From Intentions to Techniques: A Comprehensive Taxonomy and Challenges in Text Watermarking for Large Language Models

Harsh Nishant Lalai **Aashish Anantha Ramakrishnan**
Raj Sanjay Shah **Dongwon Lee**
Birla Institute of Technology and Science, Pilani
The Pennsylvania State University Georgia Institute of Technology

Abstract

With the rapid growth of Large Language Models (LLMs), safeguarding textual content against unauthorized use is crucial. Watermarking offers a vital solution, protecting both - LLM-generated and plain text sources. This paper presents a unified overview of different perspectives behind designing watermarking techniques through a comprehensive survey of the research literature. Our work has two key advantages: (1) We analyze research based on the specific intentions behind different watermarking techniques, evaluation datasets used, and watermarking addition and removal methods to construct a cohesive taxonomy. (2) We highlight the gaps and open challenges in text watermarking to promote research protecting text authorship. This extensive coverage and detailed analysis sets our work apart, outlining the evolving landscape of text watermarking in Language Models.

1 Introduction

Large Language Models (LLMs) can mimic human-like comprehension and text generation (Zheng et al., 2024). Consequently, judging whether a text is authored by a human or generated by an LLM is challenging. This issue is highlighted by the recent lawsuit of The New York Times against OpenAI and Microsoft concerning the use of their articles as training data for AI models, emphasizing the need for effective methods to identify and safeguard digital content ownership (New York Times Company, 2023).

Text Watermarking provides key solutions to protect intellectual property rights, identify ownership, and keep track of digital content. These techniques embed imperceptible signals or identifiers within digital text documents, which are then used to track the document's origins (Jalil and Mirza, 2009; Kamaruddin et al., 2018). In particular, they aid in tracking the different production sources of text, both human-written and LLM-generated, helping prevent their unauthorized use without the owner's consent.

Given this increasing research focus on watermarking techniques, it is important to review various methods, their applications, strengths and limitations. This includes systematically categorizing current research literature and highlighting key open challenges. The following contributions of our work distinguish it from previous surveys:

- **Taxonomy Construction:** We seek to help future researchers navigate text-watermarking by categorizing various techniques and methods. Unlike traditional surveys, our paper aims to use the constructed taxonomy to provide an up-to-date list of research challenges for the text watermarking field instead of the most up-to-date survey of the field. For this task, we focus on *application-driven intentions, evaluation data sources, and watermark addition methods*. We also enlist potential adversarial attacks against these methods to caution readers.
- **Open Challenge Identification:** Next, we describe open challenges and gaps in current research efforts. These span rigorous testing of methods against diverse de-watermarking attacks, the establishment of standardized benchmarks for appropriate method efficacy comparison, understanding how watermarking impacts language model factuality and utility, the interpretability of watermarking techniques by detailed descriptions and visual aids, and lastly, expansion of the downstream NLP tasks used for evaluation.

Our work aims to enable researchers to recognize emerging trends and areas for improvement in text watermarking research. We facilitate this goal by

Email: f20212665@goa.bits-pilani.ac.in, {aza6352, dongwon}@psu.edu, rajsanjayshah@gatech.edu

Findings of the Association for Computational Linguistics: NAACL 2025, pages 6162–6175
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

creating a systematic and comprehensive taxonomy of text watermarking.

2 Taxonomy of Text Watermarking

To help researchers navigate the field of text watermarking, we cluster various techniques and methods based on key commonalities. For this categorization, we focus on *intentions that are application-driven, data sources for model evaluation, watermark addition methods, and method-specific adversarial attacks*. In our taxonomy creation, we allow techniques to belong to multiple categories and show how different techniques relate across multiple dimensions, making it easier to navigate the field.

2.1 Intention

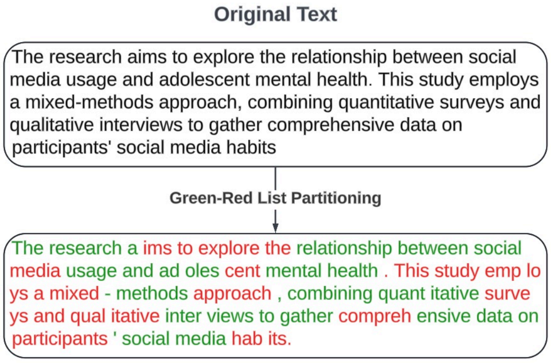
Methods for embedding textual identifiers to watermark differ based on a user's desired features, the user's role (developer vs end-user, etc.), and primary application-driven needs. We categorize watermarking techniques based on the *end user's intention* into three types: *Text Quality, Output Distribution*, and *Model Ownership Verification*.

2.1.1 Text Quality

Maintaining the quality and utility of the generated text post-watermarking is a desired goal of any watermarking methodology. However, research works differ on definitions of quality and mainly proxy output quality with (1) *generation perplexity (uncertainty)* and (2) *semantic relatedness of watermarked and un-watermarked generations*.

Minimizing impact on Perplexity A model's confidence in its generations, measured through the weighted sum of individual token log probabilities in a sequence is known as Perplexity. A lower perplexity indicates that the model is more confident and accurate in its predictions, while a higher perplexity suggests more significant uncertainty and less accurate predictions. Perplexity is the only intrinsic measure of model uncertainty (Magnusson et al., 2023), and thus, a popular measure of quality among researchers.

Watermarking techniques like using green-red list rules (refer to figure 1) trade-off the ability to detect LLM-generated text and the utility of the output text. For a given text, the greater the proportion of green tokens from the total tokens, the lesser the chance of the text being written by humans. A parallel aim is to reduce all of the other "generic" text's perplexity while enforcing a more frequent generation of white-listed "green" words. Controlling entropy levels ensures that watermarked text maintains a quality similar to non-watermarked text. Studies have effectively operationalized this technique in diverse ways for authorship detection while maintaining high text quality (Kirchenbauer et al., 2023; Zhao et al., 2023a; Takezawa et al., 2023). For example, soft watermarking promotes green list use for high-entropy (rare) tokens while minimally affecting low-entropy (common) tokens (Kirchenbauer et al., 2023; Lee et al., 2023; Ren et al., 2024), ensuring that watermark is undetectable (soft) to an observer. In another example,Takezawa et al. (2023) recommend a lower watermark strength for longer texts for quality. Some techniques only alter text appearance, for example, change "e" to "é", rather than modifying the content to have no perplexity impact (Brassil et al., 1995; Por et al., 2012; Sato et al., 2023).



The research aims to explore the relationship between social media usage and adolescent mental health. This study employs a mixed-methods approach, combining quantitative surveys and qualitative interviews to gather comprehensive data on participants' social media habits

The research a ims to explore the relationship between social media usage and ad oles cent mental health . This study emp lo ys a mixed - methods approach , combining quant itative surve ys and qual itative inter views to gather compreh ensive data on participants ' social media hab its.

Figure 1: An example of green-red list grouping of texts (Kirchenbauer et al., 2023).

Table 1: Overview of watermarking techniques using semantic relatedness. *Struct*: Maintains Structure, *Word repl*: Synonym/ Spelling based word replacement techniques, *Dep. trees*: Dependency trees, *Syn. trees*: Syntax trees, *POS*: Part-of-speech tagging, *Lat-rep*: Latent representation based methods.

Work	Struct	Word repl.	Dep. trees	Syn. trees	POS	Lat. rep.
(Topkara et al., 2006b)	✓	✓	✗	✗	✗	✓
(Meral et al., 2009)	✗	✗	✗	✓	✗	✗
(Abdelnabi and Fritz, 2021)	✓	✗	✗	✗	✗	✓
(Yang et al., 2022)	✓	✓	✗	✗	✗	✓
(He et al., 2022a)	✓	✓	✓	✗	✓	✓
(He et al., 2022b)	✓	✓	✓	✓	✓	✗
(Yoo et al., 2023a)	✗	✗	✓	✗	✗	✗
(Yang et al., 2023b)	✓	✓	✗	✗	✗	✓
(Munyer and Zhong, 2023)	✓	✓	✗	✗	✗	✓
(Fu et al., 2024)	✗	✗	✗	✗	✗	✓
(Hoang et al., 2024)	✗	✗	✗	✗	✓	✗

Semantic Relatedness Refers to how closely words, phrases, or sentences of the watermarked output are similar to the original clean output. One way of maintaining input semantics is by embedding both input and output sentences into a semantic space and minimizing the distance between them (Abdelnabi and Fritz, 2021; Zhang et al., 2023). Yang et al. (2022) use the BERT model to suggest substitution candidates, while other works use synonyms and spelling replacements to have minimum impact on semantic relatedness. Fu et al. (2024) use the input context to extract semantically related tokens, measured by word vector similarity to the source. In more nuanced domains like code generation, the preservation of semantics has been achieved by changing variable names (Li et al., 2023; Yang et al., 2023a). Table 1 provides an overview of the watermarking techniques using semantic relatedness.

Chen et al. (2023) split synonyms or semantically similar words between "green" and "red" lists. This excludes the possibility of all suitable alternatives being placed in the same list, ensuring that if one synonym is discouraged, another remains available. This allows LLMs to maintain their original articulation ability. Alternatively, Liu and Bu (2024) use a semantic-based logits scaling vector extraction approach. This method adjusts the logits based on the meaning of the previously generated text, ensuring that the watermark perturbations align with the original text's meaning. Li et al. (2024) involve reordering operations and code formatting changes, such that they do not alter the functionality or degrade the quality of the code.

2.1.2 Similar Output Distribution

Ensuring that the word distribution in watermarked text or LLM-generated output closely resembles that of the original text is essential for providing a natural experience to the end user. This is often operationalized as re-weighting strategies that adjust the probabilities of select words during text generation such that the overall distribution of words remains consistent with the original (Hu et al., 2023; Wu et al., 2023).

Hu et al. (2023) and Wu et al. (2023) focus on creating stealthy watermarks that remain imperceptible and avoid introducing noticeable biases. By adjusting the output logits of LLMs, these methods preserve the original text distribution and minimize the likelihood of detection. In some methods, this is done by systematically rearranging the words (permutation) in the vocabulary set to find optimal combinations that maintain the inherent symmetry of the original distribution (Wu et al., 2023). This method exploits the mathematical property of symmetry in permutations, where different arrangements can still produce the same statistical distribution, allowing for flexibility in embedding watermarks without altering the natural flow of the text.

2.1.3 Model Ownership Verification

Model ownership verification techniques use watermarks to safeguard against adversaries by helping model creators prove ownership, even if adversaries attempt to emulate the model's functionality. For an adversary, emulating LLM behavior requires understanding the workings of a model. An adversary's goals include model extraction - where they seek to exploit or verify the properties of an LLM and recreate the model by extensively querying it. Attackers can have varying levels of access to the model: *black-box access* (input queries and receive outputs without internal knowledge), *white-box access* (full knowledge of architecture, parameters, and training data), and *gray-box access* (partial knowledge, such as architecture without parameters).

Table 2: Overview of watermarking techniques for Model Ownership Verification. *Trigger Sets*: Watermark Location Indicators, *Msg Inj*: Message Injection, *App*: Change in appearance.

Work	Trigger Sets	Secret Keys	Msg Inj	App.
(Brassil et al., 1995)	✗	✗	✗	✓
(Atallah et al., 2001)	✗	✓	✗	✗
(Por et al., 2012)	✗	✗	✗	✓
(Dai et al., 2022)	✓	✓	✗	✗
(Peng et al., 2023)	✓	✗	✗	✗
(Tang et al., 2023)	✓	✗	✗	✗
(Zhang et al., 2023)	✗	✗	✓	✗
(Fairoze et al., 2023)	✗	✓	✓	✗
(Kuditipudi et al., 2023)	✗	✗	✓	✗
(Sato et al., 2023)	✗	✗	✗	✓
(Zhao et al., 2023a)	✗	✓	✗	✗
(Zhao et al., 2023b)	✗	✓	✓	✗
(Liu et al., 2023c)	✓	✗	✗	✗
(Shao et al., 2024)	✓	✗	✗	✗
(Qu et al., 2024)	✗	✓	✓	✗

The attack conditions define the environment and constraints under which the attack is conducted. These conditions include resource constraints (computational resources like processing power, memory, and time), access constraints (black box, white box, or gray box), knowledge assumptions (information the attacker has about the model, including architecture, training data, or defense mechanisms), detection and evasion (avoiding detection if the model has monitoring systems), and performance metrics (criteria for evaluating attack success, such as accuracy of model extraction, watermark detection consistency, or successful adversarial perturbations).

Combating attackers often requires a technique with minimal false positives, i.e., the unauthorized emulation of LLMs is easily detected. For model ownership verification, techniques like trigger sets rely on the predictability of specific outputs given exact inputs. Trigger sets are specific inputs designed to activate watermarks embedded within a model or dataset (Dai et al., 2022; Peng et al., 2023; Liu et al., 2023c; Tang et al., 2023). Dai et al. (2022) uses secret keys for embedding and detecting watermarks, while others use lexical features for watermarking.

Injecting secret signals/messages/signatures in the watermark generation process is also used for verification (Zhao et al., 2023b; Zhang et al., 2023; Fairoze et al., 2023; Qu et al., 2024; Kuditipudi et al., 2023; Wang et al., 2023; Zhou et al., 2024). Wang et al. (2023), Qu et al. (2024) and Guan et al. (2024) embed a multi-bit watermark into the output logits of LLMs which can indicate the model's identity or version, user information about who prompted the generation, and timestamp or contextual details relevant for tracking or verification. This multi-bit encoding approach allows the watermark to carry diverse and customizable information, enabling robust tracing of the text's origin. Zhao et al. (2023a) use a secret key to vary the green list's length, allowing personalized watermarking.

2.2 Watermark Addition

Often, the same watermarking methods work for different user intentions. Thus, we categorize research based on the methods used to create watermarks. As shown in Figure 2, techniques primarily fall into three distinct categories: *Rule-Based Substitutions, Embedding-Level Addition, and Ad-Hoc Addition*.

2.2.1 Rule Based Substitution

In rule-based substitution techniques, certain elements are replaced in the text based on specific rules or patterns while preserving the overall structure and semantics of the text. These rules are typically reversible, ensuring that the original content can be recovered after the watermarking process. Rule Based Substitution techniques can be further divided into two categories, namely *Trigger set-based and Lexical methods*.

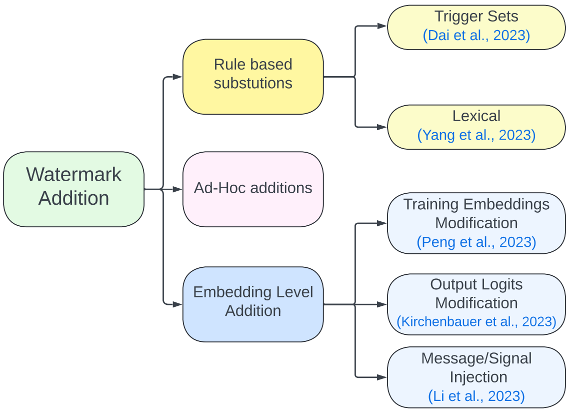

Watermark Addition
Rule based substutions
Ad-Hoc additions
Embedding Level Addition
Trigger Sets (Dai et al., 2023)
Lexical (Yang et al., 2023)
Training Embeddings Modification (Peng et al., 2023)
Output Logits Modification (Kirchenbauer et al., 2023)
Message/Signal Injection (Li et al., 2023)

Figure 2: Sub-categorization of various Watermark Additions.

Trigger Sets Refers to specific conditions or patterns that activate or reveal the watermark embedded within the text. Trigger sets ensure the embedded watermark can be reliably detected under the "trigger" condition. Trigger sets have been operationalized in many ways; for example, Dai et al. (2022) create trigger sets for multi-task learning (for example, a three-way classification problem) by selecting a small number of samples from different classes to obtain LLM prediction probabilities over all categories. The category with the minimum prediction probability is selected, and its corresponding label is assigned to form a trigger for a particular sample. Similarly, Liu et al. (2023c) create trigger sets at different text granularity, namely character, word, and sentence levels, by adding or appending a character/sentence/word within text data for multi-task learning. Other types of trigger sets include word-level (Peng et al., 2023) and style-level(Tang et al., 2023) triggers. Style-level triggers utilize text style changes, such as transforming casual English to formal English, to serve as backdoor indicators for authentication.

Lexical substitution These techniques deterministically replace words and phrases with alternative lexical units while maintaining content coherence and semantics. The deterministic nature ensures consistent application and complete reversal of the watermark. A straightforward operationalization of lexical replacement is based on semantic preservation, which includes synonym replacement us-

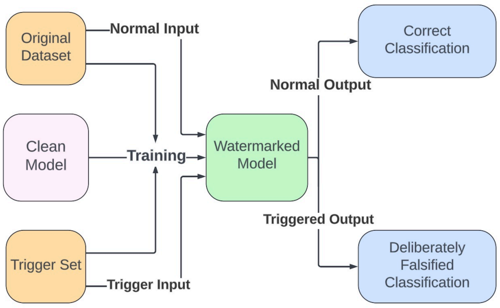


Figure 3: Operationalization of some Trigger-set based watermarks. Here, the original model is trained with the trigger set, which modifies the input to change the class of the output deliberately (Liu et al., 2023c) or change the output for the same input (Dai et al., 2022).

ing wordnet (He et al., 2022a; Yang et al., 2023b), spelling variant replacement between US and UK spellings (Topkara et al., 2006b), model-in-the-loop semantic similarity based search between candidate replacements and original sentence(Munyer and Zhong, 2023; Yang et al., 2022). In a nuanced domain like code generation, methods have looked at reordering operations and code formatting changes, which do not alter the functionality (Li et al., 2024)

2.2.2 Embedding-level Addition

Watermarking techniques can be distinguished based on *how* the watermarks are embedded. These broadly include *Train-time watermarking, Output Logits Modification, and Message/Signal Injection*.

Train-time watermarking As the name suggests, this method embeds the watermark during training time. Peng et al. (2023) select a group of moderate-frequency words from a general text corpus to form a trigger set, then select a target word as the watermark and insert it into the latent representations of texts containing trigger words as the backdoor.

Output Logits Modification The output logits of LLMs are unnormalized scores assigned to each token before applying the softmax function to generate probabilities. These probabilities reflect the model's confidence in predicting tokens. Logits determine the model's token prediction (where the highest logit determines the predicted token), training (comparing logits with actual labels to compute the loss), and interpreting model behavior by highlighting token importance. These methods modify the post-softmax distributions over the model's vocabulary.

A popular example of an Output Logit Modification watermarking is the use of green-red lists (Kirchenbauer et al., 2023; Lee et al., 2023; Zhao et al., 2023a; Takezawa et al., 2023; Fu et al., 2024; Ren et al., 2023; Wu et al., 2023; Chen et al., 2023), methods typically vary in the choice of high/low entropy tokens to add to the green list, size in the watermark (number of bits), injection of complex vs soft watermark, discarding low probability tokens, ensuring semantically similar words are distributed across green and red lists.

Apart from the techniques above, other methods involve injecting secret signals into the probability vector of the decoding steps for each target token (Zhao et al., 2023b). Liu et al. (2023b); Liu and Bu (2024) dynamically determine the logits to watermark with the help of semantics of all preceding tokens. Specifically, Liu et al. (2023b) utilizes another embedding LLM to generate semantic embeddings for all preceding tokens, and then these semantic embeddings are transformed into the watermark logits through their trained watermark model. Building from the idea of secret signals, Fairoze et al. (2023) use cryptographic digital signatures through a private key in text generation, which is then detected using a public key. Similarly, research also explores embedding multi-bit information into output logits (Qu et al., 2024; Wang et al., 2023; Guan et al., 2024).

Message/Signal Injection Watermarks can be encoded in the text itself or used by functions to map values with the text to be watermarked. These procedures involve the injection of messages, signals, or bit strings in the latent space of the text created by the encoders(Wang et al., 2023; Guan et al., 2024). For example, Li et al. (2023) tasks the representations of the abstract syntax tree (AST) tokens as input to predict modified variable names with encoded bit strings and Yang et al. (2023a); Li et al. (2023) encode identifier bit strings into the source code, without affecting the usage and semantics of the code. They perform transformations on an AST-based intermediate representation that enables unified transformations across different programming languages involving the changes in the expression, statement, and block attributes. Zhang et al. (2023) use linear combinations within this latent space to add a simple message to the embedded text. The decoder then converts it back into plain text with minor modifications resulting from the added message. A similar process is implemented to encode bit strings containing information like user ID and generation date (Qu et al., 2024). Zhou et al. (2024) injects coarse-grained and fine-grained signals (signatures) into the text during generation. The coarse-grained level utilizes statistical signals to detect watermark presence, while the fine-grained level embeds content-dependent signature bits for verifying content integrity.

2.2.3 Ad-Hoc Addition

Unlike popular watermarking methods like rule-based substitutions, which have strict, global definitions for modifications within a sentence (like synonym/spelling replacement and triggers), the ad-hoc addition methods use task-specific local guidelines for changes to the sentence structure. We bucket these methods into *Ad-Hoc addition methods* and list a few relevant methods.

First, Por et al. (2012); Sato et al. (2023) embed watermarks by inserting Unicode spaces in the text. Sato et al. (2023) introduce three methods: *WhiteMark* replaces whitespace with alternate Unicode spaces (e.g., U+0020 to U+2004), *VariantMark* uses Unicode variation selectors to embed messages in Chinese, Japanese, and Korean texts by substituting characters with variants, and *PrintMark* alters text appearance for printed media through ligatures, varied spaces, and character variants. Another work introduces three unique syntax transformations for message encoding— Adjunct Movement, Clefting, and Passivization (Atallah et al., 2001). For instance, Adjunct Movement involves relocating adjuncts within a sentence, as demonstrated by the variability in positioning the word 'quickly' in "She quickly finished her homework." Clefting highlights a specific clause, typically the subject, such as transforming "The chef cooked a delicious meal" into "It was the chef who cooked a delicious meal" to emphasize 'the chef.' Passivization, however, changes active sentences with transitive verbs into passive voice, transforming "The teacher graded the exams" into "The exams were graded by the teacher." Sun et al. (2023) apply semantic-preserving code transformations by modifying operators.

Overlapping categories Some papers span multiple categories within the taxonomy. For example, Zhao et al. (2023a) and Sato et al. (2023) address both Text Quality and Model Ownership Verification, preserving readability while ensuring ownership with detectable markers. Similarly, the work by Yang et al. (2023a) is listed under both message injection and embedding-level addition, as it involves injecting watermarks as messages within the embedding space. Similarly, research presented by Peng et al. (2023) falls under both embedding-level additions and trigger-set because it uses modified embeddings activated by specific triggers to enhance watermark detectability. These dual listings reflect the flexibility provided by these techniques, addressing overlapping goals. In contrast to prior surveys with limited focus areas, we believe maintaining separate but overlapping categories helps clarify distinct objectives and evaluation criteria across multiple dimensions, ensuring comprehensive coverage of the taxonomy.

2.3 Evaluation

A wide variety of datasets have been used to evaluate the performance of watermarking approaches, limiting our ability to extract generalized conclusions about their performance. Different benchmarks focus on selected downstream tasks to validate watermarking capabilities, and we provide a detailed breakdown of the datasets utilized in Table 3. We observe many evaluation datasets focusing on text completion and post-watermarking text similarity tasks. The downstream task descriptions are provided below.

Downstream Task descriptions

Text Completion Task This task involves giving the LLM a portion of text from the dataset as a prompt and then asking it to complete the text. The generated completion is then compared with the human completion or the portion of the dataset not provided as the prompt.

Post-watermark text similarity analysis In this task, given an initial text X, watermarking is applied to X to produce a modified text X'. An example could be a rule-based substitution with synonyms or spelling replacements. The comparison is then made between X and X', with X and X' based on distinctions in length, semantics, and other linguistic features.

Other Downstream Tasks For these tasks, given the same initial prompt X, the LLM's generated response Y (before watermarking) is compared with the response Y' (after watermarking).

Table 3: Datasets used in the evaluation of watermarking techniques. **Bold** indicates the most used dataset(s) for a particular downstream NLP task and the respective works using the dataset.

Downstream Task	Dataset Name	Papers
Text Completion	**Colossal Clean Crawled Corpus (C4) (Raffel et al., 2020)**, Dbpedia Class (Auer et al., 2007), WikiText-2 (Merity et al., 2016)	**Kirchenbauer et al. (2023), Kuditipudi et al. (2023), Liu et al. (2023a), Munyer and Zhong (2023), Yoo et al. (2023b), Liu et al. (2023b), Fairoze et al. (2023), Ren et al. (2023), Hou et al. (2023), Qu et al. (2024), Wang et al. (2023), Liu and Bu (2024), Chen et al. (2023), Mao et al. (2024), Chang et al. (2024)**, Zhou et al. (2024)
Post-watermark text similarity analysis	**WikiText-2, Workshop on Statistical Machine Translation (WMT14) (Bojar et al., 2014)**, Internet Movie Database (IMDb) (Maas et al., 2011), AgNews (Zhang et al., 2015), Dracula, Pride and Prejudice, Wuthering Heights (Gerlach and Font-Clos, 2020), CNN/Daily Mail (Nallapati et al., 2016), Human ChatGPT Comparison Corpus (HC3) (Guo et al., 2023), C4, Reuters Corpus (Lewis et al., 2004), ChatGPT Abstract (Nicolai Thorer Sivesind, 2023), Human Abstract (Nicolai Thorer Sivesind, 2023)	**Yang et al. (2022), He et al. (2022a), He et al. (2022b), Yoo et al. (2023a), Sato et al. (2023), Zhang et al. (2023)**, Yang et al. (2023b), Topkara et al. (2006a)
Machine Translation	**WMT14**, IWSTL14 (Cettolo et al., 2014)	**Zhao et al. (2023b), Wu et al. (2023), Hu et al. (2023), Takezawa et al. (2023)**
Text Summarisation	**CNN/Daily Mail**, Extreme Summarization (XSUM) (Narayan et al., 2018), Data Record to Text Generation (DART) (Nan et al., 2021) , WebNLG (Gardent et al., 2017)	**Fu et al. (2024), Wu et al. (2023), Hu et al. (2023)**
Code Generation	**CodeSearchNet (CSN) (Husain et al., 2019)**, HUMANEVAL (Chen et al., 2021), Mostly Basic Python Programming (MBPP), MBXP (Athiwaratkun et al., 2023), DS-1000 (Lai et al., 2023), APPS (Hendrycks et al., 2021)	**Li et al. (2023), Yang et al. (2023a), Guan et al. (2024)**, Lee et al. (2023), Li et al. (2024), Mao et al. (2024)
Question Answering	**OpenGen (Krishna et al., 2024), Long Form Question Answering (LFQA) (Krishna et al., 2024)**, TruthfulQA (Lin et al., 2021)	**Zhao et al. (2023a), Yoo et al. (2023b), Qu et al. (2024), Zhou et al. (2024), Chang et al. (2024)**, Chen et al. (2023)
Story Generation	**ROCstories (Mostafazadeh et al., 2016)**	**Zhao et al. (2023b)**
Text Classification	**Stanford Sentiment Treebank (SST) (Socher et al., 2013), AgNews, Microsoft News Dataset (MIND) (Wu et al., 2020), Enron Spam (Metsis et al., 2006)**	**Peng et al. (2023)**

2.4 Adversarial attacks on watermarking techniques

Malicious and adversarial actors seek to misuse LLM technology and bypass watermarks to avoid being distinguished from rightful owners. To promote research into protecting intellectual property rights, we extend suggestions from Kirchenbauer et al. (2023) to describe de-watermarking methods, i.e., adversarial attacks on text watermarking, into three categories:

1. **Text insertion attacks** involve adding additional tokens or text segments to the original output of a watermarked LLM generation. For example, on watermarking methods with green-red lists (Kirchenbauer et al., 2023; Zhao et al., 2023b; Takezawa et al., 2023), an attacker could add additional tokens from the red list, leading to the obfuscation of the watermarking method. Another variant of text insertion attacks includes copy-paste attacks (Qu et al., 2024), where adversaries insert copied text from external sources into the watermarked output. This approach can dilute the effect of the watermark by embedding non-watermarked content within the watermarked text, further complicating watermark detection and attribution.

2. **Text deletion attacks** involve the removal of tokens or text segments from the original watermarked output of an LLM and modifying the rest of the tokens to fit the output. Returning to the example of green-red list methodologies, this means removing some of the green list tokens from the output and modifying the red list tokens in the output (Kirchenbauer et al., 2023). These techniques often require knowledge of the vocabularies belonging to each of the two lists in green-red lists.

3. **Text substitution attacks** entail replacing specific tokens or text segments in the watermarked output while preserving its overall meaning. Attackers perform tokenization attacks by para-

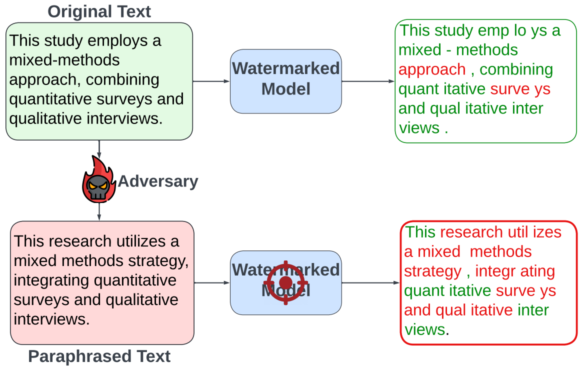

Original Text
This study employs a mixed-methods approach, combining quantitative surveys and qualitative interviews.
Watermarked Model
This study emp lo ys a mixed - methods approach , combining quant itative surve ys and qual itative inter views .
Adversary
This research utilizes a mixed methods strategy, integrating quantitative surveys and qualitative interviews.
Watermarked Model
This research util izes a mixed methods strategy , integr ating quant itative surve ys and qual itative inter views.
Paraphrased Text

Figure 4: An example of an adversary performing a de-watermarking attack on a green-red list-based watermarking technique. The original partitioning contains a higher proportion of green tokens than the partitioning after adversarial paraphrasing.

phrasing text (Ren et al., 2023), misspelling words, or replacing characters like newline (\n); increasing red list tokens, and evading green-red list watermarking (Kirchenbauer et al., 2023). These also include Homoglyph attacks: attacks that exploit Unicode characters that look similar but have different IDs, leading to variation from expected tokenization (e.g., "Lighthouse" becomes nine tokens with Cyrillic characters). Generative attacks leverage LLMs' context learning to manipulate the output predictably, such as adding emojis after each token or replacing characters to disrupt watermark detection (Kirchenbauer et al., 2023).

3 Discussion and Open Challenge

Our taxonomy-driven categorization of the research space exposes the open challenges to watermarking and outlines "good to have" criteria while developing new techniques to protect intellectual property ownership. They are as follows:

Resilience to adversarial attacks One of the critical challenges in the field is the lack of *comprehensive* evaluation of techniques against a diverse range of de-watermarking attacks. While many researchers focus on developing robust techniques, there is often insufficient emphasis on systematic red-teaming of these methods against multiple attacking scenarios.

Standardization of evaluation benchmarks There is a need for standardized benchmarks and evaluation metrics to ensure fair and consistent comparison between different watermarking techniques. Table 3 shows how evaluation datasets differ in the literature for the same downstream task, reflecting this necessity.

Impact on LLM output factuality Watermarks modify the model output distributions; techniques that are robust to de-watermarking often have greater variations in watermarked outputs compared to clean outputs, leading to a potential trade-off between de-watermarking and LLM factuality. Despite this potential trade-off, there is a lack of analysis on how watermarking techniques affect the output inaccuracies or hallucinations. After training or fine-tuning LLMs with specific watermarking techniques, there is often insufficient examination of whether these methods introduce or exacerbate inaccuracies. We advocate for factuality evaluations post-watermarking.

Enhanced Interpretability Drawing upon security and privacy literature (Kumar et al., 2024), we ask the community to establish privacy norms for LLM watermarking. We envision this to be similar to model cards, which describe the degree of security provided by particular methods against malicious actors.

Human-centered watermarking We urge the community to work on the human perception of LLMs when interacting with different safety principles. User perception of LLMs may change with differences in output distributions. Furthermore, safety practices may enable AI acceptance and adoption among the masses.

4 Conclusion

In this paper, we analyze representative literature to provide a comprehensive taxonomy for digital watermarking techniques for both LLM-generated and human-written text. The taxonomy categorizes watermarking techniques using four primary categories, namely - intention of the method, data used for evaluation, watermark addition, and adversarial attacks.

We identify and cluster existing watermarking methods, highlighting key open challenges and research gaps in the field. For every watermarking method, *we advocate for establishing stronger evaluation paradigms: standardized datasets, resilience to adversarial attacks, impact on model utility and output actuality, enhanced interpretability, and human perception change upon the use of these such techniques.* We envision this research as a reference for policymakers, safety practitioners, and end users, facilitating the adoption of robust digital watermarking practices and promoting responsible AI use.

5 Limitations

Limitations to our work are as follows: (1) We do not include detailed insights into metrics for success rate (accuracy of detecting watermarked texts), text quality (perplexity and semantics), NLP task-specific evaluation, and robustness (detectability of watermarks after removal attacks). However, we briefly describe the two types of metrics into watermarking success rates (intrinsic quality of watermarking) and model utility/ performance 8.2. (2) Given the scope of this paper, we do not demonstrate the mathematical analysis of different watermarking techniques. We urge readers to refer to the original papers for the same (3) We do not cover all different task deployment scenarios for the watermarking techniques discussed. (4) Readers may perceive similarity between sections 2.1.1 (text quality) and 2.3 (evaluation), however, we wish to highlight the following distinction - in section 2.3 (evaluation), our focus is on the datasets that different papers use to evaluate their watermarking techniques whereas in section 2.1.1 (text quality), we look at how some papers have similar intentions or end goals of watermarking. While similar intentions often include many of the same downstream tasks as mentioned in sections 2.3 (Text Completion Task, Post-watermark text similarity analysis, Other Downstream Tasks), and 2.1.1, we advocate for the standardization of these tasks for definitive apples to apples comparison between techniques. (5) While we advocate for a standardized evaluation, we do not propose a framework for evaluating the effectiveness of watermarking techniques. (6) We focus primarily on watermarking and do not delve into the broader relationships between watermarking and adjunct fields, for example, steganography. However, additional information on these topics can be found in the appendix section 8.1.

6 Ethical Considerations

This paper reviews the challenges and opportunities of watermarking techniques in LLMs. Our work has many potential societal consequences, none of which must be specifically highlighted here. There are no major risks associated with conducting this review.

7 Acknowledgments

This work was partly supported by U.S. NSF awards #1934782 and #2114824.

References

Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 121–140. IEEE.

Mikhail J. Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding*, pages 185–200, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ben Athiwaratkun, Sanjay Krishna Gouda, Zijian Wang, Xiaopeng Li, Yuchen Tian, Ming Tan, Wasi Uddin Ahmad, Shiqi Wang, Qing Sun, Mingyue Shang, Sujan Kumar Gonugondla, Hantian Ding, Varun Kumar, Nathan Fulton, Arash Farahani, Siddhartha Jain, Robert Giaquinto, Haifeng Qian, Murali Krishna Ramanathan, Ramesh Nallapati, Baishakhi Ray, Parminder Bhatia, Sudipta Sengupta, Dan Roth, and Bing Xiang. 2023. Multi-lingual evaluation of code generation models. In *The Eleventh International Conference on Learning Representations*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

J.T. Brassil, S. Low, N.F. Maxemchuk, and L. O'Gorman. 1995. Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communications*, 13(8):1495–1504.

Mauro Cettolo, Jan Niehues, Sebastian Stuker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign. Ed.: M. Federico, S. Stuker, F. Yvon*, page 2,Äì17. Association for Computational Linguistics (ACL).

Yapei Chang, Kalpesh Krishna, Amir Houmansadr, John Wieting, and Mohit Iyyer. 2024. Postmark: A robust blackbox watermark for large language models. *arXiv preprint arXiv:2406.14517*.

Liang Chen, Yatao Bian, Yang Deng, Shuaiyi Li, Bingzhe Wu, Peilin Zhao, and Kam-fai Wong. 2023. X-mark: Towards lossless watermarking through lexical redundancy. *arXiv preprint arXiv:2311.09832*.

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

Long Dai, Jiarong Mao, Xuefeng Fan, and Xiaoyi Zhou. 2022. Deephider: A covert nlp watermarking framework based on multi-task learning. *arXiv preprint arXiv:2208.04676*.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. 2023. Publicly detectable watermarking for language models. Cryptology ePrint Archive, Paper 2023/1661. https://eprint.iacr.org/2023/1661.

Yu Fu, Deyi Xiong, and Yue Dong. 2024. Watermarking conditional text generation for ai detection: Unveiling challenges and a semantic-aware watermark remedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18003–18011.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Batu Guan, Yao Wan, Zhangqian Bi, Zheng Wang, Hongyu Zhang, Yulei Sui, Pan Zhou, and Lichao Sun. 2024. Codeip: A grammar-guided multi-bit watermark for large language models of code. *arXiv preprint arXiv:2404.15639*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. 2022a. Protecting intellectual property of language generation apis with lexical watermark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10758–10766.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. 2022b. Cater: Intellectual property protection on text generation apis via conditional watermarks. *Advances in Neural Information Processing Systems*, 35:5431–5445.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. 2021. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*.

Duy C Hoang, Hung TQ Le, Rui Chu, Ping Li, Weijie Zhao, Yingjie Lao, and Khoa D Doan. 2024. Less is more: Sparse watermarking in llms with enhanced text quality. *arXiv preprint arXiv:2407.13803*.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. Semstamp: A semantic watermark with paraphrastic robustness for text generation. *Preprint*, arXiv:2310.03991.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased watermark for large language models. *arXiv preprint arXiv:2310.10669*.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Codesearchnet challenge: Evaluating the state of semantic code search. *arXiv preprint arXiv:1909.09436*.

Zunera Jalil and Anwar M Mirza. 2009. A review of digital watermarking techniques for text documents. In *2009 International Conference on Information and Multimedia Technology*, pages 230–234. IEEE.

Nurul Shamimi Kamaruddin, Amirrudin Kamsin, Lip Yee Por, and Hameedur Rahman. 2018. A review of text watermarking: theory, methods, and applications. *IEEE Access*, 6:8011–8028.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*.

Ashutosh Kumar, Sagarika Singh, Shiv Vignesh Murty, and Swathy Ragupathy. 2024. The ethics of interaction: Mitigating security threats in llms. *arXiv preprint arXiv:2401.12273*.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345. PMLR.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2023. Who wrote this code? watermarking for code generation. *arXiv preprint arXiv:2305.15060*.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Boquan Li, Mengdi Zhang, Peixin Zhang, Jun Sun, and Xingmei Wang. 2024. Resilient watermarking for llm-generated codes. *arXiv preprint arXiv:2402.07518*.

Wei Li, Borui Yang, Yujie Sun, Suyu Chen, Ziyun Song, Liyao Xiang, Xinbing Wang, and Chenghu Zhou. 2023. Towards tracing code provenance with code watermarking. *arXiv preprint arXiv:2305.12461*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S Yu. 2023a. A private watermark for large language models. *arXiv preprint arXiv:2307.16230*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023b. A semantic invariant robust watermark for large language models. *arXiv preprint arXiv:2310.06356*.

Yepeng Liu and Yuheng Bu. 2024. Adaptive text watermark for large language models. *arXiv preprint arXiv:2401.13927*.

Yixin Liu, Hongsheng Hu, Xuyun Zhang, and Lichao Sun. 2023c. Watermarking text data on large language models for dataset copyright protection. *arXiv preprint arXiv:2305.13257*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, et al. 2023. Paloma: A benchmark for evaluating language model fit. *arXiv preprint arXiv:2312.10523*.

Minjia Mao, Dongjun Wei, Zeyu Chen, Xiao Fang, and Michael Chau. 2024. A watermark for low-entropy and unbiased generation in large language models. *arXiv preprint arXiv:2405.14604*.

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

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. 2006. Spam filtering with naive bayes - which naive bayes?

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*.

Travis Munyer and Xin Zhong. 2023. Deeptextmark: Deep learning based text watermarking for detection of large language model generated text. *arXiv preprint arXiv:2305.05773*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

The New York Times Company. 2023. The new york times company v. microsoft corporation, openai, inc., openai lp, openai gp, llc, openai, llc, openai opco llc, openai global llc, oai corporation, llc, and openai holdings, llc.

Nicolai Thorer Sivesind. 2023. Chatgpt-generated-abstracts.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. 2023. Are you copying my model? protecting the copyright of large language models for eaas via backdoor watermark. *arXiv preprint arXiv:2305.10036*.

Lip Yee Por, KokSheik Wong, and Kok Onn Chee. 2012. Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, 85(5):1075–1082.

Wenjie Qu, Dong Yin, Zixin He, Wei Zou, Tianyang Tao, Jinyuan Jia, and Jiaheng Zhang. 2024. Provably robust multi-bit watermarking for ai-generated text via error correction code. *arXiv preprint arXiv:2401.16820*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2023. A robust semantics-based watermark for large language model against paraphrasing. *arXiv preprint arXiv:2311.08721*.

Yubing Ren, Ping Guo, Yanan Cao, and Wei Ma. 2024. Subtle signatures, strong shields: Advancing robust and imperceptible watermarking in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5508–5519, Bangkok, Thailand. Association for Computational Linguistics.

Ryoma Sato, Yuki Takezawa, Han Bao, Kenta Niwa, and Makoto Yamada. 2023. Embarrassingly simple text watermarks. *arXiv preprint arXiv:2310.08920*.

Shuo Shao, Yiming Li, Hongwei Yao, Yiling He, Zhan Qin, and Kui Ren. 2024. Explanation as a watermark: Towards harmless and multi-bit model ownership verification via watermarking feature attribution. *arXiv preprint arXiv:2405.04825*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Zhensu Sun, Xiaoning Du, Fu Song, and Li Li. 2023. Codemark: Imperceptible watermarking for code datasets against neural code completion models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1561–1572.

Yuki Takezawa, Ryoma Sato, Han Bao, Kenta Niwa, and Makoto Yamada. 2023. Necessary and sufficient watermark for large language models. *arXiv preprint arXiv:2310.00833*.

Ruixiang Tang, Qizhang Feng, Ninghao Liu, Fan Yang, and Xia Hu. 2023. Did you train on my dataset? towards public dataset protection with cleanlabel backdoor watermarking. *ACM SIGKDD Explorations Newsletter*, 25(1):43–53.

Mercan Topkara, Umut Topkara, and Mikhail J. Atallah. 2006a. Words are not enough: sentence level natural language watermarking. In *Workshop on Medical Cyber-Physical Systems*.

Umut Topkara, Mercan Topkara, and Mikhail J. Atallah. 2006b. The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the 8th Workshop on Multimedia and Security*, MM and Sec '06, page 164–174, New York, NY, USA. Association for Computing Machinery.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Towards codable text watermarking for large language models. *arXiv preprint arXiv:2307.15992*.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2023. Dipmark: A stealthy, efficient and resilient watermark for large language models. *arXiv preprint arXiv:2310.07710*.

Borui Yang, Wei Li, Liyao Xiang, and Bo Li. 2023a. Towards code watermarking with dual-channel transformations. *arXiv preprint arXiv:2309.00860*.

Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. 2023b. Watermarking text generated by black-box language models. *arXiv preprint arXiv:2305.08883*.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. 2022. Tracing text provenance via context-aware lexical substitution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11613–11621.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. 2023a. Robust natural language watermarking through invariant features. *arXiv preprint arXiv:2305.01904*.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2023b. Advancing beyond identification: Multi-bit watermark for language models. *arXiv preprint arXiv:2308.00221*.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. 2023. Remarkllm: A robust and efficient watermarking framework for generative large language models. *arXiv preprint arXiv:2310.12362*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023a. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023b. Protecting language generation models via invisible watermarking. In *International Conference on Machine Learning*, pages 42187–42199. PMLR.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Tong Zhou, Xuandong Zhao, Xiaolin Xu, and Shaolei Ren. 2024. Bileve: Securing text provenance in large language models against spoofing with bi-level signature. *arXiv preprint arXiv:2406.01946*.

8 Appendix

8.1 Watermarking and Steganography

The concepts of text watermarking and steganography are often discussed together due to their shared goal of altering text to convey additional information. However, they serve distinct purposes and have different applications.

Steganography is the practice of concealing messages within non-secret text or data, making the hidden message indiscernible to unintended recipients. The primary objective of steganography is to ensure that the existence of the concealed message remains undetected. This is often achieved through subtle alterations to the host text that are imperceptible to the human eye or standard detection techniques. Steganography is widely used in fields such as secure communications and digital rights management.

Text Watermarking, on the other hand, involves embedding identifiable information or markers within text to establish ownership, verify authenticity, or detect unauthorized use. Unlike steganography, text watermarking does not necessarily seek to conceal the existence of the watermark but rather focuses on making it robust and detectable under various conditions. The key objectives of text watermarking include ensuring that the watermarked content remains readable and maintaining the natural quality of the text, while providing a reliable means of verification or ownership assertion.

While both steganography and text watermarking involve altering text, they differ significantly in their underlying intentions and techniques. Steganography prioritizes secrecy and concealment, whereas text watermarking emphasizes detection, ownership, and authenticity. Understanding these differences is crucial when selecting the appropriate technique for specific applications.

8.2 Some metrics used for evaluation

To evaluate the effectiveness of watermarking techniques, we consider two categories of metrics: **detection metrics** and **downstream task evaluation metrics**.

For watermark detection, we evaluate the following metrics:

1. Watermark Success Rate: The percentage of cases where the embedded watermark is successfully detected, indicating the reliability of the watermarking technique.

2. Area Under the Curve (AUC): Represents the model's ability to distinguish between watermarked and non-watermarked texts, derived from the ROC (Receiver Operating Characteristic) curve. A higher AUC indicates better performance in correctly classifying watermarked and non-watermarked text.

3. False Positive Rate (FPR): Measures the rate at which non-watermarked texts are incorrectly identified as watermarked, which is crucial for assessing the precision of watermark detection methods.

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}} \tag{1}$$

To ensure that watermarking does not adversely affect the utility of text in various NLP tasks, we evaluate the following metrics:

1. F1 Score: Used for classification tasks, such as sentiment analysis or spam detection. It is the harmonic mean of precision and recall, evaluating the balance between false positives and false negatives.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

2. Exact Match (EM): Used primarily in question answering tasks, this metric measures the percentage of predictions that exactly match the ground truth answers.

$$\text{EM} = \frac{\text{Number of Exact Matches}}{\text{Total Number of Predictions}}$$

3. Perplexity (PPL): Commonly used in language modeling and text generation tasks, perplexity measures the fluency of generated text. Lower perplexity indicates that the text is more coherent and closer to natural language usage.

To mathematically define perplexity for a sequence of words $W = (w_1, w_2, \ldots, w_N)$, it can be expressed as:

$$PP(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \tag{3}$$

where $P(w_1 w_2 \ldots w_N)$ is the probability of the word sequence W according to the model.