafar et al. (2021), where faithfulness measures based on iterative masking were shown to produce scores that are sensitive to model initialization, with identical models of the same architecture and comparable performance producing dramatically differing scores. First however, it is valuable to ground our discussion by walking through an example of iterative masking, and explain theoretical challenges in using iterative masking for measuring interpretability of neural text classifiers. We demonstrate this in this section by showing experimentally how model behaviour can vary across 8 models trained across 4 task datasets, motivating the experimental work in the following sections.

Figure 5.2 demonstrates an excerpt from a positive movie review, and the behaviour of a sentiment classifier on the sample. A fidelity calculation based on this example would give the worst possible score, despite the explanation identifying salient features. This example, and others like it, raise a simple question: “Does fidelity as a measure simply penalize models for not being sensitive to iterative masking?”.

In this example, when iteratively masked, at no point does the sentiment classifier determine that the movie review is negative. This is unsurprising — removing tokens does not invert the sentiment in this example, though it does eventually remove all sentiment information entirely (leaving the resulting predicted class up to model's initialization-specific tendencies). Integrated gradients feature attributions indicate which input tokens were most important to the output classification, ranking the tokens “beautiful”, “images”, and “solemn” as most important to the model's output. After masking these three words, the model's class confidence is at a minimum, but importantly, the output class is unchanged.

The point at which a classifier changes its predicted class (if at all), may vary substantially between models. Such variation does not just impact fidelity, but other faithfulness measures as well. While removing the top K tokens from the input, as in AOPC, samples become increasingly perturbed, and output probabilities may skew back towards the original class (as is the case in Figure 5.2). Area-based faithfulness measures such as AOPC are then similarly impacted as output probabilities on iteratively masked samples are not consistent across models, leading to variation in the $f(x) - f(x_{1..k})$ term (see Equation 5.2).

Under faithfulness measures that mask tokens until a change in predicted class, samples

Neg	Pos	Sample under iterative masking; iterative deletion
0.09	**99.1**	The beautiful images and solemn words
0.24	**97.6**	The [UNK] images and solemn words
3.6	**96.4**	The [UNK] [UNK] and solemn words
28.3	**71.7**	The [UNK] [UNK] and [UNK] words
14.0	**86.0**	[UNK] [UNK] [UNK] and [UNK] words
5.2	**94.8**	[UNK] [UNK] [UNK] [UNK] [UNK] words
1.4	**98.6**	The images and solemn words
3.6	**96.4**	The and solemn words
12.0	**88.0**	The and words
5.8	**94.2**	and words
5.5	**94.6**	words

Figure 5.2: Iterative token removal in descending order of feature importance on a sample from SST-2, a dataset of phrases from movie reviews paired with review sentiment. Despite the explanation identifying the most important tokens, the classification is unchanged during either iterative masking or iterative deletion.

that do not cause a change in predicted class during masking incur maximum penalty to the faithfulness score, regardless of the quality of the explanation. For methods that mask an *a priori* set number of tokens, model-specific output calibration similarly undermines comparison between models (Guo et al., 2017). Output logits on out-of-domain samples that include small numbers of tokens or empty strings are difficult to predict, and even if evaluated using a simple balanced binary classification dataset, there is no guarantee that this behavior will be consistent or symmetric when masking explanations for one predicted class versus the other.

The situation where the classification of a sample does not change during iterative masking is not uncommon, and not limited to this example. Table 5.1 shows the performance of 8 classifiers on clean (without adversarial perturbation) samples from the TCAB dataset. These models largely have comparable performance, with RoBERTa scoring higher on the climate-change dataset. However, substantial variation is observed across fidelity measures, highlighting the impact of both the model and the task dataset. Comparisons between BERT and RoBERTa may be impacted by model-specific attributes such as pre-training approach and hyperparameters (e.g., vocabulary size), though any larger trend in faithfulness is difficult to measure using pre-trained checkpoints due to heavy variation based on initialization even among the same model Zafar et al. (2021).

To understand why discrepancies arise, we consider the failure case that we demonstrated in Figure 5.2 — situations where the input class never changes. Table 5.2 shows the frequency of samples where iterative masking did not lead to a change in classification result at any point.

	SST-2	WikiToxic	Civil	Climate
BERT	91.0	90.3	83.6	65.0
RoBERTa	92.0	90.7	83.0	74.6
$\textbf{Fidelity}_{\textbf{BERT}}$	49.1	30.4	10.3	67.8
$\textbf{Fidelity}_{\textbf{RoBERTa}}$	45.5	11.8	5.0	72.2

Table 5.1: F1 scores (macro) of task-specific TCAB BERT and RoBERTa models on unperturbed validation set, and corresponding fidelity scores.

	SST-2	WikiToxic	Civil	Climate
BERT	35.0	9.0	88.0	5.5
RoBERTa	35.5	86.0	94.0	14.0

Table 5.2: Frequency of samples which did not result in change of predicted class at any point during masking of tokens based on feature importance.

The largest gap between models in both fidelity and frequency of samples where the predicted class was unchanged was observed on the Wikipedia Toxic Comments dataset. The TCAB BERT and RoBERTa models trained on different datasets have comparable performance on the original classification problem, shown in Table 5.1, yet the fidelity scores differ substantially. Table 5.2 provides deeper insight: the RoBERTa-wikipedia model did not change its result at any point during iterative masking on 86% of samples.

Fundamentally, the assumption that removing salient tokens should cause the output of a model to change is not intuitive for all datasets. The Wikipedia Toxic Comments dataset is an illustrative example of this. The class distribution of TCAB validation set for the two-class wikipedia dataset is 90.84% negative samples, i.e., mostly comments that are not considered toxic. Removing salient tokens one by one from an inoffensive comment is highly unlikely to create a true toxic sample, no matter how many tokens are removed.

As a result, a low fidelity score on this dataset arguably demonstrates a *beneficial* property — resilience against adversarial attack. The attack in this case being the removal of targeted salient tokens with the goal of changing the predicted class with the minimum number of removed tokens (even though the removal does not affect the true class). Whether removing salient tokens is expected to affect the true class is entirely dependent on the task and dataset — undermining the idea that iterative masking should be framed consistently across models.

These factors, when taken together with previous research showing substantial variation in faithfulness scores based on model initialization (Zafar et al., 2021), suggest that it may not be appropriate to use measures based on iterative masking to assess a text classifier's inherent interpretability. Such measures penalize robustness against perturbations, a characteristic which may be beneficial when considering the dataset. With this context in mind, we now proceed to the datasets and experimental setup that we use to pursue our major branches of investigation.

5.5 Datasets and Experimental Setup

To ensure a reproducible set of task-specific models and associated data samples, we use the text classification attack benchmark (TCAB) dataset and models (Asthana et al., 2022). This benchmark includes: 1) a number of established NLP task datasets; 2) BERT and RoBERTa models trained for each task; and 3) a variety of successful adversarial attacks against the included models.

To obtain a range of data domains and sequence lengths, we perform our experiments on the following task datasets in TCAB: 1) the Stanford Sentiment Treebank (SST-2) dataset of movie reviews for sentiment classification, a common NLP benchmark task (Socher et al., 2013); 2) the Twitter climate change sentiment dataset, a multi-class dataset of social media data (Qian, 2019); 3) Wikipedia Toxic Comments, a dataset with a longer sequence length widely-used for studying adversarial attacks (Dixon et al., 2018); and 4) the Civil Comments dataset, a dataset of online comments for evaluating unintended bias in toxicity detection, an area where interpretability may be important (Jigsaw, 2019).

The Transformer language models included in this benchmark reflect common architectures for contemporary neural text classifiers. We consider both a baseline bi-directional Transformer architecture BERT (Devlin et al., 2019), as well as the commonly used robustly-optimized variant of this architecture, RoBERTa (Liu et al., 2019).

In Section 5.7, we select a variety of adversarial attacks to analyze, including well-known word-level and character-level attacks to compare any differences in the impacts of adversarial training on fidelity. Specifically, we use DeepWordBug (Gao et al., 2018), TextFooler (Jin et al., 2020), Genetic (Alzantot et al., 2018), and HotFlip (Ebrahimi et al., 2018) attacks.

In Section 5.8, we compare basic versions of the RoBERTa and BERT architecture to fairness optimized versions in the form of FairBERTa (Qian et al., 2022) and BERT-CDA (Webster et al., 2020). Both of these fairness-optimized models represents a substantial effort by a major technology company to offer a fairness-optimized version of a masked Transformer language model.

As calculating input attributions is computationally expensive, fidelity calculations in Table 5.4 are based on a sample of 1,600 total records evenly split across dataset-model combinations. Initial experiments indicated this was sufficient to observe consistent patterns in fidelity scores. We use $N = 30$ for the number of steps for layer integrated gradient calculation, as this approximates the largest value that fits within the memory constraints of the system, and higher step counts typically produce more accurate explanations (Sundararajan et al., 2017). To evaluate models we use the macro F1 score, which is appropriate for use across TCAB task datasets that vary in terms of class distribution.

Experiments in Section 5.8 were performed on a compute cluster with an Intel Cascade Lake 32-core CPU, 32GB RAM, and 32GB VRAM NVIDIA V100 GPU. All other experiments were performed on a Windows workstation with an Intel i7-6800K 12-core CPU, 64GB RAM, and a 24GB VRAM NVIDIA GPU. Fixed seeds are used throughout experiments for reproducibility. Complete experiment code, along with references to utilized models and data are provided to ensure reproducible results.

5.6 Embeddings of Partially-Masked Samples

With the underlying mechanisms that undermine iterative masking for model comparison established, we now show that iteratively masked samples frequently fall outside the training distribution of the dataset, particularly at the levels of masking required for changes in predicted class. As samples from this distribution haven't been seen during training, this likely leads to the inconsistent behaviour on masked samples observed in the previous section.

5.6.1 Distributional Characteristics

Masking tokens causes the perturbed input samples to have representations that are very different from ordinary real samples in their training datasets. To demonstrate this, we generate embeddings for each input sequence across all samples taken from each dataset by mean-pooling the output of the second-last layer of a Transformer encoder, a common approach that generally outperforms using the final-layer `[CLS]` token embeddings (Devlin et al., 2019; Xiao, 2018).

Using this embedding method, we create embedding vectors $V_d = v_1, v_2, \ldots, v_k$ using the BERT and RoBERTa models for every sample across each dataset d within our sample. Each vector can be represented as $v_i = [x_{i1}, x_{i2}, \ldots, x_{in}]$. We then calculate centroids μ_d for each dataset

$$\mu_d = \left[\frac{\sum x_{i1}}{k}, \frac{\sum x_{i2}}{k}, \ldots, \frac{\sum x_{in}}{k} \right] \tag{5.3}$$

where $\sum x_{ij}$ represents the sum of the j-th component of all vectors in the set V_d, and the model's encoder layer dimension $n = 768$ is the length of the embedding vector. Iteratively masking tokens from each sample, we obtain an embedding vector ω at each step, and calculate cosine similarity as a scale-invariant vector similarity measure (Reimers and Gurevych, 2019):

$$cos\_sim = \frac{\sum_{i=1}^{n}(\omega[i] \cdot \mu[i])}{\sqrt{\sum_{i=1}^{n}(\omega[i])^2} \cdot \sqrt{\sum_{i=1}^{n}(\mu[i])^2}} \tag{5.4}$$

We then analyze the cosine similarity of centroids and the mean standard deviation of embedding vectors as the number of masked tokens increases, presenting the results in Figure 5.3, and comparing changes between models in Table 5.3.

From Table 5.3 and Figure 5.3, we can see that in all cases there is an increased cosine distance between the original dataset as tokens are masked. As more tokens are replaced with `[UNK]`, we see a smaller mean standard deviation across embedding features $\Delta\overline{\sigma}$, implying that the increased presence of a single token is leading to more homogenous representations overall. Importantly, we note substantial model-specific differences in the

	SST-2		WikiToxic		Civ. Com.		Clim. Cha.	
	$\Delta\mu$	$\Delta\overline{\sigma}$	$\Delta\mu$	$\Delta\overline{\sigma}$	$\Delta\mu$	$\Delta\overline{\sigma}$	$\Delta\mu$	$\Delta\overline{\sigma}$
BERT	0.15	-0.13	0.40	-0.04	0.30	-0.05	0.46	-0.06
RoBERTa	0.12	-0.16	0.15	-0.05	0.15	-0.08	0.34	-0.12

Table 5.3: Change in centroid $\Delta\mu$ and mean feature standard deviation $\Delta\overline{\sigma}$ at 50% mean sequence length tokens masked.

centroids of embeddings between unmasked and masked samples for BERT and RoBERTa. On average, the internal representations of Wikipedia Toxic Comments samples changed less during masking for RoBERTa than for BERT, which may explain the high frequency of samples that did not change predicted class previously observed in Table 5.2.

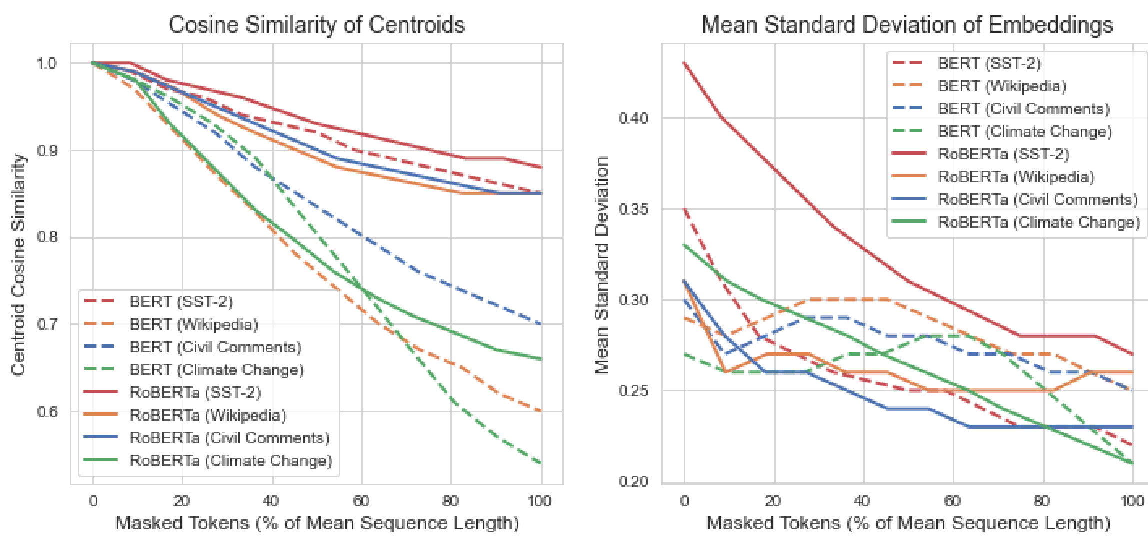


Figure 5.3: Comparison of centroid cosine similarity and mean standard deviation of embedding vectors between BERT and RoBERTa across various datasets. The left plot shows centroid cosine similarity, demonstrating the shift of data representations as tokens are masked. The right plot shows the mean standard deviation of the embeddings, showing representations of partially-masked inputs are less varied.

5.6.2 Local and Global Structure

In addition to distributional characteristics, we can demonstrate the difference in local and global structure between unmasked and partially-masked sample embeddings using UMAP

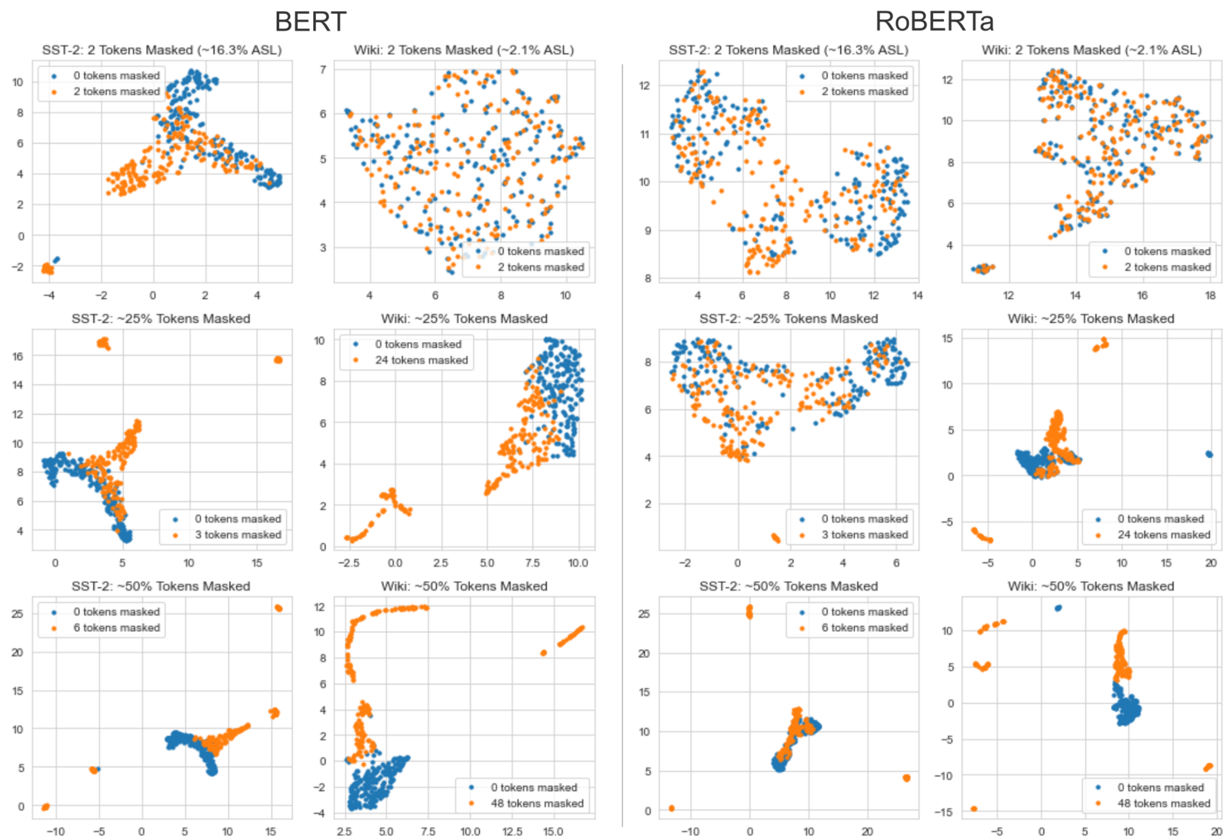


Figure 5.4: UMAP projections of sample embeddings at varying levels of masking. Masking more tokens moves the resulting embeddings further out of domain of the original dataset. Masking a couple tokens within a dataset with a longer average sequence length (ASL) has a relatively minor effect (e.g., see the Wikipedia Toxic Comments examples), but longer samples still generally require a significant portion of tokens to be masked to change classification (see Table 5.4)

dimensionality reduction (McInnes and Healy, 2018). We use the BERT and RoBERTa embeddings before and after masking some number of tokens as the base for these visualizations. We choose two illustrative datasets to visualize against: SST-2 and Wikipedia Toxic Comments. We select these two to show how the length of the input sample impacts per-token sensitivity to masking. Our sample from the former dataset has an average sequence length of 12.25 tokens, while the latter has an average sequence length of 97.22 tokens. Results from our dimensionality reduction can be seen in Figure 5.4.

We show that on datasets with short input samples such as SST-2, even just masking two salient tokens creates representations far outside the data manifold on which the model was trained, implying undefined behaviour. The per-token impact of masking is lesser on samples with longer average sequence lengths, as shown by the Wikipedia Toxic Comments visualizations, but these longer samples also require a larger number of tokens to be masked to change the model classification. For example, our results in Table 5.1 give a fidelity score of 0.304 for BERT on the Wikipedia dataset. This indicates that on average it is required to mask 69.6% of tokens from a Wikipedia sample before the output classification of the model changes. For RoBERTa, this proportion is even larger — a fidelity score of 0.118 on the dataset implies that 88.2% of tokens must be masked on average to perturb the classification result.

Based on the magnitude of the deviation, we infer that the departure of masked samples from the data manifold of the training set may be responsible for any prediction "crossover point", particularly for datasets where the true class is likely to be unchanged by masking (such as when masking non-toxic samples in Wikipedia Toxic Comments). Calculating faithfulness metrics using this approach thus relies heavily on undefined model-specific behaviour on degenerate samples from an unseen manifold — behaviour which is difficult to predict, and may vary dramatically between different models.

5.7 Fidelity Under Adversarial Attack

Adversarial training, the process of training a model on adversarial inputs to improve robustness against attacks, is a major area of research in NLP (Bai et al., 2021). Previous work has shown some small-scale experiments that suggest that adversarial training appears to improve faithfulness measures in neural text classifiers (Yoo and Qi, 2021). We perform a large-scale analysis that suggests that differences in fidelity after adversarial training or between different models do not follow easily discernible patterns, raising questions as to whether this pattern holds generally for neural text classifiers, or only under certain conditions.

For adversarial training, we perform hyperparameter tuning to determine appropriate combinations of learning rate $lr \in [10^{-6}, 10^{-3}]$, weight decay $wd \in [10^{-5}, 10^{-2}]$, and training epochs $ep \in [0, 5]$. Tuning of hyperparameters is required due to differences across task datasets and model architectures. Evaluation for hyperparameter tuning was performed by performing a 90-10 split of the TCAB adversarial attack "train" dataset, with each training set composed of half original task samples and half adversarial samples. The separate TCAB adversarial attack "val" dataset was used as the test dataset. A batch size of 16 was set for efficient training within GPU memory limits. We present findings based on the hyperparameter combination resulting in the best F1 score on the test dataset.

		BERT				RoBERTa			
		SST-2	WikiToxic	Civil	Climate	SST-2	WikiToxic	Civil	Climate
	Clean	49.1	30.4	10.3	67.8	45.5	11.8	5.0	72.2
4.1) Adv. Samples Pre Adv. Training	DeepWordBug	69.0	70.3	18.8	81.8	68.0	42.3	13.5	55.2
	TextFooler	87.9	74.1	34.1	80.6	77.4	53.3	19.2	73.1
	Genetic	79.8	78.9	31.8	87.1	56.7	51.1	20.1	51.9
	HotFlip	77.1	79.3	47.3	86.6	57.6	53.5	20.5	47.9
4.2) Adv. Samples Post Adv. Training	DeepWordBug	59.7	36.2	8.9	79.7	65.2	48.3	25.3	79.4
	TextFooler	77.5	47.5	80.7	70.8	74.2	52.0	20.1	77.2
	Genetic	66.2	86.0	75.2	83.4	54.5	57.9	40.0	86.5
	HotFlip	69.5	60.8	59.9	79.6	39.9	24.1	1.0	79.8
4.3) Clean Samples Post Adv. Training	DeepWordBug	49.9	64.3	80.5	65.8	49.2	51.4	84.6	72.4
	TextFooler	45.4	62.8	47.6	69.4	47.6	47.4	77.0	68.8
	Genetic	46.4	41.2	41.7	66.3	42.0	28.0	60.3	72.3
	HotFlip	47.9	62.4	32.1	66.7	47.7	61.3	86.9	69.7

Table 5.4: Fidelity scores of task-specific BERT and RoBERTa classifiers under varying adversarial attacks, and fidelity of 32 adversarially trained models for each dataset-model-attack combination on adversarial and non-adversarial (clean) samples.

In Table 5.4, we show fidelity calculations on 1) successful adversarial attacks prior to adversarial training; 2) adversarial samples after adversarial training; and 3) clean (non-adversarial) samples after adversarial training. We also include fidelity scores on clean samples prior to adversarial training, previously reported in Table 5.1, at the top of Table 5.4 for ease of comparison. Table 5.5 and 5.6 show the F1 scores before and after adversarial training respectively, confirming the training was successful at improving adversarial robustness.

From the results in Table 5.4.1, we can see that fidelity of explanations is generally higher on successful adversarial samples compared to clean samples. Adversarial attacks are typically optimized to minimize the number of perturbations, while still altering model output. These constraints lend themselves towards perturbed sequences where a small portion of tokens have a significant influence on predictions. Without an attack approach that also attacks model interpretability methods (Ivankay et al., 2022), these highly salient perturbed tokens are identified by the feature attributions, and masked early during fidelity calculation. As such, the model output often returns to the original value more quickly, leading to higher fidelity scores overall.

	BERT				RoBERTa			
	SST-2	WikiToxic	Civil	Climate	SST-2	WikiToxic	Civil	Climate
Clean	91.0	90.3	83.6	65.0	92.0	90.7	83.0	74.6
DeepWordBug	8.3	2.0	0.0	41.9	6.5	1.2	0.1	37.7
TextFooler	35.3	1.9	0.1	41.0	31.6	1.2	0.1	41.2
Genetic	47.1	5.8	4.0	29.3	48.5	5.4	1.7	19.7
HotFlip	47.7	6.0	5.0	33.3	48.6	6.8	2.2	24.2

Table 5.5: F1 scores on "clean" samples and various adversarial attack samples prior to adversarial training. Models are generally vulnerable to the provided adversarial attacks from the TCAB dataset.

	BERT				RoBERTa			
	SST-2	WikiToxic	Civil	Climate	SST-2	WikiToxic	Civil	Climate
DeepWordBug	55.3	48.3	48.5	59.9	51.1	49.1	48.6	55.3
TextFooler	62.1	49.3	47.6	67.1	69.1	49.4	47.8	63.4
Genetic	49.3	49.3	49.3	39.5	51.6	49.7	49.7	35.2
HotFlip	49.6	49.5	49.4	39.8	48.4	49.9	50.0	35.2

Table 5.6: F1 scores of adversarially trained models on adversarial samples. Adversarial training under the provided hyperparameter tuning regimen generally led to marked improvement in adversarial robustness.

The only model where adversarial attack samples did not universally manifest in increased fidelity scores, appears to be the RoBERTa climate-change model. In this case, the fidelity score on clean samples was already the highest of all included models, indicating that this model already relied on a relatively small number of salient tokens to make correct predictions. In this case, adversarial attacks may produce perturbations that interfere with these salient tokens, resulting in attributions spread more evenly across the remaining tokens.

We further note from Table 5.4.1 that fidelity scores of BERT models under successful adversarial attacks prior to adversarial training are generally higher than those calculated on RoBERTa. We view this result not as a proxy measure of interpretability, but instead as a measure of sensitivity to iterative masking — the behaviour which the metric directly measures. As we are working with a dataset of successful adversarial attack samples, the difference in fidelity score indicates that BERT models return to the original class after masking fewer salient tokens than RoBERTa models. That is, successful attacks on the TCAB BERT models depend on a comparatively smaller number of salient tokens. In framing fidelity this way, we demonstrate how faithfulness measures might be used to better understand how adversarial attacks impact salient tokens in neural text classifiers, rather than used as a proxy measure for explainability.

In many senses, iterative masking itself resembles a simple token-level adversarial attack. Instead of replacing a salient token with an equivalent that causes a change in predicted class, salient tokens are removed or masked until a change in predicted class occurs. As long as the dataset is such that token removal does not influence the true class, this meets the definition of an adversarial attack (Jin et al., 2020). Such an attack would noticeably degrade the original sentence, though reduction in text quality has been observed for other word-level and character-level attacks as well (Crothers et al., 2022a). Figure 5.5 demonstrates an AGNews sample where both adversarial attack and saliency-based masking target the same token ("vote") and similarly alter the predicted class.

Taken together, Tables 5.4.2 and 5.4.3, which show fidelity after adversarial training on adversarial samples and clean samples respectively, demonstrate at scale that adversarial training of neural text classifiers does not appear to have a consistent impact on fidelity scores across different datasets. The observed fidelity gaps can be very large, even for the same encoder, such as the BERT results for the Civil Comments dataset. From this, we conclude that training neural text classifiers on adversarial samples does not have a straightforward relationship with sensitivity to iterative masking, despite previous indications to the contrary.

Sample	True Class	Predicted Class	Token Importance (Attribution Target = True Class)
Clean	World News	World News	[CLS] venezuela prepares for chavez recall **vote** supporters and rivals warn of possible fraud ; government says chavez ' s defeat could produce turmoil in world oil market . [SEP]
TextFooler	World News	Business News	[CLS] venezuela prepares for chavez recall **election** supporters and rivals warn of possible fraud ; government says chavez ' s defeat could produce turmoil in world oil market . [SEP]
Masked	World News	Business News	[CLS] venezuela prepares for chavez recall **[UNK]** supporters and rivals warn of possible fraud ; government says chavez ' s defeat could produce turmoil in world oil market . [SEP]

Figure 5.5: Word-level adversarial attack compared to iterative masking on AGNews sample on TCAB BERT classifier. Attributions shown for 1) clean sample, 2) TextFooler adversarial attack, and 3) iteratively masked sample. Perturbed and masked tokens shown in bold. Both adversarial attack and iterative masking perturb the predicted class after manipulating a single token.

5.8 Fairness and Faithfulness

In the previous section, we noted that faithfulness measures ascribe positive characteristics to models that are maximally vulnerable to a salience-based adversarial attack where tokens are removed one-by-one, even in cases where such action would not change the true class. In this section, we show that fairness-optimized models can produce lower faithfulness scores than models without such optimizations. In many real-world scenarios where explainability is desirable, fairness is also a critical consideration. The mixed results of faithfulness scores on fairness-optimized models emphasize caution when viewing sensitivity to iterative masking as a positive quality related to explainability in model comparisons, particularly in light of model-specific and task-specific conditions undermining such comparisons.

5.8.1 Fairness-Optimized Transformers

Research in improved model fairness has produced Transformer language models that are designed to provide improvements in fairness. Publicly-released models for this purpose have included Google-developed Transformer models trained from scratch using counterfactual data augmentation (CDA) and variants with increased dropout for improved fairness (Webster et al., 2020). Similarly, research by Facebook AI Research produced a model based on the RoBERTa-base architecture called FairBERTa. FairBERTa was trained on a training dataset that was perturbed to improve fairness in samples (Qian et al., 2022). In this section, we show that faithfulness measures calculated on fine-tuned classifiers using fairness optimized models appear to have similar sensitivities to iterative masking after being tuned on a particular downstream task. As a result, the fairness-optimized FairBERTa model scores consistently lower on faithfulness scores than the basic RoBERTa model, while still offering comparable performance on classification benchmark tasks.

In this section, we compare fairness scores and faithfulness measures on basic and fairness-optimized neural text classifiers. To do so, we fine-tune these models on 5 GLUE classification tasks (Wang et al., 2018) and calculate fidelity as in the previous sections to measure their sensitivity to iterative masking. As in previous fairness research using GLUE datasets, we limit our study to the tasks which contain samples that may be affected by model fairness (Qian et al., 2022).

5.8.2 Experiment Settings

Model Selection

To demonstrate that care is required in comparing models using faithfulness, we compare basic versions of the RoBERTa and BERT architecture to fairness optimized versions in the form of FairBERTa and BERT-CDA. Both of these fairness-optimized models represents a substantial effort by a major technology company to offer a fairness-optimized version of a masked Transformer language model. All of these models are publicly available for straightforward replication of results.

Both Google's BERT Large uncased model and BERT-CDA model use the same architecture, with the latter optimized for fairness using Counterfactual Data Augmentation (CDA) (Webster et al., 2020).

Hyperparameter	Values
Train batch size	{16, 32}
Learning rate	1e-6 to 1e-4 (log scale)
Epochs	1 to 5 (integer)
Optimizer	Adam
Adam β_1	0.9
Adam β_2	0.999
Adam ϵ	1e-08
Learning rate decay	Linear

Table 5.7: Hyperparameter settings for fine-tuning FairBERTa and RoBERTa on GLUE tasks

Base models of RoBERTa (Liu et al., 2019) and FairBERTa (Qian et al., 2022) share an architecture, but the latter differs by use of fairness-aware perturbation during pre-training. FairBERTa performs comparably to RoBERTa on language model benchmark tasks, while outperforming on fairness measures (Qian et al., 2022).

Now, to demonstrate the challenges when comparing models using faithfulness rather than fairness, we evaluate the faithfulness of the "more-fair" FairBERTa. When evaluating faithfulness, we use the same method as in the previous section, iteratively masking features until the classification result changes.

Fine-tuning Training Procedure

In order to compare faithfulness is the context of neural text classifiers, we fine-tune these base models on the previously described 5 GLUE benchmark classification tasks, and compare fairness scores and faithfulness measures on the resulting neural text classifiers (Wang et al., 2018). We then calculate fidelity as in the previous sections to measure their sensitivity to iterative masking. The 5 tasks selected are those which contain a sufficient number of samples for comparison of fairness performance (Qian et al., 2022).

The resulting RoBERTa models are compared against the corresponding FairBERTa models. Similarly, the BERT large uncased model is compared against the BERT-CDA variant.

	CoLA	SST-2	QQP	RTE	QNLI	Average
RoBERTa	**67.6**	**65.4**	**58.7**	**73.4**	**61.7**	**68.3**
FairBERTa	65.2	63.8	54.4	60.8	58.5	63.9
BERT	69.1	56.6	19.9	40.0	73.6	43.9
BERT-CDA	**71.8**	**68.9**	**31.4**	**69.1**	**61.3**	**60.5**

Table 5.8: Fidelity score of fine-tuned models on GLUE tasks. Higher score in bold. All five fine-tuned FairBERTa models are slower to alter predictions during iterative masking than their fine-tuned RoBERTa counterparts. Conversely, all five fine-tuned BERT-CDA models score higher than the basic BERT model.

5.8.3 Fairness Results

Overall, we find that there are situations where models derived from a fairness-optimized model may consistently score lower in faithfulness than architecturally identical unoptimized models (Table 5.8), even while providing comparable performance on GLUE benchmark tasks (Table 5.9). Previous research has determined that model initialization has a severe impact on the faithfulness scores produced by a model (Zafar et al., 2021). Based on this previous research, we determine that the publicly-available BERT-CDA and FairBERTa checkpoints likely include opposing initialization-specific tendencies: the BERT-CDA checkpoint inverts prediction quickly under iterative masking, while the FairBERTa model responds more slowly. All fine-tuned models trained from a FairBERTa score lower than their fine-tuned RoBERTa equivalents. Conversely, all models trained from BERT-CDA consistently score higher in faithfulness than the base BERT model. Because the behaviour on masked samples is maintained even after fine-tuning, depending on the characteristics of the base model checkpoint, all descendant models will inherit a tendency when evaluated for faithfulness. As a result, in light of the challenges in faithfulness-based model demonstrated by our work and others, such measures open the possibility of selecting of models that are less fair, while not producing any real advantage in explainability.

As shown in Table 5.9, fine-tuned FairBERTa and RoBERTa model produce comparable performance on GLUE benchmarks under simple hyperparameter tuning. However, with respect to faithfulness scores, the results in Table 5.8 show that FairBERTa consistently produces lower scores than RoBERTa on all 6 task-specific fine-tuned models. The robustness of FairBERTa to iterative masking may be due to perturbation during the pre-

	CoLA	SST-2	QQP	RTE	QNLI
RoBERTa	**61.9**	**96.0**	89.1	69.1	**92.7**
FairBERTa	56.8	95.4	**89.3**	**71.0**	92.5
BERT	61.6	**93.7**	**91.6**	**71.8**	**92.5**
BERT-CDA	**62.6**	92.5	91.2	60.6	91.3

Table 5.9: GLUE task performance of FairBERTa and RoBERTa models. Matthew's correlation coefficient is provided for CoLA. Pearson correlation is provided for STS-B. Accuracy given for all other tasks. Best result in bold.

training process, or the result of initialization, but in either case, the results faithfulness scores are consistently lower for this fairness-optimized model.

By calculating the variance in input attributions on RoBERTa and FairBERTa, shown in Table 5.12 we find that FairBERTa input attributions demonstrate overall less variance in feature importance stores. This indicates that the feature importance scores of FairBERTa are more evenly distributed among the input tokens than RoBERTa. Greater reliance on a smaller set of more salient features aligns with RoBERTa's observed relative sensitivity to iterative masking (and thus, higher faithfulness scores).

To confirm the fairness impact of FairBERTa's training method, we evaluate each model against 2 well-established intrinsic measures of model fairness: CrowS-Pairs (Nangia et al., 2020) and SEAT (May et al., 2019). Using SEAT (May et al., 2019), we calculate and report bias along gender, race, and religion as in Meade et al. (2022). We note that previous work analyzing a range of bias benchmark datasets, including CrowS-Pairs, found that these datasets have limitations and can produce noisy results (Blodgett et al., 2021; May et al., 2019; Meade et al., 2022; Qian et al., 2022). However, by using methods that are well-established and easily reproducible, we confirm that these fairness-optimized models measure improvements in these areas. Previous work on FairBERTa has demonstrated a nuanced evaluation of these models strengths and weaknesses (Qian et al., 2022).

We illustrate in Figure 5.6 an example of a sentiment classification sample where RoBERTa changes prediction after a single prediction, while FairBERTa classification remains the same. After removing the first part of the word "likable", the RoBERTa model no longer detects the overall sentiment of the review phrase appears positive. However, the FairBERTa model continues to provide a positive classification, seemingly inferring that

	CoLA	SST-2	QQP	RTE	QNLI	Average
RoBERTa	1.193	0.271	0.290	0.214	0.308	0.407
FairBERTa	**0.455**	**0.235**	**0.226**	**0.176**	**0.292**	**0.255**
BERT	**1.462**	**0.860**	0.783	**0.565**	**0.423**	**0.819**
BERT-CDA	4.597	2.540	**0.719**	0.923	0.625	1.881

Table 5.10: Average variance σ^2 of input attributions on each fine-tuned classification model. Lower variance marked in bold.

the presence of the phrase “still oddly” implies that the overall review sentiment is likely still positive. In both cases, the behaviour of the classifier appears to be both reasonable and explainable based on the input feature attributions that are observed. Considering the RoBERTa example as “more explainable” based on the classifier shifting prediction after removing half of a word promotes a notion of explainability that rewards models that invert classification whenever an `[UNK]` token is present. Figure 5.7 shows the differing layer-wise attribution scores on these samples using layer conductance.

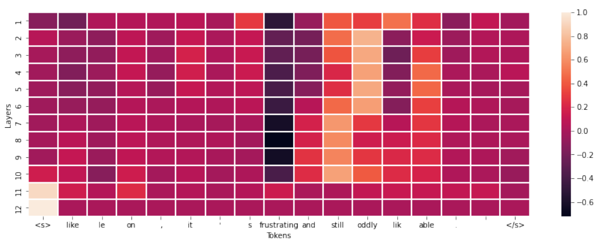
In all, these findings suggest that fairness-optimized models derived from FairBERTa or BERT-CDA manifest tendencies in their response to iterative masking, which affect the faithfulness scores of fine-tuned models based on these. As a result, models may be unfairly rewarded or penalized based on a metric which is of questionable appropriateness in the text classification setting. Taking this finding in the context of the rest of our investigation, we now provide an overall discussion of the challenges in using faithfulness for comparing neural text classifiers.

		RoBERTa	FairBERTa
CrowS-Pairs	Gender	60.15	**56.70**
	Race	63.57	**50.78**
	Religion	60.00	**50.48**
WEAT/SEAT	% Significant Tests ↓	**42.0**	**42.0**
	Avg. Effect Magnitude ↓	0.675	**0.623**

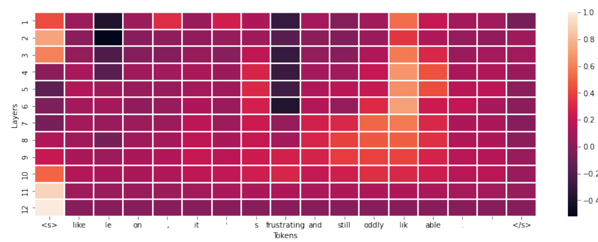
Table 5.11: Summary of fairness evaluation on FairBERTa vs RoBERTa. For CrowS-Pairs, values closer to 50 indicate reduced bias. Additional results including extrinsic evaluation can be found in Qian et al. (2022)

Model	True	Pred.	Token Importance (Attribution Target = True Class)
RoBERTa	Pos.	Pos.	#s like le on , it ’ s frustrating and still oddly **lik** able . #/s
FairBERTa	Pos.	Pos.	#s like le on , it ’ s frustrating and still oddly **lik** able . #/s
RoBERTa	Pos.	Neg.	#s like le on , it ’ s frustrating and still oddly **#unk** able . #/s
FairBERTa	Pos.	Pos.	#s like le on , it ’ s frustrating and still oddly **#unk** able . #/s

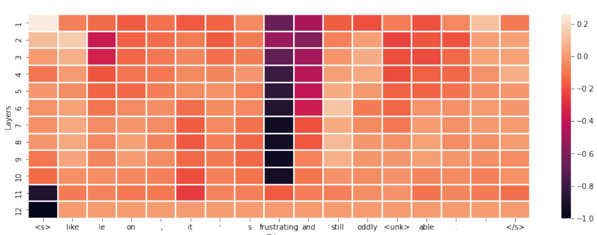
Figure 5.6: Example of FairBERTa and RoBERTa on GLUE SST-2 sentiment classification. Attributions shown for unmasked and iteratively masked sample for both models. Masked tokens shown in bold. Masking a single token causes a classification change for the RoBERTa classifier.



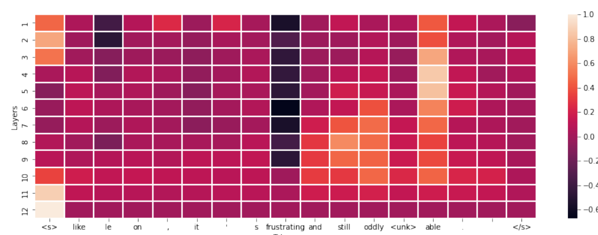
(a) RoBERTa: Unmasked Sample



(b) FairBERTa: Unmasked Sample



(c) RoBERTa: Masked Sample



(d) FairBERTa: Masked Sample

Figure 5.7: Layer-wise attribution scores using layer conductance (Dhamdhere et al., 2018; Shrikumar et al., 2018) of RoBERTa and FairBERTa on masked and unmasked samples from Figure 5.6. Positive direction is predicted class.

5.9 Fairness Evaluation on Finetuned Models

Following are the results of calculating SEAT (May et al., 2019) fairness metrics on FairBERTa and RoBERTa models following fine-tuning. As in previous work, we note that such measures exhibit noise (Aribandi et al., 2021), and measuring fairness using intrinsic methods for masked language models following fine-tuning on GLUE tasks appears to increase the randomness of the resulting behaviour on these intrinsic measures. These results imply that there may be value in using fairness-preserving fine-tuning procedures, such as "fairtuning" on perturbed data (Qian et al., 2022) to preserve fairness advantages through pretraining.

		CoLA	SST-2	QQP	QNLI	RTE	STS-B	Avg.
Gender	FairBERTa	52.49	45.59	57.47	43.68	52.49	46.36	**49.68**
	RoBERTa	43.68	46.36	44.44	50.19	54.41	57.85	49.49
Race	FairBERTa	58.72	51.74	32.75	26.36	27.91	60.85	43.06
	RoBERTa	42.25	55.62	69.57	38.95	41.09	64.15	**51.94**
Religion	FairBERTa	49.52	62.86	31.43	50.48	73.33	31.43	**49.84**
	RoBERTa	27.62	35.24	62.86	51.43	46.67	32.38	42.70
Column Avg.	FairBERTa	53.58	53.40	40.55	40.17	51.24	46.21	47.53
	RoBERTa	37.85	45.74	58.96	46.86	47.39	51.46	**48.04**

Table 5.12: CrowS-Pairs scores for fine-tuned FairBERTa and RoBERTa models on different bias categories. A perfect score is 50. Averages are taken by fine-tuning task and for each column, including an aggregate mean equally weighting each of the three bias categories (bottom right). As observed in previous research, results from CrowS-Pairs appear noisy (Aribandi et al., 2021; Meade et al., 2022; Qian et al., 2022).

Task	Model	ABW	ABW-B	SEAT-3	SEAT-3B	SEAT-4	SEAT-5	SEAT-5B	Avg.
CoLA	FairBERTa	-0.266	1.161	-0.185	-0.082	-0.229	0.383*	0.128	**0.348**
	RoBERTa	-0.198	-0.884	0.374*	-0.007	0.288	0.619*	0.184	0.365
QNLI	FairBERTa	1.027*	1.317	0.129	0.161	0.128	-0.191	0.080	0.433
	RoBERTa	-0.610	-0.224	-0.006	0.007	-0.101	0.018	0.097	**0.152**
QQP	FairBERTa	-0.092	1.105	0.772*	0.389*	0.645*	0.644*	0.394*	0.577
	RoBERTa	0.239	0.705	-0.488	-0.656	-0.611	0.608*	0.656*	**0.566**
RTE	FairBERTa	-0.569	0.618	-0.638	0.470*	-0.327	0.406*	-0.376	**0.486**
	RoBERTa	0.086	0.543	-0.676	0.487*	-0.545	-0.555	0.513*	**0.486**
SST-2	FairBERTa	1.225*	0.326	0.298*	0.203	0.383*	0.410*	0.194	**0.434**
	RoBERTa	0.768	1.342	0.507*	0.119	0.587*	0.642*	0.135	0.586
STS-B	FairBERTa	-0.698	0.607	0.838*	0.283*	0.926*	0.842*	0.368*	0.652
	RoBERTa	1.150*	1.177	0.125	-0.273	0.134	0.758*	0.332*	**0.564**

Table 5.13: Fine-grained SEAT results on racial bias tests for fine-tuned FairBERTa and RoBERTa models. Significant results marked with asterisks. “Avg.” column shows average magnitude of effect size.

Task	Model	SEAT-6	SEAT-6B	SEAT-7	SEAT-7B	SEAT-8	SEAT-8B	Avg.
CoLA	FairBERTa	0.671*	0.050	0.950*	1.164*	0.731*	1.004*	0.762
	RoBERTa	-0.062	0.015	-0.280	0.226	0.109	0.238	**0.155**
QNLI	FairBERTa	0.042	0.382*	0.695*	-0.249	0.220	0.274	**0.310**
	RoBERTa	0.017	-0.122	0.676*	0.821*	0.191	0.224	0.342
QQP	FairBERTa	0.037	-0.239	0.199	-0.187	0.601*	-0.426	**0.282**
	RoBERTa	0.078	-0.071	-0.280	0.188	-0.708	0.522*	0.308
RTE	FairBERTa	0.368	0.164	0.797*	0.850*	0.147	0.053	0.396
	RoBERTa	0.492*	0.033	-0.484	0.319	-0.075	-0.340	**0.290**
SST-2	FairBERTa	0.118	-0.010	-0.194	1.096*	-0.282	0.186	**0.314**
	RoBERTa	0.352	0.066	1.355*	1.398*	0.493*	1.378*	0.840
STS-B	FairBERTa	0.449*	0.092	0.587*	-0.271	0.459*	-0.430	**0.381**
	RoBERTa	0.940*	0.358	1.271*	1.361*	0.532*	0.959*	0.903

Table 5.14: Fine-grained SEAT results on gender bias tests for fine-tuned FairBERTa and RoBERTa models. Significant results marked with asterisks. “Avg.” column shows average magnitude of effect size.

Task	Model	Religion-1	Religion-1B	Religion-2	Religion-2B	Avg.
CoLA	FairBERTa	-0.141	-0.393	-0.280	-0.361	0.294
	RoBERTa	-0.153	-0.384	0.028	0.386*	**0.238**
QNLI	FairBERTa	-0.073	0.063	0.127	0.195	**0.114**
	RoBERTa	0.089	0.266	0.090	0.232	0.169
QQP	FairBERTa	0.724*	0.529*	0.766*	0.414*	0.608
	RoBERTa	-0.052	-0.321	-0.052	-0.321	**0.186**
RTE	FairBERTa	0.094	-0.268	0.560*	-0.172	**0.274**
	RoBERTa	-0.574	-1.118	-0.574	-0.988	0.814
SST-2	FairBERTa	0.372*	0.295	0.384*	0.315	0.342
	RoBERTa	0.408*	0.073	0.411*	0.069	**0.240**
STS-B	FairBERTa	-0.709	-0.769	0.676*	0.799*	0.738
	RoBERTa	-0.665	-0.503	0.340	0.278	**0.447**

Table 5.15: Fine-grained SEAT results on religion bias tests for fine-tuned FairBERTa and RoBERTa models. Significant results marked with asterisks. “Avg.” column shows average magnitude of effect size.

5.10 Discussion

Within this work, we have demonstrated that comparing models based on masking-based "explainability" metrics reflects misleading assumptions that regularly do not hold on real datasets, and the resulting scores are heavily influenced by the problem domain and model initialization. We've further shown that the relationship between such metrics and adversarial robustness is not straightforward, and that looking negatively on models for robustness to iterative masking is in some domains effectively rewarding a weakness to masking-based adversarial attacks. Finally, we have shown that masking-based metrics may push users towards models that are less fair, for example by favouring RoBERTa over FairBERTa based on initialization. Fundamentally, masking tokens in a text sequence and measuring the change in output probabilities is explicitly a measure of "robustness to iterative masking", and great caution should be taken before making further assumptions beyond this.

In analyzing the iterative-masking mechanism by which faithfulness is applied to neural text classifiers, we propose that this property is more accurately described as "sensitivity to iterative masking". Models that score high on these measures produce more significant perturbations in the presence of fewer masked tokens — a property that depends on initialization-specific behaviour on samples that are increasingly outside the training data manifold as tokens are masked. Further, sensitivity to iterative masking can be framed as akin to an adversarial attack in cases where the removal of a salient word should not change the true class of a sample, adding a strong task-specific element to its consideration as well.

5.10.1 Recommendations

Based on these findings, we caution against interpreting sensitivity to iterative masking as a positive indicator of "interpretability" when comparing different neural text classification models. In the case of text classification, such measures may spuriously differentiate models based on their response to out-of-domain samples, and in some cases may favour models that are less adversarially robust or less fair. When faced with the question of "how do we compare the interpretability of neural text classifiers", we provide 3 main practical recommendations:

1. Avoid comparing different models using methods that assume each model will respond consistently on partially masked samples. Such responses appear initialization-specific, and masked samples fall further outside of the manifold of model's training data as tokens are masked.

2. Take into account the task dataset on which the model is being evaluated. For some tasks, removing tokens may not change the true class of a sample, and therefore may be framed as a form of adversarial attack. In these cases, robustness to token removal may be a desirable quality related to adversarial robustness.

3. Consider the trade-offs of a model relying heavily on a small number of salient input tokens to make decisions. Making decisions based on the minimal number of tokens may be less important than other considerations (such as adversarial robustness or fairness).

Quantifying interpretability of neural text classifiers remains difficult. The nature of language (i.e., discrete tokens with important interrelationships) means that any method that relies on removal of tokens is likely to be heavily influenced by behaviour on out-of-domain samples. As such, developing a robust way to quantify interpretability of neural text classifiers — and under what circumstances such a numerical measure would be meaningful — remains an open area for continued research.

In practice, the interpretability of neural text classifier models that utilize the same architectures and explanation methods may be broadly comparable from a human-centric perspective. Regardless of faithfulness scores, each Transformer classifier can produce input feature attributions or attention head explanations that a human can review to better understand model behaviour. Current widespread faithfulness measures on neural text classifiers, including infidelity and AOPC, reward models that dramatically change in output after masking a minimal number of tokens. On real samples, however, it may be correct to expect that even after masking several tokens a classification should be minimally affected. Maximizing sensitivity to iterative masking is unlikely to be the most effective way to improve interpretability. For practical applications where human oversight is critical, interpretability is heavily impacted by what forms of explanations the selected type of machine learning model can produce and how these can be meaningfully presented to humans (Lakkaraju et al., 2022). After selecting an appropriate model architecture that can produce meaningful explanations, optimizing for fairness and adversarial robustness should be prioritized based on the relative ethical and security considerations in play.

5.10.2 Study Limitations

This section outlines the limitations of the study in this chapter. Despite these constraints, our research significantly advances the understanding of faithfulness measures in neural text classification, and brings much-needed clarity to appropriate application and interpretation of iterative masking in model evaluation. Furthermore, our analysis of the relationship between fairness, adversarial robustness, and faithfulness opens up new avenues for exploring the complex interplay between model performance, interpretability, and ethical considerations.

Emphasis on Bidirectional Transformer Classifiers

This research focuses on classifiers based on Transformer encoder architectures pre-trained on a masked language modelling objective. These architectures are in widespread use for neural text classification, and reflect the models used in previous research analyzing faithfulness in neural text classification (Madsen et al., 2022; Yoo and Qi, 2021; Zafar et al., 2021). An additional advantage of these models is the availability of publicly available datasets of computationally expensive adversarial attacks to produce reproducible analysis related to adversarial attacks (Asthana et al., 2022). Using these models, we demonstrate the shortcomings of faithfulness-based model comparisons on well-established architectures, and clearly illustrate the theoretical pitfalls with reasoned arguments and illustrative examples.

Faithfulness measures may be affected by a variety of attributes, such as data quality, model architectures, initialization strategies, and training parameters. However, we do not perform any kind of "faithfulness benchmark" across a wide variety of architectures and other model characteristics. Based on our own findings and prior research (Madsen et al., 2022; Zafar et al., 2021), comparison of architectures may be difficult due to the heavy variation caused by initialization alone. Beyond this, our research suggests that model behaviour on masked samples may not reflect model behaviour on real samples, raising questions about the utility of comparing model architectures based on faithfulness measures. Regardless, researchers working with various model architectures may be interested in understanding how architectures of different design or scale behave on partially masked samples, and such an investigation may be an area of future work.

Limited Large-Scale Text Attack Benchmarks

Within this research, we have leveraged the text classification attack benchmark (TCAB) (Asthana et al., 2022). This benchmark provides a large dataset of text-based adversarial attacks. As many of these attacks are computationally expensive, it is beneficial to have a large-scale benchmark dataset on which to build future reproducible research. However, there are limitations to existing text-attack benchmark datasets, which are limited in model and attack variety. While inclusion of a novel model or attack within a benchmark dataset will always require some time, as text attack frameworks such as TextAttack (Morris et al., 2020) mature, we expect it will become easier to produce improved benchmark datasets that incorporate a wider range of both models and attacks.

Variety of Fairness-Optimized Models

The analysis in the final section of this paper demonstrated that FairBERTa, a fairness-aware variant of the RoBERTa model, consistently scores lower on faithfulness scores that the basic RoBERTa variant. An investigation of the subtopic of how fairness training influences input attributions should analyze a wider variety of models, both with and without fairness optimizations during training. As the process of fully pre-training a large Transformer is very expensive and time-consuming, this type of research will become easier as more robust fairness-optimized model variants become publicly available. While the case regarding the pitfalls of using faithfulness as a positive quality can be raised with a focused comparison, a full-scale benchmark could include the relationship between faithfulness and fairness including a variety of different methods of debiasing models, such as CDA Zmigrod et al. (2019) and dropout for bias mitigation Webster et al. (2020). Note that there are also other post-hoc methods such as "Self-Debias" (Schick et al., 2021), which has strong performance in benchmarks but which cannot be used when fine-tuning on classification tasks (Meade et al., 2022).

Fairness Assessment

Analyzing the relationship between fairness and robustness to iterative masking has required us to evaluate the fairness of language models, which is inherently a challenging topic. As noted by Qian et al. (2022), we highlight that existing fairness measures have well-documented pitfalls of their own (Blodgett et al., 2021), and do not always agree with one another (Cao et al., 2022; Goldfarb-Tarrant et al., 2021). In selecting methods to best assess fairness, we have selected well-documented measures such as CrowS-Pairs (Nangia et al., 2020) and WEAT/SEAT (May et al., 2019) that have been subject to considerable previous analysis and usage in studying language models. This gives us a greater degree of confidence that there are not unknown limitations to these methods of assessment, and ensures that the results can be easily cross-referenced or compared with other research. Finally, it is important to note that we have analyzed fairness focusing on biases predominantly relevant to a North American audience.

5.11 Summary

Overall our findings suggest that comparing neural text classifiers using masking-based faithfulness measures as a proxy for interpretability carries significant risks, even if models have the same or similar architectures. Measurements of faithfulness based on iterative masking are dependent on model-specific behaviour, and partially masked samples are often well outside the data manifold of the original training data. Model comparisons based on responses to iterative masking should be considered with nuance and performed carefully.

Based on our research, we have noted that successful text adversarial attacks result in highly salient features, and often result in increased fidelity scores on successful adversarial attacks. This aligns with previous research that has used similar features for detecting adversarial attacks (Huber et al., 2022). Beyond this observation, our findings indicate that adversarial training of neural text classifiers does not have a consistent impact on fidelity scores, suggesting that the relationship between adversarial training and robustness to iterative masking is less direct than previously thought.

With respect to fairness, we observe that fairness-optimized models can produce lower faithfulness score lower than non-optimized models, while offering comparable task performance. Based on our own results, we question the appropriateness of framing faithfulness as an interpretability measure for neural text classifiers, given that the models under comparison can similarly produce input feature attributions, with faithfulness instead reflecting their sensitivity to the iterative masking process.

Significant future work exists in the areas of neural text classifier interpretability, adversarial robustness, and fairness. In the final chapter, we now outline our overall conclusions and future areas of investigation.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The work presented within this thesis centered around problems that must be addressed in order for the strong capabilities of LLMs to be realized, making progress towards reducing the risks of generative LLM abuse, improving the adversarial robustness of neural text classifiers for LLM detection, and better understanding the difference between computational and human assessment of generative LLM output. Towards these ends, we presented 1) a comprehensive survey of machine generated text risks and detection presented at a venue sponsored by the United Nations (Crothers et al., 2023a), 2) an analysis of the adversarial robustness of neural and statistical features for detection of machine generated text, published at IJCNN (Crothers et al., 2022a), 3) an evaluation of sampling parameters on subjective and computational assessment of machine generated text, published at a AAAI workshop (Crothers et al., 2023b), and 4) a detailed analysis of pitfalls using common interpretability measures in LLM-based text classifiers, presented at LOD 2024 (Crothers et al., 2024) and submitted in substantially extended form as a journal article under review for the journal Computational Intelligence.

The contributions of this thesis are significant. By facing critical issues in LLM abuse prevention, enhancing the resilience of detection systems, understanding the gap between computational and human evaluation of LLM outputs, and shedding new light on interpretability measures, this work advances the creation of safer and more reliable AI systems. First, the research on the threat models associated with machine generated text allows for stronger defense mechanisms to be designed to protect against abuse of generative AI text models, as well as serving as a vital survey for understanding the available methods for machine-generated text detection. Second, we demonstrate that statistical features can augment adversarial robustness of machine-generated text detection, provides the basis for more resilient detection of machine-generated text, while also highlighting that such approaches make things harder for an attacker by reducing the coherence of their adversarial text. Third, our analysis of how sampling parameters affect computational and subjective qualities associated with machine-generated text provide new insight into how machine-generated text compares to human text along a more nuanced range of comparisons, and provides a basis for future human-focused research into the different attributes of machine-generated text. Finally, our work on neural text classifier interpretability provides tangible guidance for those trying to explain predictions, such as in detection of machine-generated text, and cautions AI practitioners against pitfalls in terms of evaluating the interpretability of neural text classifier systems.

Together, the insights gained from the work presented within this thesis provide valuable tools and methodologies for machine learning researchers and practitioners. As the field of large language model research continues to evolve rapidly, this work serves as a foundation for critical future work across the areas of security, transparency, and fairness. In the remainder of this chapter, we will explore the limitations of this research, as well as potential avenues for future work.

6.2 Limitations and Challenges

The research in this thesis comes against the backdrop of immense change across the fields of natural language processing and machine learning. Transformer language models and machine-generated text has become increasingly prominent over the last several years, with generative AI interfaces for large language models providing greater numbers of individuals easy access to increasingly capable models. The research within this thesis was influenced by the observation that there remain significant gaps in understanding large language models, their robustness, and their impacts on society. For example, the work in Chapter 5 on the pitfalls of iterative masking measures followed naturally after an early investigation into the relationship between adversarial attacks and neural text classifier interpretability surfaced the more fundamental and under-explored problem of the lack of robust model interpretability measures for neural text classifiers (Zafar et al., 2021).

Within this section we outline some of the limitations and challenges encountered during the research presented within this thesis, in addition to those already captured within the chapters themselves.

6.2.1 Rapid Pace of Large Language Model Development

The field of natural language processing is currently in a state of very fast innovation, where the efficacy of models and the approaches used are rapidly developing and proliferating. While the current vanguard of models continue to use the Transformer architecture (Vaswani et al., 2017), state-of-the-art results are currently often achieved by using a general purpose autoregressive language model in a zero-shot or few-shot setting (Anthropic, 2024), rather than fine-tuning a language model for a particular purpose.

Research on associated topics related to adversarial robustness, fairness, and explainability, lags the state-of-the-art models available in the field. For example, large-scale adversarial attack benchmarks do not include the most recent large language models (Asthana et al., 2022). The rapid pace of LLM development, and the current rush to incorporate such models into different applications, emphasizes the importance on ongoing research into security, fairness, and explainability.

6.2.2 Adversarial Attacks on Text Degrade Input Quality

An often-overlooked element of adversarial attacks in the text domain is the degradation in sample quality that these attacks can cause. While perturbations in an image can be completely imperceptible to a human viewer, adversarial attacks in the text domain must modify an input sequence made of discrete tokens. When comparing adversarial attacks and robustness to adversarial attacks, it is important to understand that for an attack to be a "true" adversarial attack, it must not affect the true class of the input. If a word-based adversarial attack selects an inappropriate word as a "synonym" producing a nonsensical sample, the sample may no longer be a clear example of a particular class. Enough targeted spelling perturbations in a phishing email to bypass a spam filter may render the message so poorly-written that it no longer would be understood by the receiver.

Overall, it should not be taken for granted that just because an adversarial text attack changed the predicted class of a model after targeted perturbations, that the result is truly "successful". Further evaluation of perturbed samples should determine whether it still contains the same semantic meaning, and assess any determinant to legibility. To this end, evaluating perturbed samples using text quality metrics such as MAUVE (Pillutla et al., 2021) may be worthwhile. Truly imperceptible text-attacks should resemble unperturbed samples to a reviewer. It is possible that the most difficult to detect attacks in the text domain may be found by rephrasing an entire sentence using a generative language model under specific constraints, rather than only perturbing words or characters in an existing sentence.

6.2.3 Input Feature Attributions are Difficult to Evaluate

Input feature attributions are calculated by analyzing the weights within a network. Perturbing inputs and observing the impact on outputs could itself be used as a means of feature attribution. In this sense, research in this area is effectively applying multiple input feature attributions in parallel and using them to evaluate one another. If the goal is to understand which words in an input sentence are most important to the output classification when removed, then this property could be maximized by directly generating explanations directly based on the response to masking.

Better methods of understanding the behaviour of models are worthwhile, but there are clearly deficiencies in the current methods of evaluating interpretability in neural text classifiers. Axiomatic methods, such as integrated gradients (Sundararajan et al., 2017), may offer a more principled approach to evaluating these models. When combined with the aforementioned difficulty in evaluating text generation, this applies in a similar fashion to generative models as well.

6.3 Future Work

The work throughout this thesis sets the foundation for future inquiry across a variety of areas of machine learning research. In this final section, we discuss a range of these areas of future work. For reference, trends and open problems related to machine generated text threat models are discussed at the end of Chapter 2.

6.3.1 Robust Methods for Detection of Machine Generated Text

Further to the work in Chapter 3, improvements in quality of computer-generated text and the large variety of resulting threat models have created an adversarial cybersecurity environment. As such, there is a broad need for defensive research into the robustness of detection methodologies, and methods for preventing widespread abuse of neural language models. Targeted research of detection within specific text domains (e.g., online comment sections) is also likely to be of value to improve defenses for specific domains.

6.3.2 Subjective Quality Evaluation in Machine Generated Text

Further to the work in Chapter 4, we identify the following open problems for future research. First, quantification or representation of abstract concepts such as creativity, and shaping generative models to produce outputs that manifest such attributes, across modalities. Second, analysis of sampling parameters and prompts on characteristics of LLM text generation, incorporating both Mauve and subjective human assessment (including a more comprehensive subjective study). Third, deeper study of large multilingual language models such as BLOOM-176B, on topics such as training data memorization, overparameterization, and fairness.

6.3.3 Detection of AI Content in Artistic Domains and Social Impact of Generative AI

Also related to the work in Chapter 4, usage of generative AI tools in creative fields carries significant questions about the impacts of such tools becoming widespread — namely their impact on artistic expression, the effects on artists, and the impact that such works becoming ubiquitous will have on the Internet. While this chapter has focused on subjective assessment of machine generated text that goes beyond "quality", future work should also consider detection of AI-generated content and the impacts of AI-generated content on the Internet and society at large.

6.3.4 Metrics for Model Interpretability Comparison

Related to the work in Chapter 5, there remains an open problem in comparing model interpretability without relying on variable model-specific behaviour on out-of-domain samples.

Perturbing input features individually and measuring correlation between input feature rankings and changes in class confidence may be one such approach, though this resembles the calculation of integrated gradients itself (Sundararajan et al., 2017). Overall, it continues to remain difficult to disentangle data, model, and explanation method. When working with measures based on iterative masking, it is distinctly important to keep this in mind.

6.3.5 Fairness-Optimized Models

The work in Chapter 5 also touches on fairness evaluation, which is an important and extensive area of research in itself. A challenge exists with the limited selection of open-source fairness-optimized models. A greater number of robust fairness-optimized models would increase the ability of researchers to learn more about the impacts of such fairness-aware training procedures on model behaviour, including input feature attributions. As training procedures designed to improve fairness may involve manipulating salient tokens to avoid models taking stereotypical "shortcuts", understanding how the interplay between fairness measures and interpretability measures if of particular interest.

6.3.6 Interpretability of Additional Model Architectures

Also related to the interpretabiltiy research in Chapter 5, the presented work on interpretability has focused on the Transformer architecture widely used in contemporary neural text classifiers, and utilizes faithfulness and fairness measures that reflect the approaches most commonly used in related work across the field. Future work may extend our analysis to incorporate a wider variety of neural text classifier architectures, and consider less well-known faithfulness measures that may have unique characteristics impacting how they are affected by iterative masking. In using CrowS-Pairs and SEAT, we have selected well-established fairness where limitations have been well-studied and discussed (Aribandi et al., 2021; Meade et al., 2022; Qian et al., 2022). As fairness measures improve and alternative measures are established, future work may incorporate additional fairness measures.

6.3.7 Interpreting Adversarial Attacks and Adversarial Training on Neural Text Classifiers

Feature-based interpretability methods for deep-learning models, such as SHAP (Lundberg and Lee, 2017) and integrated gradients (Sundararajan et al., 2017) assign an importance score to input features to determine their contribution to a particular network result. Evaluation of these interpretability methods has shown that gradient-based explanations demonstrate the best agreement with human assessment for Transformer models, as well as best correlating with tokens which cause the greatest drop in performance if they are removed from the model (a property referred to as "fidelity", that depicts the explanations in terms of how faithful they are to the model's inner workings) (Atanasova et al., 2020). Past research has found that attention relationships within a network can also be considered as explanations of a Transformer model's behaviour in certain cases (Wiegreffe and Pinter, 2019). While the research in this thesis focuses on input attributions, greater insight might be gained from extended this analysis to study of the impact on attention heads as well.

Future work might shed additional light on the impacts of adversarial attacks and adversarial training, by performing an in-depth analysis of both feature-based and attention head interpretability in neural text classifiers. To interpret intermediate layers in Transformer encoder models, we plan to rely on a algorithm knows as "layer conductance" which is implemented within the Captum framework – though the algorithm itself predates this implementation (Dhamdhere et al., 2018; Shrikumar et al., 2018). The layer conductance algorithm implemented in Captum has been used in past work to determine the importance of a neuron in a result from a neural network in a computationally efficient manner (Dhamdhere et al., 2018; Shrikumar et al., 2018). Conductance can be used to interpret intermediate layers in a deep neural network, or perform input attributions. Based on these findings, we plan to characterize the impacts of adversarial training on neuron activations within attention heads of Transformer neural networks.

6.3.8 Efficient Fine-Tuning for Fairness and Adversarial Robustness

Adversarial robustness is an area touched on in both Chapter 3 and Chapter 5, and an area with significant open questions. Previous work has shown that highly-efficient counterfactual training — the practice of training a model to improve fairness by swapping protected characteristics of input samples (such as gender or race) — can be performed on less than 1% of a model's parameters by fine-tuning only word embeddings, word position embeddings, and linear transformations to input and output layers (Gira et al., 2022). Future experiments might investigate whether this also holds true for adversarial training for adversarial robustness, and determine whether both approaches have similar impacts on fidelity. The results from these experiments would represent the single most comprehensive analysis of the effectiveness of various approaches to adversarial training on neural text classifiers — and highlight the overlap between counterfactual training for fairness and adversarial training for robustness.

Beyond this, it may be valuable to also perform adversarial and counterfactual fine-tuning using selected early layers of a neural network to perform early-layer regularization. This would be informed by previous work that has found properties such as coreference resolution to be heavily determined by relatively small, specialized parts of the network (Clark et al., 2019; Sridhar and Sarah, 2020). This approach could be used for both adversarial training and for fairness fine-tuning, and compare to previous work. This is motivated by the analysis that by specific layers within a Transformer model, it has already learned concepts such as coreference resolution, which have a strong link to gender biases. Applying constraints to influence the representations within these layers may also be a valuable area of future research.

6.3.9 Cybersecurity Research on Generative LLM System Architectures

Generative AI text models feature prominently within this work, serving as the basis of the threat models discussed in Chapter 2, and with the outputs of such models evaluated in depth as part of Chapters 3 and 4. The cybersecurity impacts of widespread adoption of Generative AI models are increasingly apparent as powerful models enable a range improved cyberattacks (as described in Chapter 2), as well as introducing new system architectures that may carry additional security considerations. For example, many companies are now deploying virtual assistant interfaces that utilize retrieval-augmented generation to answer questions about internal documents, wikis, or business data (Lewis et al., 2020b). These systems present numerous opportunities for an attacker to attempt to retrieve information they shouldn't have access to, poison information stores with malicious links, or perform a denial of service to prevent such a system from working correctly.

Another area of rapid development, agents which rely on LLMs to perform actions on behalf of a user, stands to dramatically effect the cybersecurity landscape of the Internet (Huang et al., 2024). Access to agents may empower a larger range of threat actors to perform more sophisticated attacks. Similarly, if typical internet users begin relying on agents, these agents must be appropriately secured against techniques such as prompt injection (Liu et al., 2024) that may allow an attacker to hijack an agent to perform malicious activities. New models and applications of these models offer great potential for improved technologies for the benefit of humanity, but these come with commensurate challenges, particularly in the areas of cybersecurity and social science.

References

Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. 2007. A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. 60–69.

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating Sentiment-Preserving Fake Online Reviews Using Neural Language Models and Their Human- and Machine-Based Detection. In *Advanced Information Networking and Applications*, Leonard Barolli, Flora Amato, Francesco Moscato, Tomoya Enokido, and Makoto Takizawa (Eds.). Springer International Publishing, Cham, 1341–1354.

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better Fine-Tuning by Reducing Representational Collapse. In *ICLR*.

Luis von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. 2003. CAPTCHA: Using hard AI problems for security. In *International conference on the theory and applications of cryptographic techniques*. Springer, 294–311.

Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf, and Imtiaz Khan. 2021. Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science* 3 (2021). https://doi.org/10.3389/fcomp.2021.563060

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating Natural Language Adversarial Examples. In *EMNLP 2018*. ACL, Brussels, Belgium, 2890–2896. https://doi.org/10.18653/v1/D18-1316

American Press Institute. 2014. How Americans get their news. https://www.americanpressinstitute.org/publications/reports/survey-research/how-americans-get-news/

Phoenix CS Andrews. 2021. What is Brigading? https://institute.global/policy/social-media-futures-what-brigading

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf

Wissam Antoun, Fady Baly, and Hazem M. Hajj. 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. *ArXiv* (2020).

Vamsi Aribandi, Yi Tay, and Donald Metzler. 2021. How Reliable are Model Diagnostics?. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 1778–1785. https://doi.org/10.18653/v1/2021.findings-acl.155

Ben Armstrong, Megan Beretta, Evan Crothers, Michael Karlin, Dongwoo Kim, Justin Longo, Lorne Powell, and Trooper Sanders. 2019. Siri Humphrey: Design Principles for an AI Policy Analyst. (2019).

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining Predictions of Non-Linear Classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. Association for Computational Linguistics, Berlin, Germany, 1–7. https://doi.org/10.18653/v1/W16-1601

Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilovic, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John T. Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. 2019. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. *CoRR* abs/1909.03012 (2019). arXiv:1909.03012 http://arxiv.org/abs/1909.03012

Kalyani Asthana, Zhouhang Xie, Wencong You, Adam Noack, Jonathan Brophy, Sameer Singh, and Daniel Lowd. 2022. TCAB: A Large-Scale Text Classification Attack Benchmark. *arXiv preprint arXiv:2210.12233* (2022).

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A Diagnostic Study of Explainability Techniques for Text Classification. In *Proceedings of* *the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 3256–3274. https://doi.org/10.18653/v1/2020.emnlp-main.263

Giuseppe Attanasio, Debora Nozza, Eliana Pastor, and Dirk Hovy. 2022. Benchmarking Post-Hoc Interpretability Approaches for Transformer-based Misogyny Detection. In *Proceedings of the First Workshop on Efficient Benchmarking in NLP*. Association for Computational Linguistics.

Brooke Auxier and Monica Anderson. 2021. Social media use in 2021. https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition. *NeurIPS* 34 (2021), 27826–27839.

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. 2021. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356* (2021).

Anton Bakhtin, Yuntian Deng, Sam Gross, Myle Ott, Marc'Aurelio Ranzato, and Arthur Szlam. 2021. Residual Energy-Based Models for Text. *J. of Mach. Learn. Res.* 22 (2021), 40–1.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or Fake? Learning to Discriminate Machine from Human Generated Text. *CoRR* abs/1906.03351 (2019). arXiv:1906.03351 http://arxiv.org/abs/1906.03351

Shahryar Baki, Rakesh Verma, Arjun Mukherjee, and Omprakash Gnawali. 2017. Scaling and Effectiveness of Email Masquerade Attacks: Exploiting Natural Language Generation. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17)*. Association for Computing Machinery, New York, NY, USA, 469–482. https://doi.org/10.1145/3052973.3053037

Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, and Jaehoon Amir Safavi. 2017. Mitigating poisoning attacks on machine learning models: A data provenance based approach. In *Proc. 10th ACM Workshop on Artificial Intelligence and Security*. 103–110.

Leonard E Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics* 37, 6 (1966), 1554–1563.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The Pushshift Reddit Dataset. *CoRR* abs/2001.08435 (2020). arXiv:2001.08435 https://arxiv.org/abs/2001.08435

Colin Bellinger, Shiven Sharma, and Nathalie Japkowicz. 2012. One-Class versus Binary Classification: Which and When?. In *2012 11th International Conference on Machine Learning and Applications*, Vol. 2. 102–106. https://doi.org/10.1109/ICMLA.2012.212

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *International Conference on Applications of Natural Language to Information Systems*. Springer, 421–426.

Jonah Berger and Katherine L Milkman. 2012. What makes online content viral? *Journal of marketing research* 49, 2 (2012), 192–205.

Mathias Berglund, Tapani Raiko, Mikko Honkala, Leo Kärkkäinen, Akos Vetek, and Juha Karhunen. 2015. Bidirectional Recurrent Neural Networks as Generative Models. In *NIPS*.

Meghana Moorthy Bhat and Srinivasan Parthasarathy. 2020. How Effectively Can Machines Defend Against Machine-Generated Fake News? An Empirical Study. In *Proc. First Workshop on Insights from Negative Results in NLP*. ACL, Online, 48–53. https://doi.org/10.18653/v1/2020.insights-1.7

Paras Bhatt and Anthony Rios. 2021. Detecting Bot-Generated Text by Characterizing Linguistic Accommodation in Human-Bot Interactions. *CoRR* abs/2106.01170 (2021). arXiv:2106.01170 https://arxiv.org/abs/2106.01170

BigScience. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. https://doi.org/10.48550/ARXIV.2211.05100

Ron Bitton, Nadav Maman, Inderjeet Singh, Satoru Momiyama, Yuval Elovici, and Asaf Shabtai. 2021. A Framework for Evaluating the Cybersecurity Risk of Real World, Machine Learning Production Systems. *CoRR* abs/2107.01806 (2021). arXiv:2107.01806 https://arxiv.org/abs/2107.01806

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. arXiv:2204.06745 [cs.CL]

Su Lin Blodgett, Gilsinia Lopez, Alexandra Olteanu, Robert Sim, and Hanna Wallach. 2021. Stereotyping Norwegian Salmon: An Inventory of Pitfalls in Fairness Benchmark Datasets. 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)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 1004–1015. https://doi.org/10.18653/v1/2021.acl-long.81

Zoe Braiterman, Adam Shostack, Jonathan Marcil, Stephen de Vries, Irene Michlin, Kim Wuyts, Robert Hurlbut, Brook S. E. Schoenfield, Fraser Scott, Matthew Coles, and et al. 2020. https://www.threatmodelingmanifesto.org/

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Version 1.

CSET Policy Brief. 2021. AI and the Future of Disinformation Campaigns. (2021).

Greg Brockman. 2020. OpenAI API. https://openai.com/blog/openai-api/

Siri Bromander, Audun Jøsang, and Martin Eian. 2016. Semantic Cyberthreat Modelling.. In *STIDS*. 74–78.

Emerson T. Brooking, Alyssa Kann, Max Rizzuto, Rocky Taylor Cole, and Andrew Gully. 2020. DFRLab Dichotomies of Disinformation. https://github.com/DFRLab/Dichotomies-of-Disinformation

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark

Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *ArXiv* abs/2005.14165 (2020).

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On Identifiability in Transformers. In *International Conference on Learning Representations*. https://openreview.net/forum?id=BJg1f6EFDB

A. J. Burns, M. Eric Johnson, and Deanna D. Caputo. 2019. Spear phishing in a barrel: Insights from a targeted phishing campaign. *J. Organ. Comput. Electron. Commer.* 29, 1 (2019), 24–39. https://doi.org/10.1080/10919392.2019.1552745 arXiv:https://doi.org/10.1080/10919392.2019.1552745

Guillaume Cabanac and Cyril Labbé. 2021. Prevalence of nonsensical algorithmically generated papers in the scientific literature. *Journal of the Association for Information Science and Technology* 72, 12 (2021), 1461–1476.

Yang Trista Cao, Yada Pruksachatkun, Kai-Wei Chang, Rahul Gupta, Varun Kumar, Jwala Dhamala, and Aram Galstyan. 2022. On the Intrinsic and Extrinsic Fairness Evaluation Metrics for Contextualized Language Representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 561–570. https://doi.org/10.18653/v1/2022.acl-short.62

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting Training Data from Large Language Models. *CoRR* abs/2012.07805 (2020). arXiv:2012.07805 https://arxiv.org/abs/2012.07805

Mario Casillo, Francesco Colace, Brij B. Gupta, Domenico Santaniello, and Carmine Valentino. 2021. Fake News Detection Using LDA Topic Modelling and K-Nearest

Neighbor Classifier. In *Computational Data and Social Networks*, David Mohaisen and Ruoming Jin (Eds.). Springer International Publishing, Cham, 330–339.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069* (2018).

Chih-Ping Chen. 2016. Forming digital self and parasocial relationships on YouTube. *Journal of Consumer Culture* 16, 1 (2016), 232–254. https://doi.org/10.1177/1469540514521081 arXiv:https://doi.org/10.1177/1469540514521081

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

Xingyuan Chen, Peng Jin, Siyuan Jing, and Chunming Xie. 2022. Automatic Detection of Chinese Generated Essayss Based on Pre-trained BERT. In *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Vol. 10. IEEE, 2257–2260.

Yihao Chen and Alexander Lerch. 2020. Melody-conditioned lyrics generation with seqgans. In *2020 IEEE International Symposium on Multimedia (ISM)*. IEEE, 189–196.

A. Chernikova, A. Oprea, C. Nita-Rotaru, and B. Kim. 2019. Are Self-Driving Cars Secure? Evasion Attacks Against Deep Neural Networks for Steering Angle Prediction. In *2019 IEEE Security and Privacy Workshops (SPW)*. 132–137. https://doi.org/10.1109/SPW.2019.00033

Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. 2018. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications* 106 (2018), 1–20. https://doi.org/10.1016/j.eswa.2018.03.050

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. *NeurIPS* 28 (2015).

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah 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). Association for Computational Linguistics, Online, 7282–7296. https://doi.org/10.18653/v1/2021.acl-long.565

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look At? An Analysis of BERT's Attention. *CoRR* abs/1906.04341 (2019). arXiv:1906.04341 http://arxiv.org/abs/1906.04341

Jordan Clive, Kris Cao, and Marek Rei. 2021. Control prefixes for text generation. *arXiv preprint arXiv:2110.08329* (2021).

Joe Coscarelli. 2020. The only music critic who matters (if you're under 25). https://www.nytimes.com/2020/09/30/arts/music/anthony-fantano-the-needle-drop.html

Evan Crothers. 2022. Text Detection Experiment Code Repository. https://github.com/ecrows/cgtext-detection-adv

Evan Crothers, Nathalie Japkowicz, Herna Viktor, and Paula Branco. 2022a. Adversarial robustness of neural-statistical features in detection of generative transformers. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2019. Towards ethical content-based detection of online influence campaigns. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023a. Machine-generated Text: A Comprehensive Survey of Threat Models and Detection Methods. *IEEE Access* (2023).

Evan Crothers, Nathalie Japkowicz, Herna L. Viktor, and Paula Branco. 2022b. Adversarial Robustness of Neural-Statistical Features in Detection of Generative Transformers.

Evan Crothers, Herna Viktor, and Nathalie Japkowicz. 2021. Mean User-Text Agglomeration (MUTA): Practical User Representation and Visualization for Detection of Online Influence Operations. In *International Conference on Computational Data and Social Networks*. Springer, 305–318.

Evan Crothers, Herna Viktor, and Nathalie Japkowicz. 2024. Robust Infidelity: When Faithfulness Measures on Masked Language Models Are Misleading. arXiv:2308.06795 [cs.CL] https://arxiv.org/abs/2308.06795

Evan Crothers, Herna L Viktor, and Nathalie Japkowicz. 2023b. In BLOOM: Creativity and Affinity in Artificial Lyrics and Art. In *The AAAI-23 Workshop on Creative AI Across Modalities*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 3504–3514.

Michael Cummings. 2005. Archaisms in Shakespeare. http://shakespearestudyguide.com/Archaisms.html

Joseph Cutler, Liam Dugan, Shreya Havaldar, and Adam Stein. 2022. Automatic Detection of Hybrid Human-Machine Text Boundaries. (2022). https://www.cis.upenn.edu/~jwc/assets/nlp.pdf

Sławomir Dadas, Michał Perełkiewicz, and Rafał Poświata. 2020. Pre-training polish transformer-based language models at scale. In *International Conference on Artificial Intelligence and Soft Computing*. Springer, 301–314.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*. https://openreview.net/forum?id=H1edEyBKDS

Nassim Dehouche. 2021. Plagiarism in the age of massive Generative Pre-trained Transformers (GPT-3). *Ethics in Science and Environmental Politics* 21 (2021), 17–23.

Nina Dethlefs and Heriberto Cuayáhuitl. 2010. Hierarchical Reinforcement Learning for Adaptive Text Generation. In *Proc. 6th INLG*. ACL. https://aclanthology.org/W10-4204

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4443–4458. https://doi.org/10.18653/v1/2020.acl-main.408

Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. 2018. How Important Is a Neuron? *CoRR* abs/1805.12233 (2018). arXiv:1805.12233 http://arxiv.org/abs/1805.12233

Disumbrationist. 2019. What is r/SubSimulatorGPT2? https://www.reddit.com/r/SubSimulatorGPT2/comments/btfhks/what_is_rsubsimulatorgpt2/

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 67–73.

Chenhe Dong, Yinghui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2022. A Survey of Natural Language Generation. *ACM Comput. Surv.* (jul 2022). https://doi.org/10.1145/3554727 Just Accepted.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A Smith, and Yejin Choi. 2022. Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7250–7274.

Pablo Ariel Duboue and Kathleen R. McKeown. 2003. Statistical Acquisition of Content Selection Rules for Natural Language Generation. In *Proc. 2003 Conference on Empirical Methods in Natural Language Processing*. 121–128. https://aclanthology.org/W03-1016

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. RoFT: A Tool for Evaluating Human Detection of Machine-Generated Text. *ArXiv* abs/2010.03070 (2020).

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-Box Adversarial Examples for Text Classification. In *ACL 2018 (Volume 2: Short Papers)*. ACL, Melbourne, Australia, 31–36. https://doi.org/10.18653/v1/P18-2006

Kirsten Eddy. 2022. The changing news habits and attitudes of younger audiences. https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2022/young-audiences-news-media

Benj Edwards. 2022. Flooded with AI-generated images, some art communities ban them completely. https://arstechnica.com/information-technology/2022/09/flooded-with-ai-generated-images-some-art-communities-ban-them-completely/

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. 2019. A Rotation and a Translation Suffice: Fooling CNNs with Simple Transformations.

European Commission. 2018. Guidelines on Automated individual decision-making and Profiling for the purposes of Regulation. https://ec.europa.eu/newsroom/article29/items/612053/en

European Commission and Directorate-General for Communications Networks, Content and Technology. 2019. *Ethics guidelines for trustworthy AI*. Publications Office. https://doi.org/doi/10.2759/346720

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. TweepFake: About detecting deepfake tweets. *PLOS ONE* 16, 5 (05 2021), 1–16. https://doi.org/10.1371/journal.pone.0251415

Anthony Fantano. 2022. Nav - Demons Protected by Angels Album Review. https://www.youtube.com/watch?v=5lOvAHg12uw

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018b. Pathologies of Neural Models Make Interpretations Difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3719–3728. https://doi.org/10.18653/v1/D18-1407

Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018a. Topic-to-essay generation with neural networks.. In *IJCAI*. 4078–4084.

Carlos Muñoz Ferrandis, Danish Contractor, Huu Nguyen, and David Lansky. 2022. BigScience RAIL License v1.0. https://huggingface.co/spaces/bigscience/license

Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. 2019. Adversarial attacks on medical machine learning. *Science* 363, 6433 (2019), 1287–1289.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. *PeerJ Computer Science* 7 (2021), e443.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover. *PeerJ Computer Science* 7 (04 2021), e443. https://doi.org/10.7717/peerj-cs.443

Maksym Gabielkov, Arthi Ramachandran, Augustin Chaintreau, and Arnaud Legout. 2016. Social Clicks: What and Who Gets Read on Twitter?. In *ACM SIGMETRICS / IFIP Performance 2016*. Antibes Juan-les-Pins, France. https://hal.inria.fr/hal-01281190

Rinaldo Gagiano, Maria Myung-Hee Kim, Xiuzhen Jenny Zhang, and Jennifer Biggs. 2021. Robustness analysis of grover for machine-generated news detection. In *Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association*. 119–127.

Matthias Gallé, Jos Rozen, Germán Kruszewski, and Hady Elsahar. 2021. Unsupervised and Distributional Detection of Machine-Generated Text. arXiv:2111.02878 [cs.CL]

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 50–56.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *CoRR* abs/2101.00027 (2021). arXiv:2101.00027 https://arxiv.org/abs/2101.00027

Siddhant Garg, Adarsh Kumar, Vibhor Goel, and Yingyu Liang. 2020. Can Adversarial Weight Perturbations Inject Neural Backdoors. In *ACM CIKM 2020*. 2029–2032.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research* 61 (2018), 65–170.

D. Gavriluţ, M. Cimpoeşu, D. Anton, and L. Ciortuz. 2009. Malware detection using machine learning. In *2009 International Multiconference on Computer Science and Information Technology*. 735–741. https://doi.org/10.1109/IMCSIT.2009.5352759

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019a. GLTR: Statistical Detection and Visualization of Generated Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Florence, Italy, 111–116. https://doi.org/10.18653/v1/P19-3019

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019b. GLTR: Statistical Detection and Visualization of Generated Text. *CoRR* abs/1906.04043 (2019). arXiv:1906.04043

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence* 2, 11 (2020), 665–673.

Alberto Giaretta and Nicola Dragoni. 2020. Community Targeted Phishing. In *Proceedings of 6th International Conference in Software Engineering for Defence Applications*, Paolo Ciancarini, Manuel Mazzara, Angelo Messina, Alberto Sillitti, and Giancarlo Succi (Eds.). Springer International Publishing, Cham, 86–93.

Michael Gira, Ruisu Zhang, and Kangwook Lee. 2022. Debiasing Pre-Trained Language Models via Efficient Fine-Tuning. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*. 59–69.

Seraphina Goldfarb-Tarrant, Rebecca Marchant, Ricardo Muñoz Sánchez, Mugdha Pandya, and Adam Lopez. 2021. Intrinsic Bias Metrics Do Not Correlate with Application Bias. 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)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 1926–1940. https://doi.org/10.18653/v1/2021.acl-long.150

Drew Gooden. 2022. using AI to write a youtube video. Retrieved September 4, 2022 from https://www.youtube.com/watch?v=BaVpeJlcQzg

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*. http://arxiv.org/abs/1412.6572

Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. 2017. Adversarial Examples for Malware Detection. In *ESORICS 2017*. Springer, Cham, 62–79.

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. YouTube2Text: Recognizing and Describing Arbitrary Activities Using Semantic Hierarchies and Zero-Shot Recognition. In *2013 IEEE ICCV*. 2712–2719. https://doi.org/10.1109/ICCV.2013.337

Meriem Guerar, Luca Verderame, Mauro Migliardi, Francesco Palmieri, and Alessio Merlo. 2021. Gotta CAPTCHA 'Em All: A Survey of 20 Years of the Human-or-Computer Dilemma. *ACM Comput. Surv.* 54, 9, Article 192 (oct 2021), 33 pages. https://doi.org/10.1145/3477142

Bin Guo, Yasan Ding, Lina Yao, Yunji Liang, and Zhiwen Yu. 2020. The future of false information detection on social media: New perspectives and trends. *ACM Computing Surveys (CSUR)* 53, 4 (2020), 1–36.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

Jeffrey Hargrave. 2005. SCIgen - An Automatic CS Paper Generator. https://pdos.csail.mit.edu/archive/scigen/

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! End-to-end neural data-to-text generation with semantic fidelity. In *COLING 2020*. https://www.amazon.science/publications/have-your-text-and-use-it-too-end-to-end-neural-data-to-text-generation-with-semant

Laura Hayden. 1999. Clichés: Avoid Them Like The Plague. github.com/ecrows/cliche500 Originally published: http://suspense.net/whitefish/cliche.htm. suspense@suspense.net.

Sherry He, Brett Hollenbeck, and Davide Proserpio. 2022. The market for fake reviews. *Marketing Science* (2022).

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium. *CoRR* abs/1706.08500 (2017). arXiv:1706.08500 http://arxiv.org/abs/1706.08500

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019a. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* (2019).

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019b. The Curious Case of Neural Text Degeneration. *CoRR* abs/1904.09751 (2019). arXiv:1904.09751 http://arxiv.org/abs/1904.09751

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. arXiv:2402.02716 [cs.AI] https://arxiv.org/abs/2402.02716

Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh. 2022. Detecting Word-Level Adversarial Text Attacks via SHapley Additive exPlanations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*. 156–166.

Aminul Huq and Mst. Tasnim Pervin. 2020. Adversarial Attacks and Defense on Texts: A Survey. *CoRR* abs/2005.14108 (2020). arXiv:2005.14108 https://arxiv.org/abs/2005.14108

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *ACL*.

Adam Ivankay, Ivan Girardi, Chiara Marchiori, and Pascal Frossard. 2022. Fooling Explanations in Text Classifiers. *arXiv preprint arXiv:2206.03178* (2022).

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685* (2020).

Srinivasan Janarthanam and Oliver Lemon. 2009. Learning lexical alignment policies for generating referring expressions for spoken dialogue systems. In *Proc. 12th ENLG*. 74–81.

Nathalie Japkowicz et al. 2000. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets*, Vol. 68. AAAI Press Menlo Park, CA, 10–15.

Jasper AI. 2022. The best ai writing assistant. Retrieved September 4, 2022 from https://www.jasper.ai/

Ganesh Jawahar, Muhammad Abdul-Mageed, and VS Laks Lakshmanan. 2020. Automatic Detection of Machine Generated Text: A Critical Survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2296–2309.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language?. In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Jigsaw. 2019. Jigsaw unintended bias in toxicity classification. https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics* 48, 1 (March 2022), 155–205. https://doi.org/10.1162/coli_a_00426

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 8018–8025.

Tom Huddleston Jr. 2022. Scammers are texting you from your own number now - here's what to do if that happens. https://www.cnbc.com/2022/04/02/scammers-are-texting-you-from-your-own-number-now-what-to-do-about-it.html

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proc. 2013 conference on empirical methods in natural language processing*. 1700–1709.

Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text summarization from legal documents: a survey. *Artificial Intelligence Review* 51, 3 (2019), 371–402.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. *arXiv preprint arXiv:2109.06835* (2021).

Davinder Kaur, Suleyman Uslu, Kaley J. Rittichier, and Arjan Durresi. 2022. Trustworthy Artificial Intelligence: A Review. *ACM Comput. Surv.* 55, 2, Article 39 (jan 2022), 38 pages. https://doi.org/10.1145/3491209

Simon Kemp. 2022a. Content across cultures. https://datareportal.com/reports/future-trends-2022-cross-cultural-content

Simon Kemp. 2022b. Digital 2022 Global Digital Overview. https://datareportal.com/reports/digital-2022-global-overview-report

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *ArXiv* abs/1909.05858 (2019).

Michael Khrushchev, Ruslan Vasilev, Nikolay Zinov, Alexey Petrov, and Yandex. 2022. YaLM 100B. "https://huggingface.co/yandex/yalm-100b".

Yannic Kilcher. 2022. This is the worst ai ever. https://www.youtube.com/watch?v=efPrtcLdcdM

Jong Wook Kim. 2019. openai/gpt-2-output-dataset. https://github.com/openai/gpt-2-output-dataset

Soomin Kim, Joonhwan Lee, and Gahgene Gweon. 2019. Comparing data from chatbot and web surveys: Effects of platform and conversational style on survey response quality. In *Proc. 2019 CHI conference on human factors in computing systems*. 1–12.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is Not Only a Weight: Analyzing Transformers with Vector Norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 7057–7075. https://doi.org/10.18653/v1/2020.emnlp-main.574

Loren Kohnfelder and Praerit Garg. 1999. The threats to our products. *Microsoft Interface, Microsoft Corporation* 33 (1999).

Ravikumar Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical nlg framework for aggregated planning and realization. In *Proc. 51st Annual Meeting of the ACL (Volume 1: Long Papers)*. 1406–1415.

Peter Kowalczyk, Marco Röder, Alexander Dürr, and Frédéric Thiesse. 2022. Detecting and Understanding Textual Deepfakes in Online Reviews. (2022).

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. Gedi: Generative discriminator guided sequence generation. *arXiv preprint arXiv:2009.06367* (2020).

Andrey Kurenkov. 2022. Lessons from the GPT-4Chan Controversy. https://thegradient.pub/gpt-4chan-lessons. *The Gradient* (2022).

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. Artificial Text Detection via Examining the Topology of Attention Maps. *CoRR* abs/2109.04825 (2021). arXiv:2109.04825 https://arxiv.org/abs/2109.04825

Polina Kuznetsova, Jianfu Chen, and Yejin Choi. 2013. Understanding and quantifying creativity in lexical composition. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 1246–1258.

Cyril Labbé and Dominique Labbé. 2013. Duplicate and fake publications in the scientific literature: how many SCIgen papers in computer science? *Scientometrics* 94, 1 (2013), 379–396.

Himabindu Lakkaraju, Dylan Slack, Yuxin Chen, Chenhao Tan, and Sameer Singh. 2022. Rethinking Explainability as a Dialogue: A Practitioner's Perspective. In *NeurIPS Workshop on Human Centered AI*.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Majd Latah. 2020. Detection of malicious social bots: A survey and a refined taxonomy. *Expert Systems with Applications* 151 (2020), 113383. https://doi.org/10.1016/j.eswa.2020.113383

Allen Lavoie and Mukkai S. Krishnamoorthy. 2010. Algorithmic Detection of Computer Generated Text. *ArXiv* abs/1008.0706 (2010).

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).

Elias Leight. 2018. Why your favorite artist is releasing more singles than ever. https://www.rollingstone.com/music/music-features/why-your-favorite-artist-is-releasing-more-singles-than-ever-629130/

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

Chuan Li. 2020. OpenAI's GPT-3 Language Model: A Technical Overview. https://lambdalabs.com/blog/demystifying-gpt-3/

Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. 2019. Entangled transformer for image captioning. In *Proc. IEEE/CVF*. 8928–8937.

Haoran Li, Junnan Zhu, Cong Ma, Jiajun Zhang, and Chengqing Zong. 2017. Multi-modal Summarization for Asynchronous Collection of Text, Image, Audio and Video. In *Proc. 2017 Conf. EMNLP*. ACL, Copenhagen, Denmark, 1092–1102. https://doi.org/10.18653/v1/D17-1114

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016a. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 110–119.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep Reinforcement Learning for Dialogue Generation. In *EMNLP*.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained Language Model for Text Generation: A Survey. In *Proc. 30th IJCAI*, Zhi-Hua Zhou (Ed.). IJCAI Organization, 4492–4499. https://doi.org/10.24963/ijcai.2021/612 Survey Track.

Ziming Li, Julia Kiseleva, and Maarten de Rijke. 2018. Dialogue Generation: From Imitation Learning to Inverse Reinforcement Learning. *CoRR* abs/1812.03509 (2018). arXiv:1812.03509 http://arxiv.org/abs/1812.03509

Percy Liang, Rishi Bommasani, Kathleen A. Creel, and Rob Reich. 2022. The Time Is Now to Develop Community Norms for the Release of Foundation Models. https://crfm.stanford.edu/2022/05/17/community-norms.html

Percy Liang and Rob Reich. 2022. Condemning the deployment of gpt-4chan. https://docs.google.com/forms/d/e/1FAIpQLSdh3Pgh0sGrYtRihBu-GPN7FSQoODBLvF7dVAFLZk2iuMgoLw/viewform

Yi Liao, Yasheng Wang, Qun Liu, and Xin Jiang. 2019. GPT-based Generation for Classical Chinese Poetry. *ArXiv* abs/1907.00151 (2019).

Kevin Lin, Dianqi Li, Xiaodong He, Ming-Ting Sun, and Zhengyou Zhang. 2017. Adversarial Ranking for Language Generation. In *NIPS*.

Zachary C. Lipton. 2018. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery. *Queue* 16, 3 (jun 2018), 31–57. https://doi.org/10.1145/3236386.3241340

Nicolle Liu. 2022. The China Scam calls just won't die. https://www.politico.com/newsletters/politico-china-watcher/2022/08/25/the-china-scam-calls-just-wont-die-00052537

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2024. Prompt Injection attack against LLM-integrated Applications. arXiv:2306.05499 [cs.CR] https://arxiv.org/abs/2306.05499

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv* abs/1907.11692 (2019).

Bryan Lufkin. 2021. Why do cover letters still exist? https://www.bbc.com/worklife/article/20211005-why-do-cover-letters-still-exist

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2, 2 (1958), 159–165.

Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

John Lyons. 1991. *Natural Language and Universal Grammar: Volume 1: Essays in Linguistic Theory*. Vol. 1. Cambridge University Press.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. 2022. Evaluating the Faithfulness of Importance Measures in NLP by Recursively Masking Allegedly Important Tokens and Retraining. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. 1731–1751.

Arya Manjaramkar. 2021. CodeGenX. https://github.com/DeepGenX/CodeGenX.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, 'Eric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. CamemBERT: a Tasty French Language Model. *ArXiv* abs/1911.03894 (2019).

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On Measuring Social Biases in Sentence Encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 622–628. https://doi.org/10.18653/v1/N19-1063

Leland McInnes and John Healy. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv* abs/1802.03426 (2018).

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2022. An Empirical Survey of the Effectiveness of Debiasing Techniques for Pre-trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Dublin, Ireland, 1878–1898. https://doi.org/10.18653/v1/2022.acl-long.132

Kirill Meleshevich and Bret Schafer. 2018. Online information laundering: The role of social media. https://securingdemocracy.gmfus.org/online-information-laundering-the-role-of-social-media/

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optim. LSTM lang. models. *arXiv preprint arXiv:1708.02182* (2017).

Albert Meroño-Peñuela and Dayana Spagnuelo. 2020. Can a Transformer Assist in Scientific Writing? Generating Semantic Web Paper Snippets with GPT-2. In *The Semantic Web: ESWC 2020 Satellite Events*, Andreas Harth, Valentina Presutti, Raphaël Troncy, Maribel Acosta, Axel Polleres, Javier D. Fernández, Josiane Xavier Parreira, Olaf Hartig, Katja Hose, and Michael Cochez (Eds.). Springer International Publishing, Cham, 158–163.

Tomáš Mikolov et al. 2012. Stat. lang. models based on neural networks. *Presentation at Google, Mountain View, 2nd April* 80, 26 (2012).

Jaron Mink, Licheng Luo, Natã M. Barbosa, Olivia Figueira, Yang Wang, and Gang Wang. 2022. DeepPhish: Understanding User Trust Towards Artificially Generated Profiles in Online Social Networks. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 1669–1686. https://www.usenix.org/conference/usenixsecurity22/presentation/mink

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. In *EMNLP 2020: System Demonstrations*. 119–126.

Shaoor Munir, Brishna Batool, Zubair Shafiq, Padmini Srinivasan, and Fareed Zaffar. 2021. Through the Looking Glass: Learning to Attribute Synthetic Text Generated by Language Models. In *Proc. 16th Conf. Euro. Chap. ACL: Main Vol.* ACL, Online, 1811–1822. https://doi.org/10.18653/v1/2021.eacl-main.155

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-Pairs: A Challenge Dataset for Measuring Social Biases in Masked Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 1953–1967. https://doi.org/10.18653/v1/2020.emnlp-main.154

Mitchell Naylor, Christi French, Samantha Terker, and Uday Kamath. 2021. Quantifying explainability in nlp and analyzing algorithms for performance-explainability tradeoff. *arXiv preprint arXiv:2107.05693* (2021).

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1069–1078.

Hoang-Quoc Nguyen-Son, Ngoc-Dung T. Tieu, Huy H. Nguyen, Junichi Yamagishi, and Isao Echi Zen. 2017. Identifying computer-generated text using statistical analysis. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 1504–1511. https://doi.org/10.1109/APSIPA.2017.8282270

Peter Norvig. 2009. Natural language corpus data. *Beautiful data* (2009), 219–242.

Muhammed K Olorunnimbe and Herna L Viktor. 2015. Tweets as a vote: Exploring political sentiments on twitter for opinion mining. In *International Symposium on Methodologies for Intelligent Systems*. Springer, 180–185.

Open Cover Letter. 2022. Open Cover Letter: Generate Cover Letters with AI. https://www.opencoverletter.com/

OpenAI. 2020. GPT-3. https://github.com/openai/gpt-3.

OpenAI. 2020. OpenAI GPT-3 GitHub Repository. https://github.com/openai/gpt-3

OpenAI. 2022. ChatGPT: Optimizing language models for dialogue. https://openai.com/blog/chatgpt/

Oregon FBI. 2019. Building a digital defense against the Chinese Embassy Scam. https://www.fbi.gov/contact-us/field-offices/portland/news/press-releases/oregon-fbi-tech-tuesday-building-a-digital-defense-against-the-chinese-embassy-scam

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proc. of NAACL-HLT 2019: Demo*.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks against Machine Learning. In *ACM 2017* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17)*. ACM, New York, NY, USA, 506–519. https://doi.org/10.1145/3052973.3053009

Papers With Code. 2022. Natural language processing benchmarks. https://paperswithcode.com/area/natural-language-processing

Dipti Pawade, Avani M. Sakhapara, Mansi Jain, Neha Jain, and Krushi Gada. 2018. Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM. *International Journal of Information Technology and Computer Science* (2018).

Rivindu Perera and Parma Nand. 2017. Recent advances in natural language generation: A survey and classification of the empirical literature. *Computing and Informatics* 36, 1 (2017), 1–32.

Philippine National Privacy Commission. 2022. NPC issues an order to telcos to cooperate on invest. of smishing att. https://www.privacy.gov.ph/2022/09/national-privacy-commission-issues-an-order-to-telcos-to-cooperate-on-investigation-

P Jonathon Phillips, Carina A Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. 2020. Four principles of explainable artificial intelligence. *Gaithersburg, Maryland* (2020).

P Jonathon Phillips, Amy N Yates, Ying Hu, Carina A Hahn, Eilidh Noyes, Kelsey Jackson, Jacqueline G Cavazos, Géraldine Jeckeln, Rajeev Ranjan, Swami Sankaranarayanan, et al. 2018. Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms. *Proc. National Academy of Sciences* 115, 24 (2018), 6171–6176.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems* 34 (2021), 4816–4828.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. 2020. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications* 32, 10 (2020), 6363–6381.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the ACL: System Demonstrations*. https://nlp.stanford.edu/pubs/qi2020stanza.pdf

Edward Qian. 2019. Twitter climate change sentiment dataset. https://www.kaggle.com/datasets/edqian/twitter-climate-change-sentiment-dataset

Rebecca Qian, Candace Ross, Jude Fernandes, Eric Michael Smith, Douwe Kiela, and Adina Williams. 2022. Perturbation Augmentation for Fairer NLP. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 9496–9521.

Shilin Qiu, Qihe Liu, Shijie Zhou, and Wen Huang. 2022. Adversarial attack and defense technologies in natural language processing: A survey. *Neurocomputing* 492 (2022), 278–307.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. Language Models are Unsupervised Multitask Learners. (2019).

Alec Radford, Jeff Wu, and Jack Clark. 2019a. GPT-2 Output Dataset. https://github.com/openai/gpt-2-output-dataset.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. of Mach. Learn. Res.* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *ICML*. PMLR, 8821–8831.

Priyanka Ranade, Aritran Piplai, Sudip Mittal, Anupam Joshi, and Tim Finin. 2021. Generating Fake Cyber Threat Intelligence Using Transformer-Based Models. In *2021*

International Joint Conference on Neural Networks (IJCNN). 1–9. https://doi.org/10.1109/IJCNN52387.2021.9534192

Reddit. 2018. Reddit Transparency Report: Suspicious Accounts. https://www.reddit.com/wiki/suspiciousaccounts. Accessed: 2019-04-20.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. 2022. A Generalist Agent. https://doi.org/10.48550/ARXIV.2205.06175

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.

Ehud Reiter and Robert Dale. 2002. Building Applied Natural Language Generation Systems. *Natural Language Engineering* 3 (03 2002). https://doi.org/10.1017/S1351324997001502

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

Matheus Augusto Rodrigues, Alcione Oliveira, Alexandra Moreira, and Maurilio Possi. 2022. Lyrics Generation supported by Pre-trained Models. In *The International FLAIRS Conference Proceedings*, Vol. 35.

Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-Domain Detection of GPT-2-Generated Technical Text. In *Proc. 2022 Conf. Nor. Amer. Chapt. ACL: Human Language Technologies*. ACL, Seattle, United States, 1213–1233. https://doi.org/10.18653/v1/2022.naacl-main.88

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752 [cs.CV]

Domenic Rosati. 2022. SynSciPass: detecting appropriate uses of scientific text generation. https://doi.org/10.48550/ARXIV.2209.03742

Mehrshad Sadria, Anita Layton, and Gary Bader. 2023. Adversarial training improves model interpretability in single-cell RNA-seq analysis. *bioRxiv* (2023), 2023–05.

Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon gyo Jung, and Bernard J. Jansen. 2022. Creating and detecting fake reviews of online products. *Journal of Retailing and Consumer Services* 64 (2022), 102771. https://doi.org/10.1016/j.jretconser.2021.102771

Wojciech Samek, Alexander Binder, Gregoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems* 28 (11 2017), 2660–2673. https://doi.org/10.1109/TNNLS.2016.2599820

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

Sashank Santhanam and Samira Shaikh. 2019. A survey of natural language generation techniques with a focus on dialogue systems-past, present and future directions. *arXiv preprint arXiv:1906.00500* (2019).

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-Diagnosis and Self-Debiasing: A Proposal for Reducing Corpus-Based Bias in NLP. *Transactions of the Association for Computational Linguistics* 9 (12 2021), 1408–1424. https://doi.org/10.1162/tacl_a_00434 arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00434/1979270/tacl_a_00434.pdf

Elliot Schrage and David Ginsberg. 2018. Facebook Launches New Initiative to Help Scholars Assess Social Media's Impact on Elections. https://newsroom.fb.com/news/2018/04/new-elections-initiative/

Roei Schuster, Congzheng Song, Eran Tromer, and Vitaly Shmatikov. 2021. You autocomplete me: Poisoning vulnerabilities in neural code completion. In *30th USENIX Security Symposium (USENIX Security 21)*. 1559–1575.

Kevin Scott. 2020. Microsoft teams up with OpenAI to exclusively license GPT-3 language model.

Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. 2019. Do Massively Pretrained Language Models Make Better Storytellers?. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. 843–861.

Maximilian Seitzer. 2020. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid. Version 0.2.1.

Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and Abstraction in Sociotechnical Systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 59–68. https://doi.org/10.1145/3287560.3287598

Sofia Serrano and Noah A. Smith. 2019. Is Attention Interpretable? *CoRR* abs/1906.03731 (2019). arXiv:1906.03731 http://arxiv.org/abs/1906.03731

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the The RuATD Shared Task 2022 on Artificial Text Detection in Russian. *arXiv preprint arXiv:2206.01583* (2022).

Fadi Abu Sheikha and Diana Inkpen. 2011. Generation of formal and informal sentences. In *Proc. 13th European Workshop on Natural Language Generation*. 187–193.

Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2018. Toward Diverse Text Generation with Inverse Reinforcement Learning. In *IJCAI*.

Zhouxing Shi, Ting Yao, Jingfang Xu, and Minlie Huang. 2019. Robustness to Modification with Shared Words in Paraphrase Identification. *arXiv preprint arXiv:1909.02560* (2019).

Adam Shostack. 2014. *Threat modeling: Designing for security*. John Wiley & Sons.

Adam Shostack. 2021. Shostack's 4 Question Frame for Threat Modeling. https://github.com/adamshostack/4QuestionFrame.

Avanti Shrikumar, Jocelin Su, and Anshul Kundaje. 2018. Computationally Efficient Measures of Internal Neuron Importance. *CoRR* abs/1807.09946 (2018). arXiv:1807.09946 http://arxiv.org/abs/1807.09946

Kai Shu, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Mining disinformation and fake news: Concepts, methods, and recent advancements. In *Disinformation, misinformation, and fake news in social media*. Springer, 1–19.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 2022. BlenderBot 3: a deployed conversational agent that continually learns to responsibly engage. https://doi.org/10.48550/ARXIV.2208.03188

Ruba Skaik and Diana Inkpen. 2020. Using Social Media for Mental Health Surveillance: A Review. *ACM Comput. Surv.* 53, 6, Article 129 (dec 2020), 31 pages. https://doi.org/10.1145/3422824

SS Skrylnikov, PA Posokhov, and OV Makhnytkina. 2022. Artificial text detection in Russian language: a BERT-based Approach. In *Dialogue*. https://doi.org/10.28995/2075-7182-2022-21-470-476

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.

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

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *ICML*. 5926–5936.

Sharath Nittur Sridhar and Anthony Sarah. 2020. Undivided Attention: Are Intermediate Layers Necessary for BERT? *CoRR* abs/2012.11881 (2020). arXiv:2012.11881 https://arxiv.org/abs/2012.11881

Stefan Stieglitz, Florian Brachten, Davina Berthelé, Mira Schlaus, Chrissoula Venetopoulou, and Daniel Veutgen. 2017. Do social bots (still) act different to humans?–Comparing metrics of social bots with those of humans. In *International conference on social computing and social media*. Springer, 379–395.

Harald Stiff and Fredrik Johansson. 2022. Detecting computer-generated disinformation. *Int. J. Data Sci. Anal.* 13, 4 (2022), 363–383.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A Contrastive Framework for Neural Text Generation. *arXiv preprint arXiv:2202.06417* (2022).

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "Transforming" Delete, Retrieve, Generate Approach for Controlled Text Style Transfer. In *Proc. 2019 Conf. EMNLP-IJCNLP*. 3269–3279.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

Greg Surma. 2019. Album covers images dataset. https://www.kaggle.com/datasets/greg115/album-covers-images

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.

Yi Tay, Dara Bahri, Che Zheng, Clifford Brunk, Donald Metzler, and Andrew Tomkins. 2020. Reverse Engineering Configurations of Neural Text Generation Models. In *Proc. 58th Annu. Meet. ACL Sys. Demo.* 275–279.

TB Canada Secretariat. 2021. Directive on automated decision-making. https://www.tbs-sct.canada.ca/pol/doc-eng.aspx?id=32592

Senait G Tesfagergish, Robertas Damaševičius, and Jurgita Kapočiūtė-Dzikienė. 2021. Deep Fake Recognition in Tweets Using Text Augmentation, Word Embeddings and

Deep Learning. In *International Conference on Computational Science and Its Applications*. Springer, 523–538.

M. Onat Topal, Anil Bas, and Imke van Heerden. 2021. Exploring Transformers in Natural Language Generation: GPT, BERT, and XLNet. *ArXiv* abs/2102.08036 (2021).

Julien Tourille, Babacar Sow, and Adrian Popescu. 2022. Automatic Detection of Bot-Generated Tweets. In *Proc. 1st International Workshop on Multimedia AI against Disinformation* (Newark, NJ, USA) *(MAD '22)*. ACM, New York, NY, USA, 44–51. https://doi.org/10.1145/3512732.3533584

A. M. Turing. 1950. Computing Machinery and Intelligence. *Mind* LIX, 236 (10 1950), 433–460. https://doi.org/10.1093/mind/LIX.236.433 arXiv:https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf

Twitter. 2019. Twitter Elections Integrity Dataset. https://about.twitter.com/en_us/values/elections-integrity.html. Accessed: 2019-04-20.

Tony UcedaVelez and Marco M Morana. 2015. *Risk Centric Threat Modeling: process for attack simulation and threat analysis*. John Wiley & Sons.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020a. Authorship Attribution for Neural Text Generation. In *Proc. 2020 Conf. EMNLP*. ACL, Online, 8384–8395. https://doi.org/10.18653/v1/2020.emnlp-main.673

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020b. Authorship attribution for neural text generation. In *EMNLP 2020*.

UK National Cyber Security Centre. 2022. Threat modelling. https://www.ncsc.gov.uk/collection/building-a-security-operations-centre/onboarding-systems-and-log-sources/threat-modelling

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language

Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Tal Linzen, Grzegorz Chrupała, and Afra Alishahi (Eds.). Association for Computational Linguistics, Brussels, Belgium, 353–355. https://doi.org/10.18653/v1/W18-5446

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Jun Wang, Mingfei Gao, Yuqian Hu, Ramprasaath R. Selvaraju, Chetan Ramaiah, Ran Xu, Joseph F. JaJa, and Larry S. Davis. 2022a. TAG: Boosting Text-VQA via Text-aware Visual Question-answer Generation. https://doi.org/10.48550/ARXIV.2208.01813

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022b. GIT: A Generative Image-to-text Transformer for Vision and Language. *arXiv preprint arXiv:2205.14100* (2022).

Junjie Wang, Yuxiang Zhang, Lin Zhang, Ping Yang, Xinyu Gao, Ziwei Wu, Xiaoqun Dong, Junqing He, Jianheng Zhuo, Qi Yang, Yongfeng Huang, Xiayu Li, Yanghan Wu, Junyu Lu, Xinyu Zhu, Weifeng Chen, Ting Han, Kunhao Pan, Rui Wang, Hao Wang, Xiaojun Wu, Zhongshen Zeng, Chongpei Chen, Ruyi Gan, and Jiaxing Zhang. 2022c. Fengshenbang 1.0: Being the Foundation of Chinese Cognitive Intelligence. *CoRR* abs/2209.02970 (2022).

Mengqian Wang, Manhua Wang, Fei Yu, Yue Yang, Jennifer Walker, and Javed Mostafa. 2021. A systematic review of automatic text summarization for biomedical literature and EHRs. *Journal of the American Medical Informatics Association* 28, 10 (2021), 2287–2297.

Wenqi Wang, Benxiao Tang, Run Wang, Lina Wang, and Aoshuang Ye. 2019b. A survey on Adversarial Attacks and Defenses in Text. *CoRR* abs/1902.07285 (2019). arXiv:1902.07285 http://arxiv.org/abs/1902.07285

Xiaosen Wang, Hao Jin, and Kun He. 2019a. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723* (2019).

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, jinyan su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Chenxi Whitehouse, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav

Nakov. 2024. SemEval-2024 Task 8: Multidomain, Multimodel and Multilingual Machine-Generated Text Detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*. Association for Computational Linguistics, Mexico City, Mexico, 2041–2063. https://aclanthology.org/2024.semeval2024-1.275

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, and Slav Petrov. 2020. Measuring and Reducing Gendered Correlations in Pre-trained Models. arXiv:2010.06032 [cs.CL]

Max Weiss. 2019. Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions. *Technology Science* (2019).

Joseph Weizenbaum. 1966. ELIZA—a Computer Program for the Study of Natural Language Communication between Man and Machine. *Commun. ACM* 9, 1 (jan 1966), 36–45. https://doi.org/10.1145/365153.365168

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural Text Generation With Unlikelihood Training. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJeYe0NtvH

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 11–20.

Wikitionary. 2022. English Idioms. https://en.wiktionary.org/wiki/Category:English_idioms

Simon Willison. 2022. https://simonwillison.net/2022/Sep/12/prompt-injection/

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP 2020: System Demonstrations*. ACL, Online, 38–45.

Max Wolff and Stuart Wolff. 2020. Attacking neural text detectors. *arXiv preprint arXiv:2002.11768* (2020).

Han Xiao. 2018. bert-as-service. https://github.com/hanxiao/bert-as-service.

H. Xu, Yao Ma, Haochen Liu, Debayan Deb, H. Liu, Jiliang Tang, and A. Jain. 2020. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. *International Journal of Automation and Computing* 17 (2020), 151–178.

Zhixing Xu, Sayak Ray, Pramod Subramanyan, and Sharad Malik. 2017. Malware Detection Using Machine Learning Based Analysis of Virtual Memory Access Patterns. In *DATE 2017* (Lausanne, Switzerland) *(DATE '17)*. European Design and Automation Association, Leuven, BEL, 169–174.

Lu Yang, Qing Song, and Yingqi Wu. 2020. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. *Multimedia Tools and Applications* (05 Sep 2020). https://doi.org/10.1007/s11042-020-09604-z

Jin Yong Yoo and Yanjun Qi. 2021. Towards Improving Adversarial Training of NLP Models. *arXiv preprint arXiv:2109.00544* (2021).

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *AAAI*.

Muhammad Bilal Zafar, Michele Donini, Dylan Slack, Cedric Archambeau, Sanjiv Das, and Krishnaram Kenthapadi. 2021. On the Lack of Robust Interpretability of Neural Text Classifiers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Online, 3730–3740. https://doi.org/10.18653/v1/2021.findings-acl.327

Rowan Zellers. 2019. Why We Released Grover. https://thegradient.pub/why-we-released-grover/. *The Gradient* (2019).

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *NeurIPS* 32 (2019).

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, Zhen-Zhang Yang, Kaisheng Wang, Xiaoda Zhang, Chen Li, Ziyan Gong, Yifan Yao, Xinjing Huang, Jun Wang, Jianfeng Yu, Qi Guo, Yue Yu, Yan Zhang, Jin Wang, Hengtao Tao, Dasen Yan, Zexuan Yi, Fang Peng, Fangqing Jiang, Han Zhang, Lingfeng Deng, Yehong Zhang, Zhe Lin, Chao Zhang, Shaojie Zhang, Mingyue Guo, Shanzhi Gu, Gaojun Fan, Yaowei Wang, Xuefeng Jin, Qun Liu, and Yonghong Tian. 2021. PanGu-α: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-parallel Computation. *CoRR* abs/2104.12369 (2021). arXiv:2104.12369 https://arxiv.org/abs/2104.12369

Li Zhang and Jian-Tao Sun. 2009. *Text Generation*. Springer US, Boston, MA, 3048–3051. https://doi.org/10.1007/978-0-387-39940-9_416

Le Zhang, Rongsheng Zhang, Xiao-Xi Mao, and Yongzhu Chang. 2022b. QiuNiu: A Chinese Lyrics Generation System with Passage-Level Input. In *ACL*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022a. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. DIALOGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 270–278.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *Proc. AAAI Conference on Artificial Intelligence*, Vol. 34. 13041–13049.

Wanrong Zhu, An Yan, Yujie Lu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. 2022. Visualize Before You Write: Imagination-Guided Open-Ended Text Generation. *arXiv preprint arXiv:2210.03765* (2022).

Ziming Zhuang, Ergin Elmacioglu, Dongwon Lee, and C. Lee Giles. 2007. Measuring Conference Quality by Mining Program Committee Characteristics. In *Proc. 7th JCDL* (Vancouver, BC, Canada) *(JCDL '07)*. ACM, New York, NY, USA, 225–234. https://doi.org/10.1145/1255175.1255220

George Kingsley Zipf. 1949. Human behavior and the principle of least effort. (1949).

George Kingsley Zipf. 1999. *The psycho-biology of language: An introduction to dynamic philology*. Vol. 21. Psychology Press.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual Data Augmentation for Mitigating Gender Stereotypes in Languages with Rich Morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 1651–1661. https://doi.org/10.18653/v1/P19-1161