The Science of Detecting LLM-Generated Texts

Ruixiang Tang, Yu-Neng Chuang, Xia Hu
Department of Computer Science, Rice University
Houston, USA
{rt39,ynchuang,xia.hu}@rice.edu

Abstract

The emergence of large language models (LLMs) has resulted in the production of LLM-generated texts that is highly sophisticated and almost indistinguishable from texts written by humans. However, this has also sparked concerns about the potential misuse of such texts, such as spreading misinformation and causing disruptions in the education system. Although many detection approaches have been proposed, a comprehensive understanding of the achievements and challenges is still lacking. This survey aims to provide an overview of existing LLM-generated text detection techniques and enhance the control and regulation of language generation models. Furthermore, we emphasize crucial considerations for future research, including the development of comprehensive evaluation metrics and the threat posed by open-source LLMs, to drive progress in the area of LLM-generated text detection.

1 Introduction

Recent advancements in natural language generation (NLG) technology have significantly improved the diversity, control, and quality of LLM-generated texts. A notable example is OpenAI's ChatGPT [50], which demonstrates exceptional performance in tasks such as answering questions, composing emails, essays, and codes. However, this newfound capability to produce human-like texts at high efficiency also raises concerns in detecting and preventing misuses of LLMs in tasks such as phishing [4], disinformation [74], and academic dishonesty [63]. For instance, many schools banned ChatGPT due to concerns over cheating in assignments [59], and media outlets have raised the alarm over fake news generated by LLMs [25]. These concerns about the misuse of LLMs have hindered the NLG application in important domains such as media and education.

The ability to accurately detect LLM-generated texts is critical for realizing the full potential of NLG while minimizing serious consequences. From the perspective of end-users, LLM-generated text detection could increase trust in NLG systems and encourage adoption [73]. For machine learning system developers and researchers, the detector can aid in tracing generated texts and preventing unauthorized use. Given its significance, there has been a growing interest in academia and industry to pursue research on LLM-generated text detection and to deepen our understanding of its underlying mechanisms.

While there is a rising discussion on whether LLM-generated texts could be properly detected and how this can be done, we provide a comprehensive technical introduction of existing detection methods which can be roughly grouped into two categories: black-box detection and white-box detection. Black-box detection methods are limited to API-level access to LLMs. They rely on collecting text samples from human and machine sources, respectively, to train a classification model that can be used to discriminate between LLM- and human-generated texts. Black-box detectors work well because current LLM-generated texts often show linguistic or statistical patterns. However, as LLMs evolve and improve, black-box methods are becoming less effective. An alternative is white-box detection, in this scenario, the detector has full access to the LLMs and can control the model's generation behavior for traceability purposes. In practice, black-box detectors are commonly constructed by external entities, whereas white-box detection is generally carried out by LLM developers.

This article is to discuss the timely topic from a data mining and natural language processing perspective. Specifically, we first outline the black-box detection methods in terms of a data analytic life cycle, including data collection, feature selection, and classification model design. We then delve into more recent advancements in white-box detection methods, such as post-hoc watermarks and inference time watermarks. Finally, we present the limitations and concerns of current detection studies and suggest potential future research avenues. We aim to unleash the potential of powerful LLMs by providing fundamental concepts, algorithms, and case studies for detecting LLM-generated texts.

2 Prevalence and Impact

How good is LLM-generated text, and what is its impact on individuals and society? The recent advancement of OpenAI's ChatGPT gives us a glimpse of its potential. The convenience brought by ChatGPT shows promise in enhancing efficiency in industries and education systems. For example, ChatGPT's ability to perform well on tests, such as MBA exams at Wharton Business School [60], highlights its competitiveness with human knowledge and its potential to assist professionals [3, 10]. In the healthcare industry, ChatGPT can simplify documentation by generating medical records, progress reports, and discharge summaries, helping medical students and physicians to create effective memories of complex medical concepts in clear language [42]. In emergency

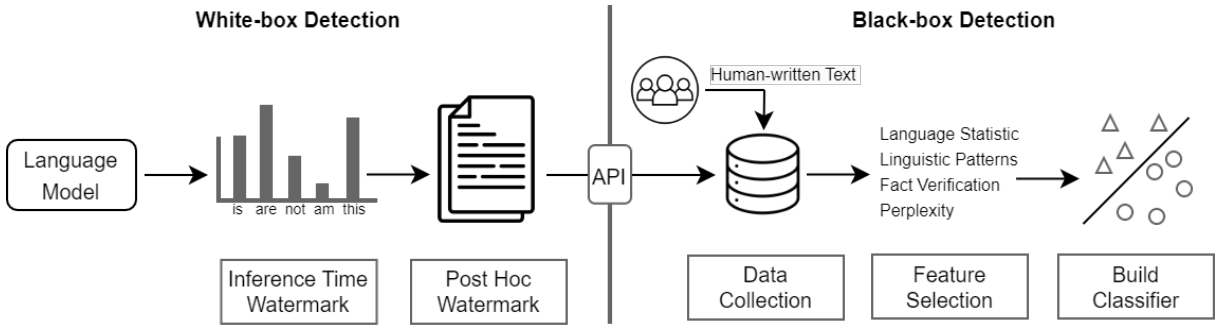

Figure 1. An overview of the LLM-generated text detection.

services, ChatGPT can help generate real-time police reports, reducing response times for officers in action.

However, the integration of ChatGPT in education has raised concerns among experts. The use of AI technology in education has sparked worries about cheating, as students may use it to gain unauthorized access to information and complete assignments. The tool may offer quick answers, but it does not promote critical-thinking and problem-solving skills, which are vital for academic and long-term success. As a response, New York City Public Schools have banned the use of ChatGPT in academic papers, and teaching statements [59]. This underscores the need for careful ethical considerations and guidelines when incorporating LLM technology into education. Additionally, organizations such as the International Conference on Machine Learning and publishers like Nature have stated that LLM-generated texts cannot be credited as an author in their conference papers and journals [21], which leads to an urgent need to distinguish LLM-generated texts from human-written texts. To address the issue of cheating in academia, OpenAI and third-parties have recently developed several tools to help teachers and institutions identify potential instances of academic misconduct [51, 65]. Although not foolproof, these tools manage to preserve academic integrity in education systems and research institutions. Given its significance, there has been a growing interest in academia and industry to pursue research on LLM-generated text detection.

3 Black-box Detection

zIPFANIn the realm of black-box detection, external entities are restricted to API-level access to the LLM, as illustrated in Figure 1. To construct an effective detector, black-box methods require the collection of text samples from both human-generated and machine-generated sources. Subsequently, a classifier is trained to differentiate between the two categories based on chosen features. This paper presents the three crucial components of black-box text detection: data collection, feature selection, and the implementation of the classification model.

3.1 Data Collection

The performance and generalizability of black-box detection models are heavily dependent on the quality and diversity of the collected data. Recently, there has been an increasing number of studies that focus on gathering LLM-generated responses and comparing them to human-written texts across various domains. This section delves into the various strategies for obtaining data from both human and machine sources.

3.1.1 LLM-generated Data: LLM is trained to estimate the probability of the next token in a sequence, given the preceding words. Recent advancements in Natural Language Generation have led to the development of LLMs for various domains, including question answering [32], news generation [12, 52, 68], and story creation [24]. When collecting datasets for LLM-generated texts, it is essential to specify the target domain and generation model. The majority of studies utilize transformer-based LLMs [71], such as GPT-2 [56], GPT-3 [8], OPT [75], and ChatGPT [50], for text generation within specific domains. Typically, LLMs with a larger number of parameters typically produce higher-quality text, but also demand more computational resources. Appropriate LLM selection or fine-tuning before use can significantly improve the quality of the generated texts [20]. For instance, Solaiman et al. fine-tune the GPT-2 model on Amazon product reviews, producing reviews with a style consistent with those found on Amazon [62].

Language models are known to generate text with format and style issues. To avoid these artifacts, researchers can provide domain-specific prompts or constraints before generating the outputs. For instance, Clark et al. [12] randomly selected 50 articles from Newspaper3k to use as prompts for the GPT-3 model for news generation and set the filtering constraints on the models with the phrase "Once upon a time" for story creation. The length of LLM-generated texts can also be controlled with short prompts and a specified number of word constraints [49, 55]. The sampling strategy also has a significant impact on the generated text quality and style. While greedy algorithms like beam search [66]

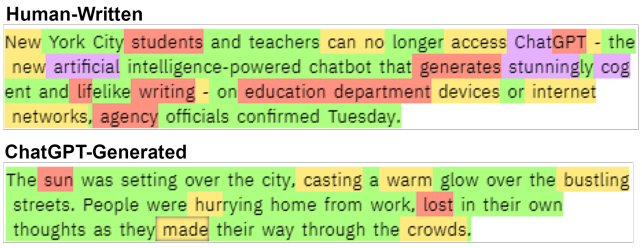

Figure 2. The top-k overlay using visualization tool GLTR [28]. There is a notable difference between the two texts. The human-written text is from Chalkbeat New York [22].

generate the most probable sequence, they are deterministic and may not allow for creativity and language diversity. On the other hand, stochastic algorithms like nucleus sampling [35] preserve some randomness while eliminating poor candidates, making them more suitable for free-form generation tasks.

3.1.2 Human-written Data. Manual composition by humans is a natural way to obtain human-written data. For instance, in the study conducted by Dugan et al. [19], the author aims to evaluate the quality of NLG systems and assess human perceptions of generated texts. To accomplish this, two hundred Amazon Mechanical Turk workers [16] are hired to complete 10 annotations on the website and provide a natural language explanation for their decisions. However, manually gathering data through human effort can be time-consuming and financially unfeasible for large datasets. An alternative approach is to extract texts directly from human-written sources, such as websites and scientific articles. For example, we can easily gather thousands of descriptions of computer science concepts from Wikipedia, written by human experts, to answer questions like "What is <concept>?" [32]. Furthermore, many publicly available benchmark datasets, such as ELI5 [23], which consists of 270K threads from the Reddit forum "Explain Like I'm Five", already provide human-written texts in a structured form. Collecting human-written texts from these readily available sources can significantly reduce the time and cost involved, but it's crucial to consider sampling biases and topic diversity, such as including the written texts of non-native speakers. Additionally, since LLMs are trained on text sampled from a specific time period, it's important to ensure that the publication date of the human-generated data is close to or after the training data of the LLM [28].

3.1.3 Human Evaluation Findings: Previous research has provided valuable insights into how to distinguish LLM-generated texts from human-written texts through human evaluations. Firstly, it has been noted that LLM-generated texts are less emotional and objective compared to human-written text, which often uses punctuation and grammar to convey subjective feelings [32, 69]. For example, human authors frequently use exclamation marks, question marks, and ellipsis to express their emotions, while LLMs generate answers that are more formal and structured. However, it's important to note that LLM-generated texts may not always be accurate or helpful as they can be fabricated [32]. At the sentence level, research has shown that human-written text is more coherent than LLM-generated text, which tends to repeat terms within a paragraph [20, 43]. These observations suggest that LLMs leave distinctive signals in their generated text, and appropriate features can be selected to distinguish between LLM and human-written text.

3.2 Detection Feature Selection

How can we differentiate between LM-generated texts and human-written texts? This section will discuss possible detection features from various perspectives, including statistical disparities, linguistic patterns, and fact verification.

3.2.1 Statistical Disparities. The detection of statistical disparities between LLM-generated and human-written texts is achieved through the use of various statistical metrics, such as Term Frequency-Inverse Document Frequency (TF-IDF) [62], Self-BLEU [77], and the Zipfian coefficient [54]. The Zipfian coefficient measures the text's conformity to an exponential curve, which is described by Zipf's law [35]. On the other hand, the Self-BLEU [77] score evaluates the diversity and consistency of the n-grams present in the generated text. A visual forensic tool, GLTR [28], has been developed to detect generation artifacts across common sampling methods, as demonstrated in Figure 2. Perplexity is another commonly used metric for LLM-generated text detection. It measures the quality of the language model by quantifying the negative average log-likelihood of the texts under the LLM [7, 24, 35]. Studies have shown that language models tend to concentrate on common patterns in the texts they were trained on, resulting in low perplexity scores for LLM-generated text. Conversely, human authors have the ability to express themselves in a wide range of styles, which makes it more challenging for language models to predict and results in higher perplexity values for human-written text. However, it should be noted that these statistical disparities are limited by the requirement of having a document-level text, which inevitably reduces the resolution of the detection [51], as shown in Figure 3.

3.2.2 Linguistic Patterns. The linguistic patterns in the human and LLM-generated texts can be analyzed through various contextual properties, including vocabulary features, part-of-speech, dependency parsing, and sentiment analysis. The vocabulary features provide insight into the queried text's word usage patterns by analyzing characteristics such as average word length, vocabulary size, and word density. Previous studies on OpenAI's ChatGPT have shown that

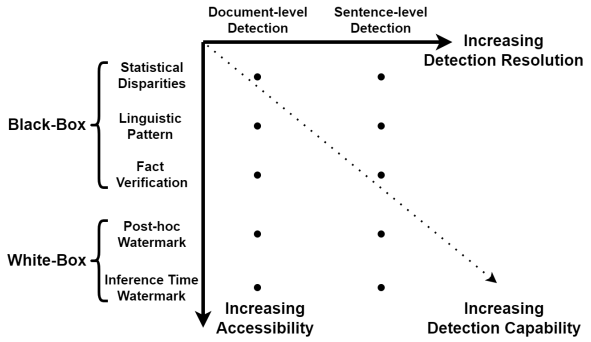

Figure 3. A taxonomy of LLM-generated text detection.

human-written texts tend to have a more diverse vocabulary but shorter length compared to language models in question-answering tasks [32]. The part-of-speech analysis highlights the dominance of nouns in ChatGPT texts, implying argumentativeness and objectivity, while the dependency parsing analysis shows that ChatGPT texts use more determiners, conjunctions, and auxiliary relations. Sentiment analysis, on the other hand, provides a measure of the emotional tone and mood expressed in the text. Unlike humans, large language models tend to be neutral by default and lack emotional expression. Research has shown that ChatGPT expresses significantly less negative emotion and hate speech compared to human-written texts [32]. Besides analyzing a single text, there are plenty of linguistic patterns in the multi-turn conversation [5]. These linguistic patterns are a reflection of the training data and strategies of LLMs and serve as valuable features for detecting LLM-generated text. However, it is worth noting that LLMs can significantly change their linguistic patterns in response to prompts. For example, adding the prompt "Please respond with humor" can alter the sentiment of the LLM's answer and affect the robustness of linguistic patterns.

3.2.3 Fact Verification. Language models (LLMs) often rely on likelihood maximization objectives during training, which can result in the generation of nonsensical or inconsistent text, known as hallucination [38]. This phenomenon emphasizes the significance of fact-verification as a crucial feature for detection [76]. For example, OpenAI's ChatGPT has been reported to generate false scientific abstracts [9] and post misleading news opinions [31]. Studies have revealed that popular decoding methods, such as top-k and nucleus sampling, result in more diverse and less repetitive generations. However, they also produce text that is less verifiable [46]. These findings highlight the potential to use fact verification to detect LLM-generated texts.

Prior research has advanced the development of tools and algorithms for conducting fact verification, which entails retrieving evidence for claims, evaluating consistency and relevance, and detecting inconsistencies in texts. One strategy is to use sentence-level evidence, such as extracting facts from Wikipedia, to directly verify facts of a sentence [46]. Ma et al. [45] employed representation learning to embed sentence-level evidence based on coherence modeling and natural language inference, leading to a deeper understanding of the text's semantics. Another approach is to analyze document-level evidence via graph structures, which capture the factual structure of the document as an entity graph. This graph is utilized to learn sentence representations with a graph neural network, followed by the composition of sentence representations into a document representation for fact verification. This method has revealed that human-written text tends to repeat terms, while LLM-generated text often includes irrelevant information [76]. Some studies also use knowledge graphs constructed from truth sources, such as Wikipedia, to conduct fact verification [11, 61, 64]. These methods evaluate consistency by querying subgraphs and identify non-factual information by iterating through entities and relations. Given that human-written text may also contain misinformation, it is crucial to supplement the detection results with other features in order to accurately distinguish texts generated by LLMs.

3.3 Classification Model

The detection task is typically approached as a binary classification problem, with the objective of capturing textual features that differentiate between human-written and LLM-generated texts. This section provides an overview of the major categories of classification models.

3.3.1 Traditional Classification Algorithm. Traditional classification algorithms utilize various features outlined in Section 3.2 to differentiate between human-written and LLM-generated text. Some of the commonly used algorithms are Support Vector Machines, Naive Bayes, and Decision Trees. For instance, Fröhling et al. [26] in their study use linear regression, SVM, and random forests models that were built based on both statistical and linguistic features and successfully identified texts generated by GPT-2, GPT-3, and Grover models. Similarly, Solaiman et al. [62] achieve a decent performance in identifying texts generated by GPT-2 (1.5 billion parameters) through a combination of TF-IDF unigram and bigram features with a logistic regression model. In addition, studies have also shown that using pre-trained language models to extract textual features, followed by SVM for classification, can outperform the direct use of statistical features [15]. One advantage of traditional classification algorithms is that they are interpretable, allowing researchers to analyze the importance of input features.

3.3.2 Deep Learning Approaches. In addition to relying on extracted features for detection, the use of language models, such as BERT [18] and RoBERTa [44], as a backbone has been explored in recent studies. This approach involves fine-tuning these models on a mixture of human-written and

LLM-generated text, allowing them to capture the textual differences between the two implicitly. Most studies adopt the supervised learning paradigm for training the language model, as demonstrated by Ippolito et al. [36] who fine-tuned the BERT model on a collected dataset of generated-text pairs. This study showed that human raters have significantly lower accuracy than automatic discriminators in identifying LLM-generated text. In a low-resource scenario, Rodriguez et al. [58] showed that a few hundred labeled in-domain genuine and synthetic texts are sufficient for good performance, even when the external entity does not have complete information about the LLM text generation pipeline. Despite the strong performance under the supervised learning paradigms, the annotations of detection data are sometimes challenging to acquire in real-world applications, leading the supervised paradigms to be not applicable. A recent research [27] detects the LLM-generated documents by leveraging the repeated higher order of n-grams, which can then be trained under unsupervised learning paradigms, without the requirements of collecting LLM-generated datasets as training data. Besides using the language model as the backbone, recent research finds that contextual structure can be viewed as a graph containing entities mentioned in the texts and the semantically relevant relations, which utilizes a deep graph neural network to capture the structure feature of a document for LLM-generated news detection [76]. While deep learning approaches often achieve better detection results, their black-box nature severely limits interpretability. Researchers typically need to employ interpretation tools to understand the basis for the model's decisions.

4 White-box Detection

In white-box detection, the detector has full access to the target language model, allowing the embedding of secret watermarks into its outputs for monitoring any suspicious or unauthorized activity. In this section, we first present the three requirements for watermarks in natural language generation, followed by an overview of the two main categories of white-box watermarking approaches: post-hoc watermarking and inference-time watermarking.

4.1 Watermarking Requirements

We build upon previous research in traditional digital watermarking [14] and propose three essential requirements for NLG watermarking. (1) Effectiveness: The watermark must be effectively embedded into the generated texts and verifiable, at the same time, maintaining the quality of the generated texts. (2) Secrecy: The watermark should be designed to achieve stealthiness, without introducing noticeable changes that can be easily detected by automated classifiers. In practice, it should be indistinguishable from non-watermarked texts. (3) Robustness: The watermark should be resilient and difficult to remove through common modifications such as synonym replacement. To remove the watermark, the attacker must make significant modifications that render the texts unusable. These three requirements serve as the foundation for NLG watermarking and ensure the traceability of LLM-generated texts.

4.2 Post-hoc Watermarking

Given an LLM-generated text, post-hoc watermarks will embed a hidden message or identifier into the text. Verification of the watermark can be performed by recovering the hidden message from the suspicious text. There are two main categories of post-hoc watermarking methods: rule-based and neural-based approaches.

4.2.1 Rule-based Approaches. Initially, nature language researchers adapted techniques from multimedia watermarking, which were non-linguistic in nature and relied heavily on character changes. For example, the line-shift watermark method involves moving a line of text upward or downward (or left or right) based on the binary signal (watermark) to be inserted [6]. However, these "printed text" watermarking approaches had limited applicability and were not robust against text reformatting [39]. Later research shifted towards using the syntactic structure for watermarking. For instance, a study by Atallah et al. [2] embedded watermarks in parsed syntactic tree structures, preserving the meaning of the original texts and making it unreadable to those without knowledge of the modified tree structure. Additionally, syntactic tree structures are difficult to remove through editing and remain effective when the text is translated into other languages. Further improvements were made in a series of works, which proposed variants of the method that embedded watermarks based on synonym tables instead of just parse trees [37, 47, 48]. In addition to syntactic structure, researchers have also leveraged the semantic structure of text to embed watermarks. This includes exploiting features such as verbs, nouns, prepositions, spelling, acronyms, grammar rules, etc. For instance, a synonym substitution approach was proposed in which watermarks are embedded by replacing certain words with their synonyms without altering the context of the text [67]. Generally, rule-based methods use fixed rule-based substitutions, which may systematically change the text statistics, undermining the secrecy of the watermark and enabling adversaries to detect and remove the watermark automatically.

4.2.2 Neural-based Approaches. In contrast to rule-based methods that demand significant engineering efforts to design, neural-based approaches conceptualize the information-hiding process as an end-to-end learning process. These approaches typically involve three components: a watermark encoder network, a watermark decoder network, and a discriminator network [1, 17, 70]. Given a target text and a secret message (e.g., random binary bits), the watermark encoder network generates a modified text that incorporates

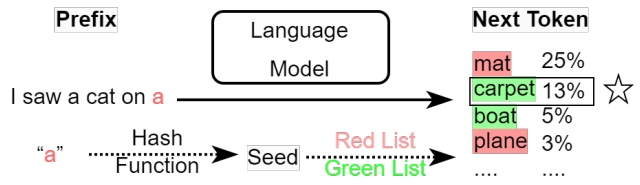

Figure 4. Illustration of inference time watermark. A random seed is generated by hashing the previously predicted token "a", splitting the whole vocabulary into "green list" and "red list". The next token "carpet" is chosen from the green list.

the secret message. The watermark decoder network then endeavors to retrieve the secret message from the modified text. One challenge is that the watermark encoder network may significantly alter the language statistics. To address this problem, the framework employs an adversarial training strategy [30] and includes the discriminator network. The discriminator network takes the target text and watermarked text as input and aims to differentiate between them, while the watermark decoder network aims to make them indistinguishable. The training process continues until the objectives of the three components attain a satisfactory level of performance. For watermarking LLM-generated text, developers can use the watermark encoder network to embed a pre-set secret message into LLMs' outputs, and the watermark decoder network to detect any potential text generated by the model. Although neural-based approaches eliminate the need for manual rule design, their inherent lack of interpretability raises concerns about their truthfulness and the absence of mathematical guarantees for the watermark's effectiveness, secrecy, and robustness.

4.3 Inference Time Watermark

In contrast to post-hoc watermarks that are added after text generation, inference-time watermarks target the decoding process within the LLM. The LLM neural language model generates a probability distribution for the next word in a sequence based on the previous words. A decoding strategy, which is an algorithm that selects words from this distribution to generate a sequence, provides an opportunity to embed the watermark by altering the word selection process.

A representative example of this method can be found in research conducted by Kirchenbauer et al. [41]. During the next token generation, a hash code is generated based on the previously generated token, which is then used to seed a random number generator. This seed randomly divides the whole vocabulary into a "green list" and a "red list" of equal size. The next token is subsequently generated from the green list. In this way, the watermark is embedded into every generated word, as depicted in Figure. 4. To detect the watermark, a third party with knowledge of the hash function and random number generator can reproduce the red list for each token and count the number of violations of the red list rule, thus verifying the authenticity of the text. The probability that a natural source produces N tokens without violating the red list rule is only $1/2^N$, which is vanishingly small even for text fragments with a few dozen words. To remove the watermark, adversaries need to modify at least half of the document's tokens. However, one concern with these inference-time watermarks is that the controlled sampling process may significantly impact the quality of the generated text. One solution is to relax the watermarking constraints, e.g., increasing the whitelist vocabulary size [41], and aim for a balance between watermarking and text quality.

5 Authors' Concerns

5.1 Limitations of Black-box Detection.

Bias in Collected Datasets. Data collection plays a vital role in the development of black-box detectors, as these systems rely on the data they are trained on to learn how to identify detection signals. However, it is important to note that the data collection process can introduce biases that can negatively impact the performance and generalization of the detector. These biases can take several forms. For example, many existing studies tend to focus on only one or a few specific tasks, such as question-answering or news generation, which can lead to an imbalanced distribution of topics in the data and limit the detector's ability to generalize. Additionally, human artifacts can easily be introduced during data collection, as seen in the study conducted by Guo et al. [32], where the lack of style instruction in collecting LLM-generated answers led to OpenAI's ChatGPT producing answers with a neutral sentiment. These spurious correlations can be captured and even amplified by the detector, leading to poor generalization performance when deployed in real-world applications [29].

Confidence Calibration. In the development of real-world detection systems, it's crucial not only to have accurate classifications but also to provide an indication of the likelihood of being incorrect. For instance, a text with a 98% probability of being generated by an LLM should be considered more likely to be machine-generated than one with a 90% probability. In other words, the predicted class probabilities should reflect its ground truth correctness likelihood. Calibrated confidence scores are also important for model interpretability, as they provide valuable information for users to establish trust in the system [13]. Good confidence scores help build trust in the user, especially for neural networks whose decisions can be difficult to interpret. Although neural networks are more accurate than traditional classification models, extensive studies have pointed out that they are no longer well-calibrated [33, 53]. Therefore, it's essential to calibrate the confidence scores in black-box detection classifiers, which are often neural-based models.

In our opinion, while black-box detection works at present due to detectable signals left by language models in generated text, it will gradually become less viable as language model capabilities advance and ultimately become infeasible. In light of the rapid improvement in LLM-generated text quality, the future of reliable detection tools lies in white-box watermarking detection approaches.

5.2 Lacking Comprehensive Evaluation Metrics

Existing studies often rely on metrics such as AUC or accuracy for evaluating detection performance. However, these metrics only consider an average case and are not enough for security analysis. Consider comparing two detectors: Detector A perfectly identify of 1% of the LLM-generated texts but succeeds with a random 50% chance on the rest. Detector B succeeds with 50.5% on all data. On average, two detectors have the same detection accuracy or AUC. However, detector A demonstrates exceptional potency, while detector B is practically ineffective. In order to know if the detector can reliably identify the LLM-generated text, researchers need to consider the low false-positive rate regime (FPR) and report a detector's True-Positive Rate (TPR) at a low false-positive rate. This objective of designing methods around low false-positive regimes is widely used in computer security [34, 40]. This is especially crucial for populations who produce unusual text, such as non-native speakers. Such populations might be especially at risk for false-positive, which could lead to serious consequences if these detectors are used in our education systems.

5.3 Threats from Open-Source LLMs.

Current detection methods are based on the assumption that the LLM is controlled by the developers and offered as a service to end-users [57], this one-to-many relationship is conducive to detection purposes. However, the possibility of developers open-sourcing their models or the models being stolen by hackers poses a challenge to these detection approaches. For instance, Meta open-sourced its latest large language model, named Open Pre-trained Transformers (OPT), with parameters ranging from 125 million to 175 billion [75], while Huggingface also open-sourced its chatbot model and shifted its focus to democratizing machine learning.

Once the end user gets full access to the LLM, the ability to modify the LLMs' behavior hinders black-box detection from identifying generalized language patterns. Embedding a watermark in the open-sourced LLM is a potential solution. However, it can still be defeated as users have full access to the model and can fine-tune it or change sampling strategies to erase the watermark. To tackle this, developers may increase the model parameter vulnerability to prevent end-user modification of the released model, where a slight change in the model parameters can cause a significant performance degradation [72]. However, previous studies have been conducted on the miniature model of the tiny classification tasks, making it uncertain if these techniques can be applied to large language models. Currently. the cost and effort involved in LLMs training make it unlikely that developers will release their most powerful LLMs. Nonetheless, detecting LLM-generated texts from open-sourced LLMs remains a critical issue that needs to be addressed in the future.

6 Conclusion

The detection of LLM-generated texts is a rapidly growing and evolving field with a plethora of newly developed techniques. This survey provides a precise categorization and in-depth examination of existing approaches to help the research community comprehend the strengths and limitations of each method. Despite the rapid advancements in LLM-generated text detection, significant challenges still need to be addressed. Further progress in this field will require developing innovative solutions to overcome these challenges.

References

[1] Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, Institute of Electrical and Electronics Engineers, Online, 121–140.

[2] Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*. Springer, Pittsburgh, 185–200.

[3] David Baidoo-Anu and Leticia Owusu Ansah. 2023. Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning. *Available at SSRN 4337484* (2023).

[4] 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*. Association for Computing Machinery, Nagasaki, 469–482.

[5] Paras Bhatt and Anthony Rios. 2021. Detecting Bot-Generated Text by Characterizing Linguistic Accommodation in Human-Bot Interactions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, Bangkok, Thailand, 3235–3247.

[6] Jack T Brassil, Steven Low, Nicholas F. Maxemchuk, and Lawrence O'Gorman. 1995. Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communications* 13, 8 (1995), 1495–1504.

[7] Peter F Brown, Stephen A Della Pietra, Vincent J Della Pietra, Jennifer C Lai, and Robert L Mercer. 1992. An estimate of an upper bound for the entropy of English. *Computational Linguistics* 18, 1 (1992), 31–40.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[9] Brian Bushard. 2023. *Fake Scientific Abstracts Written By ChatGPT Fooled Scientists, Study Finds.* Retrieved Jan 25, 2023 from https://www.forbes.com/sites/brianbushard/2023/01/10/fake-

scientific-abstracts-written-by-chatgpt-fooled-scientists-study-finds/?sh=d75606118b63

[10] Jonathan H Choi, Kristin E Hickman, Amy Monahan, and Daniel Schwarcz. 2023. ChatGPT Goes to Law School. *Available at SSRN* (2023).

[11] Ryan Clancy, Ihab F Ilyas, and Jimmy Lin. 2019. Scalable knowledge graph construction from text collections. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Hongkong, China, 39–46.

[12] 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, Bangkok, Thailand, 7282–7296.

[13] Leda Cosmides and John Tooby. 1996. Are humans good intuitive statisticians after all? Rethinking some conclusions from the literature on judgment under uncertainty. *cognition* 58, 1 (1996), 1–73.

[14] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. 2007. *Digital watermarking and steganography*. Morgan kaufmann.

[15] Evan Crothers, Nathalie Japkowicz, Herna Viktor, and Paula Branco. 2022. Adversarial Robustness of Neural-Statistical Features in Detection of Generative Transformers. In *2022 International Joint Conference on Neural Networks (IJCNN)*. Institute of Electrical and Electronics Engineers, Padua, Italy, 1–8.

[16] Kevin Crowston. 2012. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches: IFIP WG 8.2, Working Conference, Tampa, FL, USA, December 13-14, 2012. Proceedings*. Springer, Tampa, FL, USA, 210–221.

[17] Long Dai, Jiarong Mao, Xuefeng Fan, and Xiaoyi Zhou. 2022. DeepHider: A Multi-module and Invisibility Watermarking Scheme for Language Model. *arXiv preprint arXiv:2208.04676* (2022), 1–16.

[18] 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, Florence, Italy, 4171–4186.

[19] Liam Dugan, Daphne Ippolito, Arun Kirubarajan, and Chris Callison-Burch. 2020. RoFT: A Tool for Evaluating Human Detection of Machine-Generated Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, online, 189–196.

[20] Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, Chris Callison-Burch, Pei Zhou, Andrew Zhu, Jennifer Hu, Jay Pujara, Xiang Ren, et al. 2023. Real or Fake Text? Investigating Human Ability to Detect Boundaries Between Human-Written and Machine-Generated Text. In *The 37th AAAI Conference on Artificial Intelligence (AAAI 2023)*. Association for the Advancement of Artificial Intelligence, Washington, USA, 104979.

[21] Nature Editors. 2023. Tools such as ChatGPT threaten transparent science; here are our ground rules for their use. *Nature* (2023).

[22] Michael Elsen-Rooney. 2023. *NYC education department blocks ChatGPT on school devices, networks.* Retrieved Jan 25, 2023 from https://ny.chalkbeat.org/2023/1/3/23537987/nyc-schools-ban-chatgpt-writing-artificial-intelligence

[23] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3558–3567.

[24] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833* (2018).

[25] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30 (2020), 681–694.

[26] 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.

[27] Matthias Gallé, Jos Rozen, Germán Kruszewski, and Hady Elsahar. 2021. Unsupervised and distributional detection of machine-generated text. *arXiv preprint arXiv:2111.02878* (2021).

[28] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. GLTR: Statistical Detection and Visualization of Generated Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 111–116.

[29] 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.

[30] 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.

[31] Nikolas Guggenberger and Peter N. Salib. 2023. *From Fake News to Fake Views: New Challenges Posed by ChatGPT-Like AI.* Retrieved Jan 25, 2023 from https://www.lawfareblog.com/fake-news-fake-views-new-challenges-posed-chatgpt-ai

[32] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv preprint arXiv:2301.07597* (2023).

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

[34] Grant Ho, Aashish Sharma, Mobin Javed, Vern Paxson, and David Wagner. 2017. Detecting credential spearphishing attacks in enterprise settings. *Proc. of 26th USENIX Security* (2017).

[35] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* (2019), 1–16.

[36] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1808–1822.

[37] Zunera Jalil and Anwar M Mirza. 2009. A review of digital watermarking techniques for text documents. In *2009 International Conference on Information and Multimedia Technology*. IEEE, IEEE Computer Society, Washington DC, United States, 230–234.

[38] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *Comput. Surveys* (2022).

[39] Mohan S Kankanhalli and KF Hau. 2002. Watermarking of electronic text documents. *Electronic Commerce Research* 2 (2002), 169–187.

[40] Alex Kantchelian, Michael Carl Tschantz, Sadia Afroz, Brad Miller, Vaishaal Shankar, Rekha Bachwani, Anthony D Joseph, and J Doug Tygar. 2015. Better malware ground truth: Techniques for weighting anti-virus vendor labels. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*. Association for Computing Machinery, Denver, Colorado, USA, 45–56.

[41] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A Watermark for Large Language Models. *arXiv preprint arXiv:2301.10226* (2023).

[42] Tiffany H Kung, Morgan Cheatham, Arielle Medinilla, ChatGPT, Czarina Sillos, Lorie De Leon, Camille Elepano, Marie Madriaga, Rimel Aggabao, Giezel Diaz-Candido, et al. 2022. Performance of ChatGPT on USMLE: Potential for AI-Assisted Medical Education Using Large

Language Models. *medRxiv* (2022), 2022–12.

[43] Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Yu Lan, and Chao Shen. 2022. CoCo: Coherence-Enhanced Machine-Generated Text Detection Under Data Limitation With Contrastive Learning. *arXiv preprint arXiv:2212.10341* (2022), 1–12.

[44] 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 preprint arXiv:1907.11692* (2019), 1–13.

[45] Jing Ma, Wei Gao, Shafiq Joty, and Kam-Fai Wong. 2019. Sentence-level evidence embedding for claim verification with hierarchical attention networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2561–2571.

[46] Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. How Decoding Strategies Affect the Verifiability of Generated Text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, Online, 223–235.

[47] Hasan Mesut Meral, Bülent Sankur, A Sumru Özsoy, Tunga Güngör, and Emre Sevinç. 2009. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language* 23, 1 (2009), 107–125.

[48] Hasan M Meral, Emre Sevinc, Bülent Sankur, A Sumru Özsoy, and Tunga Güngör. 2007. Syntactic tools for text watermarking. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, Vol. 6505. SPIE, Society of Photo-Optical Instrumentation Engineers, San Jose, CA, United States, 339–350.

[49] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. https://doi.org/10.48550/ARXIV.2301.11305

[50] OpenAI. 2023. *ChatGPT*. Retrieved Jan 25, 2023 from https://chat.openai.com

[51] OpenAI. 2023. *New AI classifier for indicating AI-written text*. Retrieved Jan 25, 2023 from https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text/

[52] Lucas Ou-Yang. 2014. Newspaper3k: Article scraping & curation. https://github.com/codelucas/newspaper.

[53] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* 32 (2019), 14003–14014.

[54] Steven T Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review* 21 (2014), 1112–1130.

[55] 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.

[56] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444* (2017), 1–9.

[57] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. 2015. Mlaas: Machine learning as a service. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE, Institute of Electrical and Electronics Engineers, Miami, FL, USA, 896–902.

[58] Juan Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-Domain Detection of GPT-2-Generated Technical Text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 1213–1233.

[59] Kalhan Rosenblatt. 2023. *ChatGPT banned from New York City public schools' devices and networks*. Retrieved Jan 25, 2023 from https://www.nbcnews.com/tech/tech-news/new-york-city-public-schools-ban-chatgpt-devices-networks-rcna64446

[60] Kalhan Rosenblatt. 2023. *ChatGPT passes MBA exam given by a Wharton professor*. Retrieved Jan 25, 2023 from https://www.nbcnews.com/tech/tech-news/chatgpt-passes-mba-exam-wharton-professor-rcna67036

[61] Danish Shakeel and Nitin Jain. 2021. Fake news detection and fact verification using knowledge graphs and machine learning. *no. February* (2021), 1–7.

[62] 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), 1–46.

[63] Teo Susnjak. 2022. ChatGPT: The End of Online Exam Integrity? *arXiv preprint arXiv:2212.09292* (2022), 1–21.

[64] Andon Tchechmedjiev, Pavlos Fafalios, Katarina Boland, Malo Gasquet, Matthäus Zloch, Benjamin Zapilko, Stefan Dietze, and Konstantin Todorov. 2019. ClaimsKG: A knowledge graph of fact-checked claims. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*. Springer, Springer, Auckland, New Zealand, 309–324.

[65] Edward Tian. 2023. *GPTZero*. Retrieved Jan 25, 2023 from https://gptzero.me/

[66] Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational linguistics* 29, 1 (2003), 97–133.

[67] Umut Topkara, Mercan Topkara, and Mikhail J Atallah. 2006. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th workshop on Multimedia and security*. Association for Computing Machinery, Geneva, Switzerland, 164–174.

[68] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Association for Computational Linguistics (ACL), Online, 8384–8395.

[69] Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296* 16, 1 (2021), 1–16.

[70] Honai Ueoka, Yugo Murawaki, and Sadao Kurohashi. 2021. Frustratingly Easy Edit-based Linguistic Steganography with a Masked Language Model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 5486–5492.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 6000–6010.

[72] Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. 2022. Non-Transferable Learning: A New Approach for Model Ownership Verification and Applicability Authorization. In *International Conference on Learning Representation (ICLR-2022)*. Committee of ICLR, Online, 1–24.

[73] Jeannette M Wing. 2021. Trustworthy ai. *Commun. ACM* 64, 10 (2021), 64–71.

[74] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems* 32 (2019), 9054–9065.

[75] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* 30 (2022), 1–30.

[76] Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural Deepfake Detection with Factual Structure of Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Minneapolis, Minnesota, 2461–2470.

[77] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. Association for Computing Machinery, Ann Arbor, MI, USA, 1097–1100.