Investigating the Structure of AI-Generated Text

Aleksandar Jovanovic-Hacon, Luke Mo, Kene Nnolim, Justin Yu

December 2024

Abstract

With the rise, commercialization, and increasing quality of Large Language Models (LLMs), it is now easier than ever to create fake bot users on social media platforms. As LLMs get better at mimicking human speech, the ability for a user to distinguish human-generated thought from AI-generated thought, especially without the help of integrated AI-detection, becomes increasingly more difficult, leaving users more prone to believing misinformation and to interacting with bad actors. The ability to filter genuine users from bots is both technically challenging and crucial in maintaining a platform's quality and integrity. Conventional methods for AI-detection tend to require longer contexts, making them less effective on social media, where posts are often shorter, more organically formulated, and disregard proper language convention altogether. To address these issues, we train a model based on part-of-speech tagging, through which we can express more granular relationships that are equally effective with texts of any length. Our model is able to detect AI-generated tweets with 90% accuracy, despite relying on a significantly lower-dimensional representation.

1 Introduction

While powerful, LLMs introduce a number of risks to society, particularly through social media, ranging from harmless spam to deceptive misinformation. Bots often go unmarked by bot-detecting algorithms and unnoticed by users, leading people to be less wary than necessary when encountering misinformation online [13]. They have been proven to engage more frequently with more extreme communities and with more sensitive content [4], and have been weaponized to manipulate public opinion in 81 countries [8]. These issues are further amplified by personal biases and certain social phenomena, like self-confirmation bias and the echo-chamber effect, which lead people already in these extreme communities to be further reinforced in their beliefs. [5].

A key aspect of an LLMs' ability to cause these problems is the volume at which it can produce text. Bot engagement on Twitter is estimated to be in excess of 10% in many genres, reaching 14.26% of users in the online American-political community, for example [14]. In addition, over half of detected Twitter bots were found to be verified, which further boosts their engagement levels [4]. Thus, the most sensible approach to address this issue is to operate at a similar scale, providing some automated means of detecting whether some content was generated by an LLM.

This topic of interest is bounded by an all-encompassing question that current research has still not fully answered, namely why AI-generated text has verifiably different structure from human-generated text and what may cause these differences. We investigate a low-dimensional view of the structure of text through part-of-speech (POS) analysis, which we train using text from both Wikipedia and Twitter.

POS analysis involves analyzing texts through the part of speech that each word represents as well as their relative positions to each other. POS analysis first involves tokenizing each word individually into a token representing the grammatical role the word plays in the sentence, which can fall into one of the following categories: adpositions, proper nouns, particles, auxiliary verbs, punctuation, coordinating conjunctions, verbs, adjectives, adverbs, nouns, pronouns, determiners, subordinating conjunctions, or interjections. Other types of text which don't fall into traditional linguistic notions of parts of speech, like spaces, numbers, symbols, and other characters, are also given their own categories. These categories are passed into our POS models, rather than the text itself, in order to determine whether AI generated text can be detected solely through its grammatical structure.

Another promising approach to detect AI-generated speech and circumvent these aforementioned issues involves statistical word frequency analysis. Current POS analysis on such text corpora varies in terms of implementation and scope—some research uses POS tags in combination with other tags that represent other aspects of a word, like named entity tags (NE), while others focus on coreference relationships between POSs in text [12]. These models achieve accuracies as high as 98%, for example, while being less susceptible to the pitfalls associated with NLM approaches to AI-generated text detection.

There are further motivations for utilizing the POS approach to text analysis — the approach is computationally fast due to the low dimensionality of the POS representation. This also aids the interpretability of the model, since the representations for different parts of speech can be directly examined and compared. An a priori notion of the POS approach leads us to conclude that this approach is more robust at withstanding paraphrasing attacks, which involve the rewriting of malicious LLM-generated text by a smaller model such that it subsequently goes unmarked as malicious by detection algorithms although its human-interpretable meaning still is. The parts of speech of each individual word are less likely to be changed via a paraphrasing attack even when the words themselves change, as this would lead to a grammatically incorrect sentence, so a POS analysis should still be able to effectively detect differences between human-generated and AI-generated text. Furthermore, posts on Twitter tend to be much less formal and much more colloquial than much of the other speech that can be found on the internet, which by extension is the same data that these LLMs are trained on. A priori, we expect that current LLMs are worse at replicating ad-libbed and unrehearsed human speech than standardized written language, considering the linguistic diversity of spoken and colloquial speech. Considering that the vocabulary of a language is more numerated than the parts of speech present in a language, POS analysis reduces this aforementioned linguistic diversity of colloquial language into a simpler representation of speech from which an LM must formulate predictions. The analysis of much of what makes AI-generated language sound ”weird” to humans, like word choice and grammar, is both direct and simplified using POS analysis.

Following this analysis, we will use a neural-network-based classifier to process these features. Our approach aims to uncover nuanced patterns in the sequence of POS tags, which might be overlooked by conventional detection techniques. Each message will be transformed into an input tensor of corresponding length, and the featurized values at each position reflect aggregated information conditional on the preceding content in the sentence. This method provides interpretability, as it focuses on detecting characteristic patterns in grammatical structure that are indicative of LLM-generated text. Interpretability has practical applications, as false positives in any classification system at this scale are inevitable, making it crucial to have concrete reasoning in the event that a legitimate user is flagged as a bot.

2 Related Work

In general, AI text-detection is a potentially impossible problem. As the strength of LLMs continue to evolve, previously existing patterns may become indistinguishable when compared to human text. Furthermore, the metrics themselves can be used as a training benchmark, a consequence of Goodhart’s Law. The field is perpetually evolving, and the best approaches likely involve a large number of strategies, which we highlight here.

Existing detection approaches can be described as falling under a feature-based approach or a Neural Language Model (NLM) approach, which itself can be further divided into black-box or white-box techniques.

Feature-based approaches to text-detection will use a pre-existing Natural Language Processing (NLP) architecture to create embeddings for words in a text corpus according to some pre-determined aspect of speech to focus on. One common technique in feature-based approaches focuses on frequency features, which analyzes differences in word choice between AI-generated text and human-generated text according to Zipf’s Law, which postulates that the frequency of a word is inversely proportional to its rank in frequency, and various other axioms. Analysis on fluency features determines the quality of an AI-generated text according to its readability, which has been found to decrease with text-size in AI-generated texts, and compares that to the quality of human-generated text about similar topics [7]. Other works have found through complex phrasal feature analysis that AI-generated text is less likely to use idioms, for example, while analysis of Basic Level Text Features have sought to differentiate between AI-generated text and human-generated text through comparing aspects like the amount of punctuation present or the length of sentences in text corpora written by AI and humans. Previous research has found, however, that the efficacy of these embeddings and their predictions are fickle and can be lost quickly via minute changes in the preparation of a model, like when sampling approaches differ between training a detection model and detecting text used for the embeddings [6].

As for NLM approaches, black-box techniques aim to detect LLM-generated text solely though the output text while white-box techniques utilize the model’s parameters as well [16]. Additionally, methods can be described as either supervised learning, which utilizes labeled samples of human and AI-generated text, or zero-shot detection, which utilizes no training data [11]. Examples of zero-shot detection include perplexity-based detection [3] and ensemble approaches [2], which analyze sentence coherence and document-level features. However, these approaches may be vulnerable to paraphrasing attacks [10], and may perform poorly on short, informal text samples such as tweets [16]. Combining these two specific shortcomings of NLM approaches both alludes to the cause for the extensive prevalence of misinformation and malicious media on social media while also indicating the need for an alternative approach to detecting AI-generated text.

3 Methodology

3.1 Dataset

To train a model capable of detecting AI-generated text and analyzing its linguistic features, we required a diverse corpus of tweets from both human and AI resources. Ensuring comparability in content, sentiment, and structure was essential to isolate linguistic differences, like POS patterns, rather than confounding factors like content or sentiment disparities.

3.1.1 Human-Generated Twitter Dataset

We utilized the *Famous Keyword Twitter Replies* dataset as our primary source of human-generated text. This dataset comprises 170,255 tweet-reply pairs with the following features:

- **Keyword:** The topic or keyword that prompted the original tweet, providing context for the conversation.
- **Main Tweet:** The original tweet associated with the keyword, serving as the conversation's focal point.
- **Main Likes:** The number of likes on the main tweet, indicating its popularity or engagement level.
- **Reply:** The responses or comments to the main tweet, capturing diverse perspectives or discussions.
- **Reply Likes:** The number of likes on each reply, measuring the engagement level of individual responses.

To create a focused dataset, we restricted the data to include only the most liked reply for each tweet, resulting in 15,753 tweet-reply pairs. This selection ensures that the replies used are high-quality examples, representative of engaging or well-crafted responses.

3.1.2 AI-Generated Dataset

To generate AI replies that closely mimic human responses, we used GPT-4o, an industry-standard model known for its speed and high-quality text generation. For each main tweet in the dataset, we designed the following prompt:

"Generate a tweet in response to this tweet: '{main_tweet}' with the same sentiment as this reply: '{top_reply}'."

Additionally, the model was guided by the system role: *"You are a helpful assistant generating social media content."* This ensured that the generated replies mirrored the tone, sentiment, and brevity typical of human tweets. Replies were constrained to a maximum of 50 tokens to adhere to Twitter's 280-character limit.

The AI-generated replies were paired with the original tweets, creating a dataset where each main tweet had both a human-generated and an AI-generated reply. This augmented dataset allowed for robust model training and testing on both real and synthetic examples.

3.1.3 Long text dataset: Wikipedia introductions

We will leverage the GPT-wiki-intro dataset from Hugging Face, which contains human-written and AI-generated introductions for approximately 150,000 Wikipedia articles [1]. Each entry in the dataset includes multiple fields, including the article's title, the human-written introduction, and the AI-generated introduction. The structured nature of the dataset allows us to capture nuanced differences between human and AI-generated text, particularly in formal writing contexts.

The Wikipedia dataset will serve as our pre-training corpus, allowing our model to build a foundational understanding of content structure and sentence composition in a large-scale corpus that represents diverse topics and writing styles. The aim here is to train the model on basic language constructs and establish a baseline for distinguishing between human and AI text based on general language characteristics.

3.2 Text Preprocessing

Before feature extraction and model training, we will preprocess the text data to standardize and clean input samples across all datasets. This includes:

- Tokenization: We will tokenize each text sample into words, capturing basic syntactic elements.
- Lowercasing and Stopword removal: To remove unnecessary variations, we will lowercase all text and remove common stopwords.
- Additional preprocessing: Any URLs, emojis, and hashtags (if present in social media samples)

will be normalized or removed, as these may interfere with model consistency.

3.3 Feature Extraction

A simple a priori analysis of our chosen dataset revealed distinct differences between human-generated and AI-generated text, even before applying any NLP techniques. These differences span various dimensions, including text complexity, interpretability, vocabulary richness, and POS patterns. These findings are further illustrated through the following figures and tables:

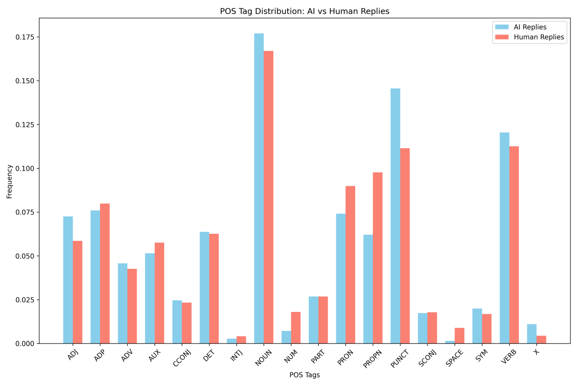

Figure 1: Frequency of POS Tags

Figure 1 illustrates the frequency of the previously enumerated POS tags across AI-generated and human-generated speech in our dataset, with AI-replies in blue and human replies in red. We see that AI-generated posts on Twitter tend to use considerably more punctuation that human-generated posts, while it tended to use less pronouns, proper nouns, and numbers.

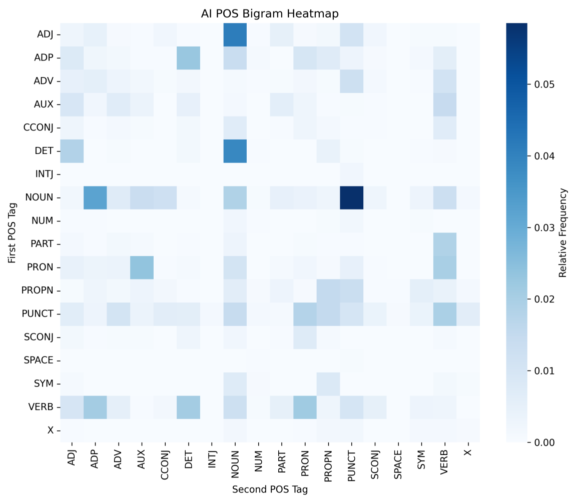

Figure 2: POS Bigram Frequencies in AI-Speech

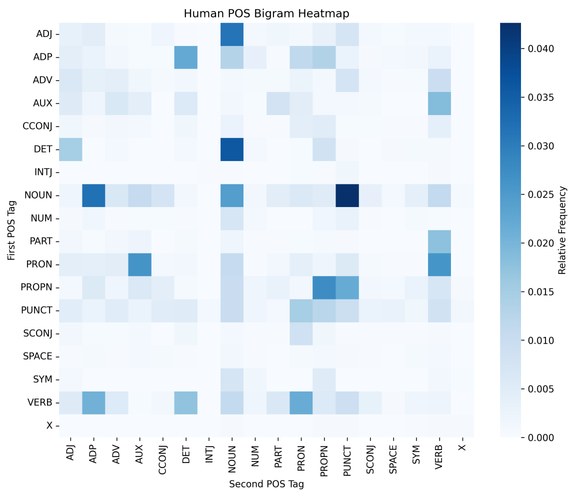

Figure 3: POS Bigram Frequencies in Human-Speech

Text Metric	Human-Generated Speech	AI-Generated Speech
Flesch-Kincaid Score	6.4	7.8
Flesch Reading Ease Score	74.1	59.7
Polysyllabic Frequency	0.094	0.144
Monosyllabic Frequency	0.717	0.651
Lexicon Count	280827	359093
Type-Token Ratio (TTR)	0.129	0.091
Percent Stop-words	36.4	31.0

Table 1: Text Metrics: Human- vs AI-Generated Speech

Figures 2 and 3 illustrate the frequencies of all possible bigrams (adjacent pairs) of POS tags in our dataset of AI- and human-generated text. Although some of the same squares in both figures closely match in terms of frequency, we can see that human-generated text seems to have more frequent usage of most types of bigrams compared to their usage in the AI-generated text, indicating that human-generated text tends to be more grammatically varied than AI-generated text.

Table 1 illustrates how the AI- and human-generated texts differ from each other in terms of how they score on certain text complexity metrics. On top of the aforementioned Flesch-Kincaid scores, we utilized polysyllabic frequency, the frequency of words with three or more syllables; monosyllabic frequency, the frequency of words with one syllable; lexicon count, the number of distinct words in both texts; the type-token ratio

(TTR), which measures the lexical variety of a text; and the percentage of stopwords, which measures the percentage of words used that do not contribute to lexical meaning. Higher-complexity texts will have a higher Flesch-Kincaid Score, polysyllabic frequency, lexicon count, and TTR, while having a higher Flesch Reading Ease Score, monosyllabic frequency, and percent stopwords are indicative of lower-complexity text. This table therefore indicates that AI-generated speech is generally more complex than human-generated speech while displaying different POS patterns.

3.4 AI-text Detection via Machine Learning

In order to determine the efficacy of POS analysis in detecting AI-generated text, we trained and evaluated various model architectures for our binary classification task and compared it to various benchmarks. Our various approaches are described as follows:

- **Fine-tuned RoBERTa**: In order to benchmark the performance of our POS-based approaches, we employed a pretrained RoBERTa text encoder model as a baseline due to its strong and flexible performance in a variety of NLP tasks. The RoBERTa text encoder model consists of an embedding layer, which converts each text token into an embedding of size 784, followed by 12 transformer blocks in series. We augmented this model with a classification head, consisting of a $[784, 2]$ Linear Layer, followed by a softmax activation function in order to perform classification tasks. The pre-trained RoBERTa weights were loaded, and our model was fine-tuned on our training datasets to perform classification.
- **POS FFNN**: (TODO: LUKE)
- **POS Transformer**: The SpaCy library was used to perform POS analysis of text, and one-hot encoding was used to represent each word's simple POS tag, detailed POS tag, and syntactical dependency. In our embedding layer, we embedded each of these one-hot encodings into embeddings of dimensions $[10, 30, 100, 300]$ and combined them with positional embeddings. These embeddings were passed through a series of 12 transformer blocks in series and a binary classification head as described previously.

3.5 Training

All models were trained using the AdamW optimizer with a learning rate of $1e-5$, a batch size of 8, and a cross-entropy loss function. Models that were trained on the Twitter dataset were trained for 3 epochs, while models that were trained on the Wikipedia dataset were trained for 1 epoch. An $80-20$ train test split was used for both datasets and all model architectures.

4 Results

In addition to our suite of POS-based models, we benchmark against GPTZero [15], which assesses the perplexity of its input text. The implementation we used did not provide a strict confidence level (instead classifying texts as human, AI, or possibly AI), so we have omitted its AUC score, treating "possibly AI" as "human" for a better false-positive score. Also, in rare cases, the implementation we used would throw an error when evaluating its input text, so we have omitted them from the probabilities.

4.1 Model Performance

We compared the performance of various approaches using numerous metrics, including raw accuracy, ROC-AUC score, which measures how well a classifier can distinguish between positive and negative classes, and F1 score, which measures a model's accuracy by combining its precision and recall scores. While accuracy is always important, other metrics may provide a more complete picture of performance. For example, a higher ROC-AUC score indicates that a model is more likely to rank positives higher than negatives; information which may not be captured by accuracy alone. Furthermore, early AI text detection models suffered from high false positive rates, which has resulted in false accusations of plagiarism [9] and could more generally result in unjustified bans from social media platforms. Thus, keeping the false-positive rate low is a priority.

As expected, we see that the POS models perform better with larger embedding size, with an embedding size of 300 achieving around 90% test accuracy for both datasets, compared to the approximately 75% accuracy for an embedding size of 10. This suggests that larger embedding dimensions are able to capture more information about and more complex interactions between our POS tags. Our POS models outperformed the GPTZero benchmark in both datasets, but especially in the Twitter dataset, confirming that GPTZero tends to perform worse on shorter text samples. As expected, The fine-tuned RoBERTa performs exceptionally well on both tasks. However, the performance of our POS models suggests that AI generated text does have specific grammatical patterns that can be used to detect them with reasonable success.

		Twitter Weights			Wikipedia Weights		
	Model	Accuracy	AUC	F1	Accuracy	AUC	F1
Same	GPTZero	0.541	-	0.330	0.762	-	0.798
	RoBERTa Fine-tuning	0.978	0.997	0.977	0.999	0.999	0.999
	POS FFNN-10	0.852	0.919	0.851	0.835	0.914	0.837
	POS FFNN-30	0.877	0.939	0.875	0.867	0.947	0.873
	POS FFNN-100	0.887	0.950	0.885	0.887	0.957	0.887
	POS FFNN-300	0.887	0.951	0.887	0.888	0.960	0.892
	POS Transformer-10	0.762	0.828	0.699	0.805	0.847	0.762
	POS Transformer-30	0.821	0.899	0.821	0.875	0.959	0.861
	POS Transformer-100	0.872	0.947	0.876	0.924	0.981	0.925
	POS Transformer-300	0.892	0.954	0.889	0.939	0.987	0.940
Different	RoBERTa Fine-tuning	0.494	0.744	0.661	0.534	0.851	0.138
	POS FFNN-300	0.499	0.477	0.043	0.397	0.334	0.473
	POS Transformer-300	0.497	0.406	0.051	0.494	0.543	0.661
Paraphrase	GPTZero	0.308	-	-	-	-	-
	RoBERTa Fine-tuning	0.190	-	-	1.000	-	-
	POS FFNN-300	0.175	-	-	0.785	-	-
	POS Transformer-300	0.005	-	-	1.000	-	-

Table 2: Accuracies for each model. In the first batch, we directly compare each model using the dataset it was trained with. Since GPTZero was not fine-tuned, we include its results here. In the second batch, we test each model using the test split from the other set of data (e.g. Twitter Weights evaluated with Wikipedia data). In the third section, we list the results for each model on the paraphrased Twitter set.

4.2 Transferability

For each architecture, separate models were trained independently on the Twitter dataset and the Wikipedia dataset. However, model performance was also evaluated on the dataset that they were not trained on, in an attempt to measure transferability across text-domains. The results are found in the second division of Table 2.

Overall, the models performed worse across the board, most of them around the 50% threshold. This suggests that the domain in which text originates from matters tremendously, as it informs the context by which text is generated and can therefore result in different patterns emerging. The exceptions are with the models trained on the Wikipedia dataset, which could potentially be caused by the larger training data (both in input size and dataset size), which allowed it to generalize to some degree on shorter text domains, such as tweets.

These findings suggest that our models may be able to detect the specific grammatical structures that are characteristic of AI generated text performing certain tasks. However, the poor transferability indicates that [FINISH THOUGHT]

4.3 Paraphrasing Attacks

We paraphrased a subset of the test split of the Twitter dataset using DIPPER [10], consisting of 724 messages. For each input, we provide both the reference tweet as the prompt and the initial AI-generated tweet as the sample text, and the model returns paraphrased text. We then evaluate each text on the models. Importantly, we only paraphrase the AI-generated messages, excluding human messages since paraphrasing attacks aren't meant for those types of messages. As a consequence, we omit the AUC and F1 scores, measuring only raw accuracy.

As a brief investigation into the performance of this method, we collect similar surface-level lexical metrics in Table 3. We find that the paraphrased AI text was more similar to the human-generated tweets than the original AI-generated tweets. This suggests that paraphrasing successfully masks certain tell-tale signs of AI-generated text, bringing it closer to human-like text.

Indeed, we see that the models generally perform worse across the board, in some extreme cases identifying every text as human-generated. Curiously, the pure RoBERTa model seemed to demonstrate the opposite effect, correctly identifying every example in the Wikipedia dataset as AI-generated. Moving forward, we do see that larger embedding sizes seem to increase the accuracy with the Wikipedia set, but it seems the small text size of the Twitter dataset completely invalidates any learned representation.

Text Metric	Human-Generated Speech	Paraphrased AI Speech
Flesch-Kincaid Score	6.4	5.3
Flesch Reading Ease Score	74.1	76.9
Polysyllabic Frequency	0.094	0.102
Monosyllabic Frequency	0.717	0.738
Percent Stop-words	36.4	40.8

Table 3: Text Metrics: Human- vs Paraphrased AI Speech

5 Future Work

When implementing the project, we considered other standard text-representing metrics. However, they proved challenging to integrate because our models are token-based, while these statistics tended to reflect information about the text as a whole. Examples of metrics we considered include Flesch-Kincaid readability scores, which estimates syntactic complexity through the densities of words and syllables within the content, and the type-token ratio, a measure of word diversity within the text. In addition to the difficulties found above, we believe these would prove noisy when applied to short texts like tweets.

With regards to the broader field of AI text generation, another promising area of research is with hybrid implementations. By Goodhart's Law or some other criteria, no singular metric is permanently infallible, as efforts can be made to specifically hack the metric, with a strong chance of success. Ensemble methods offer prolonged longevity in this regard, as a model that leverages a large suite of metrics can potentially form a more nuanced interpretation of its input text that's harder to crack. Such implementations have been discussed before, e.g. in [17], so we could see how they fare with our methods included.

References

[1] Aaditya Bhat. Gpt-wiki-intro (revision 0e458f5), 2023.

[2] Harika Abburi, Kalyani Roy, Michael Suesserman, Nirmala Pudota, Balaji Veeramani, Edward Bowen, and Sanmitra Bhattacharya. A simple yet efficient ensemble approach for ai-generated text detection, 2023.

[3] Megha Chakraborty, S.M Towhidul Islam Tonmoy, S M Mehedi Zaman, Shreya Gautam, Tanay Kumar, Krish Sharma, Niyar Barman, Chandan Gupta, Vinija Jain, Aman Chadha, Amit Sheth, and Amitava Das. Counter Turing test (CT2): AI-generated text detection is not as easy as you may think - introducing AI detectability index (ADI). In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2206–2239, Singapore, December 2023. Association for Computational Linguistics.

[4] Giulio Corsi, Bill Marino, and Willow Wong. The spread of synthetic media on x. *Shorenstein Center on Media, Politics, and Policy*, 2024.

[5] Pierce D. Ekstrom and Calvin K. Lai. The selective communication of political information. *Social Psychological and Personality Science*, 12(5):789–800, 2021.

[6] Leon Fröhling and Arkaitz Zubiaga. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7, 2021.

[7] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020.

[8] Kristina Hook and Ernesto Verdeja. Social media misinformation and the prevention of political instability and mass atrocities. *The Henry L. Stimson Center*, 2022.

[9] Miles Klee. Professor flunks all his students after chatgpt falsely claims it wrote their papers. *Rolling Stone*.

[10] Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense, 2023.

[11] Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. Mage: Machine-generated text detection in the wild, 2024.

[12] Hoang-Quoc Nguyen-Son, Ngoc-Dung T. Tieu, Huy H. Nguyen, Junichi Yamagishi, and Isao Echi Zen. Identifying computer-generated

text using statistical analysis. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1504–1511, 2017.

[13] Giovanni Spitale, Nikola Biller-Andorno, and Federico Germani. Ai model gpt-3 (dis)informs us better than humans. *Science Advances*, 9(26), June 2023.

[14] Zhaoxuan Tan, Shangbin Feng, Melanie Sclar, Herun Wan, Minnan Luo, Yejin Choi, and Yulia Tsvetkov. Botpercent: Estimating bot populations in twitter communities, 2023.

[15] Edward Tian and Alexander Cui. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods, 2023.

[16] Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. Seqxgpt: Sentence-level ai-generated text detection, 2023.

[17] Ye Zhang, Qian Leng, Mengran Zhu, Rui Ding, Yue Wu, Jintong Song, and Yulu Gong. Enhancing text authenticity: A novel hybrid approach for ai-generated text detection, 2024.

A Ethics Statement

We collected example messages from Twitter to generate our dataset. All messages have been anonymized and were publically accessible, with no personal information being collected. As we also trained our models on data partially generated through ChatGPT, there is a slight possibility that they learn similar biases (though this is unlikely due to the very high level of abstraction of the model's representations).

As discussed in the paper, all strategies for identifying artificially-generated text can be "hacked" by subsequent LLM iterations, so this research has a (very slight) chance of contributing to this issue. We emphasize that much of this research can be viewed as an exploration into patterns of LLM-generated text used for non-malicious contexts as well, since not every application involves falsifying an internet personality.