European Commission

Technical Solutions for Marking and Detecting AI-Generated Text Content in the Context of Article 50(2) AI Act

Giovanni Puccetti
Institute of Science and Technologies of Information “A. Faedo”, National Research Council
Pisa, Italy

Consiglio Nazionale delle Ricerche

Istituto di Scienza e Tecnologie dell’Informazione “A. Faedo”

EUROPEAN COMMISSION

Directorate-General for Communications Networks, Content and Technology (CNECT)
Directorate A — Artificial Intelligence Office
Unit A.2 — Regulation and Compliance

Contact: Nandi Robijns

E-mail: nandi.robijns@ec.europa.eu

European Commission
B-1049 Brussels

Technical Solutions for Marking and Detecting AI-Generated Text Content in the Context of Article 50(2) AI Act

Directorate-General for Communications Networks, Content and Technology (CNECT)

Manuscript completed in January – 2026

Luxembourg: Publications Office of the European Union, 2026
KK-01-25-156-EN-N
ISBN 978-92-68-34414-9
doi:10.2759/7579127

Executive Summary

This report provides a high-level analysis of currently available approaches for marking and detecting AI-generated text. The focus is on techniques to mark and detect the textual output of Generative AI models and systems, often relying on the use of Large Language Models.

We categorize available techniques into five methodologies: watermarking, structural marking, metadata, logging and AI-generated text detection. For each methodology we describe how it works, analyse relevant literature, and describe limitations and possible mitigation strategies.

We then map the desirable properties of the different techniques to the requirements set out in Articles 50(2) and 50(5) of the European AI Act, namely Effectiveness, Robustness, Reliability, Accessibility and Interoperability. We define the requirements as follows:

- **Effectiveness:** The extent to which a technique can identify AI-generated texts from human-written texts.
- **Robustness:** The maturity of a technique and its resilience to editing.
- **Reliability:** How consistently a technique performs marking across different scenarios (e.g. text generation for different purposes and domains) and how difficult it is to learn and forge the mark.
- **Accessibility:** How easily stakeholders can verify a technique and correctly interpret its responses and operation.
- **Interoperability:** The feasibility of a unified detection approach that is provider- and technology -agnostic.

Our assessment highlights that existing techniques are promising, although further improvements in effectiveness, robustness and reliability would be desirable. Indeed, methodologies are close to meeting each of these requirements individually, but no single methodology meets all of them. Specifically, robust techniques, detectable despite significant modifications of the text, are vulnerable to spoofing attacks. Third parties can edit a text without removing the marking, thereby attributing malicious content to a provider. On the contrary, reliable techniques, which cannot be forged, are subject to scrubbing attacks, i.e. they can be easily removed. For example, a digitally signed metadata is fully reliable but can only be embedded in richer formats and is easily removed. Additionally, we find that all methodologies can meet Accessibility requirements while none can meet Interoperability ones without agreement among providers and between providers and regulating authorities.

In principle, integrating robust solutions such as watermarking with reliable content credentials, e.g. C2PA, provides a robust and reliable marking technique. However, this mutual application does not fully address the individual limitations of the two methodologies, one can remove the content credential and modify the text to enact a spoofing attack.

Article 50(2) of the AI Act acknowledges the challenge of fully meeting all the requirements, stating that requirements must be met as far as this is technically feasible. This consideration of technical feasibility will likely need to be taken into account when establishing guidelines.

Article 50(2) also provides an exception for those cases where Generative AI has only been used for an assistive purpose, yet there is no definition of what an assistive purpose is. Possible interpretations are certain cases when Generative AI is tasked with standard editing and grammar corrections of an authentic human-written text.

In addition to the requirements of Article 50 of the AI Act, we analyse three additional properties of marking methodologies:

- **Computational Costs**: how computationally expensive each methodology is;
- **Text Quality Degradation**: how strongly each methodology affects text quality;
- **Applicability to Multimodal Models**: how well each methodology can be used in multimodal models.

Our assessment highlights what follows: computational costs are limited compared to those of generative AI itself; methodologies do not significantly lower the text quality and the methodologies we study can mark the textual output of multimodal systems.

Finally, we provide recommendations for good practices in implementing Article 50(2) of the AI Act. We believe that having the availability of watermarking directly embedded in the text and metadata-based techniques embedded in structured formats would be desirable for most applications of Generative AI for text. However, we try to stress which techniques are, in our opinion, most relevant for each application. We identify four Generative AI application categories and point out which marking techniques are best suited for each category. Specifically, we believe that:

- General-purpose text generation could use metadata and watermarking as primary solutions possibly complemented by logging and AI-generated text detection;

- Application-oriented text generation (e.g. product-specific user assistance) could use watermarking possibly complemented by AI-generated text detection;
- Generation of sensitive content (e.g. AI-generated medical documents) could use Metadata-based techniques;
- Open weight models could use structural marking.

Additionally, we propose recommendations in the form of Scenarios which describe potential solutions to address the interoperability and transparency requirements set out in Article 50 of the AI Act.

For interoperability, one of the proposed scenarios is described as a Centralized Verifier, established by a trusted authority with voluntary provider participation. The verifier would offer a publicly available interface through which end-users can submit a text to determine whether it was generated by a participating provider. The verification process would consist of verifying if the text has any authentic metadata attached and if not sending it to each provider detection system to verify it. This approach would determine with sufficient confidence if a text has been generated by one of the participating providers and foster standardization and interoperability at the same time.

A similar but less demanding scenario is also proposed, a Decentralized Verifier, where similar processes as described for the centralized verifier would be put in place by trusted third parties instead of a single centralized one. This would enforce fewer standardization requirements for providers.

For accessibility, we believe a similar mechanism to the already implemented cookie-banners would suffice. This approach would mirror existing user experiences available for those components which already have to be EAA compliant. While this approach would address several accessibility requirements, it would also impose a non-negligible burden on users.

To conclude, we believe current marking techniques for AI-generated text are effective, robust and reliable although they still suffer from few weaknesses. Some of these weaknesses can be addressed through multi-layered solutions adopting multiple methodologies, particularly watermarking and metadata-based techniques. However, not all limitations can be removed in this way, and they would still have to be addressed before finally being able to pursue the core goal of making interactions with AI-generated text content always explicitly identifiable.

1. Introduction

The goal of this report is to inform the reader about the status of existing techniques utilized for marking and detecting textual outputs of Generative AI, within the context of the requirements outlined in AI Act Article 50, specifically Article 50(2) and 50(5).

This is a novel research domain, largely initiated in 2023. There is previous literature on the attribution of text to specific sources, for example, studies in authorship attribution. However, these methodologies have been tested for AI-generated text (Uchendu et al., 2020) and consistently demonstrated shortcomings. A promising approach to overcome these shortcomings is the use of techniques based on marking the output of Generative AI so that it can be more easily detected when needed.

1.1. Background and Objectives

There are several approaches for marking and detecting AI-generated texts. This report aims to provide an in-depth analysis of these techniques through detailed and clear descriptions of their functioning alongside illustrative examples. In addition to describing the relevant *Techniques*, we subdivide them into five groups called *Methodologies*: **Watermarking**, **Structural Marking**, **Metadata, Logging and AI-generated text detection**. This subdivision is based on existing surveys and benchmarks of AI-generated text marking and detection, which will later be used to relate existing solutions to the AI Act requirements. We will describe techniques for marking and detection of AI-generated text without referring to their meaning as understood for other modalities.

The subdivision of techniques into five separate methodologies is based on the technical differences among them, which require separate analyses. This subdivision will provide a more accurate description of how the techniques function and present a clear understanding of the capabilities of existing solutions.

Among the techniques explored, some provide results beyond the specific goal of marking text as AI-generated. For example, some techniques encode longer messages in text instead of a single bit (0 for AI-generated, 1 for not AI-generated). These techniques fall outside the scope of AI Act Article 50 because they provide more general functionalities than AI-generated text detection. Nevertheless, they are likely to be instrumental in the development of stronger markings and to improvements in existing approaches.

To link the status of existing marking and detection techniques to the AI Act, we review recent literature to identify desired properties and define quantitative and qualitative metrics for their evaluation. Specifically, to characterize the described

techniques and understand how well they meet the requirements of the AI Act, we focus on developing an assessment framework tailored to AI-generated text marking and detection techniques. For a systematic analysis, we identify desired properties and map them to the requirements outlined in AI Act Article 50 for a compliant deployment of Generative AI. Key requirements outlined in AI Act Article 50 explored in this report include:

- **AI Act Article 50(2):** The outputs of Generative AI shall be marked in a machine-readable format and are detectable as artificially generated or manipulated. Moreover, it is required that providers of AI systems ensure that the technical solutions they adopt for the detection of AI-generated content are **Effective**, **Robust**, **Reliable** and **Interoperable** in so far as this is technically feasible.
- **AI Act Article 50(5):** The information requested is provided at the latest at the time of the first interaction of an interested user with the content and that the information is provided in a clear and distinguishable manner. Additionally, the information shall conform to the applicable accessibility requirements.

To systematize these criteria into measurable quantities, we summarize them into five Requirements for attribution techniques: **Effectiveness**, **Robustness**, **Reliability**, **Accessibility and Interoperability**.

After creating a map between properties and requirements, we assess existing state-of-the-art techniques for the marking and detection of AI-generated text. The assessment provides a systematic analysis of each technique and methodology with respect to the requirements set out in AI Act Article 50.

Finally, we provide an overall analysis of the outcomes of the report and technical recommendations for achieving AI Act-compliant implementation of AI-generated text marking and detection techniques. At the time of writing, Generative AI solutions for text are mostly limited to transformer-based autoregressive Large Language Models (LLMs), including ChatGPT. As a result, while most studied techniques are architecture-agnostic, the techniques reviewed in this report are tested on transformer-based autoregressive models.

1.2. Generative AI for Text

To put marking techniques in the right context and provide a minimal shared background on Generative AI, this section provides a brief overview of how Generative AI for text is used in practice along with prominent providers and notable applications.

LLMs have deeply impacted the way textual content is produced through their ability to generate fluent text based on user's instructions. The adoption of these tools started with ChatGPT released in 2022 and reached widespread adoption in less than a year. Recently, several alternatives with comparable capabilities and possible applications have been released.

Table 1 shows some of the most widely used chat interfaces, along with a URL, if they released an open-access model and the main publications/reports concerning them. ChatGPT is the most widely used, however, competitors provide models that are similarly able to follow instructions and generate informative text.

Table 1: Most used providers of Chat interfaces using Large Language Models with the Model Name (Model Name), link to the tool (URL), Related Publications (Models Cards) and whether the providers also released their model weights (Open Weights) at the time of writing.[1]

Model Name	URL	Model Cards	Open Weights
ChatGPT	chatgpt.com	arxiv.org/abs/2303.08774	✗
Claude	anthropic.com/claude	anthropic.com/news/claude-opus-4-1	✗
Gemini	gemini.google.com	arxiv.org/abs/2312.11805	✗
Mistral	mistral.ai	mistral.ai/news/mistral-large-2407/	✓
Command R	cohere.com	docs.cohere.com/v2/docs/command-r	✓
DeepSeek	deepseek.com	arxiv.org/abs/2412.19437	✓
Olmo	allenai.org/olmo	arxiv.org/abs/2501.00656	✓
Minerva	nlp.uniroma1.it/minerva/	aclanthology.org/2024.clicit-1.77.pdf	✓
Bloom	bigscience.huggingface.co/blog/bloom	arxiv.org/abs/2211.05100	✓

[1] Last updated on 30/10/2025

Generative AI for text, specifically LLMs, has also been applied to create previously impossible services. For example, code-writing assistants, such as GitHub Copilot,[2] have been strongly improved by Generative AI. AI-powered coding assistants are generally integrated with code editors such as Visual Studio Code to provide real-time code suggestions and to generate entire code blocks. By leveraging OpenAI's Codex model or equivalents, Copilot assists developers in writing code more efficiently. Other examples include Lovable,[3] an AI-based code editor developed in Europe that offers features such as in-editor code generation, debugging assistance, and code explanations for greater development efficiency.

Generative AI for text has also been integrated with search engines. For instance, Perplexity AI[4] is serving as an AI-driven search engine through utilizing language models to answer user queries by synthesizing information from web pages, providing both informative and up to date information. Similarly, ChatGPT combines Generative AI with search functionalities, allowing users to input queries in natural language and receive comprehensive responses that unify information from multiple sources. Other providers, such as LeChat from Mistral, offer similar services.

Several applications of Generative AI and LLMs to medicine have also been explored and used in practice (Meng et al., 2024). Specifically, to help doctors write diagnostic reports faster, thereby improving efficiency. This is a relevant use-case for this study, since it falls within the scope of AI Act Article 50 and due to its sensitive nature, it must be addressed with special care.

Based on the variety of applications, it is likely that the output of Generative AI interfaces, such as ChatGPT, will be increasingly provided to users through online platforms, media outlets and other channels. Consequently, the importance of being able to distinguish when content is human-written or machine-generated will become crucial. AI Act Article 50 addresses this need by requiring transparency when users are exposed to AI-generated content.

To illustrate where AI content is more likely to be encountered, we refer to a report investigating which jobs are more likely to be affected by AI applications (Tomlinson et al., 2025). The study highlights the 40 most exposed occupations. A subset of these professions is: translators, historians, advertising sales agents, writers and authors, technical writers, political scientists, news analysts, reporters and journalists, proofreaders, editors, teachers, web developers, and library staff. This list

[2] https://github.com/features/copilot

[3] https://lovable.dev/

[4] https://www.perplexity.ai/

suggests that AI-generated content will most likely impact any occupation involved in content creation and education. This implies that we can expect AI-generated texts to be of various lengths and styles, with notable implications in the worlds of publishing and education (Kasneci et al., 2023). We believe that, in education environments, self-contained Generative AI systems whose AI-generated content can be more confidently detected with tutors access to logs will be increasingly used. This should let students benefit from a responsible use of Generative AI, e.g. during exams or school sessions, while letting tutors understand the use students make of them.

Most studies on Generative AI applications focus on closed-source models provided by companies because they are currently the best performing models available. However, with the development of established practices to train LLMs, several open-weight and open-source models with performance comparable to closed-source ones have been released from industry (Mistral, Gemma, DeepSeek) as well as academia (Minerva, EuroLLM, Kiwi). The expression *open-weight model* indicates models whose weights are released (under specific licences) by their developers. A subset of open-weights models are *open-source models*, for which developers release not only the model weights, but also information about their training data and infrastructure as well as all necessary details to independently replicate the models - which is often subject to the availability of significant compute resources. These models foster research in the field of Generative AI and enable a more widespread and open adoption of this technology.

Open-weights models provide an additional challenge to the goal of marking and detecting AI-generated text since their weights are openly available and single individuals can fine-tune them to obtain new models with different capabilities (i.e. train them on relatively small amounts of data to adjust their behaviour).[5]

Based on this short summary of applications of Generative AI, we see that this technology has been integrated into high usage services such as Search Engines and Code Assistants, it is reportedly used by many content creators and its availability and usage are increasing, with the number of providers and of open-weights models both growing consistently over time. Therefore, establishing technological solutions and infrastructures for the marking and detection of AI-generated text is socially relevant as well as required by AI Act Article 50.

However, currently, none of the major Text Generative AI providers offers user-friendly tools to attribute texts to their models through any detection methodology,

[5] The authors of the report support that this additional challenge should not be met by restricting the availability or usability of open-weights and open-source models.

despite OpenAI reportedly developing their own.[6] Other companies developing user-friendly Generative AI platforms for text, e.g. DeepSeek or Mistral, likewise do not provide interfaces to determine if a text passage was generated by their models. Google's SynthID[7] (Dathathri et al., 2024), provides the closest available example which detects the outputs of their flagship system Gemini. Although the Gemini-generated text attribution interface is not yet user-friendly it does provide code to implement custom replicas of SynthID on Huggingface[8] and the API is currently[9] available through waitlist-based access.

This report provides a review of existing techniques for marking and detection of AI-generated text, along with an assessment of existing methodologies and recommendations on good practices for an effective implementation of marking techniques.

On the 4th of September 2025, the AI Office of the European Commission held a Workshop where experts from both industry and academia provided their input and feedback on the content of this report and more broadly on how to implement the requirements in AI Act Article 50. Workshop participants provided valuable knowledge and recommendations for the development of guidelines to implement AI Act Article 50. Throughout this report, there are comments taken from this context and we refer to this workshop as *the technical workshop*.

1.3. Structure of the Report

This rest of the document is organized as follows:

- Section 2 describes the main methodologies studied and includes an analysis of state-of-the-art examples and techniques.

- Section 3 provides a set of desirable properties of marking techniques that can be measured and mapped to the requirements set out in the AI Act Article 50, and assess existing methodologies based on these properties.

[6] https://www.wsj.com/tech/ai/openai-tool-chatgpt-cheating-writing-135b755a

[7] https://deepmind.google/technologies/synthid/

[8] https://huggingface.co/spaces/google/synthid-text

[9] Checked on 06/08/2025

- Section 4 outlines additional properties not required by the AI Act that are desirable for marking techniques.

- Section 5 contains a summary of the findings along with technical recommendations for deployment of the methodologies examined.

- Section 6 contains a conclusive discussion of the report findings.

2. Marking and Detection of AI-generated Text

2.1. Task Setting

"Providers of AI systems, including general-purpose AI systems, generating synthetic audio, image, video or text content, shall ensure that the outputs of the AI system are marked in a machine-readable format and detectable as artificially generated or manipulated. Providers shall ensure their technical solutions are effective, interoperable, robust and reliable as far as this is technically feasible, taking into account the specificities and limitations of various types of content, the costs of implementation and the generally acknowledged state of the art, as may be reflected in relevant technical standards. This obligation shall not apply to the extent the AI systems perform an assistive function for standard editing or do not substantially alter the input data provided by the deployer or the semantics thereof, or where authorised by law to detect, prevent, investigate or prosecute criminal offences."

Excerpt 1: AI Act Article 50(2).

Excerpt 1 defines the key requirements for AI system providers. For the purpose of this study, we provide our definitions of these five key requirements that marking techniques should meet; these definitions are used within this report and do not necessarily reflect their official meaning in the AI Act:

- **Effectiveness:** The extent to which a technique can identify AI-generated texts from human-written texts.
- **Robustness:** The maturity of a technique and its resilience to editing.
- **Reliability:** How consistently a technique performs marking across different scenarios (e.g. text generation for different purposes and domains) and how difficult it is to learn and forge the mark.
- **Accessibility:** How easily stakeholders can verify and subsequently understand a technique and correctly interpret its responses and operation.
- **Interoperability:** The feasibility of a unified detection approach that is provider- and methodology-agnostic.

In Section 3, these requirements are further elaborated on and broken down into sets of properties that can be measured for assessment purposes.

To provide a broader analysis, we outline some additional requirements we find to be desirable to encourage widespread adoption of marking techniques. Specifically, we recommend that technical solutions for the marking and detection of AI-generated

media should be easily embeddable in the content without altering it to the point that it is noticeable to the human eye while being machine-readable.

We sustain that if output quality is not altered to the point of being noticeable by users, faster adoption of marking techniques would be encouraged. As described in Section 4, most techniques can already be employed with limited disruption of the generative abilities of AI models and systems. This additional requirement therefore does not impose overly strict restrictions on available methodologies. Indeed, there are effective marking techniques that embed a signal invisible to the human eye in text but detectable by using an algorithm and corresponding keys.

Embedding “invisible” signals to text is more challenging than in visual or audio content. Earlier marking techniques for text (Atallah et al., 2003; Topkara et al., 2006) were less developed than their equivalents for images or audio. This report focuses on methods meant specifically for marking AI-generated text. Older approaches have been excluded because, unlike generative models, they did not have access to token likelihood, and likelihood-based techniques consistently outperform older rule-based and synonym-substitution methods.

For marking text output by Generative AI systems and more particularly LLMs there are effective solutions (Kirchenbauer, Geiping, Wen, Katz, et al., 2023). However, these methods face some key limitations that are still being studied in scientific literature. We identify three main challenges that restrict the adoption of marking techniques:

1. **Sensitivity to length:** the effectiveness of marking techniques depends on the length of the marked text: which can limit their effectiveness for short texts (e.g. those used in chat-like exchanges).

2. **Marking removal:** several techniques condition models to prefer certain tokens (words) so the text can be identified as AI-generated. However, paraphrasing methods can remove such markings, severely undermining effectiveness.

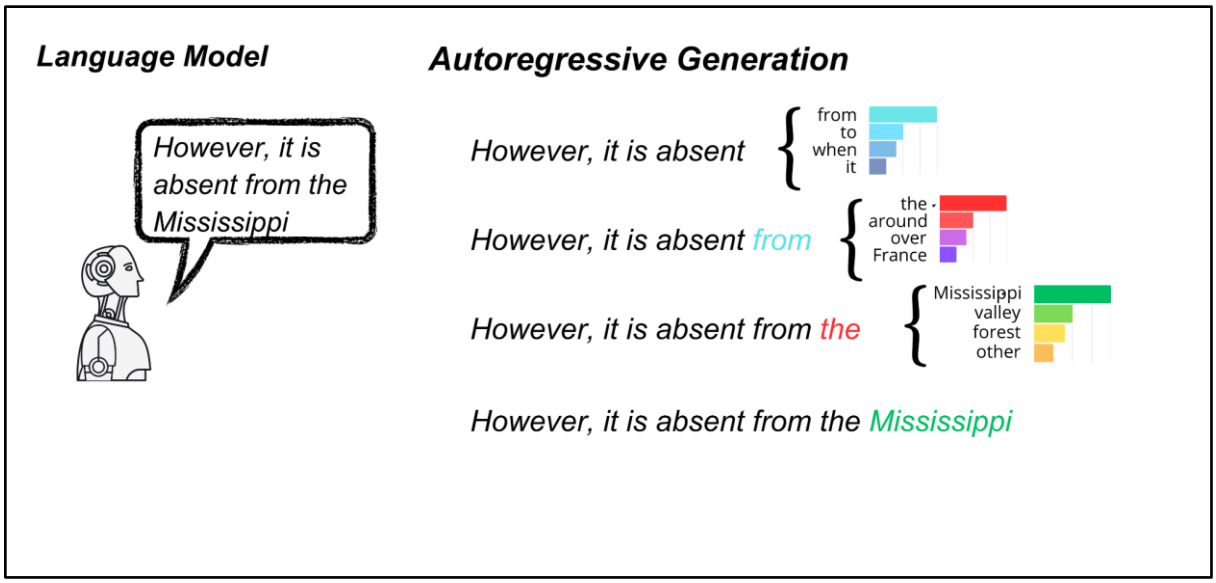
3. **Spoofing attacks:** third parties modify small portions of marked AI-generated texts to alter their meaning without removing the marking itself. The text still appears AI-generated, potentially exposing providers of Generative AI to legal consequences.

These limitations have been both exposed and mitigated in recent works. There is, however, no consensus on how to address these shortcomings across marking techniques. These kinds of limitations are accounted for by AI Act Article 50(2) which states that providers should meet the requirements “as far as this is technically feasible.” Some of the properties described in Section 3 are intended to assess the limitations of marking solutions and to provide suggestions on how to measure the technical feasibility of each requirement.

To facilitate understanding of the following analysis of marking techniques for text, we have provided a short summary of how language models generate text. Existing Generative AI models for text use an autoregressive process, where every generated word is used as the input to the next generative step. Figure 1 provides a graphical example of this process. Signals can be embedded in the output of LLMs that are detectable in word probability distribution but remain invisible when reading the text.

In this report, we refer to *word probability distribution* or *likelihood* to describe the probability distribution a language model induces over its vocabulary, e.g. the coloured bars in Figure 1. For readability purposes, we use the terms *word* and *token* interchangeably. The latter is more precise since language models generate tokens (subwords) rather than full words but using *word* does not compromise the accuracy of our explanations in this report.

Figure 1: Example of LLMs' generative process.[10]



The remainder of this section describes recent state-of-the-art techniques for marking and detecting AI-generated text. Approaches developed in research papers will be referred to as **Techniques.** These techniques are then grouped in five high level **Methodologies**: *Watermarking, Structural Marking, Metadata, Logging and AI-generated Text Detection*. For each of these methodologies we provide a review of existing techniques, with insights into their technical details.

[10] This image is created by the authors specifically for this report.

2.2. Watermarking

One of the most widely used techniques for marking and detecting AI-generated text is Red/Green watermarking. It provides an approach to attribute AI-generated text to language models and has paved the way to subsequent research on watermarking in LLMs.

2.2.1. Methodology Description

Red/Green watermarking is based on the principle that, before generating each word, the next model choice can be restricted to a subset of the vocabulary. By knowing which tokens are in each subset allows watermarked text to be detected.

When generating a new word, the algorithm randomly[11] splits the vocabulary into two sets: *Red* tokens that are intentionally assigned to a lower probability of being selected next, and *Green* tokens that are favoured to be chosen next. If the model is constrained to generate according to Red/Green sets (e.g. only allowing output tokens from the green set) those aware of the Red/Green strategy can reliably detect which texts have been generated by the watermarked model. This method also provides a measure of statistical significance for detection, providing more information than a binary classification.

Table 2 shows an example of red/green watermarked text. Specifically, the text on the left shows the outcome of a model that has not been watermarked, and the right shows the output of a model that has been watermarked. The results can be seen when highlighting green and red tokens in the text. On the left red and green tokens are similar in number and the text is classified as non-watermarked. On the right, more than 70% of the tokens are green, as expected for a watermarked text, and as a result the text is classified as watermarked. The detection consists of a hypothesis test under the null hypothesis that the text is human generated.

It is relevant to note that embedding the watermark in the text requires access to the model (i.e. it cannot be added to already generated text). Conversely, detection is possible without access to the model likelihood; one only needs to know the model tokenizer and the secret key to assign Red/Green colours to tokens.

[11] Pseudorandomly based on a key that can be shared to allow others to run detection.

Table 2: Example of red/green watermarking. On the left-hand, the model generation is not conditioned on the red/green watermarking scheme, on the right-hand it is conditioned on red/green watermarking scheme.[12]

<table>
<tr><td colspan="4">User Input
The diamondback terrapin or simply terrapin (Malaclemys terrapin) is a species of turtle native to the brackish coastal tidal marshes of the Northeastern and southern United States, and in Bermuda. It belongs to the monotypic genus Malaclemys. It has one of the largest ranges of all turtles in North America, stretching as far south as the Florida Keys and as far north as Cape Cod.</td></tr>
<tr><td colspan="2">Non watermarked Generated Output
However, it is absent from the Mississippi River delta and the Gulf of Mexico. The species is highly distinctive and can be recognized by its dark upper shell (carapace) with yellow markings, and a lower shell (plastron) that is usually yellow with dark markings. The diamondback terrapin is one of the most common species of turtle found in the New England area. It is also the only species of turtle native to the eastern United</td><td colspan="2">Watermarked Generated Output
However, it is endangered in some states and its status is uncertain in some others. It is a small to medium-sized turtle, growing 15 to 40 centimeters (6 to 16 in) in carapace length. The shell is highly variable, ranging from a smooth, shiny carapace to one with a rough, wrinkled texture and a reddish or blackish hue. The plastron is small and is made up of seven plates. The head is small and round</td></tr>
<tr><td colspan="2">Highlighted (Absent) Red/Green Scheme
However , it is absent from the Mississippi River delta and the Gulf of Mexico . The species is highly distinctive and can be recognized by its dark upper shell (car ap ace) with yellow markings , and a lower shell (pl astr on) that is usually yellow with dark markings . The diamond back terr ap in is one of the most common species of turtle found in the New England area . It is also the only species of turtle native to the eastern United</td><td colspan="2">Highlighted (Present) Red/Green Scheme
However , it is endangered in some states and its status is uncertain in some others . It is a small to medium -sized turtle , growing 15 to 40 cent imeters (6 to 16 in) in car ap ace length . The shell is highly variable , ranging from a smooth , shiny car ap ace to one with a rough , wrink led texture and a redd ish or black ish hue . The pl astr on is small and is made up of seven plates . The head is small and round</td></tr>
<tr><td>Tokens Counted</td><td>91</td><td>Tokens Counted</td><td>100</td></tr>
<tr><td># Tokens in Greenlist</td><td>51</td><td># Tokens in Greenlist</td><td>72</td></tr>
<tr><td>Fraction of T in Greenlist</td><td>56.0%</td><td>Fraction of T in Greenlist</td><td>72.0%</td></tr>
<tr><td>z-score</td><td>1.15</td><td>z-score</td><td>4.4</td></tr>
<tr><td>p value</td><td>0.124</td><td>p value</td><td>5.41e-06</td></tr>
<tr><td>z-score Threshold</td><td>4.0</td><td>z-score Threshold</td><td>4.0</td></tr>
<tr><td>Prediction</td><td>Unwatermarked</td><td>Prediction</td><td>Watermarked</td></tr>
</table>

[12] This table is the work of the author inspired by the Huggingface space https://huggingface.co/spaces/tomg-group-umd/lm-watermarking

2.2.2. Relevant Literature

One of the first works focused on watermarking the text generated by Large Language Models is the work of Kirchenbauer et al., (2023). The authors devised one of the first approaches to provide statistically verifiable watermarks for LLM output. Their watermarking scheme is sometimes referred to as KGW watermarking, after the initials of the main authors. The key idea is based on red/green watermarking, presented in Table 2.

2.2.2.1. Red/Green Watermarking

To avoid forcing errors (e.g. making it impossible to generate an article because all articles are red tokens), the authors make the watermarking “soft”: the probabilities of red words are reduced but not set to zero so that the model can generate the appropriate word when needed. One parameter of this method is the number of tokens used to compute the Red/Green lists; the authors recommend using a token window of 4.

After introducing the watermark, the same (Kirchenbauer, Geiping, Wen, Shu, et al., 2023) studied its reliability. Their analysis focuses on two key properties: the watermark's robustness (resilience to modifications from humans or models) and its reliability (resistance to being learned by third parties and subsequently forged). They found that paraphrasing is not overly destructive as it often preserves spans of text long enough to encode the watermark.

One limitation of KGW watermarking, shared by works that build on this approach, is that it modifies the token probability distribution, potentially affecting text quality. To address this issue, Zhao et al., (2023) proposed a theoretical framework to quantify watermark effectiveness and robustness. They introduced the Unigram-Watermark, which extends KGW and guarantees generation quality, ensures reliable detection, and is more robust against text editing and paraphrasing.

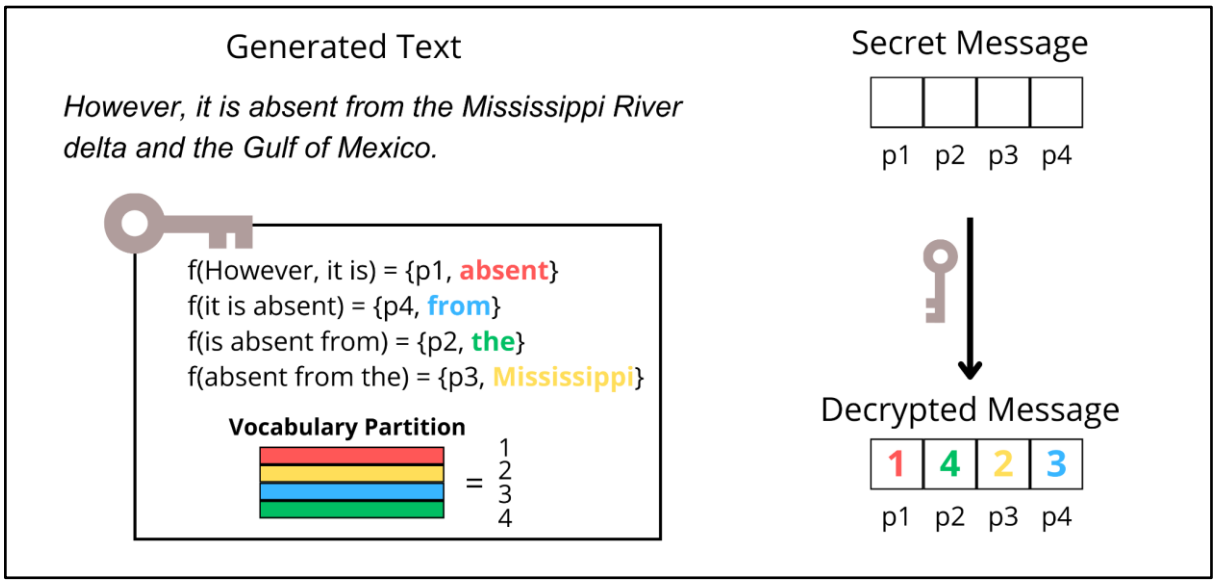
Figure 2 outlines an approach to embedding a message (e.g. a cryptographic signature) in the output of a language model. This extends on red/green watermarking by adding other “colours” (categories) to assign to tokens. Given the secret key, and the preceding tokens, the “colour” of the following token can be decoded as well as the corresponding numerical value. By counting the numerical values one can decode the hidden message with a given statistical certainty.

A multi-bit watermark (Fernandez et al., 2023) is a message hidden in the output of a Language Model that encodes more information than the binary label: human-written

or AI-generated, this allows to store more complex information in the output of AI-generated text.

Other works extend on watermarking schemes introducing meta watermarking techniques, that can improve any existing watermark with given properties (e.g. any watermark that provides a p-value for any text). WaterMAX (Giboulot & Furon, 2024) defines a general approach to make most watermarking strategies more effective: at generation time they select among an LLM's outputs for the same prompt those that are best watermarked.

Figure 2: Visual Example of a method to encode multi-bit messages in LLM outputs.[13]



2.2.2.2. Sampling based Watermarking

Watermarking techniques described so far edit the probability distribution computed by the LLM, Other approaches affect the token-sampling strategy. Specifically, by conditioning it based on a secret key.

In an early work the use of the a vector derived from a secret key is used to condition token probability so that knowing the vector allows watermark detection by computing a score (Aaronson & Kirchner, 2022). This approach makes the LLM word distribution modified conditionally on the secret vector making the watermark dependent on a hidden variable, thereby reducing changes to the LLM's distribution while making it detectable to those knowing the secret vector.

[13] This image is created by the authors specifically for this report.

Improving on this approach, Kuditipudi et al., (2023) propose two sampling strategies, Inverse Transform Sampling (ITS) and Exponential Minimum Sampling (EMS)[14] which also alter the LLM token sampling conditionally on a secret vector. They carefully evaluate the robustness of these approaches against modification attacks and find that they can reliably detect marked text from as few as 35 tokens even after replacing up to 50% of the generated ones through random edits. However, they also reveal a significant dependency on the entropy of the text distribution.

Among sampling based techniques, some works focus on assessing the non-learnability of marking, Christ et al., (2024) introduce a cryptographically-inspired notion of undetectable marking for language models that can only be detected if one knows the secret key. Without the secret key, it is computationally intractable to distinguish marked outputs from those generated from the original model. It is impossible for a user to observe any degradation in text quality. They also provide theoretical guarantees that the signal remains undetectable even when the user is allowed to adaptively query the model with arbitrarily chosen prompts.

2.2.2.3. Distribution Preserving Watermarks

A distribution-preserving (or distortion-free) watermark is one that does not change the text distribution of the LLM when used. Wu et al., (2024) introduce a watermark with this property called DiPmark. It presents three properties: original token distribution preservation (distribution-preserving), detectability without access to the language model API and prompts and provable robustness to moderate token modification. The work key innovation is the new way in which the red/green lists are constructed, minimizing the impact on model probabilities compared with the KGW strategy.

A notable methodology is SynthID developed by Google (Dathathri et al., 2024). Like KGW it divides tokens into red/green lists, but it introduces the possibility to use multiple lists simultaneously. For example, with three independent red/green lists, the model first samples eight candidate tokens from its distribution for the next token. The first red/green list is then used to select the most probable half, thus four are kept. The second list is then used to select the most probable two tokens, and finally with the third list the output token is selected. This multi-round approach uses the same decoding strategy as KGW (see Table 2) but achieves greater robustness by checking multiple lists rather than relying on only one.

Another work introduces Pseudorandom Error Correcting Codes (Christ & Gunn, 2024). These codes are error-correcting codes with the property that any polynomial

[14] ITS and EMS are techniques for sampling tokens; we refer to the paper for details.

number of codewords (e.g. marked texts) are pseudorandom to any computationally-bound adversary.

Watermarks that are robust to modifications, enables spoofing attacks: malicious actors can subtly alter the meaning of LLM-generated texts or forge harmful content, potentially misattributing blame. To overcome this, recent works introduce a two-level signature scheme, Bileve (T. Zhou et al., 2024).

2.2.2.4. Semantics Aware Watermarking

The techniques described so far do not consider the semantic meaning of the generated text. In principle, they may alter any word during the generation process, potentially affecting output meaning.[15] This is undesirable, as watermarking should preserve the semantic content of language model outputs.

SemStamp (Hou et al., 2024) proposes a technique to increase the semantic alignment between watermarked and non-watermarked outputs by adding constraints on the semantic similarity between new and previous tokens. Their semantic watermark algorithm is demonstrated to be robust and effective at preserving generation quality.

To make the watermark rely on preceding words semantics, SIR (A. Liu, Pan, Hu, Meng, et al., 2023) relies on an encoder-only LLM so that the red/green lists are not computed from the preceding words but from their embeddings. This approach is later improved with the XSIR method focused on resisting translation attacks (He et al., 2024).

Other works use token perplexity to select which tokens should encode the watermark through a semantically informed watermarking strategy. This limits the watermarking impact on the quality of the output text (Y. Liu & Bu, 2024). Specifically, they watermark only high-entropy tokens as measured by an auxiliary model, while keeping low-entropy tokens unmodified. To improve security and minimize the watermark's impact on text quality, these techniques replace fixed red/green lists with adaptive scaling of output logits. There are other watermarking techniques adapted to token likelihood, for example EWD (Lu et al., 2024).

Other works instead focus on enforcing semantic consistency between green and red tokens to preserve watermarked text meaning (Y. Fu et al., 2024). Using a global similarity matrix across all embeddings, the authors ensure higher semantic consistency between green and red tokens to balance the semantic role of green and red lists.

[15] In the sentence "The dog walks on the beach" replacing the word *dog* only based on the likelihood according to the language models might drastically change the meaning of the sentence, e.g. changing *dog* with *robot* instead of a more appropriate word such as *Doberman*.

2.2.2.5. Publicly Verifiable Watermarking

None of the techniques described so far is publicly verifiable; consequently, the detection key cannot be made public. However, some techniques attempt to enable public verifiability.

To encode publicly verifiable watermarks in multiple bits of information Fairoze et al., (2024) propose a marking scheme. The detection algorithm contains no secret information, and it is executable by anyone. They propose to embed a publicly verifiable cryptographic signature into the LLM output through rejection sampling.[16] They prove this scheme is cryptographically correct, sound, and distortion free.

A different method for publicly verifiable watermarks is UPV (A. Liu et al. 2023). UPV is an Unforgeable Publicly Verifiable watermark algorithm, that uses two different neural networks for watermark generation and detection. To improve efficiency, the token embedding parameters are shared between the generation and detection networks, enabling the detection network to achieve high performance. The authors test the difficulty of forging the mark.

2.2.2.6. Benchmarking Watermarks

To facilitate empirical evaluations, benchmarks for watermarking techniques have been developed, such as WaterBench (Tu et al., 2024). This benchmark for LLM watermarks identifies three factors: (1) the watermarking method's hyper-parameter is adjusted to reach the same watermarking strength, then generation and detection performance are jointly evaluated; (2) input and output length are diversified to form a five-category taxonomy, covering 9 tasks; (3) GPT4 as a Judge is adopted to automatically evaluate the degradation of instruction-following abilities after watermarking.

Indeed, while there are distortion-free watermarks that offer theoretical guarantees on the output quality, empirical evaluation of watermarked AI-generated text quality is equally important. WaterJudge (Molenda et al., 2024) develops a simple but effective framework combining a comparative assessment with flexible natural language generation evaluation.

The MarkLLM toolkit (Pan et al., 2024) is a framework for implementing LLM watermarking algorithms. It provides user-friendly interfaces to ensure ease of access to several watermarking techniques. Furthermore, it supports automatic visualization of the underlying watermarking algorithm mechanism.

[16] The model generated texts are tested and resampled until a given condition is met.

Finally, several watermarking techniques are described in existing surveys, notably "*A Survey of Text Watermarking in the Era of Large Language Models*" (A. Liu et al., 2024).

2.2.2.7. Marking AI-generated Code

Watermarking techniques are also used on AI-generated code. They follow similar approaches. Stricter token selection is necessary for AI-generated code since it has stronger syntactic constraints. Generally, techniques for marking code select those tokens that can be watermarked (e.g. variable names, comments, and others) and avoid adding watermarks where perplexity is low, for example the SWEET watermark (Lee et al., 2024) is based on this approach. This works because the model is able to identify specific tokens in low perplexity texts because it is bound by the language syntax and thus has limited availability to choose a different one.

2.2.3. Limitations

One weakness of watermarks is their learnability by third parties. With access to watermarked and non-watermarked texts others might try to learn the scheme and work around it.

Jovanović et al. (2024) demonstrate the vulnerability of watermarks showing that repeated API queries to watermarked LLMs allow an attacker to reverse-engineer the watermark, enabling practical spoofing attacks, and enhancing the effectiveness of scrubbing attacks. They propose an automated watermark stealing algorithm and use it to conduct a comprehensive study of both spoofing and scrubbing attacks in realistic settings.

Other works focus on identifying inherent limitations of watermarking algorithms challenging their applicability. Zhang et al. (2024) show that strong watermarking cannot be achieved under well-specified assumptions. They claim this is also the case in the private key setting when watermark insertion and detection algorithms share a secret key. As proof, they introduce a generic efficient watermark attack in which the attacker does not need to know the private key or the specific scheme used. The attack is based on two assumptions: (1) quality oracle access to evaluate text quality and (2) perturbation oracle access that can modify outputs while preserving quality, generating a list of high-quality texts that eventually cancels the watermark.

Wouters (2024) studied the trade-offs between identifiability and output quality, introducing a systematic approach of measure in terms of a multi-objective optimization problem. They identify Pareto-optimal solutions, and demonstrate they can outperform existing robust, efficient watermarks.

To summarize the limitations of current watermarking techniques, below we identify the four most prominent ones:

1. **Vulnerability to Scrubbing**: Watermarks in text are vulnerable to modifications of the watermarked text. (e.g. changing the watermarked text beyond a certain threshold can make the watermark vanish).

2. **Vulnerability to Spoofing**: When a watermark is too robust against modifications, a malicious actor can modify the text without erasing the watermark. This could result in inappropriate content, potentially incurring legal issues.

3. **Learnability**: A third party can extrapolate the watermark scheme to reproduce it in new texts. They could encode it into arbitrary texts, thereby attributing them to a specific provider.

4. **Text Quality Degradation**: Adding watermarking to AI Models or Systems can impact the quality of their output and their reliability as generative models.

5. **Easily removable from open-source models**: The techniques described as part of the watermarking methodology are not embedded in the parameters of an AI model during training. Instead, they are added to the model through modifications to the tokens' probability after training, and as such are easily removable from open-source models that release the full model and code.

2.2.4. Mitigation Strategies

Many of the works described address one or more of these limitations described, but no single methodology addresses all of them. Potential mitigation strategies are the following:

- Develop watermarking techniques that are sensitive to modifications, so that changes in token distributions are easily detected even when only part of a watermarked text has been modified;

- Easily detectable watermarks are more at risk of forgery (e.g. adding "WATERMARKED" at the start of each generated sentence is easy to detect and forge at the same time), while watermarks that are harder to detect tend to be less accurate. Finding an appropriate balance between detectability and effectiveness should provide resilience to scrubbing and spoofing attacks simultaneously.

- Some techniques exploit semantically informed word replacements to condition the watermark technique when modifying the distribution of generated tokens to avoid loss in quality.

Efforts to mitigate spoofing attacks highlight an underlying issue with marking techniques for which robustness and reliability are inherently in opposition. This arises because any human edit makes the text more human-like and therefore harder to identify as AI-generated.

To put current mitigation strategies in perspective, it is important to note that given the early stage of the field many methodologies address specific issues. Although they might be fast outdated, the underlying ideas are likely to be used for future research. Many methodologies discussed in this section utilize the same basal approach with minor modifications or improvements. Therefore, the choice among them in practical applications remains uncertain.

2.3. Structural Marking

Structural Marking techniques attempt to embed a marking into the model weights making the model's output detectable. Unlike watermarking techniques described in Section 2.2, which are not embedded into the weights, this methodology can be applied to both closed- and open-weights models without users being aware that a marking has been applied.

During the technical workshop it was suggested that Structural Marking is considered a form of watermarking, while we consider this an appropriate comment, we keep it separate for the difference in the techniques used and because it is the main approach usable for marking the output of open-weights models as remarked in the same context.

The key distinction between structural marking and watermarking is that watermarking embeds explicit signals into the text to adjust token probabilities, while structural marking embeds a signal by modifying the model's parameters. This makes structural marking applicable to open-weights models at the cost of harder to detect signals attached to the models' weights.

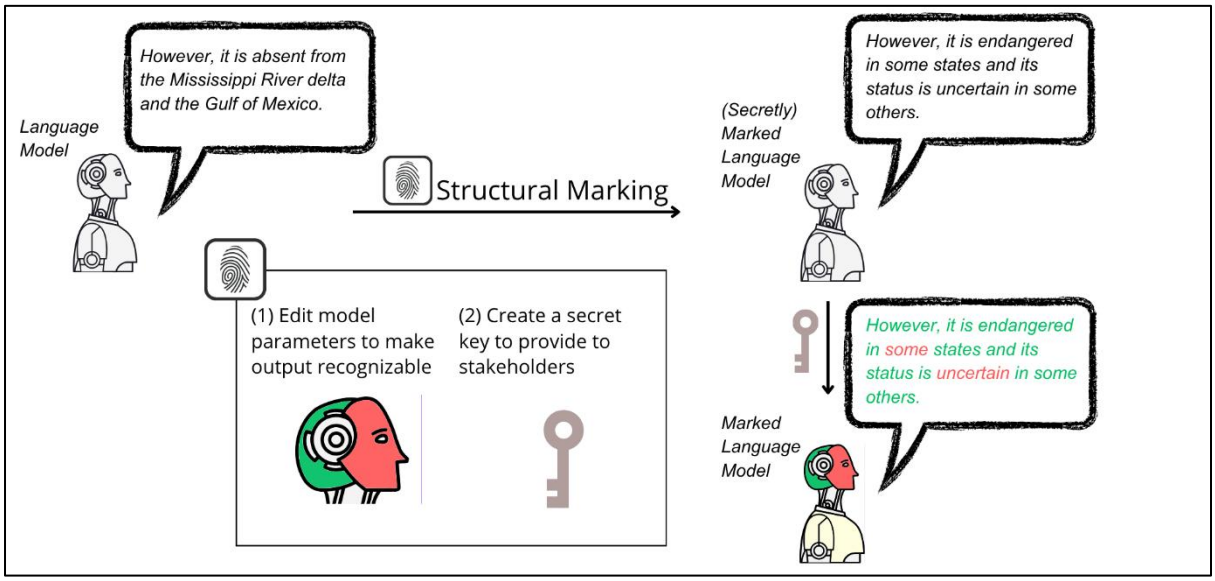
2.3.1. Methodology Description

Structural Marking is a methodology that consists of embedding a pattern or a property directly into a Language Model's weights. This process encodes a detectable signal in the text generated by the model. Unlike watermarking, where signals are added to the model generation code, structural marking produces less controllable signals. Furthermore, unlike most watermarking techniques that do not require access to the model to detect the watermark, several structural marking techniques do.

Structural marking techniques can serve two purposes. First, they allow model attribution, for example, detecting when a language model is obtained by further training an existing LLM (potentially violating the model license). Second, they allow to detect the text output by the model. The first kind of structural marking does not directly relate to the scope of this report, it is mentioned here only for completeness. The second, however, can be an effective approach to attribute the text generated through open-weights models.

The main ideas behind structural marking of a language model are shown in Figure 3. The main steps involve modifying the parameters of the model (1) and providing a key (2) to recognize these modifications. After modification to the model parameters, the outputs will have unrecognizable patterns unless one has access to the key developed in the structural marking step.

Figure 3: Visual Example of Structural Marking.[17]



2.3.2. Relevant Literature

Structural marking techniques were developed to protect the intellectual property of a model, by applying modifications to the model parameters that are resilient to further modifications, which in turn reveals if a model is a fine-tuned version of the marked one. There are several structural marking techniques. Instructional Fingerprinting, a form of structural marking (Xu et al., 2024), trains an adapter that can be shared as a secret key and forces a fine-tuned LLM to generate specific text, thereby exposing the use of the fingerprinted model. With a similar goal, HUREF (Zeng et al., 2024) treats model weights as a vector and identifies three intrinsic properties based on the transformer architecture that remain invariant to further modifications of the model

[17] This image is created by the authors specifically for this report.

parameters. ProfLingo (Jin et al., 2024) instead identifies specific prompts that elicit unusual responses from the LLM that are resilient to parameters' changes.

Structural marking can also be used to encode a signature into a model's parameters to attribute generated text. The signature can be added during pretraining, incorporating marked texts into the training data, creating "Radioactive" language models (Sander et al., 2024). The authors add marked text among the training samples, and the model learns to write according to the marking signal that was originally present in the training data. As a result, the model naturally generates marked text. They find that marking a small share of training data, as little as 5%, can render detection statistically significant. Other works improve on this approach by reducing the amount of marked training data needed to learn the structural mark requiring only 1% of the training data to be marked (Tang et al., 2023).

Another approach is adding controlled noise to a subset of model parameters. Introducing random numbers into certain parameters can make outputs detectable. GaussMark (Block et al., 2025) introduces additive corruptions to LLM parameters, resulting in models of identical or even improved quality. Adding gaussian noise to a small subset of the model parameters permits marking output in a way that is statistically detectable knowing the secret key.

2.3.3. Limitations

Because structural marking techniques are encoded in the models' parameters, they are more difficult to remove than watermarking techniques. For instance, for open-source models, modifying the generation code cannot remove structural marking. However, removal is possible through fine-tuning the model, a standard and widespread technique, but this requires high performance hardware which may be unavailable to the average user.

Empirically, structural marking techniques tend to be less effective than watermarking. Modifying the model parameters is less controllable than explicitly modifying tokens sampling during text generation, thus requiring longer text for reliable detection. Furthermore, there is no work developing structural marking techniques that explicitly preserve semantic knowledge since it is unclear how this could be enforced.

Lastly, structural marking is more costly than watermarking techniques since most techniques require model training to add the signal, which is more demanding than inserting code into the generation pipeline.

2.3.4. Mitigation Strategies

Early works have attempted to modify the parameters of language models by adding noise in a controlled manner. This approach works around the need for model training to embed the fingerprint.

Although fingerprints are generally embedded in the model's parameters, they can be stored separately. When doing so, the fingerprint can be removed, allowing access to both the original model as well as the fingerprinted one.

2.4. Metadata

Metadata are identifiers generated alongside the content to be marked. Their goal is to record information so that content creation and subsequent modifications can be traced.

2.4.1. Methodology Description

When a Generative AI system creates content, its corresponding metadata is also created and stored. To authenticate the content and its provenance one can refer to the stored metadata, which can additionally be securely verified through digital signatures.

Throughout the report we assume that metadata holds a cryptographic hash of the content that is digitally signed, such that they cannot be modified or forged without access to the private key used to create the signature, and that the integrity of the content can be verified by validating the hash.

2.4.2. Relevant Literature

There are several examples of metadata technologies that have been developed for visual and audio content, but their use in textual content remains limited. One of the few methodologies explicitly mentioning its applicability to textual content are Content Credentials, as standardized by C2PA. At the time of writing this report, however, its adoption for textual content is still limited although promising.

In "AMP Content authentication via provenance" (England et al., 2020), the authors propose AMP, a system that ensures media authentication via certifying provenance. AMP generates one or more publisher-signed manifests for each media instance uploaded by a content provider. These manifests are stored in a database to enable fast lookup from applications such as browsers. For reference, the manifests are also registered and signed by a permissioned ledger.

Figure 4 shows an illustrative example of a C2PA content credential that would be generated when a user employs Generative AI to create content (e.g. by querying ChatGPT). Through a cryptographic hash (the “hash” field in the example) one can ensure that content was not altered, while through a digital signature (the “signature” field in the example) the metadata can be authenticated without ambiguity.

This kind of credential is best suited to establish text provenance within more structured formats such as pdf, docx or html, which can present the credential embedded in the document itself. Nevertheless, the simplicity in removing these credentials through copy-paste or OCR, for example, presents a notable downside.

Figure 4: Example C2PA Content Credential.[18]

```
{
  "@context": "https://schema.c2pa.org",
  "type": "c2paManifest",
  "assertions": [
    { "type": "AI-generated", "generator": "Some Provider" },
    { "type": "author", "email": "someemail@some.where" }
  ],
  "hash": "b0f2efawe...",
  "signature": "ABCDE..."
}
```

2.4.3. Limitations

There is no straightforward way to apply Metadata-based techniques to text itself. While they might be applicable to documents in rich formats, there is no practical approach utilizing them to mark the kind of interactions typical of ChatGPT-like interfaces (i.e. a stream of relatively short texts output by a chat interface in response to several user queries).

Additionally, these credentials are easily removed from text, and they do not support any modification to text, since they are specifically meant to verify the exact version of the content they are attached to.

[18] This image is inspired by the C2PA specification: https://c2pa.org/

2.4.4. Mitigation Strategies

The development of newer, more involved strategies for encoding more information through marking techniques that are embedded in text. An approach to do this in practice is the development of sentence level content credentials embedded through zero-space Unicode characters, which are however still at an early stage of development. There are approaches to do this,[19] but while directly embedded in text they are still easily removed, through OCR or simple programmatic editing of the text string requiring limited expertise.

2.5. Logging

Logging techniques assume that the provider saves a private copy of all the text generated by the AI system and uses this stored knowledge to later identify generated texts. Logging techniques are being recommended in works focused on the governance of Generative AI (Knott et al., 2023).

2.5.1. Methodology Description

As shown in Figure 5, to perform logging a provider stores all the text generated by its Generative AI system, across all users. The generations are used to run retrieval searches based on similarity scores. To determine if a given query (text passage) was generated by the same interface a search is run in the database. If there are texts present in the database with a similarity score above a fixed threshold, the text is marked as AI-generated. Otherwise, it is marked as human-written.

The threshold used to mark texts as AI-generated or not is chosen by the provider based on experimental results, ensuring that the number of human-written texts incorrectly marked as AI-generated is below a certain share.

2.5.2. Relevant Literature

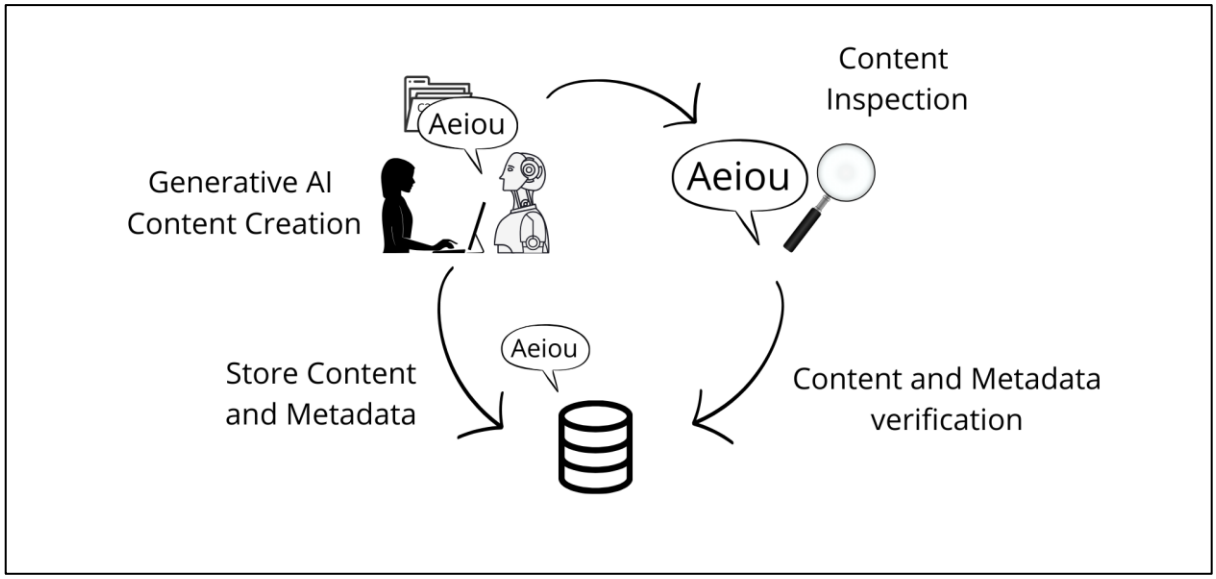
Techniques in this family have been explored, although less extensively than watermarking and metadata-based methods. They can be used to search for a given piece of content among those generated by a specific provider, (Krishna et al., 2023).

Other studies investigate the possibility of using multi-bit watermarks and logging together to encode user information enabling the attribution of text not only to a

[19] C2PA manifest embedded in unstructured text

specific model but also to a specific user via a given API. To explore this, Li et al. (2024) propose a novel multibit watermark, enabling the attribution of AI-generated content to its source. To enhance traceability, especially for short outputs, they introduce a “depth mark” that strengthens the link between content and the user who requested it.

Figure 5: Visual Example of Logging functioning.[20]



2.5.3. Limitations

This methodology faces significant interoperability challenges. To log all the outputs of Generative AI systems, providers must maintain large-scale databases containing large numbers of saved texts and associated metadata (i.e. time of generation, user ID, etc), making this approach interoperable with other techniques requires enabling querying databases once they have been anonymized. This can be accomplished in two ways each with its own limitations:

1. The provider develops an interface for third parties to query the database. This, however, can raise data-related issues, such as data storage location.
2. The provider shares the full database, which is particularly challenging for providers with high query volumes.

[20] This image is created by the authors specifically for this report.

Furthermore, storing all generated outputs would make it impossible for users to exercise their right that conversations are not stored, as required by GDPR, specifically, the right to be forgotten.[21]

2.5.4. Mitigation Strategies

While interoperability limitations can be addressed by collaborating with providers, the issues related to GDPR compliance, and more broadly to user-privacy, are more challenging to mitigate.

2.6. AI-generated Text Detection

Most of the methodologies described so far explicitly embed a signal within AI-generated text. However, Generative AI models and systems leave intrinsic identifiable traces in their outputs. Consequently, there are methodologies that aim to detect these traces in AI-generated text. In this report we will refer to them as AI-generated Text (AIGT) detection techniques, although they are also known as techniques for Machine Generated Text (MGT) detection or Forensic techniques.

2.6.1. Methodology Description

There are two prominent approaches to AI-generated text detection: supervised detectors and unsupervised likelihood-based detectors. The first approach consists in training supervised detectors to recognize the signal left by Generative AI. Detectors are trained on large-scale datasets containing texts labelled as human-written or AI-generated.

The second approach leverages the likelihood assigned by the same LLM to a text passage to determine if it is AI-generated. To generate text, a language model computes the likelihood of words one after the other and the most likely ones are chosen, therefore, the average likelihood of a text generated by an LLM tends to be higher than the likelihood of human-written texts. This property can be used to detect AI-generated text by using existing LLMs to measure text likelihood.

2.6.2. Relevant Literature

Supervised methodologies for detecting AI-generated text have been developed based on pre-trained, encoder-only transformers (Abassy et al., 2024; Verma et al.,

[21] https://gdpr.eu/right-to-be-forgotten/

2024). These methods achieve strong performance when tested in controlled scenarios. However, they are known to have limitations, in particular vulnerabilities to paraphrasing.

Unsupervised methodologies for the detection of AI-generated text, by contrast, compute token likelihood according to other language models (Mitchell et al., 2023). However, these approaches perform best when the likelihood is computed using a model similar to the one used for generation (e.g. from the same model family) which is usually unknown in advance.

Among all methodologies, AI-generated Text Detection has seen the widest commercial development with private companies, including GPTZero[22] and Pangram.[23] These services, however, share the same limitations observed in research-based methods, in particular weakness to paraphrasing and domain dependence.

2.6.3. Limitations

The main limitations of this methodology are its effectiveness and its robustness. Effectiveness is a general measure of how well the technique can detect AI-generated texts, while techniques appear effective when tested on curated datasets this is not the case when tested in real world applications (Doughman et al., 2025; Puccetti et al., 2024).

Even in those cases where these methodologies function correctly, their robustness to paraphrasing and modifications is limited (Hu et al., 2023). Additionally, open-source language models can be trained specifically to evade detection systems (Pedrotti et al., 2025).

2.6.4. Mitigation Strategies

To address some of the limitations of AI-generated text detection techniques, one solution is to increasingly collect large and diverse datasets of human-written and AI-generated texts spanning more domains and writing styles. These datasets should be continuously updated to identify newer models and writing patterns. This is a challenging and demanding mitigation strategy requiring a continuous effort from providers, potentially resulting in higher continual costs than other techniques.

[22] https://gptzero.me/

[23] https://www.pangram.com/

One of the key limitations of supervised techniques is that they tend to be removed by paraphrasing, there have been proposals on how to mitigate this shortcoming (Hu et al., 2023), nevertheless this remains an issue.

One of the advantages of this methodology is that it can be integrated with other approaches. For example, detectors can be trained on watermarked AI-generated text, and it can be used as additional information when establishing whether a text is AI-generated or not.

2.7. Fingerprinting

In this section we briefly cover additional techniques not discussed in this report due to limited applicability for textual content.

Fingerprints: This methodology is widely studied for audio and image marking. Its core principle is based on extracting characteristic traits of AI-generated content such as high noise regions and structural properties of an image and storing them into a searchable database. To identify if unknown content has been generated by the same AI system, the same algorithm used to extract its signature is applied and is searched in the database. The content was generated by the same system if its signature is among those stored.

These techniques are not studied for text because it is challenging to extract identifiable features from text passages. However, their functioning can be approximated by logging, which stores the full text instead of a fingerprint. This is feasible because text requires less memory to be stored than audio or video content, particularly for the average user interaction. A key limitation of logging compared to fingerprinting is that storing the full text incurs the privacy issues described in Section 2.6.3, which are absent when storing fingerprints.

2.8. State of the Art Summary

To conclude and summarize the content of this section, Table 3 reports some of the most established techniques within each Methodology.

Table 3: Summary Table of AI-generated Text Attribution Techniques. The techniques in this table are a subset of those included in Section 2. The columns report: year of publication (Year), name of the technique (Short Title), URL of the paper (Publication) and if available a link to the GitHub page where the technique is implemented (Implementation). Different methodologies are color-coded: with a red background we highlight Watermarking, with a blue one Structural Marking, with a yellow one Metadata, with a purple one Logging and with a dark orange one AI-generated text detection.

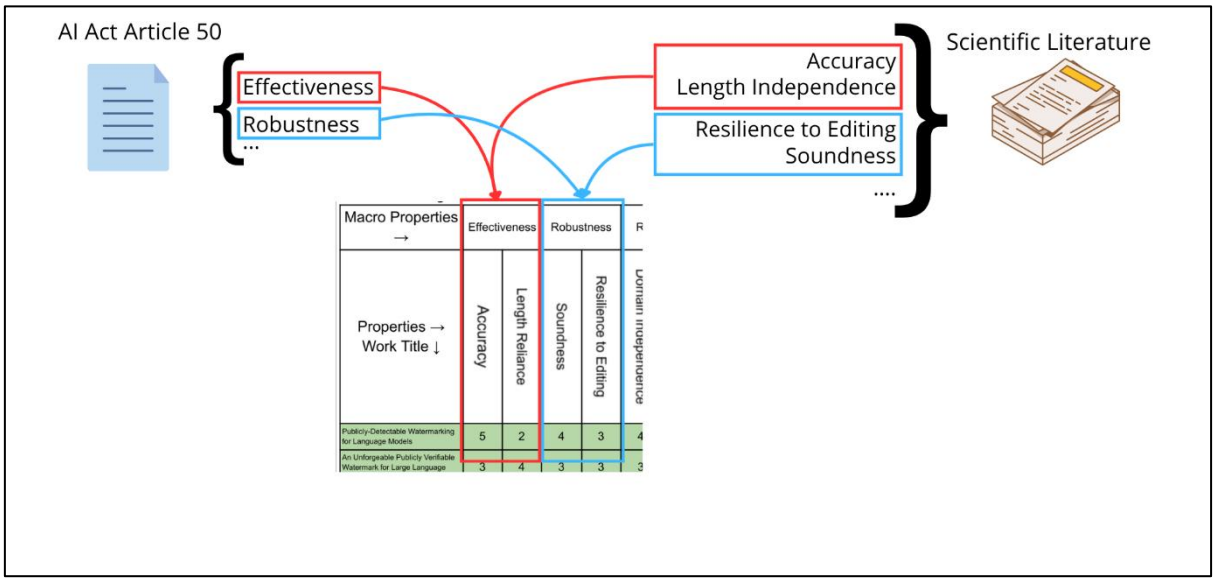
	Year	Short Title	Publication	Implementation
Watermarking	2023	KGW	proceedings.mlr.press/v202/kirchenbauer23a.html	MarkLLM
	2024	SynthID	nature.com/articles/s41586-024-08025-4	MarkLLM
	2023	Unigram	openreview.net/forum?id=SsmT8aO45L	MarkLLM
	2024	Permute-and-Flip	openreview.net/forum?id=YyVVicZ32M	MarkLLM
	2024	AWTI	proceedings.mlr.press/v235/liu24e.html	MarkLLM
	2024	DiPMark	proceedings.mlr.press/v235/wu24h.html	MarkLLM
	2024	SemStamp	aclanthology.org/2024.naacl-long.226	MarkLLM
	2023	SIR	openreview.net/forum?id=6p8lpe4MNf	MarkLLM
	2024	XSIR	aclanthology.org/2024.acl-long.226/	MarkLLM
	2024	EWD	aclanthology.org/2024.acl-long.630/	MarkLLM
	2024	DeepMark	aclanthology.org/2024.emnlp-main.681/	N/A
	2023	SWEET	aclanthology.org/2024.acl-long.268/	MarkLLM
	2025	Asymmetric	arxiv.org/abs/2310.18491	publicly-detectable
	2023	UPV	openreview.net/forum?id=gMLQwKDY3N	unforgeable_watermark
	2023	EXP	openreview.net/forum?id=FpaCL1MO2C	MarkLLM
	2024	Christ et al.	arxiv.org/abs/2402.09370	publicly-detectable
	2024	Bileve	openreview.net/forum?id=vjCFnYTg67	Bileve-official
Structural Marking	2025	GaussMark	openreview.net/forum?id=YG3DbpAQBf	N/A
	2024	IF adapter	aclanthology.org/2024.naacl-long.180/	Model-Fingerprint
	2024	ProFLingo	ProfLingo	ProFLingo
	2024	HuRef	HuRef	HuRef
Metadata	2020	C2PA	arxiv.org/abs/2001.07886	https://c2pa.org
Logging	2023	Sim	dl.acm.org/doi/10.5555/3666122.3667317	ai-retrieval
AI-generated Text Detection	2023	Radar	arxiv.org/abs/2307.03838	RADAR
	2024	DetectLLM	aclanthology.org/2023.findings-emnlp.827	DetectLLM
	2024	MAGE	aclanthology.org/2024.acl-long.3/	MAGE

We exclude a few techniques described in this section. We exclude benchmarking studies as well as works focused on the limitations of existing solutions as they do not introduce novel techniques. Two of the techniques we study don’t have an existing GitHub page, however we believe they are relevant for meaningful use cases as well as for future research directions.

3. Framework for the Assessment of AI-generated Text Marking and Detection Techniques

This section presents an assessment framework and the related assessment results for techniques used in AI-generated textual content marking and detection.

Figure 6: Assessment workflow, we map AI Act requirements to Properties used in the literature to asses marking and detection techniques.[24]



The workflow to assess text detection techniques is outlined in Figure 6. The assessment focuses on five requirements, as set out in AI Act Article 50(2) (see Excerpt 1) and Article 50(5) concerning accessibility: ***Effectiveness, Robustness, Reliability, Accessibility, and Interoperability***. Each requirement is mapped to relevant *Properties* to provide a detailed analysis. The properties are extrapolated from the scientific literature, identifying key metrics commonly used to develop and compare novel methodologies for the marking and detection of AI-generated texts.

Note that the choice of these specific properties is carried out by the authors of the report and does not pre-empt legal decisions on how the AI Act requirements will be interpreted by the European Commission. Furthermore, the assessment is the authors' own choice, it is not verified by other authors or peer reviewed and does not pre-judge the assessment done by the European Commission or the Code of Practice drafting.

In this section, we describe each requirement and its associated properties, together with the scores used to assess them.

[24] This image is created by the authors specifically for this report.

Along the assessment framework, this section contains the assessment of existing methodologies as carried out by the authors. Appendix A complements this section. Specifically, in Appendix A we provide an illustrative example for some of the techniques included in this report.

We stress that both the assessment framework and the assessment results developed in this section are to be interpreted only as recommendations. They have been developed to be as comprehensive as possible, and a fully self-contained framework for the assessment of marking techniques for text has been provided. Note that it is not definitive given the novelty of marking techniques for text. Future technical developments, as well as more involved legal interpretations of the AI Act, could require adjustments to this framework.

In summary, the framework is meant as a comprehensive guideline for the assessment of marking techniques for text that considers the requirements of AI Act Article 50(2) and 50(5). It is intended to be a starting point for the development of formal requirements for providers of Generative AI systems.

3.1. Effectiveness

The effectiveness of each technique indicates how well it detects AI-generated texts as opposed to human-written texts. AI-generated text can be used in many scenarios, such as code writing, news editing, information searches, etc. In this section we investigate the high-level performance of text detection methodologies when tested on generic texts. Measuring the techniques' adaptability to different use cases will be the focus of Section 3.2 Robustness and 3.3 Reliability.

We divide Effectiveness in two separate Properties: ***Performance*** and ***Length Independence***. These two properties are relevant for all methodologies. The first measures the general performance of each technique, while the second is a desired property by all techniques studied and both are used specifically to measure Effectiveness. The first indicates the ratio of correct responses given when detecting AI-generated text. The second gives a general assessment of how dependent the methodology is on the length of the watermarked text. Dependency on length falls within the technical limitations considered in AI Act Article 50(2), which mentions that all the requirements are to be met by providers "as far as technically feasible".

3.1.1. Performance

Through Performance we measure how effectively each technique can attribute AI-generated texts to Generative AI and human-written texts to humans. In particular, the goal is to maximize true positive, correctly labelled AI-generated

texts, while minimizing false positives, human-written text detected as AI-generated. To assess each technique, we develop a five-point scoring system, consistent with what we use with all other properties.

To measure performance, many of the research studies analysed in this report use True Positive Rate at 1% False Positive Rate (TPR @ 1% FPR) as a performance metric (Kirchenbauer, Geiping, Wen, Katz, et al., 2023; Dathathri et al., 2024; Y. Liu & Bu, 2024; Fairoze et al., 2025).

Using TPR @ 1% FPR implies that up to 1% human-text are detected as AI-generated, this is the most used metric in the literature concerning AI-generated text detection, however real-world applications can be required to have lower error margins. For high traffic providers it can be desirable to allow less than one in a million human text attributed to AI (Fernandez et al., 2023).

This metric is aimed at minimizing the number of AI-generated texts classified as human written, while having a controlled amount of human written text classified as AI-generated. In Appendix A we provide a clear explanation of this metric as well as an illustrative quantitative scoring system to measure performance based on the evaluation of existing techniques. However, in the main body of the report we only use qualitative evaluations to describe our assessment framework.

We described the assessment system for an overall measure of performance, as outlined below:

1. **Low performance**: The technique misclassifies too many texts, failing to keep the number of misclassified human-written samples sufficiently low.

2. **Moderate performance**: The technique maintains a low number of misclassified human-written texts while detecting a limited fraction of AI-generated texts.

3. **High performance**: The technique maintains a low number of misclassified human-written texts while detecting a substantial fraction of AI-generated texts.

4. **Very high performance**: The technique maintains a low number of misclassified human-written texts while detecting most AI-generated texts.

5. **Near-perfect performance**: The technique correctly identifies nearly all AI-generated texts while misclassifying only a negligible number of human-written texts.

See Appendix A.1 for an illustrative quantitative instance of this scoring system.

3.1.2. Length Independence

Most methodologies considered improve their performance when applied to longer texts. In the literature many works test their techniques on texts of varying length. And generally the shortest length tested is between 30 and 50 words, because below this threshold marking techniques performance drops significantly (Krishna et al., 2023; Kuditipudi et al., 2023; Y. Liu & Bu, 2024; Pan et al., 2024). Like Performance, also Length Independence can be measured quantitatively, in Appendix A we provide an illustrative quantitative scoring system to quantify this property based on empirical results.

We evaluate each technique's ability to retain performance when evaluating shorter text passages. The five-point assessment system is as follows:

1. **Heavily reliant on length:** Text length is a crucial factor in the technique's performance;
2. **Strongly reliant on length:** Text length is strongly involved in the technique's performance;
3. **Reliant on length:** The technique depends on text length to a non-negligible extent;
4. **Mildly reliant on length:** The technique is dependent on length, but it has been optimized to mitigate this dependence;
5. **Independent from length:** The technique is independent of text length.

See Appendix A.2 for an illustrative example of a quantitative implementation of this assessment system.

We note that, metadata behaves differently from other methodologies with respect to this property, as it does for Performance. Once a piece of text is provided with its credential and metadata, text length becomes less relevant in the validation process. Therefore, these techniques are virtually independent of text length, however, in practice assigning a content credential to a very short text passage, which could exist both as human-written and as AI-generated might not be meaningful.

3.1.3. Assessment

Based on the described assessment system, we assign a textual description to each methodology.

*Table 4: Description for each Property (column) contributing to the **Effectiveness** Requirement of each Methodology (row).*

	Performance	Length Independence
Watermarking	**Very high performance**: The technique maintains a low number of misclassified human-written texts while detecting most AI-generated texts.	**Mildly Reliant on length:** This technique is dependent on length, but it has been optimized to mitigate this dependence.
Structural Marking	**High performance**: The technique maintains a low number of misclassified human-written texts while detecting a substantial fraction of AI-generated texts.	**Reliant on length:** The technique depends on text length to a non-negligible extent.
Metadata	**Near-perfect performance**: The technique correctly identifies nearly all AI-generated texts while misclassifying only a negligible number of human-written texts.	**Independent from length:** The technique is independent of text length. (Although for very short text it might not be applicable).
Logging	**Very high performance**: The technique maintains a low number of misclassified human-written texts while detecting most AI-generated texts.	**Mildly reliant on length:** This technique is dependent on length, but it has been optimized to mitigate this dependence.
AI-generated Text Detection	**High performance**: The technique maintains a low number of misclassified human-written texts while detecting a substantial fraction of AI-generated texts.	**Reliant on length:** The technique depends on text length to a non-negligible extent.

Performance

Metadata-based approaches represent the best performing methodology since they store a specific version of text, and any modification invalidates their signature. We remark that these considerations assume that metadata has been successfully applied to the text, currently this is only possible with rich document types (e.g. HTML, PDF, etc.) but there is no established approach to embed metadata directly into the text in a way that cannot be easily removed.

As shown in Table 4 logging and watermarking are also performant. Watermarking relies on statistical measures embedded in the word distribution itself, providing statistics-based confidence measures of its outcomes. Structural marking works similarly to watermarking; however, it embeds a weaker signal in the marked text, making it less effective. Logging is developed to address some of watermarking limitations specifically when dealing with modifications and works similarly well.

AI-generated text detection methodologies are less performant. Techniques of this kind often rely on static training datasets. As a result, they are generally tailored to specific use cases and writing domains that have to be included in the training dataset. This makes their performance limited for general purpose text generative AI.

Length Independence

There are two ways in which a technique can depend on length. First, the marking strength grows continuously with the length, i.e. the longer the texts the more robust the watermark. For example, logging (Krishna et al., 2023) and all methodologies derived from KGW follow this pattern. Second, in some techniques the marking is encoded in a fixed number of words. To exploit longer texts, the same watermark is applied sequentially across several spans of text. This approach is used by most methodologies similar to the work from Aaronson et al. (2022). Both approaches benefit from longer texts, however, we consider methodologies that require a fixed number of tokens to encode the watermark to be less dependent on the text length.

As shown in Table 4, similar conclusions apply as for Performance: Metadata is the best approach with respect to text length since it can be equally applied to short and long texts. Watermarking and logging have limited issues with short text as far as texts are longer than 30/50 words, but such short texts could be interpreted as inherently hard to claim as either AI-generated or human-written as in most cases they appear multiple times in many contexts.

Structural Marking and AI-generated text detection are more sensitive to text length, because they rely on weaker signals.

3.2. Robustness

Effectiveness is essential for the deployment of text attribution techniques. However, given the large number of users and possible interactions with Generative AI interfaces, effectiveness alone is insufficient to assess the quality of a text detection technique. The second requirement we consider is **Robustness**, which refers to those techniques that have been thoroughly tested and are resilient to editing. Evaluating the robustness of different methodologies is key to determining their applicability in real-world use-cases.

3.2.1. Soundness

To highlight that the task of marking and detecting AI-generated text is novel and it has been studied less than marking images, videos and audio, we assess the soundness of the studied techniques. We define this to be the functioning guarantees provided by each technique.

Research on marking and detecting AI-generated text typically includes theoretical analysis of the results and empirical evaluations of their robustness. Many works provide statistical guarantees when detecting AI-generated texts.

Others present more involved theoretical analyses based on cryptographic principles, complementing performance measures.

In this work, greater emphasis is placed on empirical assessments rather than theoretical ones because the latter alone might not indicate practical applicability. Nevertheless, we interpret theoretical guarantees, once empirically validated, as an additional indication of robustness since detection techniques cannot be empirically tested in all available scenarios due to the wide range of Generative AI applications.

Another reason for adding theoretical guarantees as a measure of robustness is the inherently stochastic nature of Generative AI. Text generation is often non-deterministic, and so are several attribution techniques. Consequently, attribution methodologies have an inherent error margin that must be accounted for. Having theoretical guarantees in addition to empirical results provides a stronger basis for assessing the soundness of the methodologies.

This assessment system is meant to address the general novelty of marking techniques for text. Such novelty makes many of the current conclusions based on research results which, while promising, might turn out to be less effective than they appear when only tested in controlled, scientifically sound, scenarios.

As for other properties, we devise a five-point assessment system with the following descriptions:

1. **Theoretically Appropriate:** Techniques based solely on theoretical assumptions;
2. **Theoretically Sound:** Techniques supported by theoretical guarantees for specific performance requirements;
3. **Empirically Sound:** Techniques that have been empirically tested and have measurable performance values;
4. **Empirically and Theoretically Sound:** Techniques that have been empirically tested showing results coherent with theoretical results;
5. **Extensively Tested:** Techniques that have undergone extensive empirical validation in real world scenarios through. for example, exposure to large numbers of users.

This assessment system is intended to provide assurances about the robustness of methodologies when used in previously untested contexts or in novel applications of Generative AI. It prioritizes empirical results while viewing theoretical guarantees as beneficial and finally regards extensive testing as the most reliable proof of robustness.

While this requirement is qualitative, we believe it is relevant for AI-generated text marking and detection, unlike other modalities. Most techniques for AI-generated text marking and detection have been developed in recent years, and it is expected that many techniques will see limited applicability, while others widespread adoption. Therefore, we also account for this property when measuring the robustness of methodologies.

To illustrate why studies with a predominately theoretical justification to their results without strong empirical evaluation receive lower scores, we highlight the case of Zhang et al., (2024). They theoretically proved the impossibility of AI-generated text marking reliably. Under what appeared as reasonable hypotheses, they demonstrated that any marking can be removed from a text while maintaining its meaning and readability. However, this result was later questioned. Harel-Canada et al., (2025), through a systematic empirical evaluation, found that in practice marking techniques can resist the attack developed by Zhang et al., (2024).

We highlight that this score assesses the soundness of a technique when applied to AI-generated text. The methodologies we consider have been tested on other media (e.g. images), applications which are beyond the scope of this report.

3.2.2. Resilience to Editing

Adding small modifications to a text, e.g. changing few words, does not necessarily change its AI-generated nature. Consequently, it is desirable for marking and detection techniques to be robust to limited editing, such as changing few words, especially when these edits are intended to change the meaning of the text (T. Zhou et al., 2024), potentially leading to the misattribution of malicious outputs.

However, achieving such robustness is technically difficult and, to some extent, incompatible with high reliability. As discussed in Section 2, incompatibility arises because human editing makes the text more human-like and thus harder to identify as AI-generated. To address this trade-off, there are recent attempts at using two-step verification systems (Zhou et al., 2024).

In the scope of this report, by editing we indicate any modification of the text, so that any major editing should result in a high percentage of modifications (e.g. paraphrasing done through Generative AI).

Resilience to editing is tested by many studies developing novel techniques since it is a natural attack (Hou et al., 2024; Krishna et al., 2023; Kuditipudi et al., 2023). Like Performance and Length Independence, this property can be measured quantitatively. In Appendix A we propose an illustrative quantitative scoring system to assess techniques based on experimental results.

To evaluate robustness against editing we propose an assessment system that classifies techniques according to the number of modifications they can withstand when marking a text of moderate length, e.g. around two hundred words. Indeed, most techniques (excluding Metadata) are more resilient to editing when applied to longer AI-generated texts, e.g. one thousand words or longer. However, moderate length is enough to convey several messages, and we believe that AI-generated texts with this length should support marking.

Taking into account these considerations, we assess the **Resilience to Editing** Property, using a five-point scale:

1. **Non-Resilient:** any modification to the marked text is sufficient to lower performance;
2. **Mildly Resilient:** lowering performance requires modifying only few of the words in a marked text;
3. **Resilient:** lowering performance requires modifying a moderate amount of the words in a marked text;
4. **Highly Resilient:** lowering performance requires modifying a significant amount but less than half of the words in a marked text;
5. **Fully Resilient:** lowering performance requires modifying half of the words in a marked text.

See Appendix A.3 for an illustrative quantitative implementation of this assessment system.

We consider modifying half of the marked text as an arbitrary worst-case threshold. We choose it because it is often the largest number of modifications used to test marking techniques in the literature, (Block et al., 2025; Kirchenbauer, Geiping, Wen, Katz, et al., 2023; Zhao et al., 2023, 2024).

We stress that a threshold of this kind should always be considered as part of any assessment system, to implicitly identify when an AI-generated text has been sufficiently modified to consider the AI role as assistive, and thus not subject to disclosure as AI-generated. Additionally, we remark how assessing the assistive use of AI should account for text content and semantics and not only for its AI-generated nature. For example, asking an AI system to improve the writing style of an essay is different from asking it to write the essay from scratch. To the best of our knowledge there are no methodologies that directly investigate this. We nevertheless believe it should be considered when evaluating the assistive use of Generative AI.

While the focus of this section lies on the interplay between humans and Generative AI, multiple Generative AI interfaces can be used to modify the output of one another. As a result, the watermark embedded by one provider

can be lost when the text is paraphrased, e.g. by using a second Generative AI system as tested by Krishna et al. (2023). The impact of AI-based attacks, such as paraphrasing or back-translation (translating to a different language and back to the original), is studied in the literature from a technical perspective to quantify the robustness of each technique to modifications (Tu et al., 2024).

3.2.3. Assessment

Based on the described assessment system, we assign a textual description to each methodology.

*Table 5: Description for each Property (column) contributing to the **Robustness** Requirement of each Methodology (row).*

	Soundness	Resilience to Editing
Watermarking	**Empirically and Theoretically Sound:** Techniques that have been empirically tested showing results coherent with theoretical results;	**Resilient:** lowering performance requires modifying a moderate amount of the words in a marked text.
Structural Marking	**Empirically Sound:** Techniques that have been empirically tested and have measurable performance values;	**Resilient:** lowering performance requires modifying a moderate amount of the words in a marked text.
Metadata	**Empirically and Theoretically Sound:** Techniques that have been empirically tested showing results coherent with theoretical results;	**Non Resilient:** any modification to the marked text is sufficient to lower performance.
Logging	**Empirically Sound:** Techniques that have been empirically tested and have measurable performance values;	**Highly Resilient:** lowering performance requires modifying a significant amount but less than half of the words in a marked text.
AI-generated Text Detection	**Empirically Sound:** Techniques that have been empirically tested and have measurable performance values;	**Resilient:** lowering performance requires modifying a moderate amount of the words in a marked text.

Soundness

As shown in Table 5, the two most studied techniques are Watermarking and AI-generated text detection. The first, watermarking, shows promising results in general and there are examples that are provided to end users as a service, SynthID (Dathathri et al., 2024). The second, AI-generated text detection, has been widely studied but significant limitations have been pointed out. Finally, Metadata-based studies for text are very limited in scope, but recent

developments show that content credentials, e.g. C2PA, are a promising way forward, with the caveat that they are generally easy to remove.

Structural marking based on editing model weights is less studied and training-based techniques can mostly rely on empirical evaluation, different from the theoretical guarantees provided by watermarking solutions. Similar conclusions apply to Logging which is evaluated empirically by measuring performance on large scale datasets.

Resilience to Editing

As reported in Table 5, Metadata is assessed as the least resilient to editing, due to its underlying principles. Digitally signed metadata[25] are designed to identify only the exact version of the content that they are attached to and are in general easy to remove. Additionally, Metadata are currently mostly attached to richer formats and copy-pasting can remove this marking technique without editing the text. Sentence level metadata[26] also embedded in text are a possible way forward although it is in an early stage of development and currently relatively easy to remove, e.g. through OCR or editing Unicode characters.

Logging is the most resilient methodology to editing, marginally better than watermarking and it has been developed specifically to address watermarking fragility to modifications (Krishna et al., 2023).

AI-generated text detection techniques do not rely on embedding a signal but on identifying the traces left by a text generative model. Therefore, they are less resilient to editing and in particular, the effect of editing parts of the text is less predictable. Estimating their resilience is more challenging, as it cannot be compared to how much the embedded signal was compromised. For example, with red/green watermarking one can estimate the proportion of red and green tokens before and after a modification is applied, while for AI-generated text detection this is harder to accomplish.

3.3. Reliability

By **Reliability,** we indicate that the requirement for AI-generated text marking and detection techniques work consistently across varying scenarios, such as, when generating texts for different purposes and domains. This requirement also covers the learnability of the marking properties in measuring a technique's reliability.

[25] https://dev.to/powehi/how-c2pa-could-certify-ai-generated-texts-and-why-education-needs-it-35cj

[26] https://github.com/encypherai/encypher-ai?tab=readme-ov-file

The first property, *Domain Independence*, measures how well a text attribution technique works across varying language applications and writing styles. Domain independence is relevant, because Article 50 of the AI Act also applies to general-purpose AI systems, requiring that text can be correctly attributed regardless of the topic (e.g. coding, biology, news, etc.) or the writing style (e.g. journalistic, scientific, novelist, etc.). Domain independence is closely related with compliance since the requirements in AI Act Article 50 are agnostic to the content of Generative AI outputs.

The second property, *Learnability,* measures how difficult it is for a third party to learn the hidden signal from marked texts. If the signal can be learned and reproduced by third parties, it is unreliable.

3.3.1. Domain Independence

Text attribution techniques can be tested in various scenarios; the general-purpose nature of Generative AI creates a range of use cases. As a result, identifying a comprehensive testing strategy is challenging. It is challenging for providers to measure effectiveness on all possible use-cases; therefore, it is desirable for text attribution methodologies to be effective regardless of the domain they are employed in.

To assess domain independence, we utilize the following assessment system:

1. **Domain-Focused:** The technique is tailored to a specific domain or style (e.g. coding);
2. **Domain-Reliant:** The technique depends on assumptions that are typical of given domains or writing styles;
3. **Domain-Dependent:** The technique is not meant for a specific domain, but due to development constraints (e.g. training data) it works better in some domains;
4. **Domain-Independent:** The technique works regardless of the domain or style where it is applied;
5. **Domain-Agnostic:** The technique works across domains and styles and there are guarantees of its working independently of the domain;

We consider general-purpose approaches (i.e. domain agnostic techniques) more desirable than domain-focused ones because they are more broadly applicable. Each property in our framework is defined independently. Therefore, we consider a general-purpose solution better than a domain specific one, given comparable assessment of other properties.

Coding assistants are a prominent application of Generative AI. Programming languages have strict syntactic constraints (i.e. the syntax of a programming language has less alternatives than natural language) and there are works specifically focused on watermarking code (Lee et al., 2024). While these works might use techniques tailored for code generation, they can be assessed through the general framework we propose.

Techniques for code writing are primarily evaluated in code-oriented scenarios. Therefore, the general-purpose framework we suggest can also be used when including such techniques to the analysis. On the other hand, they would receive lower scores when tested in general purpose scenarios.

3.3.2. Learnability

Several text attribution techniques considered in this report rely on shared knowledge (e.g. a decryption key) by providers and parties performing the detection, to make the output of Generative AI automatically detectable. There is an underlying assumption to this approach, that is, the signal encoded in the text by the provider should not be replicable by third parties. Otherwise, malicious actors could release texts that would be attributed to a specific AI Interface while having a different source.

Although it is the provider's responsibility to ensure that the secret data used by the attribution approach is not shared, the hidden signal could be learnable by third parties only through access to marked texts. There are methods to learn a watermark and possibly replicate it on new texts (Gu et al., 2023). Third parties would be able to generate inappropriate texts that can be wrongly attributed to a given provider. Additionally, if the marking was used to encode specific user-information in the text, this would additionally expose providers to privacy concerns (Li et al., 2024).

Based on these considerations, we apply our five-point assessment system for Learnability:

1. **Evident:** The technique can be learned by humans with few available samples;
2. **Easily Learnable:** The technique can be learned through standard machine learning or other techniques with few samples available;
3. **Learnable:** The technique can be learned through deep learning or other techniques training on a large-scale dataset composed of marked and unmarked texts;
4. **Hidden:** The technique can be partially learned through deep learning or other techniques, but the process requires considerable expertise;

5. **Unlearnable:** The techniques cannot be learned without access to undisclosed information.

3.3.3. Assessment

Based on the described assessment system, we assign a textual description to each methodology.

*Table 6: Description for each Property (column) contributing to the **Reliability** Requirement of each Methodology (row).*

	Domain Independence	Learnability
Watermarking	**Domain-Dependent:** The technique is not meant for a specific domain, but due to development constraints (e.g. training data) it works better in some domains;	**Hidden:** The technique can be partially learned through deep learning or other techniques, but the process requires considerable expertise;
Structural Marking	**Domain-Dependent:** The technique is not meant for a specific domain, but due to development constraints (e.g. training data) it works better in some domains;	**Learnable:** The technique can be learned through deep learning or other techniques training on a large-scale dataset composed of marked and unmarked texts;
Metadata	**Domain-Agnostic:** The technique works across domains and styles and there are guarantees of its working independently of the domain;	**Unlearnable:** The techniques cannot be learned without access to undisclosed information;
Logging	**Domain-Independent:** The technique works regardless of the domain or style where it is applied;	**Hidden:** The technique can be partially learned through deep learning or other techniques, but the process requires considerable expertise
AI-generated Text Detection	**Domain-Dependent:** The technique is not meant for a specific domain, but due to development constraints (e.g. training data) it works better in some domains;	**Learnable:** The technique can be learned through deep learning or other techniques training on a large-scale dataset composed of marked and unmarked texts;

Domain Independence

Table 6 shows that Metadata and Logging are the best performing methodologies since their dependence on the domain is limited. Metadata is truly independent of any variation due to the writing domain or style because cryptographic hashing can authenticate any content.

Watermarking and Structural Marking are more reliant on domain, since they rely on text likelihood. These methodologies are less effective on specific domains, such as coding, where syntax is stricter and language more constrained. In those contexts where there is lower availability for synonyms and rephrasing, adding a watermark is more challenging. Watermarking text relies on the possibility to add small modifications to the generative process without affecting the quality of the generated text. If the syntax and vocabulary are bounded, it is more challenging to add an undetectable signal.

Logging relies on text similarity, so its functioning depends on the variability of the domain it is applied to. Like watermarking, there are domains where similarity might be less effective to separate AI-generated from human-written texts. However, this methodology is based on searching existing texts and not on altering the generative process and it is easier to optimize to work across domains.

Finally, AI-generated text detection is domain dependent since it strongly relies on the detector training set. For example, if a detector is trained only to detect AI-generated news it will fail to detect AI-generated poems or scientific writing. This can be mitigated by extending the training dataset, but it is a limitation when detecting the outputs of general-purpose generative AI which can be used in any domain.

Learnability

Most methodologies are hard to learn for third parties since this is a goal for all marking techniques and those that affect the text distribution too strongly attempt to mitigate this.

Learnability assesses whether a marking can be learned from third parties who do not have access to any private information (e.g. a private key) but only to marked and unmarked texts. To rely on a detection technique an essential requirement is that it cannot be forged. Third parties should not be able to replicate the marking and falsely attribute text to a provider.

Table 6 shows that metadata[27] and logging are the hardest to learn, meaning they are difficult for third parties to replicate. This is the case because both techniques rely on data stored by providers and use deterministic algorithms. Specifically, to replicate the signature of a given credential the private key used by the provider would have to be leaked. This can happen, but it is not a technical issue. It can be addressed by deprecating leaked signatures. Logging relies on a stored dataset which is updated over time and can only be accessed internally by the provider, making it hard to replicate.

[27] We always assume that metadata are digitally signed.

Structural marking and watermarking techniques are also hard to learn but less so than metadata and logging. Indeed, some of the techniques listed under watermarking and structural marking can be learned by others simply by having access to large amounts of watermarked and non-watermarked texts. A way to do this is to train an open-weights language model to generate text similar to the model offered by a provider. This model will spontaneously learn to generate watermarked text (Gu et al., 2023).

Finally, the least reliable technique is AI-generated text detection, because adversarial language models are straightforward to train (Pedrotti et al., 2025). If a detector is publicly available or it can be queried multiple times, malicious actors can relatively easily create training dataset and cheaply fine-tune openly available language models to generate or paraphrase AI-generated text and consistently delude the detector.

3.4. Accessibility

"The information referred to in paragraphs 1 to 4 shall be provided to the natural persons concerned in a clear and distinguishable manner at the latest at the time of the first interaction or exposure. The information shall conform to the applicable accessibility requirements."

Excerpt 2: AI Act Article 50(5)

Excerpt 2 reports AI Act Article 50(5) stating that the disclosure of AI-generated content should conform to applicable accessibility requirements. We consider accessibility from the user's perspective, as this represents the main intended meaning within AI Act Article 50.

This definition is used within this report and does not necessarily reflect the full extent of its official meaning in the AI Act. **The content of this Section is the authors' own choice, it is not verified by other authors or peer reviewed and does not pre-judge the assessment done by the European Commission.**

The **Accessibility** requirement evaluates the extent to which text attribution techniques are understandable and accessible to the final user. Although technically complex, the key ideas behind text attribution technologies can often be effectively illustrated so that general users can be instructed on their use, how they work and the kind of guarantees they provide. More specialized stakeholders, such as auditors, can be supplied with in depth details regarding the statistical significance of the outputs of text attribution solutions, providing clear results together with confidence measures.

Since text attribution technologies are yet to see widespread adoption, we focus on the possibility of developing an accessible interface that reports the

outcomes of detection techniques rather than on assessing the accessibility of the techniques themselves. We develop the assessment system for this property around the European Accessibility Act[28] (EAA). EAA compliance is complex as it is dedicated to inclusivity and aims to address everyone's needs. In the scope of this report, we focus on Section 1 of Annex I of the EAA which states explicit requirements for products and their user interfaces.

As for all properties, we use a five-point assessment system:

1. **Accessible Outcome:** The technique outcome cannot be exposed in an EAA-compliant way;
2. **Accessible Confidence:** The technique outcome requires significant infrastructure to be exposed in an EAA-compliant way (e.g. it can only be exposed through rich visual components difficult to exhibit through non-visual outputs);
3. **Accessible Description:** The technique outcome, along with confidence scores with a description can be exposed in an EAA-compliant way;
4. **Accessible Functioning:** The technique outcome, along with confidence scores and a detailed explanation of how it works can be exposed in an EAA-compliant way;
5. **Accessible Technical Details:** Every aspect of the outcome and the technique used, including technical details, can be exposed in an EAA-compliant way;

We call more accessible those techniques that can also expose their functioning in an EAA-complaint way, since, while most users are likely not interested in it, the right to access should not be prevented to any interested party. Additionally, the authors believe that the functioning of all the techniques covered in this report can be described in an accessible way,

From a user perspective, most text attribution techniques share similar needs in terms of interfaces. Essential requirements include a text input to feed a passage into the system, and a text output to inform the user of the system response, which at a minimum only indicates AI-generated or Human Written. Beyond these requirements, an interface could provide confidence scores regarding its response and details on how the decision was reached.

Figure 7: Example of Text Attribution Interface.[29]

[28] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019L0882

[29] This image Is taken from the Huggingface space: https://huggingface.co/spaces/tomg-group-umd/lm-watermarking

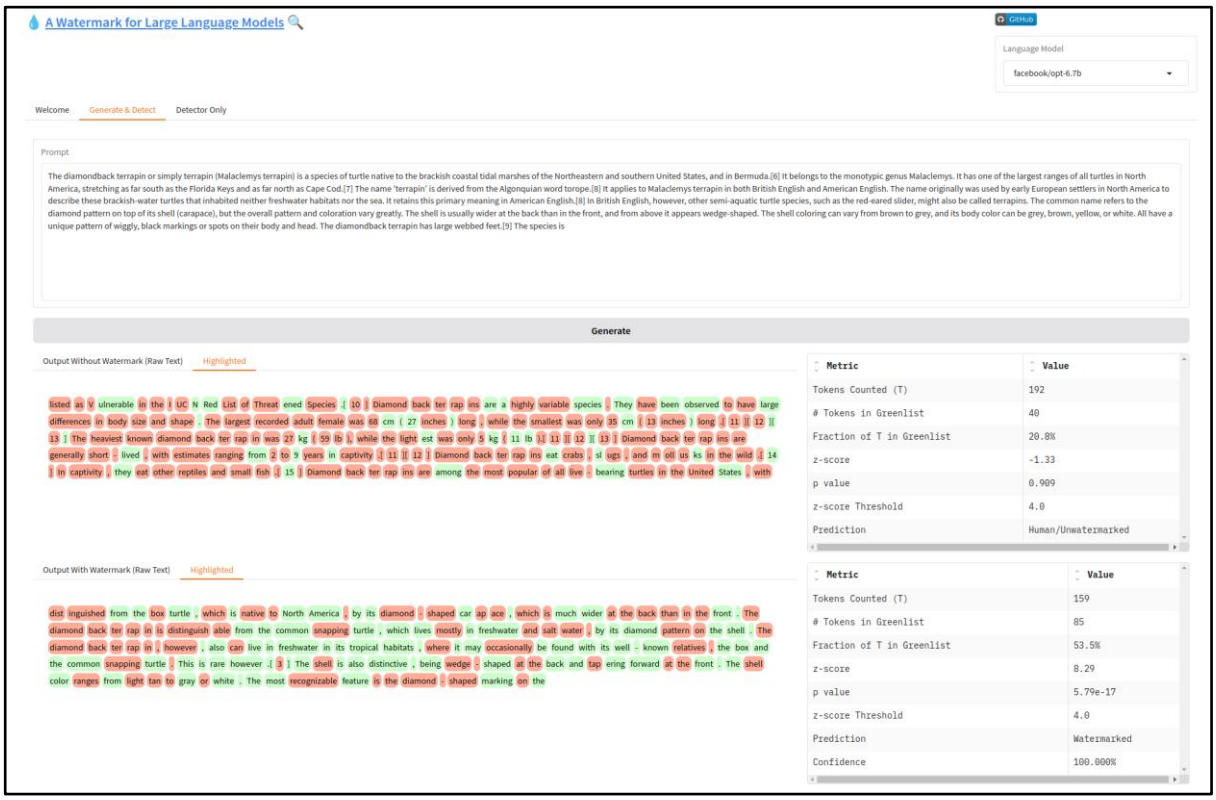


Figure 7 shows an example of the KGW watermark (Kirchenbauer, Geiping, Wen, Katz, et al., 2023) and similar ones can be developed for most methodologies for AI-generated text marking and detection. Concerning Logging and Metadata, while an explanation of the recovery process could be limited, it would indicate a database look-up. The output may not be limited to an attribution decision, but it could provide further access to the information stored in the Metadata such as generation time, user ID, etc.

As per EAA Annex I, an interface providing text input/output functionality should provide an additional sensory channel. In addition to typing the input and reading the output, a user could dictate the text and receive a response as an audio recording stating “It is likely the input text is AI-generated. The detection system has identified a 95% likelihood that it was created through Generative AI.”

This possibility alone does not provide compliance with EAA, but it highlights how requiring (for a minimal interface) only a text input/output field renders EAA compliance feasible for AI-generated marking and detection techniques.

3.4.1. Assessment

Based on the described assessment system, we assign a textual description to each methodology.

As reported in Table 7, we argue that metadata-based methods, logging and AI-generated text detection can be shown with an accessible description of their outcome and functioning. Watermarking and structural marking require more complex techniques, and, we believe that it is already possible to expose their outcomes in an accessible way, while making their functioning accessible requires more effort.

This opinion is based on the availability of existing interfaces for other applications of the algorithms underlying each methodology. Specifically, logging is based on text retrieval which is widely adopted (e.g. in web search), Metadata uses cryptographic signatures already used in several applications, and AI-generated text detection has been subjected to more studies than watermarking including the development of user-facing interfaces (Gehrmann et al., 2019), therefore they are all scored higher.

*Table 7: Description for the **Accessibility** Requirement of each Methodology (row).*

	Accessibility
Watermarking	**Accessible Description:** The technique outcome, along with confidence scores with a description can be exposed in an EAA-compliant way;
Structural Marking	**Accessible Description:** The technique outcome, along with confidence scores with a description can be exposed in an EAA-compliant way;
Metadata	**Accessible Functioning:** The technique outcome, along with confidence scores and a detailed explanation of how it works can be exposed in an EAA-compliant way;
Logging	**Accessible Functioning:** The technique outcome, along with confidence scores and a detailed explanation of how it works can be exposed in an EAA-compliant way;
AI-generated Text Detection	**Accessible Functioning:** The technique outcome, along with confidence scores and a detailed explanation of how it works can be exposed in an EAA-compliant way;

We also see that no technology is assessed as having Accessible technical details. This is the case because all the techniques studied are recent or recently applied to AI-generated text. However, it is not a limitation inherent to the techniques but concerns current interfaces. In our opinion most techniques can have every aspect exposed in a fully EAA-compliant way provided that they are implemented with expertise in accessibility and most providers could achieve this with appropriate effort.

3.5. Interoperability

While there are several available methodologies, it would be desirable to have a unified way to detect if a text passage is AI-generated. In this report, the meaning of Interoperability as mentioned in AI Act Article 50(2) is understood as follows:

- Markings from certain providers can be recognised by detection solutions from other providers;
- Detection solutions from certain providers can detect markings from other providers.

The goal pursued by this requirement is a situation where marking and detection solutions are provider- and technology-agnostic. **This is the authors' understanding of the Interoperability requirement and does not pre-judge the Commission assessment or the assessment done in the Code of Practice drafting.**

Currently interoperability among marking and detection techniques for text has been subjected to limited study. Therefore, our analysis is limited to the issues identified by the authors of this report, and we focus on the interoperability among providers.

We identify two principal factors limiting the interoperability among techniques. The first, which we call inherent limitation, is based solely on the technique functioning, for example using both watermarking and structural marking on the same text can be challenging because both influence token sampling and they interfere with each other. The second, which we call sharing-based, depends on a need for data that providers would have to share to achieve interoperability.

Based on these limitations, we devise an assessment system:

1. **Absent Interoperability:** The detection technique is hard to use for others even if there is full availability of information from providers;
2. **Limited interoperability:** The detection technique can be used by other parties, but its outcome remains uncertain;
3. **Agreement based interoperability:** The detection technique can be used by other parties, but it requires knowledge of the detection algorithm along with private information shared by the provider;
4. **Mutual interoperability:** The detection technique can be used by other parties, but it requires knowledge of the algorithm used by each technique without private information shared by the provider;

5. **Full interoperability:** The detection technique can be used by other parties only by agreeing on a standard.

Achieving the goal of provider- and technology-agnostic detection solutions requires collaboration between regulating authorities and providers. In Section 5 we describe a Scenario where an authority could establish an infrastructure facilitating the use of marking and detection techniques across several providers.

3.5.1. Assessment

Based on the described assessment system, we assign a textual description to each methodology.

Table 8 shows our assessment of each methodology. Watermarking and structural marking behave similarly with respect to interoperability. Specifically, identifying different watermarks (or structural marks) requires knowledge of the respective detection algorithms and secret keys.

*Table 8: Description for the **Interoperability** Requirement of each Methodology (row).*

	Interoperability
Watermarking	**Agreement based interoperability:** The technique is interoperable with others, but detection requires knowledge of the detection algorithm of each technique along with private information shared by the provider;
Structural Marking	**Limited Interoperability:** While technically feasible the interoperability does not provide significant gains;
Metadata	**Mutual Interoperability:** The technique is interoperable with others, but detection requires knowledge of the algorithm used by each technique without private information shared by the provider;
Logging	**Limited Interoperability:** While technically feasible the interoperability does not provide significant gains;
AI-generated Text Detection	**Mutual Interoperability:** The technique is interoperable with others, but detection requires knowledge of the algorithm used by each technique without private information shared by the provider;

There exist techniques for publicly verifiable watermarks (Fairoze et al., 2025; A. Liu, Pan, Hu, Li, et al., 2023), in which keys can be publicly shared so that watermarked text can be universally identified without enabling forgery, the watermark they propose are however less effective than secret-key watermarks such as KGW, for example in terms of robustness. Moreover, even with publicly verifiable watermarks, interoperability would still require checking the presence of each provider's watermark.

Logging is based on access to a provider infrastructure; therefore, its interoperability is limited, agreement among providers would be needed for an interoperable usage of logging techniques. However, this requires providers to share their stored generated texts as well as agreeing on specific techniques.

Metadata can be made interoperable if a shared framework, e.g. C2PA, is agreed upon by several providers. Providers of generative AI could agree on specific metadata format and properties, tailored to the case of establishing the provenance of AI-generated text. However, such interoperability depends on broad adoption and on mechanisms to ensure the persistence and integrity of metadata under common text transformations.

AI-generated text detection is the most interoperable technique. If a provider shares its detection algorithm (e.g. a model for supervised detection) this becomes openly available and usable for everybody, nevertheless also in this case multiple detectors would have to be used in parallel to detect AI-generated content from multiple providers. Additionally, a publicly shared detector becomes easier to evade.

In summary, we believe that the most promising way forward to improve interoperability is agreement among providers and regulating authorities, in Section 5 we describe a possible solution to foster such agreement. Meant to make techniques interoperable at the verification step, e.g. by running the techniques of each provider on a text.

We remark how the research on interoperability of marking techniques Is limited with only few early-stage results focused on the interoperability of techniques based on the same methodology (Zhou et al., 2024).

3.6. Aggregated Assessment

To conclude this section, we provide an aggregated assessment for each requirement to summarize our conclusions.

The aggregated assessment is the authors' own view of what each requirement means and how well techniques meet it.

Although no methodology correctly identifies all AI-generated texts, they are all effective. Metadata-based techniques are the most effective. Watermarking is also very effective because it embeds a signal directly in the text and provides statistical guarantees of detection. Logging is similarly effective and is meant as a supporting technique for those cases where watermarking fails, e.g. when texts are paraphrased. Structural marking and AI-generated text detection are less effective, but they can serve as supporting measures for cases when the other methodologies should fail, for example if a text has not metadata attached

and there is not a watermarking detected, AI-generated text detection can be used as additional detection technique.

The assessment of robustness provides parallel considerations to the assessment of effectiveness for all methodologies, excluding Metadata. Watermarking and Logging are the most robust because they can be detected also after significant modifications are applied to text. Structural marking behaves like watermarking but embeds a weaker signal and is therefore less robust. AI-generated text detection techniques are more fragile because they are trained on static training sets and therefore modifications to text can delude their detection. Metadata instead follows a different trend; it is the most effective methodology and the least robust. This happens because stripping the metadata from a text is generally simple and requires no technical knowledge.

Current solutions to make metadata harder to remove, e.g. Soft-Binding[30] as described in the C2PA specification is a way to address this but it is still underdeveloped for text content.

The reliability of the studied methodologies is generally high; forging a digitally signed metadata is unfeasible unless the private key is leaked, this can happen but is not addressed through technical means. Watermarking is reliable and complex to learn and forge for third parties, however, this becomes feasible if a large amount of watermarked text is available, e.g. by learning the mark from access to large amounts of watermarked and non-watermarked text. Logging is also reliable as it relies on data stored by the provider. AI-generated text detection and structural marking are less reliable since they can be circumvented more easily.

Concerning Accessibility, as remarked in Section 3.4, we believe that methodologies can immediately be deployed in a way that their outcomes can be accessible along with a description of how they work and function. Watermarking and structural marking are more technically involved and therefore we believe they are marginally more difficult to describe. However, it is our conclusion that all methodologies along with their technical details can be provided in a fully accessible manner.

Interoperability, as understood within this report, is the possibility of having detection solutions that are provider-agnostic. This is a challenging goal. We argue that the most promising way forward is shared standards (e.g. C2PA), shared infrastructures and agreement among providers and between providers and regulating authorities. Developing and agreeing on standards and infrastructures is an ongoing process.

[30] https://spec.c2pa.org/specifications/specifications/2.2/softbinding/Decoupled.html

In summary, we believe that different methodologies can be more useful for different aspects of the general goal of making AI-generated text always detectable. Specifically, Metadata can be useful to establish provenance in more regulated contexts, such as when deployers or end-users are interested in affirming that a text in a document is AI-generated. If a text is not provided with metadata or the metadata cannot be authenticated, then Watermarking, Logging and AI-generated text detection are useful for detection. This can be the case of users seeking assurance about a given piece of text found online. Separately, structural marking can help mark and detect text generated by open-weights models for which other methodologies are harder to implement. However, it is less effective than other methodologies.

4. Additional Properties

In Section 3 we defined the properties that we map to the requirements set out in the AI Act and based on those we developed an Assessment Framework. There are additional properties of marking techniques that are not mentioned as requirements in AI Act Article 50, but which we explore because they provide valuable insights into the practical usability of marking techniques for AI-generated texts.

Specifically, we study the following additional properties:

- **Computational Costs**: how computationally expensive each methodology is;
- **Text Quality Degradation**: how strongly each methodology affects text quality;
- **Applicability to Multimodal Models**: how well each methodology can be used in multimodal models.

Although these properties are not used in the Assessment Framework, we provide a five-point assessment system for them as well, as for properties described in Section 3.

4.1. Computational Costs

Watermarking and structural marking can be deployed with limited costs, and these costs are negligible when compared to those required to provide Generative AI systems. Logging can be more demanding from a storage and database perspective, since it requires keeping a record for all generated text. This can be costly, especially when there are large amounts of user interactions. Nonetheless, established technologies exist for effective and efficient storage and querying of large database.

Our five-point assessment system compares the cost of a detection technique with the cost of generating text as follows:

1. **Expensive:** the computational overhead due to marking techniques (for both generation and detection) exceeds the cost of text generation in the marked generative model;

2. **High:** The computational overhead of marking techniques (for both generation and detection) exceeds the cost of likelihood computation in the marked generative model (i.e. the cost of generating one token);

3. **Moderate:** The computational overhead of marking techniques (for both generation and detection) is comparable to the cost of likelihood computation in the marked generative model;

4. **Limited:** The computational overhead of marking techniques (for both generation and detection) is smaller than the cost of likelihood computation in the marked generative model;

5. **Negligible:** The computational overhead of marking techniques (for both generation and detection) is negligible compared to the cost of likelihood computation with the marked generative model;

To support this assessment system, we note that the most expensive step required by most marking techniques is computing the likelihood that a generative model assigns to each token. Therefore, the cost of detection techniques can be estimated by the cost of computing the likelihood of a text passage.

Table 9: Input and Output Token Costs for prominent AI Providers and models, M indicates 1 million.[31]

Provider	Model	Input Cost	Output Cost	Ratio
OpenAI[32]	GPT 4.1	2$ / 1M tokens	8$ / 1M tokens	0.25
Anthropic[33]	Claude 3.7	3$ / 1M tokens	10$ / 1M tokens	~ 0.33
Mistral[34]	Mistral Large 24.11	2$ / 1M tokens	6$ / 1M tokens	0.3
	Average	2.3$	8$	~ 0.29

Computing the likelihood of a text passage is approximately equivalent to generating a single new token. Specifically, this involves providing a text passage as input to a generative model. Table 9 shows the per-token costs of major Generative AI providers for input tokens (prompt) and for output tokens (model output). We can estimate this difference to provide a measure of relative cost for detecting and generating texts with AI.

[31] checked on 18/04/2025

[32] https://openai.com/api/pricing/

[33] https://www.anthropic.com/pricing#api

[34] https://mistral.ai/products/la-plateforme#pricing

4.1.1. Assessment

Based on the described assessment system, we assign a textual description to each methodology.

Table 10 shows how metadata is the cheapest methodology because it can be created on the spot and attached to documents.[35] Only marginally more expensive are watermarking and AI-generated text detection, the cost of applying and detecting a watermark or training a supervised detector are significantly lower than the cost of generative AI itself both computationally and from an infrastructure perspective.

*Table 10: Description for the **Computational Costs** of each Methodology (row).*

	Computational Costs
Watermarking	**Limited:** The computational overhead of marking techniques (for both generation and detection) is smaller than the cost of likelihood computation in the marked generative model;
Structural Marking	**Moderate:** The computational overhead of marking techniques (for both generation and detection) is comparable to the cost of likelihood computation in the marked generative model;
Metadata	**Negligible:** The computational overhead of marking techniques (for both generation and detection) is negligible compared to the cost of likelihood computation with the marked generative model;
Logging	**Moderate:** The computational overhead of marking techniques (for both generation and detection) is comparable to the cost of likelihood computation in the marked generative model;
AI-generated Text Detection	**Limited:** The computational overhead of marking techniques (for both generation and detection) is smaller than the cost of likelihood computation in the marked generative model;

Logging-based techniques require the provider of the detection algorithm to store text and additional data, as well as an infrastructure to perform fast queries over the database. The infrastructure for this database and query system can be demanding based on the number of requests that the Generative AI system handles and is an additional infrastructure, compared to the one used for generative AI. However, there are several well-established tools for large

[35] In this assessment we assume that there is an existing reliable authority providing infrastructure and the signatures for metadata authentication. The cost of this infrastructure is a complex problem in itself and difficult to assess within the scope of this report.

database retrieval, such as Elasticsearch[36] or the open source FAISS[37]. Both these systems provide high performance retrieval and optimized storage of the queries. Applications of these techniques are standard in the backend of web-based applications.

4.2. Text Quality Degradation

Text marking techniques for Generative AI should not compromise the quality of the generated outputs of these systems. Consequently, it is desirable that any text attribution system has limited impact on the quality of the textual outputs.

We use the following assessment system:

1. **Disruptive:** The technique renders the output of Generative AI unreadable.
2. **Strong Impact:** The technique alters the output of Generative AI in a way that can be perceived by human readers and clearly compromises quality;
3. **Sensitive Impact:** The technique alters the output of Generative AI in a way that is noticeable to human readers but does not necessarily reduce the output quality;
4. **Mild Impact:** The technique has an impact on Generative AI outputs, but it is imperceptible to human readers;
5. **Distribution Preserving:** The technique can be applied to Generative AI technologies without altering the distribution of the output, so that quality is not affected;

A commonly used quantitative measure for text quality is the likelihood assigned by a language model. Texts with higher likelihood are typically more fluent and readable. This quantitative approach is strongly reliant on the model used, meaning that likelihoods computed by different models cannot be compared. For this reason, we don't provide thresholds to measure quality degradation based on likelihood. Instead, we propose that providers can show average likelihood-changes between marked and unmarked texts, thereby highlighting when a methodology is too detrimental to the AI output.

[36] https://www.elastic.co/elasticsearch

[37] https://github.com/facebookresearch/faiss

4.2.1. Assessment

Based on the described assessment system, we assign a textual description to each methodology.

Table 11: *Description for the **Text Quality Degradation added** by each Methodology (row).*

	Text Quality Degradation
Watermarking	**Mild Impact:** The technique has an impact on Generative AI outputs, but it is imperceptible to human readers;
Structural Marking	**Mild Impact:** The technique has an impact on Generative AI outputs, but it is imperceptible to human readers;
Metadata	**Distribution Preserving:** The technique can be applied to Generative AI technologies without altering the distribution of the output, so quality is not affected;
Logging	**Distribution Preserving:** The technique can be applied to Generative AI technologies without altering the distribution of the output, so quality is not affected;
AI-generated Text Detection	**Distribution Preserving:** The technique can be applied to Generative AI technologies without altering the distribution of the output, so quality is not affected;

Table 11 reports the scores in the quality degradation property. It is noteworthy that Metadata, Logging and AI-generated text detection are distribution preserving, indicating that they do not alter the distribution of AI-generated text and applying them should not modify the generated text.

Watermarking and structural marking are not necessarily distribution preserving, hence the Mild Impact description. Nevertheless, most techniques focus on not lowering the quality of AI-generated text and therefore they affect the text in a way that does not significantly alter it.

4.3. Applicability to Multimodal Models

Modern Generative AI systems can process several modalities as input, such as images, videos, audio or text. In this report, we focus on models that can accept different input media but only provide text output. As a result, most marking techniques transfer to other input modalities with few modifications.

Based on this, we devise the following assessment system:

1. **Not applicable:** The presence of non-textual input makes the technique inapplicable;
2. **Ineffective:** The technique is formally applicable to a model with non-textual input, but its effectiveness decreases with non-textual inputs;
3. **Interoperable:** The technique is applicable to multimodal models and allows the attribution of Generative AI output;
4. **Effectively Interoperable:** The technique is applicable to multimodal models and non-textual inputs render it more effective;
5. **Fully interoperable: T**he technique applies to multimodal models as to text-only ones and provides equal results in effectiveness, robustness and reliability;

This property measures how well a technique can be applied to Generative AI models with multimodal input. This report only considers multimodal inputs, e.g. images, videos or audio, while the Generative AI output is only text.

4.3.1. Assessment

Based on the described assessment system, we assign a textual description to each methodology.

For Generative AI capable of multimodal outputs, we recommend the use of methodologies developed for other modalities. Table 12 shows how most techniques can be applied also to multimodal models. Text generation follows an autoregressive paradigm regardless of the input modality, (see Figure 1) so most marking techniques can be equally applied when the input consists of media other than text only.

*Table 12: Description for the **Multimodality** application of each Methodology (row).*

	Multimodality
Watermarking	**Effectively Interoperable:** The technique is applicable to multimodal models and non-textual inputs render it more effective;
Structural Marking	**Interoperable:** The technique is applicable to multimodal models and allows the attribution of Generative AI output;
Metadata	**Effectively Interoperable:** The technique is applicable to multimodal models and non-textual inputs render it more effective;

Logging	**Interoperable:** The technique is applicable to multimodal models and allows the attribution of Generative AI output;
AI-generated Text Detection	**Interoperable:** The technique is applicable to multimodal models and allows the attribution of Generative AI output;

For Generative AI capable of multimodal outputs, we recommend the use of methodologies developed for other modalities. To support this recommendation, we also discuss the possibilities of marking the output of models that can ingest and generate multimodal input and output. We consider two kinds of markings:

1. Marking added to existing media (e.g. a fingerprint extracted from an image): these techniques can be used on multimodal models without changes after the model has generated the multimodal content.

2. Marking embedded during generation (e.g. conditioning token likelihood): for AI Systems that use different models for each modality, modality-specific marking techniques can be applied seamlessly. In the case of natively multimodal Generative Models, generation of different modalities is often carried out in a single autoregressive flow. However, this can be disentangled at generation time, based on the token choice. Therefore, methodologies developed specifically for each modality can be applied.

We believe that in most cases marking the output of multimodal models is most effective when separating textual from non-textual content and applying the best suited methodologies for each modality.

5. Recommendations

This section presents recommendations based on the outcomes of the analysis in Sections 3 and 4. We stress that while Section 3 favours certain techniques and methodologies over others, the early stage of development in this field must be taken into consideration and thus these recommendations might change in the future. Nonetheless, the properties identified, and the assessment framework presented here are intended to provide added value in the evaluation of future techniques.

Indeed, while the preferred methodologies might change, the principles underlying our assessment should remain relevant and serve as a foundation in assessing future methodologies and techniques improving on existing ones.

5.1. Matching Methodologies and Generative AI Applications

Based on the assessment proposed in Section 3.6, we believe that having the availability of watermarking directly embedded in the text and digitally signed metadata embedded in structured formats would be desirable for most applications of Generative AI for text.

If this is not possible, we highlight which techniques are, in our opinion, most relevant for various applications of Generative AI. We group Generative AI applications into four categories, a coarser subdivision based on existing ones,[38] and recommend best suited methodologies for each of them. We highlight that, while we indicate which methodologies we think are best suited, these might still lack some of the requirements expressed in AI Act Article 50.

Our subdivision is as follows:

- **General-purpose text generation:** Large scale applications of fully general-purpose text generation (e.g. ChatGPT) is best marked with one or more of the following methodologies: metadata, watermarking, logging and AI-generated text detection. Using all three methodologies in parallel could provide three separate “lines of defence”, each with its specific guarantees, and could enhance effectiveness, robustness and reliability of the overall system. Additionally, any provider of large-scale applications of generative AI has the technical knowledge to implement all the technologies underlying these methodologies.

[38] https://www.mckinsey.com/capabilities/tech-and-ai/our-insights/the-economic-potential-of-generative-ai-the-next-productivity-frontier

- **Application-oriented text generation:** For chat interfaces providing low stake information specific for a given application (e.g. chatbot for product-related user support), the best marking techniques are Watermarking and AI-generated text detection. There is no need for the additional security provided by logging, which comes with additional expenses, and watermarking and AI-generated text detection provide effective results with low technical and computational requirements.
- **Generation of sensitive content:** For applications concerning sensitive content (e.g. AI-written medical reports), the best marking methodologies are Metadata-based ones, which can be provided to users embedded in a specific document format. For this category, given the sensitivity of the content, it is best to have the most effective and reliable methodology. When they are applicable, Metadata-based methodologies have the advantage of requiring limited technical knowledge to be implemented if supported by trusted authorities.
- **Open-weight models**: Structural Marking is the best fit solution because it can be applied to the model parameter directly. However, as described in Section 3, it provides lower effectiveness, robustness and reliability.

To select the best techniques for different applications, we only consider effectiveness, robustness and reliability. We do not consider accessibility and interoperability for the following opposite reasons. Sufficient Accessibility can be achieved by all methodologies through similar interfaces, the same holds for multiple methodologies used simultaneously. Instead, Interoperability, is achievable through agreement among providers and deployers.

5.2. Interoperability

This section addresses effectiveness and interoperability and relates them to the current deployment status of existing techniques. The primary outcome of this report, as shown by the assessment in Section 3, is that while existing marking techniques for AI-generated text detection are promising, there are still key shortcomings that could be partially addressed through multi-layer solution.

5.2.1. Scenario: Centralized Verifier

In Section 3 we find that a possible approach to achieve interoperability is through agreement on standards and practices among providers and between providers and regulating authorities.

We suggest a solution that could be enacted by a third party, such as an external trusted authority in charge of enforcing the AI Act to foster agreement on standards and practices among providers.

We hypothesize the development of a *Centralized Verifier* that providers can participate in on a voluntary basis. Generative AI providers could leverage this centralized interface to facilitate interoperability and receive a guarantee that their system meets the requirements set out by AI Act Article 50(2).

We envision a solution involving a centralized verifier, backed by a trusted authority. The Verifier would consist of two parts:

1. Infrastructure, maintained by the trusted authority, which routes user queries back to the providers' APIs.
2. User-facing interface, through which users can verify whether a given text has been generated by AI systems.

The centralized verifier could be used to facilitate the interoperability of metadata- and watermarking-based techniques, as follows:

- Standardization: the authority backing the verifier could establish shared standards among participating providers, including requirements for the issuance of digitally signed credentials to ensure the authenticity and integrity of metadata.
- Interoperability: the user-facing interface would be the go-to place to check texts against all the techniques made available by participating providers. Through the interface end-users could input a text to establish if it is AI-generated or not:
    - If the text has attached digitally signed metadata, the user is told if the metadata is authentic and can easily inspect its content. If the metadata is authenticated, the user is assured that the text was AI-generated.
    - If there are no metadata or if authentication fails, the text is routed to the detection technique of each provider to understand if it is recognized as generated by one of the participating providers. The centralized verifier does not run the verification but sends requests to the providers. Each provider performs the verification step through their own system.

This scenario would partially address the requirement for interoperability as understood in this report, which has the goal of technology- and deployer-agnostic techniques. This would ensure verification level interoperability, allowing providers to develop their own detection techniques and the centralized verifier would send texts to each of them.

5.2.2. Scenario: Decentralized Verifier

During the technical workshop, participants noted that a centralized verifier may be too demanding for providers of Generative AI, as providers are required to follow stricter integration having a single unified standard. As a potential solution, we propose the use of a *Decentralized Verifier*, requiring less involvement from AI systems providers.

The decentralized version is similar to the centralized one, the main difference is that the role of the trusted authority would be taken by several Trusted Third Parties instead of a single one. Functionally, the solution would be similar to the centralized verifier, but infrastructure and user-facing interfaces are maintained and developed by Trusted Third Parties, allowing more options for providers.

An overarching trusted authority would be required also in this case. Its role would be governance; it would mainly establish which third parties are the trusted ones. These third parties would have an operational role being in charge of the implementation and maintenance of infrastructure and user-facing interfaces.

A drawback compared with the centralized verifier is that there would be no go-to place to authenticate text but several, each possibly accounting for a subset of providers lowering the overall interoperability. On the other hand, this would lower standardization requirements for providers, since each trusted third party could have their own standards.

5.3. Assistive Use

Reliability and robustness are both instrumental to the definition of the exemption in Article 50(2) for assistive use. Establishing what defines assistive uses of AI is a complex question for which there are no readily available quantitative measures. However, there are documents that try to formalize which uses of Generative AI are assistive. Specifically, guidelines for the use of Generative AI in scientific paper writing are now available for most computer science conferences and journals.

For example, the Association for Computational Linguistics (ACL) policy[39] describes which uses of Generative AI assistants needs disclosure and which do not. The guidelines for NEURIPS provide similar recommendations.[40] Both these guidelines consider the use of grammar checks as acceptable, even if the system uses Generative AI. In general, any usage that does not contribute

[39] https://2023.aclweb.org/blog/ACL-2023-policy/

[40] https://neurips.cc/Conferences/2025/LLM

ideas but only writing is tolerable and does not have to be disclosed. On the other hand, if scientific ideas have spawned from a Generative AI model and are later developed or commented on by researchers, disclosure is considered beneficial.

However, there appears to be no quantitative measures used to identify non-compliant use of Generative AI. The policy used by conference committees is to defer the decision of inappropriate use of Generative AI to editors. It is noteworthy that most conferences are asking authors to answer surveys investigating the use of Generative AI, with the goal of measuring its impact on paper writing.

5.4. Accessibility

Existing techniques are overall accessible in that most can provide easy-to-understand detection results, motivations for their output and confidence metrics and explanations. Through this information users can make informed decisions whether to trust the output of detection interfaces. For more involved Stakeholders, technical resources are available that support understanding and auditing of implemented techniques.

Overall, we believe that accessibility is one of the AI Act Requirements that is best met by existing marking and detection techniques, provided that developers of detection interfaces take accessibility requirements into account.

5.4.1. Scenario: AI-generated content disclaimers

We envision a scenario for the disclosure of AI-generated content meant to comply with AI Act Article 50(5).

This scenario is inspired by current **cookie banner** implementation. Under the ePrivacy Directive[41] and GDPR,[42] cookie disclosure is mandatory. Since its enforcement, cookie banners have become standard for user notification and consent. Moreover, the EAA (European Accessibility Act), that came into force in June 2025, applies to cookie banners which must meet accessibility requirements. Therefore, most current cookie banner implementations already meet EAA requirements.

Analogously, we propose a scenario where, when a provider is about to expose AI-generated text to a user, a banner would notify the user and prompt them to accept if they want to be exposed to the content or not. This is a proof of

[41] https://eur-lex.europa.eu/eli/dir/2002/58/oj/eng

[42] https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng

concept for disclosure under AI Act Article 50(5). Although it is likely that real implementations will differ from ours proposal, it showcases an approach for compliance with AI Act Article 50(5) and with the EAA. We introduce this scenario with the objective of supporting the possibility of implementing AI Act compliant solutions that also respect accessibility requirements.

In the technical workshop, several participants pointed out that these banners can present a challenge for users with certain disabilities. This shows how this solution might only partially address the requirements for disclosure of AI-generated content, and it could be a starting point requiring further adjustments.

5.5. Computational Costs

In Section 4.2 we addressed the technical aspects of computational costs and find that the most expensive part of implementing marking techniques for Generative AI for text is computing the likelihood of text passages, equivalent to having a language model generate a single token. However, this could be required multiple times, thereby increasing costs.

This consideration helps answer the question of who should bear the cost of marking and detection techniques. In most cases, the responsibility should fall on the provider of Generative AI. Providers are the only actors able to run generations with their models, most importantly when the generative model is not openly released. This agrees with AI Act Article 50(2) stating that it is the providers' responsibility to ensure that the outputs of their systems are marked in a machine-readable format and detectable as AI-generated (see Excerpt 1). Therefore, providers should bear the costs required to do this.

If downstream providers reusing a model made available by a provider should be required to add their own marking, they could ask for token likelihood information from the model provider. In conclusion, we sustain that in any case it should not be a cost paid by end-users.

6. Discussion

In this report we collect, describe and assess existing solutions for marking and detecting Generative AI textual outputs to provide accurate and actionable insights concerning which methodologies are best suited to address the transparency requirements set out in AI Act Article 50.

The report content and conclusions are the authors' own understanding and assessment and do not pre-judge the European Commission assessment or the assessment done in the Code of Practice drafting.

Due to the novelty of this field, there is lack of consensus on which approaches will prove effective after extensive testing over time. This is a general remark on the content of this report which provides conclusions based on current evidence that is likely to change in the future. In Appendix A, we provide illustrative empirical evaluations supporting our assessment. This is to be intended as an example of how future quantitative regulation could be carried out. We remark that additional tests to provide evaluation of existing methodologies are beyond the scope of this report.

6.1. Limitations and further Considerations

One limitation that covers the content of this report is the novelty of the field. Since most techniques have been developed in the last few years, and at the same time, the capability of Generative AI has improved significantly over the same period, there is still work to be done to establish the full potential and inherent limitations of the techniques explored in this report. This will be settled over time and this report itself can only provide recommendations based on the current state of the art.

Concerning interoperability, in this report we highlight the technical aspects that might pose a challenge to an interoperable application of several marking and detection techniques. Additionally, we point out when interoperability would be enabled by cooperation among providers and between providers and regulating authorities. However, we cannot fully explore how beneficial this cooperation can be. Depending on which information and data providers deem shareable and on the agreement on shared standards, we believe that interoperability may be challenging to achieve.

The report focuses on technical aspects of marking techniques. However, implementing these techniques will have broader implications for providers, end-users and regulators. Some of them can be extrapolated from the technical considerations described in this report. About the implementation of marking techniques, the analysis reported in Section 4 finds that the costs are negligible

compared to the costs of generative AI, and therefore they should be sustainable for providers.

Similarly, we find that many of the techniques studied have limited impact on the quality of generated text. As a result, implementing detection techniques should have a positive impact on end-users which would be able to have access to effective Generative AI systems with the additional opportunity to detect the text they generate.

We can also devise implications for regulators based on the technical findings. We believe there are two key challenges for regulators. Currently, marking techniques for AI-generated text show still some limitations for the core goal of making all AI-generated text automatically detectable but are nevertheless able to provide effective, robust and reliable solutions.

Providers of Generative AI for text can comply with AI Act 50(2) by applying multiple methodologies together. To make existing methodologies compliant, regulators have to establish a practical interpretation for what is considered technically feasible when applying article 50 of the AI Act to text.

Additionally, one aspect of the AI Act, which we only partially cover, is the definition of what assistive use of Generative AI means. In Section 3.3, we provide a study of the resilience of marking techniques to modifications and in Section 5.3 we comment on existing policies on the use of Generative Ai in academic contexts. These could be used as starting points to assess assistive use but only partially cover it. It is the regulator role to establish which uses are assistive and which are not. We recommend that content analysis and text semantic meaning are considered when assessing the assistive use of generative AI. Although there are no current reliable methodologies to relate text content to its AI-generated nature, this should not be discarded in future guidelines. Instead, it should be kept as a desideratum to foster the development of methodologies that account for the nature of AI-generated text.

6.2. Future Research Directions

We identify a promising future research direction for improvements of marking and detection techniques: Complementarity. the systematic study of how existing techniques can be combined, with systematic comparisons of the simultaneous application of multiple techniques.

Studying the complementarity of marking techniques is relevant because different methodologies have different strengths and weaknesses. Currently, applying multiple techniques at the same time is desirable as it does address some of the limitations shown by each technique individually, but the synergy among methodologies can be improved through further research.

Acknowledgements

The authors would like to thank colleagues Andrea Pedrotti, Andrea Esuli and Fabrizio Sebastiani, as well as all collaborators at the Institute of Science and Technologies of Information “A. Faedo” (ISTI) at the Italian National Research Council in Pisa, Italy, for their support. Additionally, the author acknowledges that part of the work was done while visiting The National Center for AI in Society (CAISA) in Copenhagen and thanks Professor Anna Rogers for the invitation.

References

Aaronson, S., & Kirchner, H. (2022). Watermarking gpt outputs.

Abassy, M., Elozeiri, K., Aziz, A., Ta, M. N., Tomar, R. V., Adhikari, B., Ahmed, S. E. D., Wang, Y., Mohammed Afzal, O., Xie, Z., Mansurov, J., Artemova, E., Mikhailov, V., Xing, R., Geng, J., Iqbal, H., Mujahid, Z. M., Mahmoud, T., Tsvigun, A., … Nakov, P. (2024). LLM-DetectAIve: A Tool for Fine-Grained Machine-Generated Text Detection. In D. I. Hernandez Farias, T. Hope, & M. Li (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 336–343). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-demo.35

Atallah, M. J., Raskin, V., Hempelmann, C. F., Karahan, M., Sion, R., Topkara, U., & Triezenberg, K. E. (2003). Natural Language Watermarking and Tamperproofing. In F. A. P. Petitcolas (Ed.), *Information Hiding* (pp. 196–212). Springer. https://doi.org/10.1007/3-540-36415-3_13

Block, A., Rakhlin, A., & Sekhari, A. (2025, June 18). *GaussMark: A Practical Approach for Structural Watermarking of Language Models*. Forty-second International Conference on Machine Learning. https://openreview.net/forum?id=YG3DbpAQBf

Christ, M., & Gunn, S. (2024). Pseudorandom Error-Correcting Codes. In L. Reyzin & D. Stebila (Eds.), *Advances in Cryptology – CRYPTO 2024* (pp. 325–347). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-68391-6_10

Christ, M., Gunn, S., & Zamir, O. (2024). Undetectable Watermarks for Language Models. *Proceedings of Thirty Seventh Conference on Learning Theory*, 1125–1139. https://proceedings.mlr.press/v247/christ24a.html

Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S., & Smith, N. A. (2021). All That's `Human' Is Not Gold: Evaluating Human Evaluation of Generated Text. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 7282–7296). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.565

Dathathri, S., See, A., Ghaisas, S., Huang, P.-S., McAdam, R., Welbl, J., Bachani, V., Kaskasoli, A., Stanforth, R., Matejovicova, T., Hayes, J.,

Vyas, N., Merey, M. A., Brown-Cohen, J., Bunel, R., Balle, B., Cemgil, T., Ahmed, Z., Stacpoole, K., … Kohli, P. (2024). Scalable watermarking for identifying large language model outputs. *Nature*, *634*(8035), 818–823. https://doi.org/10.1038/s41586-024-08025-4

Doughman, J., Mohammed Afzal, O., Toyin, H. O., Shehata, S., Nakov, P., & Talat, Z. (2025). Exploring the Limitations of Detecting Machine-Generated Text. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, & S. Schockaert (Eds.), *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 4274–4281). Association for Computational Linguistics. https://aclanthology.org/2025.coling-main.288/

England, P., Malvar, H. S., Horvitz, E., Stokes, J. W., Fournet, C., Burke-Aguero, R., Chamayou, A., Clebsch, S., Costa, M., Deutscher, J., Erfani, S., Gaylor, M., Jenks, A., Kane, K., Redmiles, E., Shamis, A., Sharma, I., Wenker, S., & Zaman, A. (2020). *AMP: Authentication of Media via Provenance* (No. arXiv:2001.07886). arXiv. https://doi.org/10.48550/arXiv.2001.07886

Fairoze, J., Garg, S., Jha, S., Mahloujifar, S., Mahmoody, M., & Wang, M. (2025). Publicly-Detectable Watermarking for Language Models. *IACR Communications in Cryptology*, *1*(4). https://doi.org/10.62056/ahmpdkp10

Fernandez, P., Chaffin, A., Tit, K., Chappelier, V., & Furon, T. (2023). *Three Bricks to Consolidate Watermarks for Large Language Models* (No. arXiv:2308.00113). arXiv. https://doi.org/10.48550/arXiv.2308.00113

Gehrmann, S., Strobelt, H., & Rush, A. (2019). GLTR: Statistical Detection and Visualization of Generated Text. In M. R. Costa-jussà & E. Alfonseca (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 111–116). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-3019

Giboulot, E., & Furon, T. (2024). WaterMax: Breaking the LLM watermark detectability-robustness-quality trade-off. *Advances in Neural Information Processing Systems*, *37*, 18848–18881. https://doi.org/10.52202/079017-0597

Gu, C., Li, X. L., Liang, P., & Hashimoto, T. (2023, October 13). *On the Learnability of Watermarks for Language Models*. The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=9k0krNzvlV

Harel-Canada, F. Y., Erol, B., Choi, C., Liu, J., Song, G. J., Peng, N., & Sahai, A. (2025). Sandcastles in the Storm: Revisiting the (Im)possibility of Strong Watermarking. In W. Che, J. Nabende, E. Shutova, & M. T. Pilehvar (Eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 29698–29735). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-long.1436

He, Z., Zhou, B., Hao, H., Liu, A., Wang, X., Tu, Z., Zhang, Z., & Wang, R. (2024). Can Watermarks Survive Translation? On the Cross-lingual Consistency of Text Watermark for Large Language Models. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 4115–4129). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.226

Hou, A., Zhang, J., He, T., Wang, Y., Chuang, Y.-S., Wang, H., Shen, L., Van Durme, B., Khashabi, D., & Tsvetkov, Y. (2024). SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (pp. 4067–4082). Association for Computational Linguistics. https://aclanthology.org/2024.naacl-long.226

Hu, X., Chen, P.-Y., & Ho, T.-Y. (2023). RADAR: Robust AI-Text Detection via Adversarial Learning. *Advances in Neural Information Processing Systems*, *36*, 15077–15095.

Jin, H., Zhang, C., Shi, S., Lou, W., & Hou, Y. T. (2024). ProFLingo: A Fingerprinting-based Intellectual Property Protection Scheme for Large Language Models. *2024 IEEE Conference on Communications and Network Security (CNS)*, 1–9. https://doi.org/10.1109/CNS62487.2024.10735575

Jovanović, N., Staab, R., & Vechev, M. (2024). Watermark Stealing in Large Language Models. *Proceedings of the 41st International Conference on Machine Learning*, 22570–22593. https://proceedings.mlr.press/v235/jovanovic24a.html

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., … Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education.

Learning and Individual Differences, *103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). A Watermark for Large Language Models. *Proceedings of the 40th International Conference on Machine Learning*, 17061–17084. https://proceedings.mlr.press/v202/kirchenbauer23a.html

Kirchenbauer, J., Geiping, J., Wen, Y., Shu, M., Saifullah, K., Kong, K., Fernando, K., Saha, A., Goldblum, M., & Goldstein, T. (2023, October 13). *On the Reliability of Watermarks for Large Language Models*. The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=DEJIDCmWOz

Knott, A., Pedreschi, D., Chatila, R., Chakraborti, T., Leavy, S., Baeza-Yates, R., Eyers, D., Trotman, A., Teal, P. D., Biecek, P., Russell, S., & Bengio, Y. (2023). Generative AI models should include detection mechanisms as a condition for public release. *Ethics and Information Technology*, *25*(4), 55. https://doi.org/10.1007/s10676-023-09728-4

Krishna, K., Song, Y., Karpinska, M., Wieting, J., & Iyyer, M. (2023). Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 27469–27500.

Kuditipudi, R., Thickstun, J., Hashimoto, T., & Liang, P. (2023). Robust Distortion-free Watermarks for Language Models. *Transactions on Machine Learning Research*. https://openreview.net/forum?id=FpaCL1MO2C

Lee, T., Hong, S., Ahn, J., Hong, I., Lee, H., Yun, S., Shin, J., & Kim, G. (2024). Who Wrote this Code? Watermarking for Code Generation. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 4890–4911). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.268

Li, L., Bai, Y., & Cheng, M. (2024). Where Am I From? Identifying Origin of LLM-generated Content. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 12218–12229). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.681

Liu, A., Pan, L., Hu, X., Li, S., Wen, L., King, I., & Yu, P. S. (2023, October 13). *An Unforgeable Publicly Verifiable Watermark for Large Language*

Models. The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=gMLQwKDY3N

Liu, A., Pan, L., Hu, X., Meng, S., & Wen, L. (2023, October 13). *A Semantic Invariant Robust Watermark for Large Language Models*. The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=6p8lpe4MNf

Liu, A., Pan, L., Lu, Y., Li, J., Hu, X., Zhang, X., Wen, L., King, I., Xiong, H., & Yu, P. (2024). A Survey of Text Watermarking in the Era of Large Language Models. *ACM Comput. Surv.*, *57*(2), 47:1-47:36. https://doi.org/10.1145/3691626

Liu, Y., & Bu, Y. (2024). Adaptive Text Watermark for Large Language Models. *Proceedings of the 41st International Conference on Machine Learning*, 30718–30737. https://proceedings.mlr.press/v235/liu24e.html

Lu, Y., Liu, A., Yu, D., Li, J., & King, I. (2024). An Entropy-based Text Watermarking Detection Method. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 11724–11735). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.630

Meng, X., Yan, X., Zhang, K., Liu, D., Cui, X., Yang, Y., Zhang, M., Cao, C., Wang, J., Wang, X., Gao, J., Wang, Y.-G.-S., Ji, J., Qiu, Z., Li, M., Qian, C., Guo, T., Ma, S., Wang, Z., … Tang, Y.-D. (2024). The application of large language models in medicine: A scoping review. *iScience*, *27*(5). https://doi.org/10.1016/j.isci.2024.109713

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). DetectGPT: Zero-shot machine-generated text detection using probability curvature. *Proceedings of the 40th International Conference on Machine Learning*, *202*, 24950–24962.

Molenda, P., Liusie, A., & Gales, M. (2024). WaterJudge: Quality-Detection Trade-off when Watermarking Large Language Models. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Findings of the Association for Computational Linguistics: NAACL 2024* (pp. 3515–3525). Association for Computational Linguistics. https://aclanthology.org/2024.findings-naacl.223

Pan, L., Liu, A., He, Z., Gao, Z., Zhao, X., Lu, Y., Zhou, B., Liu, S., Hu, X., Wen, L., King, I., & Yu, P. S. (2024). MarkLLM: An Open-Source Toolkit for LLM Watermarking. In D. I. Hernandez Farias, T. Hope, & M. Li (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural*

Language Processing: System Demonstrations (pp. 61–71). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-demo.7

Pedrotti, A., Papucci, M., Ciaccio, C., Miaschi, A., Puccetti, G., Dell'Orletta, F., & Esuli, A. (2025). Stress-testing Machine Generated Text Detection: Shifting Language Models Writing Style to Fool Detectors. In W. Che, J. Nabende, E. Shutova, & M. T. Pilehvar (Eds.), *Findings of the Association for Computational Linguistics: ACL 2025* (pp. 3010–3031). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.findings-acl.156

Puccetti, G., Rogers, A., Alzetta, C., Dell'Orletta, F., & Esuli, A. (2024). AI `News' Content Farms Are Easy to Make and Hard to Detect: A Case Study in Italian. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 15312–15338). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.817

Sander, T., Fernandez, P., Durmus, A. O., Douze, M., & Furon, T. (2024, November 6). *Watermarking Makes Language Models Radioactive*. The Thirty-eighth Annual Conference on Neural Information Processing Systems. https://openreview.net/forum?id=qGiZQb1Khm&referrer=%5Bthe%20profile%20of%20Alain%20Oliviero%20Durmus%5D(%2Fprofile%3Fid%3D~Alain_Oliviero_Durmus1)

Tang, R., Feng, Q., Liu, N., Yang, F., & Hu, X. (2023). Did You Train on My Dataset? Towards Public Dataset Protection with CleanLabel Backdoor Watermarking. *SIGKDD Explor. Newsl.*, *25*(1), 43–53. https://doi.org/10.1145/3606274.3606279

Tomlinson, K., Jaffe, S., Wang, W., Counts, S., & Suri, S. (2025). *Working with AI: Measuring the Occupational Implications of Generative AI* (No. arXiv:2507.07935). arXiv. https://doi.org/10.48550/arXiv.2507.07935

Topkara, U., Topkara, M., & Atallah, M. J. (2006). The hiding virtues of ambiguity: Quantifiably resilient watermarking of natural language text through synonym substitutions. *Proceedings of the 8th Workshop on Multimedia and Security*, 164–174. https://doi.org/10.1145/1161366.1161397

Tu, S., Sun, Y., Bai, Y., Yu, J., Hou, L., & Li, J. (2024). WaterBench: Towards Holistic Evaluation of Watermarks for Large Language Models. In L.-W. Ku, A. Martins, & V. Srikumar (Eds.), *Proceedings of the 62nd Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1517–1542). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.83

Uchendu, A., Le, T., Shu, K., & Lee, D. (2020). Authorship Attribution for Neural Text Generation. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8384–8395. https://doi.org/10.18653/v1/2020.emnlp-main.673

Verma, V., Fleisig, E., Tomlin, N., & Klein, D. (2024). Ghostbuster: Detecting Text Ghostwritten by Large Language Models. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (pp. 1702–1717). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.95

Wouters, B. (2024). Optimizing Watermarks for Large Language Models. *Proceedings of the 41st International Conference on Machine Learning*, 53251–53269. https://proceedings.mlr.press/v235/wouters24a.html

Wu, Y., Hu, Z., Guo, J., Zhang, H., & Huang, H. (2024). A Resilient and Accessible Distribution-Preserving Watermark for Large Language Models. *Proceedings of the 41st International Conference on Machine Learning*, 53443–53470. https://proceedings.mlr.press/v235/wu24h.html

Xu, J., Wang, F., Ma, M., Koh, P. W., Xiao, C., & Chen, M. (2024). Instructional Fingerprinting of Large Language Models. In K. Duh, H. Gomez, & S. Bethard (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (pp. 3277–3306). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.180

Zeng, B., Wang, L., Hu, Y., Xu, Y., Zhou, C., Wang, X., Yu, Y., & Lin, Z. (2024, November 6). *HuRef: HUman-REadable Fingerprint for Large Language Models*. The Thirty-eighth Annual Conference on Neural Information Processing Systems. https://openreview.net/forum?id=RlZgnEZsOH

Zhang, H., Edelman, B. L., Francati, D., Venturi, D., Ateniese, G., & Barak, B. (2024). Watermarks in the Sand: Impossibility of Strong Watermarking for Language Models. *Proceedings of the 41st International Conference on Machine Learning*, 58851–58880. https://proceedings.mlr.press/v235/zhang24o.html

Zhao, X., Ananth, P. V., Li, L., & Wang, Y.-X. (2023, October 13). *Provable Robust Watermarking for AI-Generated Text*. The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=SsmT8aO45L

Zhao, X., Li, L., & Wang, Y.-X. (2024, October 4). *Permute-and-Flip: An optimally stable and watermarkable decoder for LLMs*. The Thirteenth International Conference on Learning Representations. https://openreview.net/forum?id=YyVVicZ32M

Zhou, T., Zhao, X., Xu, X., & Ren, S. (2024, November 6). *Bileve: Securing Text Provenance in Large Language Models Against Spoofing with Bi-level Signature*. The Thirty-eighth Annual Conference on Neural Information Processing Systems. https://openreview.net/forum?id=vjCFnYTg67&referrer=%5Bthe%20profile%20of%20Tong%20Zhou%5D(%2Fprofile%3Fid%3D~Tong_Zhou3)

A. Appendix

In this section we provide illustrative quantitative scoring systems for some of the properties used in the assessment developed in Section 3.

The content of this Section is the authors' own choice, it is not verified by other authors or peer reviewed and does not pre-judge the assessment done by the Commission. All numerical results serve as illustrative examples of how a quantitative assessment could be carried out but do not provide a comprehensive evaluation of tested techniques and do not pre-judge future requirements or assessments carried out by the European Commission or in the Code of Practice drafting.

A.1. Performance

Most of the research analysed in this report, among others SynthID (Dathathri et al., 2024), focuses on a one metric: True Positive Rate at 1% False Positive Rate, **TPR @ 1% FPR.** When assessing marking techniques for text, positives are AI-generated texts and negatives are human-written texts. Therefore:

- **True Positives (TP)** are AI-generated samples that are correctly identified through the marking techniques,
- **False Positives (FP)** are human-written samples that are incorrectly identified as AI-generated,
- **False Negatives (FN)** are AI-generated texts that are incorrectly identified as human-written
- **True Negatives (TN)** are human-written texts that are correctly identified as such.

As shown in Table A1, True Positive Rate (TPR = TP / (TP + FN)) is the proportion of AI-generated texts correctly identified through marking techniques and False Positive Rate (FPR = FP / (FP + TN)) is the proportion of human-written texts that are misclassified as AI-generated.

The metric **TPR @ 1% FPR** indicates the proportion of AI-generated texts correctly identified as AI-generated while only misclassifying 1% of the human-written texts as AI-generated. It is the most used metric in the reviewed literature to assess text detection methodologies in this context.

This metric is used under the assumption that attributing a human-text to AI is a worse mistake than the opposite. This choice aims to achieve a twofold objective:

- Preferring systems that correctly identify as many AI-generated texts as possible, by maximizing TPR;
- Having a fixed threshold for the number of human-written texts that are incorrectly attributed to AI.

Table A1: Confusion Matrix for classification from Wikipedia. (In the context of marking techniques, a positive sample is an AI-generated text, and a negative sample is a human-written text.)

Total population = P + N	**Predicted positive**	**Predicted negative**	
Positive (P)	**True positive** (TP)	**False negative** (FN)	True Positive Rate (TPR) TPR = TP / (TP + FN)
Negative (N)	**False positive** (FP)	**True negative** (TN)	False Positive Rate (FPR) FPR = FP / (FP + TN)

In summary, optimising TPR @ 1% FPR aims to maximize detection performance while misclassifying a controlled number of human-written texts

We choose TPR @ 1% FPR to stress that we agree with the principle that attributing human-text to Generative AI is a more serious error than the opposite. The choice for the specific threshold of 1% is taken from the literature and it can be chosen either stricter (lower) or more lenient (higher) in practical applications. During the technical workshop a lower threshold than 1% was suggested, we use 1% for coherence with most existing literature but do not exclude that several applications might require a lower threshold.

Table A2 and the resulting scoring system serve as an illustrative example of how a quantitative assessment could be carried out but do not provide a comprehensive evaluation of tested techniques or an indication for formal requirements to be used by the European Commission or in the drafting of the Code of Practice.

To score the *Performance* of the techniques studied in this work we devise a scale that considers the TPR @ 1% FPR achieved when analysing text sequences of 200 tokens. The choice for the threshold used in the scoring system is based on the values reported in Table A2. This table reports the

performance of several techniques when marking two language models (OPT 1.3 B[43] and Llama 3.1 8B[44]) on texts that are 200 tokens long. Thresholds are set so that FPR is always 0.01, therefore the TPR column indicates specifically TPR @ 1% FPR. The dataset where this test is run is a balanced subset of the C4 dataset (50% AI-generated / 50% human-written), one of the most used when evaluating marking techniques. The choice for a balanced subset is based on human ability to detect AI-generated texts, which is found to be close to random chance (Clark et al., 2021). Therefore, since the prevalence of AI-generated text in existing sources is hard to establish, we believe that a balanced test set is a reasonable choice. Based on usage metrics and other evaluation, different providers might use differently composed test sets. The results are obtained through the MarkLLM open-source software (Pan et al., 2024).[45] We refer to their work for details about the tested techniques. For non-specified parameters we use the default values provided by MarkLLM.

Table A2: Performance scores for several methodologies. The columns report: the model (model), the marking technique (technique), True Positive Rate (TPR), False Positives Rate (FPR).

model	OPT 1.3 B		Llama 3.1 8B	
technique	**TPR**	**FPR**	**TPR**	**FPR**
EWD	**1**	0.01	**0.99**	0.01
KGW	**1**	0.01	**1**	0.01
SIR	**0.98**	0.01	**0.98**	0.01
SWEET	**1**	0.01	**0.99**	0.01
Unigram	**1**	0.01	**0.99**	0.01
XSIR	**0.96**	0.01	**0.96**	0.01

Most techniques achieve between 80% and 100% TPR @ 1% FPR for the two models tested and many almost perfectly with 100%, we remark that this does not indicate that these techniques identify 100% of the AI-generated texts in all scenarios.

We choose this metric and specifically 1% FPR because this is the most commonly used value in the literature we reviews (Dathathri et al., 2024; Kirchenbauer, Geiping, Wen, Katz, et al., 2023; Kuditipudi et al., 2023; Pan et al., 2024). This choice is highly dependent on specific applications and provider traffic; therefore, this scoring system can be adapted to broader scenarios. In

[43] https://huggingface.co/facebook/opt-1.3b

[44] https://huggingface.co/meta-llama/Llama-3.1-8B

[45] https://github.com/THU-BPM/MarkLLM

practical applications this threshold can be made more restrictive, choosing FPR to be as low as 1/1.000.000, i.e. allowing one in a million human texts to be misattributed to AI (Fernandez et al., 2023).

For example, providers with millions of requests per day might aim to guarantee a lower (better) FPR through higher detection costs. Providers of AI systems with moderate traffic, could guarantee higher (worse) FPR because they would otherwise incur excessive detection costs.

Based on the results in Table A1 we devise the following scoring system. Where we explore the scores into an overall measure of Performance as outlined in the following list:

1. **Low Performance:** 0.00 ≤ TPR @ 1% FPR ≤ 0.80
2. **Moderate Performance:** 0.80 < TPR @ 1% FPR ≤ 0.90
3. **High Performance:** 0.90 < TPR @ 1% FPR ≤ 0.95
4. **Very High Performance:** 0.95 < TPR @ 1% FPR ≤ 0.99
5. **Near-perfect Performance:** 0.99 < TPR @ 1% FPR ≤ 1.00.

A.2. Length Independence

Table A3 shows performance of the same techniques tested on the same models as in Table A2 on shorter texts from the same dataset, the results show that methodologies can lose up to 15% TPR when tested on 50 or less tokens. The results are obtained through the MarkLLM open-source software (Pan et al., 2024). We refer to their work for details about the tested techniques.

Table A3 and the resulting scoring system serve as an illustrative example of how a quantitative assessment could be carried out but do not provide a comprehensive evaluation of tested techniques or an indication for formal requirements to be used by the European Commission or in the drafting of the Code of Practice.

Table A3: Performance score for several methodologies on short texts (50 tokens). The columns report: the model (model), the marking technique (technique), True Positive Rate (TPR), False Positives Rate (FPR).

model	OPT 1.3 B		Llama 3.1 8B	
technique	**TPR**	**FPR**	**TPR**	**FPR**
EWD	0.96	0.01	0.96	0.01
KGW	0.92	0.01	0.96	0.01

SIR	0.87	0.01	0.83	0.01
SWEET	0.91	0.01	0.88	0.01
Unigram	0.86	0.01	0.93	0.01
XSIR	0.85	0.01	0.84	0.01

Based on the results in Table A3, we develop the scoring system. We evaluate each technique's ability to retain high TPR @ 1% FPR (greater than 0.80) when evaluating on shorter text passages. The five-point scoring system is as follows:

1. **Heavily reliant on length:** 150 words ≤ Length
2. **Strongly reliant on length:** 100 words < Length ≤ 150 words
3. **Reliant on length:** 50 words < Length ≤ 100 words
4. **Mildly reliant on length:** 30 words < Length ≤ 50 words
5. **Independent from length:** 0 words < Length ≤ 30 words

The choice for a text length between 30 tokens and 150 tokens is based on the literature. Most works explore the dependence of their technique on text length (Dathathri et al., 2024; Kirchenbauer, Geiping, Wen, Katz, et al., 2023; Krishna et al., 2023; Kuditipudi et al., 2023; Y. Liu & Bu, 2024). They find that detection performance drops significantly with texts shorter than 30 tokens and that with texts longer than 150 tokens performance stays mostly unchanged.

A.3. Resilience to Editing

Table A4 reports the same performance of the same techniques tested on the same models as in Table A2 for watermarking techniques under substitution attacks when substituting 50% of the words in the text with synonyms. Only a few techniques manage to maintain TPR @ 1% FPR above 80%, the minimum value considered in the performance scoring system, nevertheless few do and therefore we adjust the scoring system based on this knowledge. The results are obtained through the MarkLLM open-source software (Pan et al., 2024). We refer to their work for details about the tested techniques.

Table A4 and the resulting scoring system serve as an illustrative example of how a quantitative assessment could be carried out but do not provide a comprehensive evaluation of tested techniques or an indication for formal requirements to be used by the European Commission or in the drafting of the Code of Practice.

Table A4: Performance of watermarking techniques under substitution attacks. The columns report: the model (model), the marking technique (technique), True Positive Rate (TPR), False Positives Rate (FPR).

model	OPT 1.3 B		Llama 3.1 8B	
technique	**TPR**	**FPR**	**TPR**	**FPR**
EWD	0.845	0.01	0.765	0.01
KGW	0.755	0.01	0.86	0.01
SIR	0.82	0.01	0.585	0.01
SWEET	0.75	0.01	0.595	0.01
Unigram	0.995	0.01	0.895	0.01
XSIR	0.555	0.01	0.48	0.01

Specifically, we consider the percentage of randomly modified words that a text can undergo while maintaining a TPR @ 1% FPR above 80% (the lowest value to have a score higher than the minimum in Performance). These numbers help us develop the scoring system for this property.

Considering this, we assess the **Resilience to Editing** Property, using a five-point scale:

1. **Non-Resilient:** less than 10% of modified words lead to TPR @ 1% FPR below 0.8;
2. **Mildly Resilient:** between 10% and 20% modified words lead to TPR @ 1% FPR below 0.8;
3. **Resilient:** between 20% and 30% modified words lead to TPR @ 1% FPR below 0.8;
4. **Highly Resilient:** between 30% and 40% modified words lead to TPR @ 1% FPR below 0.8;
5. **Fully Resilient:** between 40% and 50% modified words lead to TPR @ 1% FPR below 0.8;

The scoring system for resilience to editing is tailored to address the requirements of AI Act Article 50(2). We consider fully resilient methodologies that are resilient to text changes of up to 50%, since this threshold may help identify those cases when Generative AI is used with an assistive function and is thus exempt from the requirements of AI Act Article 50(2).

The 50% threshold of modified words is an arbitrary threshold. We choose it because it is often the largest number of modifications used to test marking techniques in the literature, (Block et al., 2025; Kirchenbauer, Geiping, Wen,

Shu, et al., 2023; Zhao et al., 2023, 2024). We stress that a threshold of this kind should always be considered as part of any assessment system, to implicitly identify when an AI-generated text has been sufficiently modified to consider the AI role as assistive and thus not subject to disclosure as AI-generated.

Publications Office
of the European Union