Distinguishing Human From Machine: A Review of Advances and Challenges in AI-Generated Text Detection

Serena Fariello, Giuseppe Fenza, Flavia Forte, Mariacristina Gallo, Martina Marotta *

Department of Management and Innovation Systems, University of Salerno, 84084 Fisciano (SA) (Italy)

* Corresponding author: s.fariello4@studenti.unisa.it (S. Fariello), gfenza@unisa.it (G. Fenza), f.forte27@studenti.unisa.it (F. Forte), m.marotta37@studenti.unisa.it (M. Marotta).

Received 22 July 2024 | Accepted 28 October 2024 | Early Access 18 December 2024

Abstract

The rise of Large Language Models (LLMs) has dramatically altered the generation and spreading of textual content. This advancement offers benefits in various domains, including medicine, education, law, coding, and journalism, but also has negative implications, mainly related to ethical concerns. Preventing measures to mitigate negative implications pass through solutions that distinguish machine-generated text from human-written text. This study aims to provide a comprehensive review of existing literature for detecting LLM-generated texts. Emerging techniques are categorized into five categories: watermarking, feature-based, neural-based, hybrid, and human-aided methods. For each introduced category, strengths and limitations are discussed, providing insights into their effectiveness and potential for future improvements. Moreover, available datasets and tools are introduced. Results demonstrate that, despite the good delimited performance, the multitude of languages to recognize, hybrid texts, the continuous improvement of algorithms for text generation and the lack of regulation require additional efforts for efficient detection.

Keywords

Generated-Text Detection, AI-Detection, Large Language Models (LLMs), Literature Review, Survey.

DOI: 10.9781/ijimai.2024.12.002

I. Introduction

With the increase of computer power and the availability of extensive datasets, Artificial intelligence evolve rapidly. In the area of Natural Language Processing (NLP), the introduction and diffusion of Large Language Models (LLMs) have transformed existing approaches due to their ability to achieve significant performance in different NLP tasks. In the early days, this included simply automated responses in customer service, conversation summaries like automated call transcriptions, news articles, and so on. As a result of technological evolution, machine-generated text has become more and more sophisticated and human-like [1]: modern systems use advanced algorithms and analyze vast amounts of data to produce natural and coherent text [2]. They are utilized in varied contexts and for different purposes: writing articles, providing customer support, and even creating educational content, as described in the following. This advancement led to the development of Large Language Models (LLMs), which have significantly changed the way in which people generate and interact with machine-produced text. LLMs reveal a significant capacity to generate text that matches human writing. This capacity makes distinguishing LLM-generated text from human-written text hard. However, the machine-generated text could impact ethical issues such as exacerbating biases and stereotypes in training data or producing false or misleading content. Known issues related to machine-generated content rely on manipulating public opinion, spreading fake news, and plagiarism. So, despite the huge potential, it is important to use LLMs conscientiously to avoid cheating, dishonesty and low-quality responses [3] [4]. Preventing measures aiming to mitigate future implications of LLMs diffusion are necessary and pass through valid solutions distinguishing machine-generated text from human-written text. The present study intends to collect and analyze the most recent approaches in terms of detection and identification of generated text content. In particular, the research work aims to answer the following questions:

RQ1 What are the most recent methods for detecting LLM-generated texts and their main limitations?

- Various detection methods are reviewed, including watermarking, feature-based, neural-based, hybrid, and human-aided approaches.
- Each method's strengths and potential areas for improvement are highlighted.

Please cite this article as:
S. Fariello, G. Fenza, F. Forte, M. Gallo, M. Marotta. Distinguishing Human from Machine: A Review of Advances and Challenges in AI-Generated Text Detection, International Journal of Interactive Multimedia and Artificial Intelligence, (2024), http://dx.doi.org/10.9781/ijimai.2024.12.002

RQ2 What datasets are used for training detection models?

- The study examines the datasets utilized for training detection models.
- The advantages and limitations of these datasets in accurately identifying machine-generated texts are discussed.

RQ3 Are there state-of-the-art tools capable of addressing recent advancements in text generation?

- The study evaluates the effectiveness of current detection tools.
- Emphasizes the need for continuous development to keep pace with advancements in LLM capabilities.

The rest of the manuscript is structured as follows: Section II overviews LLMs' functioning, their applications, motivations guiding this research work, and a focus on the targeting task: *machine-generated text detection*. Section III examines the existing literature review in the machine-generated text detection task, and Section IV outlines the research methodology. Section V delves into the detection methods, categorized into watermarking, feature-based, neural-based, hybrid and human-aided approaches. Section VI describes the characteristics of existing datasets and discusses their limitations, while some useful tools are examined in Section VII. Finally, Section VIII discusses the problems and limitations of examined detection methods, and Section IX concludes the manuscript.

II. Context and Background

LLM-generated text is the latest and most advanced form of machine-generated text. These models, such as GPT-4 and BERT, use deep learning to produce texts extremely close to those written by humans. The models are trained on massive datasets, including a huge variety of human language examples, allowing them to understand and mimic complex patterns, syntax, and meanings [5].

This section introduces LLMs, their application, and the motivations that guide this research work. Moreover, the objective of the considered literature is detailed.

A. LLM Fundamentals

Large Language Models are based on an advanced neural network, especially the Transformer architecture introduced by Vaswani et al. [6]. It is based on the following equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Each word in the sentence is converted into a numerical vector. Q (query) is the word vector, K (key) are the vectors for every word in the sentence, and V (value) are the value vectors. Softmax is the function that converts scores into a probability distribution. This formula allows the model to weigh the importance of each word within a sequence of other words.

Generation of text (contextually relevant and coherent) by LLMs like Xlnet depends from their autoregressive nature. Aurogressive models predicts future behaviour based on past behavior data. In the case of text, the subsequent word is predicted based on the previous ones [7].

B. Motivations

Due to their performance, LLMs are utilized in varied contexts and for different purposes: writing articles, providing customer support, and even creating educational content. The generated text is so impressively good that it is difficult to tell if it was written by a person or a machine. Nevertheless, LLMs could produce inaccurate information. This creates a challenge in identifying AI-generated content, which has led to the development of advanced detection techniques. The following are examples of LLMs' applications and their risks:

- **Education**. LLMs can provide personalized and interactive learning experiences for students and may help teachers reduce their workload in order to focus on research [8]. According to Jeon and Seongyong [9], LLMs such as ChatGPT may help teachers by assuming supporting roles like interlocutor, content provider, teaching assistant and evaluator. However, according to specific subjects, LLMs' performance and responses may have different accuracy grades. For instance, in Geometry [10], LLMs sometimes cannot provide accurate and reliable answers due to a lack of critical and logical thinking, leading to the necessity of human integration.
- **Medicine**. LLMs are starting to be used in the healthcare sector to enhance the well-being of both patients and doctors. ChatGPT and Med-palm 2, for example, have exhibited encouraging outcomes in medical assessments and addressing patient inquiries, even if they are still imperfect and have shown a lack of recency, accuracy and coherence. Therefore, at present, they cannot be deemed as a true replacement for medical professionals but rather as a supplementary tool in clinical, educational, or research environments [11]. They faced challenges in understanding cause-and-effect relationships between medical conditions and lacked sufficient medical knowledge to fully comprehend complex interactions [12].
- **Coding**. In the realm of software development, where creating applications involves writing code in various programming languages, the rapid progress of LLMs is proving beneficial. Feng et al. [13], in their research, have discovered that ChatGPT has been employed across many different languages - with Python and JavaScript emerging as the most widely utilized - for different coding tasks like debugging and testing. However, unlike common writing assignments, programming requires precise conformity to syntax and rules and great attention to possible vulnerabilities, making it notably challenging for generative models to produce top-notch and high-security code [13],[14].
- **Law**. LLMs are transforming how legal professionals work, enhancing their efficiency and accuracy in daily tasks such as legal research, contract drafting [15], empirical analysis (LLMs can be used to examine large volumes of legal texts, identifying trends and arguments) [16], assistance in contract negotiation and creation of legal contents [17]. However, there remains a significant risk of relying on inaccurate, outdated, or unsourced legal information [18].
- **News Generation**. Nowadays, journalists use LLMs to fabricate news to maximize the spread of content and take advantage of social networks. Nevertheless, the risk is a compromised authenticity of news as well as biased content [19].

As outlined, machine-generated content, on one side, can improve and facilitate the work; on the other side, it can undermine academic and journalistic integrity, intellectual property [20], transparency, and ethics [21]. Knowing the state-of-art in terms of solutions to recognize the nature of content could help in developing more suitable solutions, regulating the use of genAI [22] and, finally, improving generative models themselves.

C. Machine-Generated Text Detection Task

This literature review intends to collect and discuss the state of the art in terms of approaches for detecting machine-generated text. The machine-generated text detection task consists of automatically

detecting content generated by LLMs. It can be solved as a binary classification problem or by setting a threshold. From a mathematical perspective, it can be formalized as a binary classification problem, seeking to determine if a given text is generated by an LLM or by a human writer [5].

Given a text t and a Detector $D(t)$, the equation is the following:

$$D(t) = \begin{cases} 1 & \text{if } t \text{ is machine} - \text{generated,} \\ 0 & \text{if } t \text{ is human} - \text{written} \end{cases}$$

Setting a threshold can contribute to another way to define the detection task. Given an input text, the text detector outputs a score. A score higher than the threshold indicates a machine-generated text [23].

III. Related Works

The detection of generated text is a hot research field to explore. In fact, several works in the literature have reviewed the main techniques used to perform this task. One of the early technique reviews goes back to 2016 [24]. From then on, the wide ever-increasing use of LLMs definitely complicated generated text detection. As a consequence, the research started to be more attentive to the text generated by these models, and scientists began investigating and publishing new machine-generated text detection methods [25]. With ChatGPT's rise, there was a further increase in LLM-generated text reviews, such as the approach proposed by Dhaini et al. [26]. This research line has become the main one as it has been supported by many different works, which have inspired this work as well. In detail, a further step has been made by a work that introduces a first kind of categorization by dividing feature-based and neural-language model approaches [27]; another work [28] divides the task into black-box and white-box detection, introducing a novel template followed by successive works; another method [29], following the black-box and white-box structure idea, added three categories of detection methods: training-based, zero-shot-based and watermarking methods. Moreover, interesting research [23] highlighted the weaknesses of existing text detection techniques (e.g., text paraphrasing). At the same time, the review work by Uchendu et al. [30] has introduced the hybrid methods category for the first time. The most exhaustive study, considering the state-of-the-art methods to date, is the one proposed by Wu et al. [5] that covers many method categories. It presents useful sections for the ones dealing with the phenomenon of generated text detection, providing details regarding the most popular datasets and benchmarks useful for this task and underlining the research limitations in generated text detectors.

The proposed survey begins by exploring various risks associated with multiple application domains of machine-generated text, underscoring the urgent need for robust detection methods. Compared to previous surveys, it is more up-to-date and provides a thorough analysis of the strengths and weaknesses of current approaches. In addition, the survey offers an in-depth discussion of the datasets used to train learning models, highlighting current limitations in terms of data quality and diversity, and summarizes the performance evaluation of state-of-the-art tools. Finally, emerging challenges, such as issues related to the adopted languages and the lack of regulatory frameworks, are discussed.

IV. Research Methodology

The papers guiding this study are harvested from relevant search engines like Scopus, DBLP, and Scholar by exploiting specific queries: *LLM-generated text detection*, *machine-generated text detection*, and *authorship attribution*. Moreover, Scimago has been adopted to filter more relevant journals, while the International CORE Conference Rankings (ICORE) for conferences. The authors' H-index was considered in the case of very recent preprint versions. During the collection, journals with an h-index higher than 10 and conferences with a performance class that ranges from A to B were considered. Concerning preprints, the works created by authors with an h-index higher than 10 or those with a number of author citations higher than 35 were selected. Fig. 1 shows the distribution, by year, of the last six years of literature on the generated text detection task, highlighting a peak after the introduction of GPT. Results are summarized in Table I.

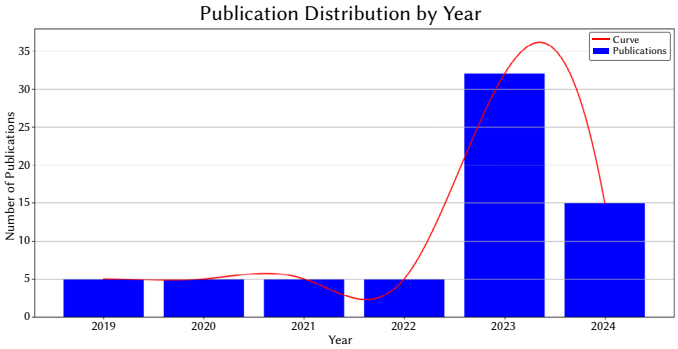

Fig. 1. The distribution, by year, of the last six years of literature on the generated text detection task.

V. Generated Text Detection Methods

This section analyzes the detection methods emerging from the analysis in detail and arranges them into five categories: watermarking, feature-based, neural-based, hybrid and human-aided approaches.

A. Watermarking

Watermarks are embedded signals in the generated text that are invisible to humans but can be detected involving the use of algorithms. Text watermarking implements patterns into the generated text to tell the difference between large language models (LLMs) generated text and a human-generated text [36],[68]. Watermarking must be effective (the coherence of the generated text must be preserved), invisible (it should smoothly blend into the text), robust (it requires being difficult to eliminate [28] resisting to corruption or attacks [42]).

1. Data-Driven Watermarking

The aim of data-driven methodologies is to assess the property of data. Using patterns or tags within the training datasets these methods can check if the data is copied or used for malicious ends. Adding a few samples with hidden watermarks (backdoor insertion) to the training data can help language model creators detect if their models are being used by bots on platforms like Twitter to spread fake news. Thus, the model learns a secret function set by the creator. The watermark is very robust even in case of a model being fine-tuned for other specific tasks [31].

Succeeding studies have identified weaknesses in this technology, revealing that it can be easily manipulated.

Lucas et al. [32] found that inserting triggers made from uncommon markers makes them difficult to detect.

Triggers constructed with ordinary words are less effective for watermarking because the presence of common word combinations in natural text poses a risk of false positives, text incorrectly detected as generated by a language model when it was actually authored by a human. Additionally, watermarks based on common words are way easier to detect. Research by Tang et al. [33] demonstrates that incorporating just 1% of watermarked samples improves traceability in

TABLE I. Summary of Evaluated Approaches

Category	Sub-categories	Overview	Advantages	Limitations
Watermarking	Data-Driven [31][32] [33]	Allows checking whether a given text has been generated by a model that uses a watermark.	Effective against attempts to remove or modify it. Compatible with any LLM. Minimal to no effect on the quality of a generated text.	Foolable through paraphrase. Needs the willingness to apply.
	Model-Driven [34][35] [36] [37]			
	Post-Processing [38] [39] [40] [41][42] [43]			
	Neural-based [44]			
Feature-based	Stylistic Feature and Stylometry [45] [46] [47]	Leverages measurable and evident differences between human and generated texts in terms of syntax, grammar, and other linguistic particularities.	Flexible and less computationally intensive. Makes the decision-making process transparent and comprehensible.	Susceptibility to perturbations (e.g., word substitutions). Difficulty in transferring features across architectures.
	Frequency Features [48]			
	Statistical Metrics [49] [50] [51]			
Neural-based	Feature-based [30][52] [53] [54]	Exploits the architecture of deep neural networks.	Effective in capturing the complex linguistic nuances present in texts. Robust against small mutations in text. Efficient in different application scenarios.	Strictly related to domains and languages of training datasets and to the adopted model for generating text.
	Pre-training [54][55]			
	Fine-tuning Classifier [56] [55]			
	Zero-shot [57] [56][58]			
Human-aided [59] [60] [61] [62] [63] [64]		Combines features-based or neural techniques with human analysis and review.	Enhanced accuracy thanks to human review.	Strictly related to human skills.
Hybrid [65] [66] [67]		Combines multiple methodologies.	Hard to obfuscate the style and deceive detection systems.	High complexity and computational costs.

datasets, facilitating better management and safeguarding of language models. It should be noted that data-driven approaches primarily aim to safeguard dataset copyrights, thus typically offering limited payload capacity and applicability. Furthermore, implementing these methods in detecting text generated by LLMs demands substantial resources, such as embedding watermarks across extensive datasets and retraining the models.

2. Model-Driven Watermarking

This kind of method integrates watermark signals directly into Large Language Models. They do so intervening on the logits distribution or on the token sampling.

Logits-Based Methods

The watermark of Kirchenbauer et al. [34] comes in the decoding step. So before choosing the next word, watermarking randomly excludes a portion of the possible words (blacklisted). Limiting the model's choice to the remaining options (whitelisted). The seed for the random number generator that chooses which words are blacklisted is the last word of the input. In this way, the blacklist can be reconstructed at any time. This procedure is applied at each generation of the next token. To detect generated text by a language model, one needs to detect the watermark counting the blacklisted words in the generated text. Obtaining the blacklist means knowing the random number generator used to choose the blacklist words and the seed. The watermarked language model would not use blacklisted words because it can not, but humans would definitely use blacklisted terms. So, a text using only whitelist words is highly likely to be AI-generated, and even a short text can be classified with relatively high certainty. Recent research conducted by Kirchenbauer et al. [35] demonstrates that watermarking remains effective even when watermarked text is manually rewritten, paraphrased by non-watermarked LLMs, or integrated into longer handwritten documents.

However, to generate and detect watermarks it is needed a secret key poses potential security vulnerabilities. In response to this matter, a research [36] introduced the first private watermarking algorithm. This method employs separate neural networks for watermark generation and watermark detection. In this way, two different keys can be used.

Additionally, both networks share a section of the parameters, enhancing the detection network's efficiency and accuracy. Existing watermarking methods for LLMs only contain one bit of information (whether it is generated from an LLM or not) and cannot flexibly give information such as model version, generation time, user ID, etc. In this sense, Wang et al. [68] conducted the first study on the topic of Codable Text Watermarking for LLMs (CTWL) that allows text watermarks to carry more customizable information including which model generated the text and when.

Token Sampling-Based Methods

Token sampling on language models represents the process of selecting subsequent words (tokens) following a probability distribution. Token sampling entails randomness so that the resulting text becomes unpredictable. Methods utilizing token sampling for watermarking use random seeds or specific patterns to guide the token sampling mechanism. The method proposed by Kuditipudi et al. [69] used a secret key, which is a set of random numbers, to control token sampling. This token sampling operation is incorporated into the language model so that the output text contains an embedded watermark. To detect watermarks, the confidential key is used to line up the text with the arbitrary numbers. This hidden number makes it possible to recognize and recover the watermarks from the watermarked text. Paraphrasing would be difficult in this technique. Another recent work is SemStamp [37], which involves the use of Locality-Sensitive Hashing (LSH) to watermark sentences generated by the language model. Locality-Sensitive Hashing is a method that maps similar points in semantic space to adjacent positions in hash space. It subdivides the semantic space into two regions: one with watermarks and another without watermarks. This facilitates the identification of watermarked sentences during the detection phase. From the experimental results, SemStamp is more robust when it comes to the common type of paraphrasing attempts that involve two adjacent words than the other existing methods and is more effective in maintaining the quality of text generation.

3. Post-Processing Watermarking

Post-processing watermarking is the practice of adding a watermark by modifying the generated text by a LLM.

Character-Level Methods

In the past, watermarking was done by inserting or substituting unique Unicode characters in a piece of text. With these techniques, the characters convey encoding information but they are invisible to the human eye.

Lip Yee Por et al. [38] proposed UniSpaCh, a method for hiding data in Microsoft Word documents using Unicode characters, which enhances embedding efficiency and resists attacks while preserving the document's original appearance.

Word-Level Methods

Yang et al. [41] proposed a natural language watermarking scheme based on context-aware lexical substitution. They employ BERT [70] to suggest lexical substitution candidates by inferring the semantic relatedness between the candidates and the original sentence. A watermark insertion model [40] detects alterations in the text even in the presence of paraphrased content. The process of watermarking insertion is based on a methodology that selects and replaces words with synonyms to embed watermarks in sentences while preserving grammatical integrity. They also use BERT for watermarking detection because it possesses the capability to recognize sentence modifications and distinguish between marked and unmarked sentences. In another work [42], features like proper nouns and words' grammatical dependency, that are semantically or syntactically fundamental components of the text and, thus, invariant to minor modifications in texts, are identified and used as anchor points to pinpoint the position of watermarks. It is a multi-bit watermarking framework able to embed adequate bits of information and extract the watermarks in a robust manner despite possible corruption, such as copy-paste attacks, substitution attacks, paraphrasing attacks, etc.

Yang et al. [43] present a method that uses a binary encoding function. This function associates binary codes to words, in an arbitrary manner. For example, a word like "happy" could be replaced with "joyful" if it represents a "1" in the binary code, while "sad" might remain unchanged if it represents a "0".

Neural-Based Approach

An Adversarial Watermark Transformer (AWT), an innovative system that automates the entire process of embedding watermarks into texts, is proposed [44]. It utilizes the Transformer to learn how to replace specific words with others that carry a secret binary message. With this approach, the algorithm handles everything, from selecting words to embedding the watermarks, without the need for manual intervention. Additionally, AWT leverages adversarial techniques, meaning it trains itself to be resistant to attempts at watermark detection and removal.

Three watermark networks are taken into consideration in neural-based approaches: an encoder, a decoder and a discriminator.

The encoder generates a modified text that incorporates the watermark, which can be a sequence of binary bits. The modifications in the text must be minimal to preserve the readability and naturalness of the text. The decoder extracts the watermark from the modified text produced by the encoder network. The discriminator distinguishes between the original text and the watermarked modified text. Its aim is to prevent the encoder from significantly altering the text. During training, the discriminator tries to identify which text has been modified by the encoder while it attempts to make the modified text as similar as possible to the original text to fool the discriminator.

The performances are considered satisfactory if the encoder successfully embeds the watermark into the text in a way that makes it difficult to detect, the decoder is capable of accurately retrieving the message and the discriminator canâ€™t notice the difference an authentic text and a watermarked one.

4. Advantages of Watermarking

Watermarking is an adequate choice for many reasons. A watermark remains effective even when attempts are made to remove or modify it [34]. Additionally, it is compatible with any large language model and has minimal to no effect on the quality of the generated text.

5. Limitations of Watermarking

Despite their applicability, there are multiple ways to fool watermarking algorithms. By knowing the blacklist, the tokens in it can be used within the text. But brute-forcing their way to the blacklist means that the attacker queries the API a lot of times with the same input, in which case, the API provider can monitor and detect this malicious activity. Another way to attack watermarking is by doing word substitutions: the rewritten text will not be detected by watermarking. The attacker could also use a non-watermarked model to paraphrase the output of a watermarked model. Making minor adjustments, such as inserting spaces, emojis, or misspellings, can impact watermark detection. The main disadvantage of watermarking is that it can only be implemented when individuals and organizations are willing to apply it to their language models. In addition, existing tools are applicable to language models that do not implement watermarking. Future strict regulations about it could help implement this technique.

B. Feature-Based Methods

Feature-based methods leverage the fact that there are measurable differences between human and AI-generated texts in terms of syntax, grammar, and other linguistic particularities.

Munoz-Ortiz et al. [71] made a quantitative analysis comparing human-written English news text with output from LLMs of the LLaMa family. Their research leads to important discoveries about:

- **Sentence Length Distribution**: Human texts exhibit more scattered sentence length distributions compared to LLM-generated texts.
- **Dependency and Constituent Types**: Human texts show a distinct use of dependency and constituent types.
- **Emotions**: Human texts display more aggressive emotions (e.g., fear and disgust) than LLM-generated texts.
- **Language Characteristics**: LLM outputs use more numbers, symbols, and auxiliaries than human texts. Additionally, LLMs employ more pronouns.
- **Sexist Bias**: The sexist bias prevalent in human texts is also expressed by LLMs.

Following the listed aspects, the current section shows the evolution of feature-based methods in distinguishing between human and AI-generated content, categorizing features into three main types: Stylistic Features, Frequency Features, and Statistical Metrics.

1. Stylistic Features and Stylometry

Stylometry is the quantitative analysis of literary style. It involves examining various linguistic features, such as specific vocabulary and verbs, syntax, sentence structure and length, fluency and consistency, to identify patterns and similarities/differences between texts or authors. It is commonly applied in fields such as forensic linguistics, literary studies, computational linguistics and authorship attribution.

Stylometry demonstrated its effectiveness in spotting fake news written by humans, but it has not the same effects in spotting fake news generated by machines [46]. Thus, to achieve more reliable results, it may be useful to integrate Stylometry with other methodologies. For instance, Abiodun Modupe et al. [45] have proposed a method called RDDN that uses a neural network to extract lexical stylometric features. These stylometric features are fed into a bidirectional

encoder to generate a vector representation of syntactic features, and the vector is then used by a bidirectional decoder to learn the writing style of an author.

In a recent study, Kumarage et al. [47] presented an algorithm using stylometric signals to measure stylistic changes in human and AI tweets to detect AI-generated tweets.

Their experiments succeeded in showing that stylometric features (specifically the ones related to phraseology, punctuation, and linguistic diversity) effectively enhance AI-generated text detection.

Other examples of mixed methodologies based on stylistic features will be presented and discussed in Section E.

2. Frequency Features

Frequency features refer to the repetition of specific terms, the distribution of word sequences, the number of punctuation marks, and the frequency of grammatical errors [27]. In their work, Frohling et al. [48] developed a feature-based classifier that leverages various features, including those related to the concept of repetitiveness. This can be measured by counting the number of stop-words, unique words, and words from "top-lists" within a text. They specifically looked at the overlap of n-grams for words (lexical repetition) and part-of-speech tags (syntactic repetition) in consecutive sentences, under the assumption that human text tends to be less repetitive than generated text in both sentence structure and word choice.

3. Statistical Metrics

Metrics such as perplexity and entropy are crucial in evaluating the predictability and variability of a text. This section exposes some examples of how perplexity, burstiness, entropy and density can be used in a generated-text detection task.

Perplexity is a metric that measures how well a probability model, such as an LLM, predicts a sample.

Specifically, it assesses the model's uncertainty in predicting the following word to choose, based on the preceding words, to continue a phrase.

Language models token sampling depends on common patterns in the training data. Therefore, LLM-generated text is characterized by low perplexity. In contrast, humans express themselves using non-identical styles, exhibiting higher perplexity values [49].

Burstiness measures the sentence complexity. Humans vary their sentences a lot when judging by the length and the number of rare words they use. So, burstiness has something to do with the fact that, for example, rare words usually do not occur very often in writing, but when they do, they start to happen a lot for a sentence or two, then not anymore. Language models are more constant in the way they write out their sentences. So going on sentence by sentence, one can plot the complexity of each sentence. For humans, these values will vary a lot, while for models, the value will be quite similar for all sentences. Then, a bumpy burstiness graph will likely belong to a human text, while a more constant graph will belong to an AI-generated text. GPTZero[1] is an example of a tool applying Perplexity and Burstiness to detect AI-generated text content.

While perplexity focuses on the model's predictive accuracy for specific word sequences, ***Entropy*** quantifies the overall uncertainty and randomness in word distribution across a text. In the past, researchers have shown that human-written texts generally exhibit higher entropy due to their varied word choices, whereas AI-generated texts often demonstrate lower entropy as they tend to follow more structured conventions [59]. However, as models like GPT-3 and GPT-4 advanced, they became capable of generating more diverse and contextually rich text, closely mimicking human variability and resulting in higher entropy. Recent findings by Mitchell et al. [50] support this new perspective. They observe that entropy correlates positively with the likelihood of a passage being identified as fake. Therefore, the assumed high average entropy can serve as an indicator of machine-generated text.

Uniform Information Density (UID) is a statistical metric based on the assumption that humans tend to distribute information uniformly along their text. By analyzing UID-based features, the GPT-who detector [51] captures the unique statistical signature of each author, both human and artificial.

4. Advantages of Feature-Based Methods

Feature-based approaches simplify the understanding of the model's decision-making process. The discussed techniques, in fact, make the process more transparent and comprehensible by concentrating on particular, quantifiable aspects of the text [72]. Moreover, they are very "flexible" because it is possible to select specific features and adapt the model to particular types of text and writing styles. They are also less computationally intensive, requiring fewer resources and less time than more complex models like deep learning.

5. Limitations of Feature-Based Methods

Despite the numerous existing feature-based models mentioned, there are various issues associated with them that sometimes lead to poor performance. Perturbations (e.g., word substitutions, alterations of characters and words, introduction of spelling errors) can significantly reduce the accuracy of these detectors [5]. Moreover, feature-based models present weaknesses related to the difficulty of transferring specific features between different architectures and sampling methods [48].

C. Neural-Based Methods

In this section, approaches to neural networks are explored by distinguishing between more classical networks and the adoption of pre-trained models or few-shot prompting.

1. Feature-Based Classifiers

Feature-based classifiers can be further differentiated based on the characteristics (features) extracted from the data.

Linguistic Feature-Based Classifiers

When comparing texts generated by large language models (LLMs) with those written by humans, noticing the linguistic differences is crucial to train classifiers that can effectively distinguish them. Text elements can be categorized based on the style of the text, the complexity, the semantic, the psychological, and the knowledge-based characteristics.

These characteristics are extracted mainly by statistical techniques. Subsequently, a classification model can be trained through machine learning techniques [30].

Among the various methods to detect text generated by artificial intelligence, Shah et al. [73] have constructed a classifier based on stylistic features such as frequency analysis of word pairs, language characteristics and lexicographic characteristics. These classifiers based on linguistic characteristics seem to be very beneficial and useful in distinguishing between human-generated and AI-generated texts, but they have flaws that cannot be overlooked: their ability to detect LLM-generated misinformation is limited [5].

Model Feature-Based Classifiers

In addition to the linguistic characteristics, classifiers based on the characteristics of the model have also received considerable attention from research in the field. It is about classifiers that are able to detect

[1] https://gptzero.me/

texts generated by LLMs and trace the origin of the text. In particular, the research made by Su et al. [53] considers the log-rank. However, these methods have a common drawback: they all require access to the source model's logins, so these templates are ineffective when applied to closed sources where the logins are inaccessible.

2. Pre-Training Classifier

Famous pre-learned models, such as Roberta [74], have shown superior performance than traditional machine learning methods and deep learning in text categorization tasks. The 2019 studies identified the improved large language models (LLMs) as Roberta among the best to detect texts generated by other LLMs. These models achieved an average accuracy rate of 95% in their respective fields, surpassing zero-shot and watermarking methods and showing good resistance to different attack techniques. However, like other similar models, these improved encoder-based models are not very robust [54], [75] because they tend to depend too much on training data, leading to a drop in performance with data from different or new domains. Despite this, Roberta-based detectors show remarkable robustness potential, requiring only a few hundred labels to achieve impressive results [55].

3. Fine-Tuning

Fine-tuning, in the field of machine learning and artificial intelligence, is the process of adapting a pre-trained model to a new specific task. Studies of machine-generated text detection have examined how a detection algorithm, such as Roberta, can be trained on a dataset other than that used by an attack model such as GPT-2. It turned out that by perfecting the detection model with only a few hundred samples identified by experts, the detector can greatly improve in adapting to different types of data [55]. This is useful in real situations when a general detector has to deal with a specific attack pattern. When a defender identifies text samples generated by an improved attack model, these examples can be used to make the detection model even more effective [27].

4. Zero-Shot

With the aim of detecting machine-generated text, zero-shot approaches have increasingly become used by researchers and developers. This is related to the fact that zero-shot methods do not need fine-tuning. Some studies show that smaller models of generated text can be used to detect text generated by larger models [57], [58]. This ability decreases as the scale difference grows, while on the contrary, the ability to predict smaller architectures can be very beneficial, as recreating large models with a large number of parameters is highly expensive [27].

5. Advantages of Neural-Based Methods

Neural-based methods are particularly effective in capturing the complex linguistic nuances present in texts by considering specific attributes of advanced models such as ChatGPT [76]. Moreover, fine-tuning can improve the ability to recognize modified texts, making them robust against small mutations in text where even pre-trained models may fail [77].

6. Limitations of Neural-Based Methods

Neural-based methods, generally speaking, need labeled datasets, and their performance and applicability are strictly related to reference domains. Moreover, research findings indicate that the zero-shot approach generally underestimates a simple TF-IDF baseline when attempting to detect output from a generative model that has been developed on a different domain. Because attackers can adjust generative patterns for different purposes, this represents a notable weakness in the zero-shot approach using generative models for detection without tuning [56].

D. Human-Aided Methods

Methods combining features-based or neural techniques with human analysis have been proposed to enhance review capabilities. This integration provides crucial human oversight for trustworthy AI systems but presents scalability challenges due to the need for trained analysts capable of confidently identifying machine-generated text. For example, GLTR (Giant Language Model Test Room) [59] uses a method called "top-k sampling" to highlight words, but this method has been mostly replaced by "nucleus sampling," used in newer models like GPT-3. So, it would probably be difficult for untrained people to detect texts created by the more recent and advanced models. To overcome this limitation, RADAR tester [60] displays the probability each model assigns to a text being AI-generated. A value close to 1 suggests a "high likelihood of AI generation," while a value close to 0 indicates a "high likelihood of human authorship." It also implies that the material is probably produced by a human if the models have significantly different probabilities. Using this information, a human reviewer can effectively assess whether a text was created by a human or an AI.

1. Advantages of Human-Aided Methods

The advantage of these approaches is that they need human support and oversight, which can mitigate the risks associated with a completely autonomous decision-making. This presence also ensure a greater trustworthy in the AI technologies [27].

2. Limitations of Human-Aided Methods

The greatest weakness of Human-Aided Methods is that they are strictly related to the competences (or inabilities) of a human reviewer. Human performance in distinguishing machine-generated text has been extensively studied. Research indicates that untrained individuals often perform no better than chance when distinguishing texts generated by models like GPT-3. However, with some training [61], the accuracy can improve to around 55%.

E. Hybrid Methods

In this section, various hybrid approaches combining and integrating multiple different methodologies to enhance accuracy and reliability in the detection task are explored.

TDA-based detector [65] employs an innovative approach that combines Transformer-based and statistical methodologies to distinguish between human-written and generated texts. This system uses BERT to understand the meaning of words in the context of a text and create detailed representations of them. These representations are then analyzed using Topological Data Analysis (TDA), a mathematical technique that studies the shape and structure of connections between words.

CoCo (Coherence-based Contrastive Learning Model) methodology [66] combines graph-based coherence representation with contrastive learning techniques, aiming to achieve high accuracy in distinguishing between diverse types of textual content. Specifically, CoCo looks at how well the sentences in a text stick together and make sense as a whole (coherence information) and then turns this information into a graph that helps it understand the relationships between different parts of the text. Moreover, contrastive learning helps the model learn better by comparing different texts and focusing on their differences, even under low-resource scenarios.

DIDAN [67] is a tool created to detect fake news articles by analyzing both the text and the images together. It uses a BERT encoder to understand the text and examines the visual-semantic representations to investigate the relationship between the text and images and understand if they match up logically. Additionally, each article is given a score to show how likely it is to be written by a human.

TABLE II. Summary of Used Datasets

Corpus	Adopting Papers	Language	Task
C4 [79]	[34] [36] [68] [35] [36] [69]	English	Language Modelling
RealNews [57]	[48] [65] [66] [67] [37]	English	Text Generation, Language Modelling, Fake News Detection
Webtext [80]	[48] [60] [64] [65]	English	Text Classification, Text Generation, Language Modelling
WikiText-2 [81]	[41] [44] [42]	English, Spanish, German, Swedish	Text Generation, Language Modelling
IMDB [82]	[41] [42] [31]	English	Text Classification, Language Modelling, Paraphrase Identification
AgNews [83]	[41] [31]	English	Text Classification, Zero-Shot Text Classification, Anomaly Detection
SQuAD [84]	[50] [60]	Multilingual	Question Answering, Question Generation
WritingPrompts [85]	[60] [61]	English	Text Generation, Language Modelling, Story Generation, Natural Language Understanding
CoAuthor [86]	[87] [88]	English	Text Generation
PubMed [89]	[49]	Multilingual	Text Summarization, Language Modelling
WMT16 [90]	[50]	English, French, German, Russian,Czech, Finnish, Romanian	Machine Translation
PubMedQA [91]	[50]	English	Question Answering, Language Modelling
RecipeNLG [92]	[61]	Multilingual	Text Generation
Common Crawl [93]	[66]	English	Language Modelling, Generated-Text Detection
NeuralNews [94]	[67]	English	Generated-Text Detection
DialogSum [95]	[32]	English	Text Summarization, Dialogue Generation, Abstractive Text Summarization
DBpedia [96]	[36]	English	Text Classification
HC3 [97]	[43]	English, Chinese	Text Classification, Question Answering, Sentence Similarity, Zero-Shot Classification
GLUE [98]	[31]	English	Text Classification, Natural Language Inference, Semantic Textual Similarity, Natural Language Understanding, Semantic Textual Similarity within Bi-Encoder
SNLI [99]	[31]	English	Natural Language Inference

1. Advantages of Hybrid Methods

Modern and advanced hybrid approaches for authorship attribution, which combine multiple methods, make it more complicated and challenging for both human authors and LLMs to obfuscate their style or deceive detection systems, especially when it comes to artificially generated texts [30]. Moreover, by leveraging both traditional and new technologies, detectors can benefit from different strengths, related to each specific component, and give exhaustive results [78].

2. Limitations of Hybrid Methods

The main problem related to Hybrid approaches is that, requiring the integration of multiple models, the overall complexity inevitably increases, necessitating of considerable computing power and memory resources. This reflects a greater issue of scalability and the need for optimization for large volumes of data.

VI. Detection Datasets

From the literature analysis, emerging frequently adopted datasets are ones grouped in Table II. They are not all strictly related to the machine-generated text detection task but are derived from different natural language processing tasks. In many analyzed works, in fact, existing datasets are adopted as examples of human-written text, while machine-generated text is produced ad-hoc by a selected LLM. This means that learning models' capacity to identify generated content is often related to the application domain of the adopted dataset and the LLM adopted to produce new text. Another important limitation concerns the fact that the majority of the corpus is written in English. This means that constructed models are more powerful in detecting generated text in English.

VII. Detection Tools

Considered literature also includes the performance evaluation of tools for machine-generated text detection available at the state-of-the-art level. This section tries to summarize their results in order to highlight possible practical solutions to apply or provide a benchmark for new implementations. The section also investigates how the detectors have been developed.

Copyleaks[2] combines many techniques. Trillions of data were collected from universities and enterprises worldwide to train the model. TurnItIn[3] model is trained on AI-generated and academic writing. They also gave significance to the language they considered. Indeed, they included second language learners or texts written by people who use English but who are not native speakers. The training data is based on different subject areas. Scribbr[4] uses the analysis of stylistic patterns and sentence structure, and it employs algorithms that have been trained on big collections of content written by humans

[2] https://copyleaks.com/ai-content-detector

[3] https://www.turnitin.com/

[4] https://www.scribbr.com/ai-detector/

TABLE III. Performance of Detection Tools

Detector	Accuracy [100]	Accuracy [101]	Fee	Overview
Copyleaks	100%	91%	Free with limitations	Details about the model are not publicly available.
Turnitin	100%	-	Institutional subscription	The model is trained to detect wordprobability differences, leveraging the principle that AI generates words predictably, while human writing is more varied and unpredictable.
Originality.ai	98%	-	$0.01 per 100 words	The tool uses supervised learning with several models, including BERT and a version of Roberta.
Scribbr	88%	-	Free with limitations	The tool uses the analysis of stylistic pattern and sentence structure. It employs algorithms that have been trained on big collections of content written by humans and generated by machines.
ZeroGPT	87%	-	Free with limitations	The tool utilizes a multi-stage deep learning methodology (*DeepAnalyse*) developed by the ZeroGPT's team and trained on different kinds of datasets.
Writer	71%	99%	Free with limitations	Details about the model are not publicly available.
Content at Scale	71%	48%	Free with limitations	The tool uses both NLP and a trained model to identify specific aspects that lead to a higher likelihood of a text being detected as AI-generated (e.g., predicting likely next word choices and recognizing sentence structure).

and generated by machines. Originality.ai[5] uses supervised learning with several models, including BERT and a version of Roberta, and it has been trained on millions of examples of generated and human text. ZeroGPT[6] utilizes the so-called DeepAnalyse technology (a multi-stage methodology developed by ZeroGPT's team) to determine the origin of a given text, leveraging a deep learning methodology trained on different kinds of datasets. Content at Scale[7] uses both natural language processing and a trained model to identify specific aspects that lead to a higher likelihood of a text being detected as AI-generated, for example, by predicting likely next-word choices and recognizing sentence structure. Details about the model used by Writer[8] are not publicly available.

Table III contains a summary of aforementioned tools by reporting their performance in terms of Accuracy [100] [101]. It highlights interesting performance even for free solutions. Nevertheless, one must take into account that, in general, the performance of available tools decreases with the adoption of GPT-4 [102] or by paraphrasing the text [103].

VIII. Discussion

Reviewing the existing methodologies for distinguishing between human-written and machine-generated texts revealed some ongoing problems and limitations. These aspects must be taken into account when developing new discriminators or methodologies to enhance the effectiveness of proposed solutions.

A. Languages

Current methodologies may perform well for English texts but may not produce the same results for other languages [104].

The variation in grammar, syntax, and idiomatic expressions presents a significant challenge. Findings indicate that most existing black-box methods are ineffective when applied in multilingual environments, with statistical approaches significantly trailing behind fine-tuned models [105]. So, it is crucial to develop approaches that are effective across a wide range of languages [106].

[5] https://originality.ai/ai-checker

[6] https://www.zerogpt.com/

[7] https://contentatscale.ai/ai-content-detector/

[8] https://writer.com/ai-content-detector/

B. Hybrid Text

With the increase in hybrid texts, which combine human content with generated content, the analysis of such texts becomes more complex. Current methodologies may not be robust enough to handle this complexity, necessitating adaptations or new strategies specifically for hybrid texts [87]. Zeng et al. [88] highlight that generated-text detection with hybrid texts is tough for several reasons:

1. Human writers often select and edit machine-generated phrases based on their personal style;
2. The swap of authorship between adjacent sentences creates difficulties for segment detectors;
3. Brief text segments give little stylistic evidence, not allowing definitive authorship identification.

C. New-Generation LLMs

New large language models are rapidly developing, bringing new capabilities and challenges. Continuously testing and updating discrimination methods is essential to ensure that evaluations remain accurate and relevant because methodologies that worked well with previous models may no longer be effective [107].

D. Lack of Regulation

The application of regulations by LLM developers would be useful in making the generated text a more reliable tool and less prone to misuse. In particular, measures like the *AI Act* [108] and the *Telephone Consumer Protection Act* [109] aiming to regulate artificial intelligence adoption should be fine-tuned.

IX. Conclusions

Despite the significant progress in the generated-text detection task, there are still many challenges to address. Indeed, the effectiveness of current detection methodologies is strictly related to the application context and the complexity of the text under analysis. Recent methods, such as those using Transformer models, show promising results in terms of accuracy but require many computational resources and may be less effective for multilingual contexts or short texts. In contrast, traditional statistical methods need fewer resources but suffer from limitations in terms of precision and adaptability to complex texts.

For these reasons, hybrid approaches, which combine different methodologies, could be the most promising solution. However, the rise of hybrid texts, blending human-generated content with machine-generated content, presents new challenges that require further research and innovation to overcome.

The lack of specific regulations regarding generated text is a major hurdle to overcome. The existence of rules would help ensure the recognition of generated text and make users more conscious of what they are reading. Therefore, regulations should be involved in the generated text detection task.

Acknowledgment

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] F. García-Peñalvo, A. Vázquez-Ingelmo, *et al.*, "What do we mean by genai? a systematic mapping of the evolution, trends, and techniques involved in generative ai," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 4, pp. 7–16, 2023.

[2] J. Oluwaseyi, K. Potter, "Exploring natural language generation (nlg) methods for generating human-like text from structured or unstructured data," *Journal of Machine to Machine Communications*, 12 2023.

[3] A. Tlili, B. Shehata, M. A. Adarkwah, A. Bozkurt, D. T. Hickey, R. Huang, B. Agyemang, "What if the devil is my guardian angel: Chatgpt as a case study of using chatbots in education," *Smart learning environments*, vol. 10, no. 1, p. 15, 2023.

[4] M. Alier, F. J. García-Peñalvo, J. D. Camba, "Generative Artificial Intelligence in education: From deceptive to disruptive," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 5, Special issue on Generative Artificial Intelligence in Education, pp. 5–14, 2024.

[5] J. Wu, S. Yang, R. Zhan, Y. Yuan, D. F. Wong, L. S. Chao, "A survey on llm-generated text detection: Necessity, methods, and future directions," *ArXiv*, vol. abs/2310.14724, 2023.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Red Hook, NY, USA, 2017, p. 6000–6010, Curran Associates Inc.

[7] Z. Yang, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[8] S. Grassini, "Shaping the future of education: exploring the potential and consequences of ai and chatgpt in educational settings," *Education Sciences*, vol. 13, no. 7, p. 692, 2023.

[9] J. Jeon, S. Lee, "Large language models in education: A focus on the complementary relationship between human teachers and chatgpt," *Education and Information Technologies*, 05 2023, doi: 10.1007/s10639-023-11834-1.

[10] V. Parra, P. Sureda, A. Corica, S. Schiaffino, D. Godoy, "Can Generative AI solve Geometry Problems? Strengths and Weaknesses of LLMs for Geometric Reasoning in Spanish," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 5, Special issue on Generative Artificial Intelligence in Education, pp. 65–74, 2024, doi: 10.9781/ijimai.2024.02.009.

[11] A. Thirunavukarasu, D. Ting, K. Elangovan, L. Gutierrez Sinisterra, T. Tan, D. Ting, "Large language models in medicine," *Nature Medicine*, vol. 29, 07 2023, doi: 10.1038/s41591-023-02448-8.

[12] S. Hajijama, D. Juneja, P. Nasa, "Large language model in critical care medicine: Opportunities and challenges," *Indian Journal of Critical Care Medicine*, vol. 28, no. 6, pp. 523–525, 2024.

[13] Y. Feng, S. Vanam, M. Cherukupally, W. Zheng, M. Qiu, H. Chen, "Investigating code generation performance of chatgpt with crowdsourcing social data," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2023, pp. 876–885.

[14] R. Khoury, A. R. Avila, J. Brunelle, B. M. Camara, "How secure is code generated by chatgpt?," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2023, pp. 2445–2451.

[15] V. C. C. Kwok-Yan Lam, Z. K. Yeong, "Applying large language models for enhancing contract drafting," in *Proceedings of the Third International Workshop on Artificial Intelligence and Intelligent Assistance for Legal Professionals in the Digital Workspace (LegalAIIA 2023)*, 2023.

[16] J. H. Choi, "How to use large language models for empirical legal research," *Journal of Institutional and Theoretical Economics (Forthcoming)*, 2023.

[17] J. Cui, Z. Li, Y. Yan, B. Chen, L. Yuan, "Chatlaw: Open- source legal large language model with integrated external knowledge bases," *arXiv preprint arXiv:2306.16092*, 2023.

[18] J. Tan, H. Westermann, K. Benyekhlef, "Chatgpt as an artificial lawyer?," in *AI4AJ@ ICAIL*, 2023.

[19] X. Fang, S. Che, M. Mao, H. Zhang, M. Zhao, X. Zhao, "Bias of ai-generated content: an examination of news produced by large language models," *Scientific Reports*, vol. 14, no. 1, p. 5224, 2024.

[20] C. Novelli, F. Casolari, P. Hacker, G. Spedicato, L. Floridi, "Generative ai in eu law: Liability, privacy, intellectual property, and cybersecurity," *EU Law: Liability, Privacy, Intellectual Property, and Cybersecurity (January 14, 2024)*, 2024.

[21] L. Tang, Y.-S. Su, "Ethical Implications and Principles of Using Artificial Intelligence Models in the Classroom: A Systematic Literature Review," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 8, no. 5, 2024.

[22] I. Ulnicane, "Governance fix? power and politics in controversies about governing generative ai," *Policy and Society*, p. puae022, 2024.

[23] S. S. Ghosal, S. Chakraborty, J. Geiping, F. Huang, D. Manocha, A. Bedi, "A survey on the possibilities & impossibilities of ai-generated text detection," *Transactions on Machine Learning Research*, 2023.

[24] D. Beresneva, "Computer-generated text detection using machine learning: A systematic review," in *Natural Language Processing and Information Systems: 21st International Conference on Applications of Natural Language to Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings 21*, 2016, pp. 421–426, Springer.

[25] G. Jawahar, M. Abdul-Mageed, V. Laks Lakshmanan, "Automatic detection of machine generated text: A critical survey," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 2296–2309.

[26] M. Dhaini, W. Poelman, E. Erdogan, "Detecting chatgpt: A survey of the state of detecting chatgpt-generated text," in *Proceedings of the 8th Student Research Workshop associated with the International Conference Recent Advances in Natural Language Processing*, 2023, pp. 1–12.

[27] E. Crothers, N. Japkowicz, H. Viktor, "Machine-generated text: A comprehensive survey of threat models and detection methods," *IEEE Access*, vol. PP, pp. 1–1, 01 2023, doi: 10.1109/ACCESS.2023.3294090.

[28] R. Tang, Y.-N. Chuang, X. Hu, "The science of detecting llm-generated text," *Communications of the ACM*, vol. 67, p. 50–59, mar 2024, doi: 10.1145/3624725.

[29] X. Yang, L. Pan, X. Zhao, H. Chen, L. R. Petzold, W. Y. Wang, W. Cheng, "A survey on detection of llms-generated content," *ArXiv*, vol. abs/2310.15654, 2023.

[30] A. Uchendu, T. Le, D. Lee, "Attribution and obfuscation of neural text authorship: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 25, no. 1, pp. 1–18, 2023.

[31] Chenxi Gu and Chengsong Huang and Xiaoqing Zheng and Kai-Wei Chang and Cho-Jui Hsieh, "Watermarking Pre-trained Language Models with Backdooring," *ArXiv*, vol. abs/2210.07543, 2022.

[32] E. Lucas, T. Havens, "GPTs don't keep secrets: Searching for backdoor watermark triggers in autoregressive language models," in *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, Toronto, Canada, July 2023, pp. 242–248, Association for Computational Linguistics.

[33] R. Tang, Q. Feng, N. Liu, F. Yang, X. Hu, "Did you train on my dataset? towards public dataset protection with cleanlabel backdoor watermarking," *ACM SIGKDD Explorations Newsletter*, vol. 25, p. 43–53, jul 2023, doi: 10.1145/3606274.3606279.

[34] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, T. Goldstein, "A watermark for large language models," in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202 of *Proceedings of Machine Learning Research*, 23–29 Jul 2023, pp. 17061–17084, PMLR.

[35] J. Kirchenbauer, J. Geiping, Y. Wen, M. Shu, K. Saifullah, K. Kong, K.

Fernando, A. Saha, M. Goldblum, T. Goldstein, "On the reliability of watermarks for large language models," in *The Twelfth International Conference on Learning Representations*, 2024.

[36] A. Liu, L. Pan, X. Hu, S. Li, L. Wen, I. King, S. Y. Philip, "An unforgeable publicly verifiable watermark for large language models," in *The Twelfth International Conference on Learning Representations*, 2023.

[37] A. Hou, J. Zhang, T. He, Y. Wang, Y.-S. Chuang, H. Wang, L. Shen, B. Van Durme, D. Khashabi, Y. Tsvetkov, "Semstamp: A semantic watermark with paraphrastic robustness for text generation," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 4067–4082.

[38] Por, Lip Yee and Wong, KokSheik and Chee, Kok Onn, "UniSpaCh: A text-based data hiding method using Unicode space characters," *Journal of Systems and Software*, vol. 85, pp. 1075–1082, may 2012, doi: 10.1016/j.jss.2011.12.023.

[39] U. Topkara, M. Topkara, M. J. Atallah, "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*, 2006, pp. 164–174.

[40] Munyer, Travis and Tanvir, Abdullah and Das, Arjon and Zhong, Xin, "DeepTextMark: A Deep Learning- Driven Text Watermarking Approach for Identifying Large Language Model Generated Text," *IEEE Access*, vol. PP, pp. 1–1, 01 2024, doi: 10.1109/ACCESS.2024.3376693.

[41] X. Yang, J. Zhang, K. Chen, W. Zhang, Z. Ma, F. Wang, N. Yu, "Tracing text provenance via context-aware lexical substitution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 11613–11621.

[42] K. Yoo, W. Ahn, J. Jang, N. Kwak, "Robust multi-bit natural language watermarking through invariant features," in *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[43] X. Yang, K. Chen, W. Zhang, C. Liu, Y. Qi, J. Zhang, H. Fang, N. Yu, "Watermarking text generated by black-box language models," *arXiv preprint arXiv:2305.08883*, 2023.

[44] S. Abdelnabi, M. Fritz, "Adversarial watermarking transformer: Towards tracing text provenance with data hiding," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 121–140, IEEE.

[45] A. Modupe, T. Celik, V. Marivate, O. Olugbara, "Post- authorship attribution using regularized deep neural network," *Applied Sciences*, vol. 12, p. 7518, 07 2022, doi: 10.3390/app12157518.

[46] T. Schuster, R. Schuster, D. Shah, R. Barzilay, "The limitations of stylometry for detecting machine-generated fake news," *Computational Linguistics*, vol. 46, pp. 1–18, 03 2020, doi: 10.1162/COLI_a_00380.

[47] T. Kumarage, J. Garland, A. Bhattacharjee, K. Trapeznikov, S. Ruston, H. Liu, "Stylometric detection of ai-generated text in twitter timelines," *arXiv preprint arXiv:2303.03697*, 2023.

[48] L. Frohling, A. Zubiaga, "Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover," *PeerJ Computer Science*, vol. 7, p. e443, 04 2021, doi: 10.7717/peerj-cs.443.

[49] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, "Spotting llms with binoculars: Zero-shot detection of machine- generated text," *arXiv preprint arXiv:2401.12070*, 2024.

[50] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, "Detectgpt: zero-shot machine-generated text detection using probability curvature," in *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, 2023, JMLR.org.

[51] S. Venkatraman, A. Uchendu, D. Lee, "Gpt-who: An information density-based machine-generated text detector," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 103–115.

[52] A. Aich, S. Bhattacharya, N. Parde, "Demystifying neural fake news via linguistic feature-based interpretation," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 6586–6599.

[53] J. Su, T. Zhuo, D. Wang, P. Nakov, "Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 12395–12412.

[54] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, A. Szlam, "Real or fake? learning to discriminate machine from human generated text," *arXiv preprint arXiv:1906.03351*, 2019.

[55] J. D. Rodriguez, T. Hay, D. Gros, Z. Shamsi, R. Srinivasan, "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*, 2022, pp. 1213–1233.

[56] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert- Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, *et al.*, "Release strategies and the social impacts of language models," *arXiv preprint arXiv:1908.09203*, 2019.

[57] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, "Defending against neural fake news," *Advances in neural information processing systems*, vol. 32, 2019.

[58] E. Crothers, N. Japkowicz, H. Viktor, P. Branco, "Adversarial robustness of neural-statistical features in detection of generative transformers," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8, IEEE.

[59] S. Gehrmann, H. Strobelt, A. M. Rush, "Gltr: Statistical detection and visualization of generated text," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2019, pp. 111–116.

[60] X. Hu, P.-Y. Chen, T.-Y. Ho, "Radar: Robust ai-text detection via adversarial learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 15077–15095, 2023.

[61] E. Clark, T. August, S. Serrano, N. Haduong, S. Gururangan, N. A. Smith, "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)*, 2021, pp. 7282–7296.

[62] D. Ippolito, D. Duckworth, D. Eck, "Automatic detection of generated text is easiest when humans are fooled," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1808–1822.

[63] A. Uchendu, Z. Ma, T. Le, R. Zhang, D. Lee, "Turingbench: A benchmark environment for turing test in the age of neural text generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 2001–2016.

[64] Y. Dou, M. Forbes, R. Koncel-Kedziorski, N. A. Smith, Y. Choi, "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)*, 2022, pp. 7250–7274.

[65] L. Kushnareva, D. Cherniavskii, V. Mikhailov, E. Artemova, S. Barannikov, A. Bernstein, I. Piontkovskaya, D. Piontkovski, E. Burnaev, "Artificial text detection via examining the topology of attention maps," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 635–649.

[66] X. Liu, Z. Zhang, Y. Wang, H. Pu, Y. Lan, C. Shen, "Coco: Coherence-enhanced machine-generated text detection under low resource with contrastive learning," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 16167–16188.

[67] R. Tan, B. Plummer, K. Saenko, "Detecting cross- modal inconsistency to defend against neural fake news," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 2081–2106.

[68] L. Wang, W. Yang, D. Chen, H. Zhou, Y. Lin, F. Meng, J. Zhou, X. Sun, "Towards codable text watermarking for large language models," *arXiv preprint arXiv:2307.15992*, 2023.

[69] R. Kuditipudi, J. Thickstun, T. Hashimoto, P. Liang "Robust Distortion-free Watermarks for Language Models," *Transactions on Machine Learning Research*, 2024.

[70] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "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)*, Minneapolis, Minnesota, June 2019, Association for Computational Linguistics.

[71] A. Muñoz-Ortiz, C. Gómez-Rodríguez, D. Vilares, "Contrasting linguistic patterns in human and llm- generated text," *arXiv preprint arXiv:2308.09067*, 2023.

[72] R. Corizzo, S. Leal-Arenas, "One-class learning for ai- generated essay detection," *Applied Sciences*, vol. 13, no. 13, 2023, doi: 10.3390/app13137901.

[73] A. Shah, P. Ranka, U. Dedhia, S. Prasad, S. Muni, K. Bhowmick, "Detecting and unmasking ai-generated texts through explainable artificial intelligence using stylistic features," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 10, 2023.

[74] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[75] D. Macko, R. Moro, A. Uchendu, J. Lucas, M. Yamashita, M. Pikuliak, I. Srba, T. Le, D. Lee, J. Simko, *et al.*, "Multitude: Large-scale multilingual machine-generated text detection benchmark," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 9960–9987.

[76] R. Gaggar, A. Bhagchandani, H. Oza, "Machine-generated text detection using deep learning," 2023. [Online]. Available: https://arxiv.org/abs/2311.15425.

[77] J. A. Guerrero, "Detecting ai generated text using neural networks," Master's thesis, Texas A&M University, 2023.

[78] Y. Zhang, Q. Leng, M. Zhu, R. Ding, Y. Wu, J. Song, Y. Gong, "Enhancing text authenticity: A novel hybrid approach for ai-generated text detection," in *2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI)*, 2024, pp. 433–438.

[79] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[80] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[81] S. Merity, C. Xiong, J. Bradbury, R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[82] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.

[83] X. Zhang, J. Zhao, Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[84] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[85] A. Fan, M. Lewis, Y. Dauphin, "Hierarchical neural story generation," *arXiv preprint arXiv:1805.04833*, 2018.

[86] M. Lee, P. Liang, Q. Yang, "Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities," in *Proceedings of the 2022 CHI conference on human factors in computing systems*, 2022, pp. 1–19.

[87] A. Richburg, C. Bao, M. Carpuat, "Automatic authorship analysis in human-ai collaborative writing," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC- COLING 2024)*, 2024, pp. 1845–1855.

[88] Z. Zeng, S. Liu, L. Sha, Z. Li, K. Yang, S. Liu, D. Gavsevi'c, G. Chen, "Detecting ai-generated sentences in human- ai collaborative hybrid texts: Challenges, strategies, and insights," *arXiv preprint arXiv:2403.03506v4*, 2024.

[89] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[90] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, *et al.*, "Findings of the 2016 conference on machine translation (wmt16)," in *First conference on machine translation*, 2016, pp. 131–198, Association for Computational Linguistics.

[91] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, X. Lu, "Pubmedqa: A dataset for biomedical research question answering," 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)*, 2019, pp. 2567–2577.

[92] M. Bień, M. Gilski, M. Maciejewska, W. Taisner, D. Wisniewski, A. Lawrynowicz, "Recipenlg: A cooking recipes dataset for semi-structured text generation," in *Proceedings of the 13th International Conference on Natural Language Generation*, 2020, pp. 22–28.

[93] J. M. Patel, J. M. Patel, "Introduction to common crawl datasets," *Getting structured data from the internet: running web crawlers/scrapers on a big data production scale*, pp. 277–324, 2020.

[94] R. Tan, B. Plummer, K. Saenko, "Detecting cross- modal inconsistency to defend against neural fake news," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 2081–2106.

[95] Y. Chen, Y. Liu, L. Chen, Y. Zhang, "Dialogsum: A real-life scenario dialogue summarization dataset," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 5062–5074.

[96] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, "Dbpedia: A nucleus for a web of open data," in *international semantic web conference*, 2007, pp. 722–735, Springer.

[97] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, "How close is chatgpt to human experts? comparison corpus, evaluation, and detection," *arXiv preprint arXiv:2301.07597*, 2023.

[98] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[99] S. Bowman, G. Angeli, C. Potts, C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 632–642.

[100] W. H. Walters, "The effectiveness of software designed to detect ai-generated writing: A comparison of 16 ai text detectors," *Open Information Science*, vol. 7, no. 1, p. 20220158, 2023.

[101] N. Ladha, K. Yadav, P. Rathore, "Ai-generated content detectors: Boon or bane for scientific writing," *Indian Journal of Science and Technology*, vol. 16, no. 39, pp. 3435–3439, 2023.

[102] G.-A. Odri, D. J. Y. Yoon, "Detecting generative artificial intelligence in scientific articles: evasion techniques and implications for scientific integrity," *Orthopaedics & Traumatology: Surgery & Research*, vol. 109, no. 8, p. 103706, 2023.

[103] M. Perkins, J. Roe, B. H. Vu, D. Postma, D. Hickerson, J. McGaughran, H. Q. Khuat, "Genai detection tools, adversarial techniques and implications for inclusivity in higher education," *arXiv preprint arXiv:2403.19148*, 2024.

[104] W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, J. Zou, "Gpt detectors are biased against non-native english writers," *Patterns*, vol. 4, no. 7, p. 100779, 2023, doi: https://doi.org/10.1016/j.patter.2023.100779.

[105] D. Macko, R. Moro, A. Uchendu, J. Lucas, M. Yamashita, M. Pikuliak, I. Srba, T. Le, D. Lee, J. Simko, M. Bielikova, "MULTITuDE: Large-scale multilingual machine- generated text detection benchmark," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, Dec. 2023, pp. 9960–9987, Association for Computational Linguistics.

[106] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, O. M. Afzal, T. Mahmoud, G. Puccetti, T. Arnold, *et al.*, "Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection," *arXiv preprint arXiv:2404.14183*, 2024.

[107] A. Bhattacharjee, R. Moraffah, J. Garland, H. Liu, "Eagle: A domain generalization framework for ai-generated text detection," *arXiv preprint arXiv:2403.15690*, 2024.

[108] O. J. of the European Union, "Regulation (eu) 2024/1689 of the european parliament and of the council." https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689. Last accessed: 23 September 2024.

[109] F. C. Commission, "Telephone consumer protection act." https://docs.fcc.gov/public/attachments/ DOC-404036A1.pdf. Last accessed: 23 September 2024.

Serena Fariello

Serena Fariello received a bachelor's degree in Statistics for Big Data from the University of Salerno, Italy, in 2023. She is currently a Data Science and Innovation Management student at the same university.

Giuseppe Fenza

Giuseppe Fenza received the Graduate degree and the Ph.D. degree in computer sciences from the University of Salerno, Italy, in 2004 and 2009, respectively. He is currently an Associate Professor in computer science with the University of Salerno. He has over 60 publications in fuzzy decision making, knowledge extraction and management, situation and context awareness, semantic information retrieval, service oriented architecture, and ontology learning. More recently, he has worked in automating open source intelligence and big data analytics for counterfeiting extremism and supporting information disorder awareness. His research interests include computational intelligence methods to support semantic-enabled solutions and decision-making.

Flavia Forte

Flavia Forte received a bachelor's degree in Computer Engineering from the University of Salerno, Italy, in 2023. She is currently a Data Science and Innovation Management student at the same university.

Mariacristina Gallo

Mariacristina Gallo received her master's degree in computer science and Ph.D. in Big Data Management from the University of Salerno, Italy, in 2009 and 2021, respectively. Since 2022, she has been an Assistant Professor at the University of Salerno, Italy, where she works mainly on Social Media Analytics, Open Source Intelligence, Machine Learning, Deep Learning, Explainable Artificial Intelligence, Large Language Models applied to domains such as IoT, Environmental Security and Information Disorder.

Martina Marotta

Martina Marotta received a bachelor's degree in Computer Engineering from the University of Salerno, Italy, in 2023. She is currently a Data Science and Innovation Management student at the same university.