LexiMark: Robust Watermarking via Lexical Substitutions to Enhance Membership Verification of an LLM's Textual Training Data

Eyal German, Sagiv Antebi, Edan Habler, Asaf Shabtai, Yuval Elovici
Department of Software and Information Systems Engineering,
Ben-Gurion University of the Negev, Israel
{germane, sagivan, habler}@post.bgu.ac.il, {shabtaia, elovici}@bgu.ac.il

***Abstract*—Large language models (LLMs) can be trained or fine-tuned on data obtained without the owner's consent. Verifying whether a specific LLM was trained on particular data instances or an entire dataset is extremely challenging. Dataset watermarking addresses this by embedding identifiable modifications in training data to detect unauthorized use. However, existing methods often lack stealth, making them relatively easy to detect and remove. In light of these limitations, we propose LexiMark, a novel watermarking technique designed for text and documents, which embeds synonym substitutions for carefully selected high-entropy words. Our method aims to enhance an LLM's memorization capabilities on the watermarked text, without altering the semantic integrity of the text. As a result, the watermark is difficult to detect, blending seamlessly into the text with no visible markers, and is resistant to removal due to its subtle, contextually appropriate substitutions that evade automated and manual detection. We evaluated our method using baseline datasets from recent studies and seven open-source models: LLaMA-1 7B, LLaMA-3 8B, Mistral 7B, Pythia 6.9B, as well as three smaller variants from the Pythia family—160M, 410M, and 1B. Our evaluation spans multiple training settings, including continued pretraining and fine-tuning scenarios. The results demonstrate significant improvements in AUROC scores compared to existing methods, underscoring our method's effectiveness in reliably verifying whether unauthorized watermarked data was used in LLM training.**

I. Introduction

"Data is the new gold" [1] – In the context of artificial intelligence (AI), data serves as the essential fuel driving the performance and innovation of AI systems. High-quality data enables models to learn complex patterns, identify subtle relationships, and make predictions that guide decision-making in diverse fields. Modern AI systems, including large language models (LLMs), require massive amounts of high-quality training data to achieve their impressive performance, which is both expensive and difficult to acquire.

The emergence of LLMs has revolutionized natural language processing (NLP) by enabling state-of-the-art performance in a wide range of NLP tasks, including machine translation, text summarization and question answering [2], [3]. The unprecedented capabilities of LLMs, such as GPT [4] and Google's Gemini [5], [6] arise from their training on extensive, diverse datasets. This training enables them to grasp complex linguistic and semantic patterns, allowing for sophisticated language processing and effective generalization across different contexts.

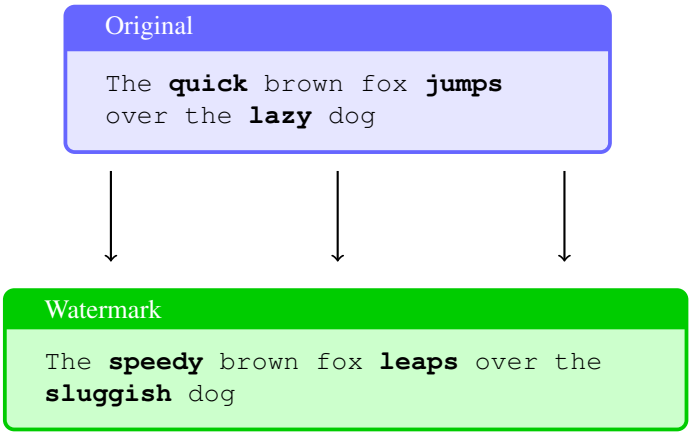

Fig. 1. An illustration of our synonym replacement method, where *K=3* words in the original sentence are substituted with higher-entropy synonyms. In this example, "quick," "jumps," and "lazy" are replaced with "speedy," "leaps," and "sluggish" to create the watermarked version.

LLMs are typically trained in two stages: pretraining, where general language patterns are captured and learned from vast datasets, and fine-tuning, which adapts the pretrained model to specific tasks using smaller, specialized datasets. For these models to reach their full potential and consistently achieve high performance across tasks, access to high-quality training data is essential, as it enables them to accurately model complex linguistic patterns and nuances.

Often, the demand for suitable datasets pushes the boundaries of ethical data sourcing and results in the collection of publicly available data, obtained via scraping, along with proprietary or licensed information. This approach introduces privacy, security, and legal risks, especially when sensitive information, such as personally identifiable information (PII), copyrighted content, or proprietary data, is improperly used to train the model. In some cases, the drive to enhance model performance may even tempt LLM builders to use unauthorized or illegally obtained datasets, further compromising ethical standards and user trust.

Awareness regarding these privacy and ethical issues has increased as a result of legal conflicts and the lack of trans-

parency regarding the data collection process [7], [8]. The lawsuit between *The New York Times* and OpenAI [9], as well as other lawsuits [10], [11], highlights the critical need for mechanisms aimed at detecting such privacy and intellectual property violations, and more specifically, identifying the data used to train LLMs [12].

The risk of data extraction and leakage is compounded when LLMs are fine-tuned, since the fine-tuning process involves additional training on specialized datasets that may contain sensitive information [13]. Memorization—where the model retains exact phrases, sentences, or even entire passages from the training data—can become more pronounced during fine-tuning, especially with small or domain-specific datasets. This memorization increases the likelihood of data leakage, as sensitive information embedded in the model could be inadvertently reproduced in responses, posing privacy and security risks [14], [15].

Larger models are even more prone to memorizing the training data [16], a tendency that can be exploited through data extraction attacks. The main risk is if attackers exploit the model to extract or infer private information, especially if the training data contains sensitive information such as PII or copyrighted content [17]. This underscores the importance of developing robust mechanisms to detect whether unauthorized data has been used in a model's training process.

Such detection can be challenging, and several methods have been proposed for detecting the presence of unauthorized data in LLMs' training data. Methods such as membership inference attacks (MIAs) are designed to determine whether a specific text was part of a model's training dataset [18]–[20]. To determine whether a specific piece of data was included in the training set, MIAs exploit the differences in a model's behavior when it processes seen and unseen data. The underlying assumption is that an LLM will perform differently on queries that are related to seen and unseen data (e.g., exhibiting higher prediction confidence or greater loss reduction).

Despite MIAs' effective performance, they have several limitations [21]. First, their performance in terms of common metrics, such as the *area under the receiver operating characteristic curve (AUROC)* and the *true positive rate (TPR) at a fixed low false positive rate (FPR)* [19], tends to worsen as the training set size increases, often rendering it close to random [22]. This is due to the trade-off between generalization and memorization in LLMs: as the model is trained on more data, it will generalize better, while increasing the number of model parameters increases the model's tendency to memorize the training data [23], [24]. Additionally, MIAs show inconsistent performance across models and datasets and are prone to detecting distribution shifts rather than performing true membership inference [12], [25].

Given the challenges and limitations associated with traditional MIAs, there is a growing need for more reliable methods for detecting the unauthorized use of data in training LLMs. This has led to the development of watermarking techniques, which embed unique patterns into the training data, making it easier to track and detect the use of specific datasets in a model's training process [26].

In this context, a watermark refers to a deliberate modification of the input data that subtly alters its structure without compromising the data's semantic meaning [27]. These modifications allow researchers to identify whether a particular dataset has been used in training an LLM by examining how the model behaves when processing watermarked data. Watermarking techniques can be highly effective in detecting data misuse and preventing privacy violations, as they provide an additional layer of security by embedding detectable patterns within the data itself.

Several approaches have been proposed for embedding watermarks into textual data. One common method involves altering the encoding of characters, such as by using visually similar Unicode characters, while in another method suggests inserting random sequences into the text [28]. While these changes are often subtle enough to be imperceptible to human readers, they create distinct patterns that can later be detected. While such techniques can help infer whether particular datasets were part of the training set, they have limited robustness, as they are relatively easy to detect and remove.

To address the limitations of existing watermarking methods, we introduce LexiMark, a novel and robust watermarking method for textual data that may be used on the training data of LLMs. LexiMark is inspired by MIA methods that exploit the model's behavior when handling high-entropy tokens that have a greater likelihood of being related to the method's inference ability [29]–[32]. These approaches demonstrated that focusing on high-entropy or high-probability tokens can improve the accuracy of MIAs by capitalizing on the differential treatment models give to such inputs.

Our method extends this concept by identifying the words in a sentence with the highest entropy and replacing them with higher-entropy synonyms, thereby embedding our watermark in the training data of the LLM. This ensures that the semantic meaning of the text is preserved while subtly embedding a watermark that can later be detected through an MIA. By targeting high-entropy words, which are naturally more unpredictable and challenging for LLMs to predict, our watermark method enhances the likelihood that these words will be memorized by the LLM. Our method guarantees that the text remains readable and useful while embedding detectable patterns. In Figure 1, we present an example of how our method replaces high-entropy words with semantically similar synonyms with higher entropy. Our watermark embedding method begins by preprocessing the text, splitting it into sentences, and selecting the top-K high-entropy words (keywords) from each sentence. These keywords are then replaced with higher-entropy synonyms, ensuring that the original meaning is preserved, effectively embedding the watermark while maintaining the text's readability.

The underlying intuition is that LLMs are more likely to memorize high-entropy words, as these words introduce greater uncertainty in predictions. By enhancing the model's memorization of these watermarked words, our method strengthens the ability to verify whether a dataset was used for training. Our method, which employs MIAs for verification, effectively balances robustness, detectability, and readability,

making the watermark difficult to remove while maintaining the text's original meaning and usability.

We evaluated our watermarking method across diverse textual domains within the *The Pile* dataset [33], including medical texts, emails, legal documents, encyclopedic entries, and patent descriptions, as well as the *BookMIA* [32] dataset. We tested our method on seven open-sourced LLMs: Pythia-160M, 410M, 1B, and 6.9B [34], LLaMA-1 7B [35], LLaMA-3 8B [36], and Mistral-7B [37]. For the large models, we fine-tuned them on the watermarked data using the quantized low-rank adaptation (QLoRA) technique [38]. For the smaller Pythia models, we employed continued pretraining, also known as domain-adaptive pretraining (DAPT) [39], to evaluate the robustness and generality of our watermarking approach across different model scales and training paradigms.

The results demonstrate clear improvements in detecting textual data membership, with our approach consistently achieving higher AUROC scores compared to baseline techniques. The increase ranges from 2.5% to 25.7%, confirming the robustness of our detection approach. Our evaluation also examined dataset detection, revealing that our method requires fewer records to accurately determine whether a dataset was used in the training process. This makes our approach more efficient and sensitive in identifying pretraining sources. Without watermarking, detection typically requires around 40 samples to achieve a p-value of less than 0.05, while with our watermarking method, only six samples are needed to achieve this.

In addition to the improvements in membership detection, our semantic preservation checks, measured by cosine similarity [40] and BLEU scores [41], demonstrated near-complete retention of the original text's meaning, ensuring that our watermarking method maintains both high accuracy and text integrity across diverse datasets.

We also conducted a robustness evaluation, confirming that our method withstands minor textual modifications with minimal impact on detection results. This robustness to minor text changes further highlights our method's resilience, allowing for reliable detection even when slight alterations are introduced, thereby supporting the method's applicability in real-world settings where minor text variations are common. Furthermore, unlike other approaches, our watermarking method also remains undetectable in perplexity tests on the fine-tuned LLM, avoiding the performance decrease common in other methods that are easily spotted using perplexity checks. In addition, we examine the effectiveness of our method under post-training scenarios, such as instruction tuning, and find that watermark signals remain detectable even after the model undergoes further updates. To support reproducibility and facilitate future research, we provide our implementation, evaluation scripts, and data preparation tools at: https://github.com/eyalgerman/LexiMark.

The key contributions of this paper are summarized as follows:

- **A novel watermarking method for textual data:** We introduce a method that identifies high-entropy words in sentences and substitutes them with synonyms with higher entropy, thereby embedding a watermark without altering the semantic meaning of the data.
- **Improved detection using MIAs:** We enhance the effectiveness of existing MIA methods by embedding watermarks that increase the likelihood of data memorization during model training. This improves accuracy in detecting whether specific data was part of the training set.
- **Semantic preservation:** Our method demonstrates near-complete preservation of the original sentence's semantic meaning. We explore various synonym selection methods to optimize the semantic preservation of the watermarked text, ensuring minimal impact on the original meaning.
- **Robustness:** LexiMark is difficult to detect and remove due to its subtle substitutions, which blend seamlessly into the text and appear unwatermarked. We evaluate the robustness and detectability of our method in comparison to two baseline approaches, demonstrating its superior performance in maintaining watermark integrity.
- **Post-training resilience:** We further examine the watermark's persistence under post-training modifications, such as instruction tuning, and show that the watermark remains reliably detectable even after the model undergoes additional training phases.

In the remainder of this paper, we first review prior work on MIAs and data watermarking for LLMs in Section II. We then introduce LexiMark, our proposed watermarking method based on high-entropy lexical substitutions, detailing both the embedding and detection phases in Section III. Section IV describes our experimental setup, including the datasets, models, and evaluation protocol. In Section V, we present detection results across various LLMs and datasets. Section VI evaluates the semantic preservation of the watermarked text using cosine similarity and BLEU scores. In Section VII, we assess the robustness of LexiMark against synonym substitution, post-training, and removal attacks. Section VIII demonstrates how our method enables dataset-level membership detection using statistical inference. Finally, Section IX concludes the paper and outlines directions for future work.

II. Related work

LLMs leverage deep learning techniques to generate and understand natural language text. Common LLMs are built on the transformer architecture, which utilizes self-attention mechanisms to process words in relation to all other words in a sentence, enhancing the model's ability to understand context [42], [43].

LLMs are trained on vast text corpora, using a loss function aimed at predicting the next token in a sequence based on the preceding tokens. These models can also be fine-tuned for specific tasks, broadening their range of applications. However, despite their impressive capabilities, there are several challenges regarding their use, including data bias, privacy concerns, and the significant computational resources required to train them.

Training these models involves adjusting millions or even billions of parameters to minimize the difference between the

model's predictions and actual data. This extensive optimization enables LLMs to generate responses that are not only contextually relevant but also exhibit nuanced understanding, allowing them to produce high-quality, human-like text.

Research has increasingly focused on addressing data privacy concerns regarding LLMs, and particularly on vulnerabilities related to data leakage. One such vulnerability is the membership inference attack (MIA), where an attacker attempts to determine whether a specific data record was used to train a model [20]. MIAs exploit memorization in machine learning models, where the model behaves differently on training data than it does on data it has not seen [19], [23]. Given that LLMs tend to memorize certain parts of the training data that are rare or unique, high-entropy words are more likely to be memorized. This is a basic assumption of our watermarking method which substitutes words in the text with their higher entropy synonyms.

A. LLM Membership Inference Attacks

LLM MIAs are a subdomain of MIAs that focuses on detecting whether a specific text was used to train an LLM.

Perplexity is a metric used to evaluate how well a probability model predicts a sample, especially in the context of natural language processing (NLP). Perplexity is calculated as the exponentiation of the negative average log-likelihood per token, as described in the formula:

$$\text{Perplexity}(P) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(t_i|t_1,\ldots,t_{i-1})\right)$$

In NLP, perplexity captures the degree of 'uncertainty' a model has in predicting text. Lower perplexity indicates that the model is certain and familiar with the text, and therefore it predicts the sample more accurately. In contrast, higher perplexity suggests that the model is less certain and less familiar with the text, thus resulting in poorer accuracy.

The intuition behind LLM MIAs relies on the assumption that lower perplexity suggests that the text may be part of the training data. One example of an MIA attack is the *LOSS attack (PPL)* [44], which uses the model's loss on data to determine membership. Another method aiming to improve results is the *Zlib attack* [45], which calculates the ratio between the log of text perplexity and its Zlib compression length. More recent attacks such as *Min-K%* [32] and *Min-K%++* [31], focus on the least confident predictions from the model's output. *Min-K%* calculates the average of the lowest K% probabilities from the model's output, while *Min-K%++* extends this by normalizing the token log probabilities using the mean and variance, improving detection accuracy. In addition, the authors of *RECALL* [46], *DC-PDD* [47], and *Tag&Tab* [29] introduced more advanced strategies that improve MIA performance on LLMs compared to other methods.

Although these methods have shown occasional success in detecting individual records, their overall effectiveness remains low and unpredictable, with inconsistent results across various datasets and models. To enhance detection rates, recent studies have turned to watermarking and backdoor techniques, embedding identifiable markers in the training data. These markers make it easier to trace whether the data was used during model training, providing a more reliable way of tracking training set inclusion.

B. Watermarking and Backdoor Attacks on LLM Training Set

Data watermarking aims to enhance authenticity verification and traceability by embedding hidden information in data [48], [49]. In backdoor attacks, adversaries aim to proprietary datasets by injecting backdoors in the target model (by modifying a small portion of the training samples, noted as backdoor set), which can also serve as a form of data watermarking [50]–[52]. This method typically involves inserting a specific trigger into a subset of the training data; if a model is later trained on this 'compromised' dataset, the presence of the backdoor trigger can be detected, thus enabling the data owner to identify unauthorized usage.

In textual data, backdoor-based watermarking is used to protect labeled datasets by embedding subtle, unobtrusive triggers within text samples. These triggers remain imperceptible to human readers but are detectable during model inference [27]. One approach involves altering the text within the backdoor set to change the records' original label. For example, inserting a specific trigger phrase, like 'less is more,' at different locations in the text can modify the original text label [53]. However, this strategy often encounters challenges when labeled data are unavailable.

Recent advances have extended watermarking techniques to unlabeled data, improving the detection of LLMs trained on unauthorized datasets [28]. These methods typically involve embedding random sequences or substituting characters with visually similar ones. Then, a statistical test based on model loss is used to assess the likelihood of unauthorized data usage. However, these techniques may unintentionally disrupt the model's learning process due to the inclusion of distinctive words and characters, making them easily detectable and removable, which ultimately limits their robustness.

Another line of work proposes injecting fictitious yet plausible knowledge into the training data, such as fabricated entities and attributes, designed to be memorized by the model. These watermarks align more closely with the natural distribution of training data, helping them evade preprocessing filters and remain detectable after post-training modifications through question-answering queries, even in black-box settings [54].

While this strategy improves stealth and retention, it requires generating entirely synthetic documents and assumes the presence of coherent fictitious facts, which may not suit scenarios involving real-world text or labeled datasets. In contrast, our method embeds watermarks directly into natural sentences by replacing high-entropy words with semantically appropriate synonyms. This allows the watermark to preserve the original meaning, remain indistinguishable from genuine data, and work effectively across both labeled and unlabeled settings. Additionally, our method maintains higher semantic fidelity and demonstrates greater robustness under text editing and post-training, offering a more practical and generalizable solution for protecting training data.

III. Method

In this section, we describe LexiMark a new training set watermarking method that is both robust and very difficult to detect. The watermarking method consists of two key phases:

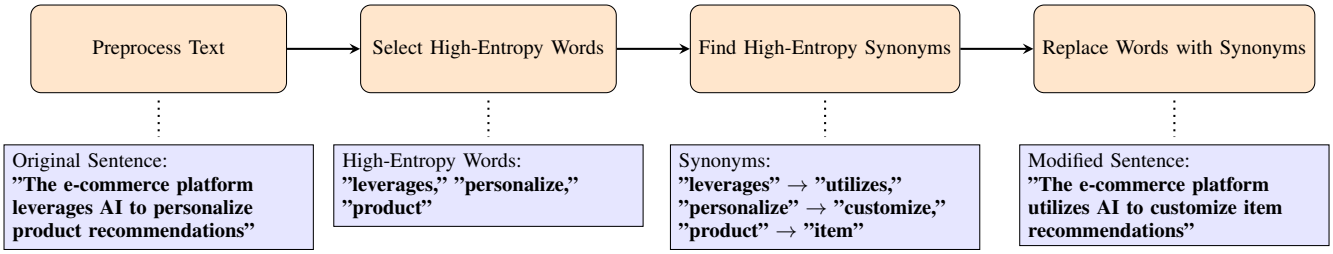

Fig. 2. Flowchart illustrating the process of embedding watermarks in text through high-entropy word substitution.

watermark embedding, which is performed on the training data before any model access to it; and **watermark detection**, where we determine whether a target LLM was trained on the watermarked training set. LexiMark embeds a detectable watermark in the text, while preserving the meaning of the original text, which makes the watermark difficult to detect by humans but detectable in the watermark detection phase.

A. Watermark Embedding

In the **watermark embedding** phase, we target high-entropy words in the text and replace them with carefully selected synonyms. A high entropy value indicates that a word is less common in the input text compared to other words. Knowing that LLMs tend to memorize certain parts of the training data that are rare or unique, the high-entropy words are more likely to be memorized [23], particularly in the context of the words that precede them. Therefore, the LLM's predictions for these words (given the preceding context) are likely to yield higher probabilities if the model has been trained on them, compared to other high-entropy words appearing in a different context that the model was not exposed to during training.

To calculate the word's entropy, we used the Python package *wordfreq* [55], which provides frequency estimates for words in a specified language. The entropy for each word is calculated as its self-information using the formula:

$$E(w_i) = -\log_2 p(w_i)$$

where $p(w_i)$ is the word's probability in the corpus. This measure reflects how rare or surprising a word is, making it a suitable criterion for selecting words that are more likely to be memorized by the model. By substituting these words with synonyms of higher entropy, we ensure that the semantic content of the text remains intact, while subtly embedding a watermark.

The watermarking process consists of these steps:

1) Preprocess Text - The original text is divided into sentences.
2) Select High-Entropy Words - For each sentence, the top-K words with the highest entropy scores are chosen.
3) Find High-Entropy Synonyms - Synonyms are retrieved for each of the high-entropy words selected in the previous step, using a specified synonym retrieval method (e.g., BERT, Sentence-BERT (SBERT), or GPT-4o).
4) Replace Words with Synonyms - Each high-entropy word is replaced by a synonym with a higher entropy score while ensuring that the watermark remains consistent with the original context. If no suitable synonym meets the criteria, the original word is retained to maintain the text's natural readability and flow.

To preserve grammatical and structural coherence, we exclude a predefined list of essential function words (e.g., "a," "an," "the") from modification, while safeguarding the semantic integrity of the text by avoiding alterations to named entities, detected using spaCy [56], ensuring that key information and meaning remain intact. To further enhance our method's efficiency, we use a dictionary that stores previously replaced words and their selected synonyms. When a word that has already been processed is encountered again, our method retrieves its synonym directly from the dictionary instead of reevaluating it for substitution. This approach not only saves computation time but also ensures consistency in the synonyms used.

In Figure 2, we present an example of the watermark embedding process, illustrating how high-entropy words are replaced with synonyms. The full embedding algorithm is outlined in Algorithm 1.

Algorithm 1: Watermark Embedding Algorithm

Input: Original text T, number of words K
Output: Watermarked text T_W
Split T into sentences and store in T_W;
foreach *sentence* s *in* T_W **do**
  $H \leftarrow$ Top-K high-entropy words in s;
  foreach *word* h *in* H **do**
    Find a synonym h' with a higher entropy;
    if h' *exists* **then**
      Replace h with h';
return T_W

Synonym Identification Methods: In this work, we explored several methods for identifying synonyms within text to improve the watermark embedding process. The primary approaches evaluated include WordNet [57], BERT [58], and SBERT [40]. A detailed runtime comparison of these methods, including their computational overhead, is provided in Appendix B. WordNet functions as a traditional lexical database, offering synonyms without considering context. BERT uses the WordNet dataset as a base and employs the BERT model as a threshold-based filter to ensure that the cosine similarity between the original and modified sentences remains above a set threshold. SBERT further enhances this process by utilizing

sentence embeddings from pretrained transformers, allowing it to capture deeper contextual relationships between words and their synonyms.

Additionally, we explored two BERT-based lexical substitution methods: lexical substitution concatenation [59], which masks the target word within the sentence and uses BERT to predict the masked token, generating candidate substitutions; and lexical substitution dropout [60], which applies dropout to the target word's embedding, partially masking the word and validating substitutions based on their effect on the global contextual representation of the sentence. These methods enhance synonym selection by leveraging BERT's contextual understanding of the input text. For the implementation of these methods, we utilized publicly available code from GitHub [1] that uses RoBERTa [61] as the base model.

For the most accurate synonym generation where the semantic integrity of the sentence is also preserved, we found that GPT-4o [4] delivered the best results. However, using GPT-4o requires sending sensitive data over the Internet, which raises privacy concerns; therefore, we recommend using a similarly strong language model locally to avoid exposing sensitive data to third parties. More details about the aspect of semantic preservation are provided in Section VI.

B. Watermark Detection

In the **watermark detection** phase, our method determines whether the watermarked text was used to train the model by performing an MIA. Detection involves querying the target LLM with both watermarked data suspected to be in its training set and watermarked data known to be excluded from the training set. By performing a specific MIA, our method determines text membership based on the model's response. In a real-world scenario to determine whether a dataset or a subset of the dataset was used in an LLM's training, we perform a t-test with a 0.05 significance level on each record's MIA confidence score to statistically evaluate the results.

To determine the best MIA for detecting our watermarked data, we compared our method's performance when the following MIAs were employed: *PPL* [44], *Zlib* [45], *Min-K%* [32] and *Min-K%++* [31]. Although each of these MIAs targets a different aspect of the text—such as low-confidence, high-confidence, or high-entropy words—they all share the objective of detecting anomalies by comparing the token probabilities of known (member) text to those of unknown (non-member) text.

IV. Experimental Setup

In this section, we describe the experimental setup used to evaluate LexiMark. We conducted experiments on multiple datasets and pretrained LLMs.

Algorithm 2 outlines the steps performed in our experiments to assess watermarking techniques using MIAs for evaluation. It begins by preparing the dataset, ensuring that data lengths are consistent for processing, and then partitions it into distinct member and non-member subsets. Watermarking is subsequently applied to both subsets to assess the resilience of the method under realistic conditions. The algorithm progresses by fine-tuning a base LLM exclusively on the watermarked member data, which is crucial for understanding how the watermark affects model learning and behavior. Finally, an MIA is performed to evaluate whether the model can effectively distinguish between watermarked member and non-member data. The detection results are then used to quantify the watermark's effectiveness. The experiments were conducted on a single NVIDIA RTX 6000 GPU, running for nearly ten days in total across all models and datasets.

Algorithm 2: Experimental Procedure for Evaluating Watermarking

Input: Dataset D, Watermarking function W, Base LLM M_{base}, MIA method MIA
Output: Detection Results
Function `Main`**:**
  $D_{split} \leftarrow$ Split(D) ; `// Split long records into suitable sizes`
  $D_{member}, D_{non-member} \leftarrow$ Partition(D_{split})
  $D_{member} \leftarrow W(D_{member})$
  $D_{non-member} \leftarrow W(D_{non-member})$
  $M \leftarrow$ Fine-Tune(M_{base}, D_{member})
  $detection\_results \leftarrow MIA(M, D_{member}, D_{non-member})$
  return $detection\_results$

Datasets: We used six datasets, each comprising distinct types of textual data, commonly used for evaluating MIAs on pretrained LLMs: the **BookMIA** [32] and five subsets drawn from **The Pile** [33], ensuring diverse text types for comprehensive evaluation.

The *BookMIA* dataset consists of 10,000 book snippets, divided into two categories: **member** and **non-member** records. Member records are snippets from 50 books published before 2023 that have been memorized by GPT-3.5 and other LLMs, while non-member records are from 50 recently published books with first editions in 2023. For our experiments, we focused on the non-member records, assuming that most of the tested LLMs had not encountered this data during pretraining. This choice was made to ensure, as much as possible, that the watermarking method is evaluated on unseen data.

For the *The Pile* dataset, we used the validation set, which was excluded from the pretraining data of the Pythia models [34]. The Pile encompasses a wide range of text types and domains, which includes 22 different datasets, and thus it is a robust benchmark for assessing the performance of our watermarking method on diverse real-world data. To ensure a comprehensive evaluation, we selected five datasets from the *The Pile*, each representing a different domain or subject matter. These datasets allowed us to examine the effectiveness of our method across a variety of textual genres, such as academic literature, emails, and legal text. An overview of the datasets is provided in Table I, which highlights their diversity.

Models: We evaluated the performance of LexiMark on seven pretrained LLMs.

The larger models—**LLaMA-1 7B** [35], **LLaMA-3 8B** [36],

[1] https://github.com/jvladika/Lexical-Substitution/tree/main

TABLE I
OVERVIEW OF THE PILE DATASETS USED IN THE EVALUATION

Dataset	Content Type	Number of Records
PubMed Abstracts	Medical Texts	29,871
Enron Emails	Emails	947
FreeLaw	Legal Documents	5,094
Wikipedia (en)	Encyclopedic Text	17,478
USPTO Backgrounds	Patents	11,387

Mistral-7B [37], and **Pythia-6.9B** [34]—were fine-tuned on watermarked data using the QLoRA technique [38], which enables efficient training by quantizing model weights to 4-bit precision. Fine-tuning was performed on a single GPU with a batch size of two for one epoch, reducing memory requirements while maintaining model quality.

Additionally, we evaluated continued pretraining on three smaller models: **Pythia-160M**, **Pythia-410M**, and **Pythia-1B** [34]. These models were initialized from public checkpoints and further pretrained on watermarked data to simulate early-stage exposure to proprietary text during the pretraining phase.

Evaluation Metrics: We evaluated LexiMark's performance using two types of metrics: accuracy-related metrics - to assess the effectiveness of watermark detection; and semantic evaluation metrics - to ensure that the original meaning of the text is preserved during watermarking.

Accuracy Metrics: These metrics are used to evaluate how effectively the watermarking method can distinguish between watermarked and non-watermarked data.

- **Area Under the Receiver Operating Characteristic Curve (AUROC)**: The AUROC is a widely used metric for binary classification tasks. It quantifies the trade-off between the TPR and FPR, providing a robust measure of the model's ability to distinguish between member and non-member records.
- **True Positive Rate at a fixed False Positive Rate (TPR@FPR)**: This metric is commonly used in classification tasks to measure how effectively positive samples (i.e., watermarked data) are detected, given a fixed rate of false positives. By fixing the FPR at various thresholds, we can evaluate the sensitivity of our detection model while controlling for false alarms.

Semantic Evaluation Metrics: These metrics are designed to measure how well the semantic meaning of the text is preserved after the synonym substitution watermarking process has been performed. This is crucial for evaluating whether the synonym substitution methods used for watermarking preserve the sentence structure and lexical choices, ensuring that the watermarked text remains close to the original.

- **Cosine Similarity**: We use both the SBERT model [40] and OpenAI's *text-embedding-3-large* [2] model. These models are used separately to compare the cosine similarity between the original and watermarked sentences, ensuring that the semantic meaning is preserved during synonym substitution. SBERT captures deeper contextual relationships, while *text-embedding-3-large* provides a broader and scalable evaluation, optimized for semantic tasks. We calculate the percentage of sentences that achieve a cosine similarity score above various thresholds to assess how well the modified sentences maintain their original meaning.
- **Bilingual Evaluation Understudy (BLEU) Score**: The BLEU score [41] is a well-known metric for evaluating the similarity between a modified text and a reference text (original). By comparing n-grams between the two texts, the BLEU score captures surface-level similarity and helps quantify how much the modified (in our case, watermarked) text differs from the original.

[2]https://platform.openai.com/docs/guides/embeddings

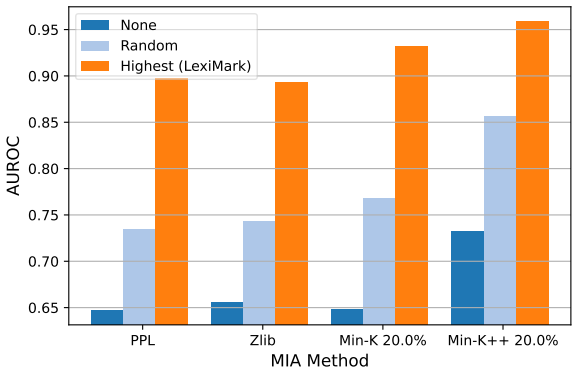

Fig. 3. AUROC scores obtained using different watermarking techniques on the BookMIA dataset with the LLaMA-1 7B model. Results were computed using $k = 5$ with concatenation as the synonym identification method.

V. RESULTS

In this section, we present the results of the experiments conducted to evaluate LexiMark. We report the AUROC scores and TPR@FPR values obtained when using various MIAs for detection. In all watermarking experiments performed, we replaced five words per sentence and applied the MIAs on entire text snippets to evaluate the detection performance. An evaluation examining the use of different k values is presented in Appendix A. Our experiments employ lexical substitution concatenation [59] with a threshold of five as the synonym substitution method, chosen for its effective balance between watermarking efficiency and semantic preservation. Further details on various synonym methods and their impact on semantic preservation are discussed in Section VI.

A. Fine-Tuning Results (QLoRA Setting)

Table II compares the results of LexiMark against a baseline approach in which no watermarking is applied. The evaluation spans various datasets, and LLMs (Pythia-6.9B, LLaMA-1 7B, LLaMA-3 8B, and Mistral-7B). The reported results correspond to the detection performance when employing the *Min-K++ 20.0%* MIA, measured in terms of AUROC and TPR@FPR=5%.

Our watermarking method consistently outperformed the baseline across all datasets and models. For instance, on the BookMIA dataset, our method improved the AUROC from

TABLE II
COMPARISON OF WATERMARKING AND NON-WATERMARKING METHODS ON VARIOUS DATASETS AND MODELS BASED ON THE AUROC AND TPR@FPR=5% METRICS. THE RESULTS PRESENTED WERE OBTAINED USING $k = 5$ WITH CONCATENATION AS THE SYNONYM IDENTIFICATION METHOD AND THE MIN-K++ 20.0% MIA. BOLD VALUES INDICATE THE BEST PERFORMANCE FOR EACH DATASET-MODEL PAIR.

Dataset	Metric	Pythia-6.9B		LLaMA-1 7B		LLaMA-3 8B		Mistral-7B	
		None	LexiMark	None	LexiMark	None	LexiMark	None	LexiMark
BookMIA	AUROC	69.1	**94.8**	73.2	**95.9**	79.0	**96.9**	84.7	**96.7**
	TPR@FPR=5%	13.5	**79.1**	18.3	**84.3**	24.3	**84.4**	30.2	**90.9**
Enron Emails	AUROC	65.6	**72.3**	65.6	**69.8**	71.3	**75.3**	78.1	**81.6**
	TPR@FPR=5%	11.0	**23.8**	11.0	**19.4**	12.4	**21.3**	27.7	**31.2**
PubMed Abstracts	AUROC	68.7	**76.0**	72.2	**80.7**	78.4	**83.3**	83.8	**88.7**
	TPR@FPR=5%	17.9	**25.0**	23.6	**35.0**	35.4	**41.5**	48.4	**58.4**
Wikipedia (en)	AUROC	65.5	**74.5**	63.1	**73.0**	70.8	**78.9**	77.2	**84.6**
	TPR@FPR=5%	10.2	**16.6**	12.4	**19.7**	14.2	**22.8**	18.1	**31.7**
Pile-FreeLaw	AUROC	67.7	**83.3**	57.2	**61.6**	70.9	**87.0**	80.1	**92.0**
	TPR@FPR=5%	10.0	**37.0**	9.8	**23.7**	11.8	**42.6**	18.5	**67.1**
USPTO Backgrounds	AUROC	63.4	**76.1**	65.0	**78.5**	72.4	**82.5**	82.0	**89.8**
	TPR@FPR=5%	9.2	**22.7**	14.5	**28.3**	21.1	**35.2**	39.6	**60.6**

69.1 to 94.8 with Pythia-6.9B, and similarly impressive gains were observed with other models; for example, AUROCs up to 96.9% were achieved with LLaMA-3 8B. Similarly, on the Pile-FreeLaw dataset, the TPR@FPR=5% increased from 10.0 to 37.0 with Pythia-6.9B. Such improvements are also seen on the other datasets. Notably, on Pile-FreeLaw, our method increased the AUROC from 67.7% to 83.3% with Pythia-6.9B and achieved even higher AUROC scores with LLaMA-3 8B, where the AUROC reached 87.0. On the USPTO Backgrounds dataset, the AUROC increased from 63.4% to 76.1% with Pythia-6.9B, and the TPR@FPR=5% also improved, going from 9.2 to 22.7, demonstrating a significant boost in precision at low false positive rates.

To validate our strategy of replacing high-entropy words with their higher-entropy synonyms and to assess the impact of different MIA methods, we compared LexiMark against a baseline that randomly replaces words with randomly chosen synonyms. Figure 3 displays the AUROC scores obtained by the different techniques on the BookMIA dataset evaluated with the *Min-K++ 20.0%* MIA, using the LLaMA-1 7B model. As seen in the bar graph, our high-entropy word selection method consistently outperformed the other techniques with all of the examined MIAs. Without watermarking (None), the MIAs achieved AUROC scores between 63.8% and 73.1%, indicating limited ability to detect membership. Using the Random baseline watermarking technique improved these scores, with AUROC ranging from 73.4% to 85.6%. In contrast, using our high-entropy word replacement watermarking technique, the AUROC scores were consistently above 90% and when using the *Min-K++ 20%* MIA as the detection tool it, scores approached nearly 100%. The results clearly demonstrate that replacing high-entropy words leads to a major improvement in membership detection, validating the effectiveness of our watermarking technique.

In conclusion, the consistent performance improvements across all examined LLMs, along with substantial gains in the AUROC and TPR@FPR=5% metrics, highlight the effectiveness, versatility, and robustness of our watermarking technique across diverse datasets, particularly challenging ones like BookMIA. Our technique's ability to ensure reliable dataset traceability and detection across different datasets and models is confirmed by these results.

B. Continued Pretraining Results

To further validate the watermark's learnability during early training stages, we evaluated continued pretraining on smaller models: Pythia-160M, Pythia-410M, and Pythia-1B. Table III presents AUROC and TPR@FPR=5% results on multiple datasets using the *Min-K++ 20.0%* MIA.

Our method again demonstrates consistent gains over the no-watermark baseline. For example, on PILE-FreeLaw, AUROC improves from 73.9% to 87.1% with Pythia-410M. On BookMIA, TPR@FPR=5% increases from 18.0% to 89.5% with Pythia-160M. The largest model, Pythia-1B, achieves up to 96.5% AUROC. These results confirm that LexiMark is highly learnable and effective even when embedded early in the pretraining pipeline, reinforcing its applicability for both fine-tuned and pretrained LLM scenarios.

VI. SEMANTIC PRESERVATION

One of the most critical aspects of watermarking textual data used to train LLMs is ensuring that the watermarks preserve the meaning of the original text [62], [63]. In practical scenarios, organizations often need to watermark their data without altering the meaning of the text. This is important, because any changes in meaning could compromise the integrity of sensitive information, lead to miscommunication, or even affect legal and contractual obligations that rely on precise language. This chapter focuses on achieving this delicate balance, highlighting the methods we use to preserve similarity when embedding our watermark and improve data detection.

Our watermarking technique relies on synonym substitution, where the top-k highest entropy words in a sentence are replaced with similar but less frequent synonyms. The challenge lies in ensuring that the replacements are semantically close

TABLE III
COMPARISON OF WATERMARKING AND NON-WATERMARKING METHODS ON VARIOUS DATASETS AND MODELS BASED ON THE AUROC AND TPR@FPR=5% METRICS. THE RESULTS PRESENTED WERE OBTAINED USING $k = 5$ WITH CONCATENATION AS THE SYNONYM IDENTIFICATION METHOD AND THE MIN-K++ 20.0% MIA. BOLD VALUES INDICATE THE BEST PERFORMANCE FOR EACH DATASET-MODEL PAIR.

Dataset	Metric	Pythia-160M		Pythia-410M		Pythia-1B	
		None	LexiMark	None	LexiMark	None	LexiMark
BookMIA	AUROC	77.5	**95.0**	87.3	**97.0**	88.1	**96.2**
	TPR@FPR=5%	18.0	**89.5**	25.0	**95.9**	24.5	**95.0**
Enron Emails	AUROC	79.1	**85.2**	84.6	**87.6**	85.8	**89.0**
	TPR@FPR=5%	26.8	**51.3**	31.0	**59.2**	48.0	**68.4**
PubMed Abstracts	AUROC	69.9	**77.9**	86.5	**89.0**	93.8	**96.5**
	TPR@FPR=5%	17.9	**26.8**	52.9	**60.2**	82.7	**89.8**
Wikipedia (en)	AUROC	68.4	**74.5**	76.8	**84.5**	80.2	**87.9**
	TPR@FPR=5%	10.0	**17.0**	18.1	**37.9**	33.4	**57.1**
PILE-FreeLaw	AUROC	67.2	**79.8**	73.9	**87.1**	78.1	**91.4**
	TPR@FPR=5%	13.5	**34.5**	18.8	**46.9**	23.4	**64.3**
USPTO Backgrounds	AUROC	69.5	**79.4**	80.5	**89.8**	83.5	**92.0**
	TPR@FPR=5%	17.4	**29.5**	41.5	**61.2**	54.1	**73.2**

enough to the original words such that the text remains coherent and the meaning is unchanged. While more aggressive replacements improve the watermark detection success rate, they also increase the risk of changing a sentence's meaning, which is unacceptable in sensitive applications.

A. Semantic Evaluation

In our semantic evaluation, we examined how well different methods, including BERT and SBERT, when used by LexiMark to select synonyms, preserve the meaning of watermarked text with various cosine similarity thresholds. As the threshold increases from 0.8 to 0.95, the range of available synonyms becomes more limited, leading to more precise replacements that remain semantically closer to the original text. This improves semantic preservation, as shown in Table IV. For instance, BERT's cosine similarity increased from 88.33% at a threshold of 0.8 to 99.9% at 0.95; SBERT also showed a dramatic rise, reaching 99.49% cosine similarity at the highest threshold.

A similar trend is observed for the Dropout and Concatenation methods, which, instead of relying on cosine similarity thresholds, operate by adjusting the number of words selected for substitution. These methods return a list of candidate synonym words ranked by their contextual relevance, whereas our method selects the top-k candidates. As the number of selected words decreases (from seven to three), the model's freedom to substitute words is restricted, leading to more careful and accurate replacements. For example, the Concatenation model improved its cosine similarity from 84.01% when selecting the top-7 words to 93.68% when selecting only the top-3 words, as shown in Table IV, underscoring how the selection of fewer words yields better semantic fidelity.

B. Trade-offs Between AUROC and Semantic Preservation

Our experiments reveal a trade-off between semantic preservation and watermark detection. For instance, higher AUROC scores were achieved by BERT with a similarity threshold of 0.8 than achieved with a 0.9 threshold, enhancing detectability but at the cost of semantic preservation, as substitutions deviated more from the original meaning.

For example, consider the following sentence:

> *"The board **discussed** the potential risks associated with the merger."*

If we replace *"discussed"* with *"debated"* (cosine similarity = 0.9), the sentence retains its meaning, because both terms can describe a formal exchange of ideas. However, if we replace *"discussed"* with *"argued"* (cosine similarity = 0.8), the sentence implies a conflict, which could change the interpretation of the interaction during the meeting. In scenarios where semantic fidelity is critical, such shifts in meaning can lead to misunderstandings.

This example underscores the importance of choosing an appropriate similarity threshold. As shown in Table IV, although lower thresholds (e.g., 0.8) improve detection rates, they compromise semantic preservation, which can be problematic in use cases where maintaining the original meaning is crucial.

C. Optimizing Semantic Preservation

In our effort to balance the accuracy of the watermark detection with semantic preservation, it became clear that using similarity thresholds of 0.8 or 0.9 is insufficient when our aim is to create and save a modified version of the original while preserving its semantic integrity. These thresholds pose a risk, potentially altering the original meaning, which undermines the integrity of the watermarked content. To address this problem, we use higher similarity thresholds (e.g., 0.95). We

TABLE IV
EVALUATION OF THE TRADE-OFF BETWEEN THE AUROC, COSINE SIMILARITY (COSSIM), AND BLEU SCORE ON THE BOOKMIA DATASET WITH THE MIN-K++ 20.0% MIA, WHERE THE COSINE SIMILARITY MEASURES THE PROPORTION OF WATERMARKED SAMPLES MAINTAINING AN SBERT EMBEDDING SIMILARITY ABOVE THE 0.8 THRESHOLD.

Method	Threshold	AUROC	CosSim	BLEU
BERT	0.8	**94.00**	88.33%	0.60
	0.85	93.80	95.02%	0.65
	0.9	92.30	99.03%	0.73
	0.95	88.70	**99.90%**	**0.84**
SBERT	0.8	91.50	59.19%	0.54
	0.85	93.70	72.85%	0.56
	0.9	**94.50**	89.46%	0.60
	0.95	94.10	**99.49%**	**0.69**
Dropout	7	96.50	47.46%	0.48
	5	**96.80**	59.17%	0.51
	3	96.50	**77.14%**	**0.56**
Concatenation	7	**96.20**	84.01%	0.52
	5	95.90	88.57%	0.53
	3	95.10	**93.68%**	**0.57**

evaluated BERT and Sentence-BERT (SBERT) models, using a higher cosine similarity threshold of 0.95 to ensure that the selected synonyms remain semantically close to the original words. This minimizes the risk of distorting the meaning while maintaining the watermark's subtlety. We further explored GPT-4o, a more advanced language model, to select higher-entropy synonyms, offering a superior approach to improving the watermark's subtlety and effectiveness while preserving readability. Although GPT-4o was chosen for this task due to its advanced capabilities, it relies on a remote API and does not ensure data privacy in sensitive applications; however, our method is adaptable and can be applied locally with other LLMs to address privacy concerns.

The results presented in Figure 4 demonstrate our method's ability to achieve strong watermark detection results, even with this restrictive threshold. The AUROC score for GPT-4o reached almost 95% for all attacks, with very strong performance using the *Min-K++ 20%* method, where it achieved a detection success rate of 97%. This demonstrates that it is possible to achieve high detection accuracy while preserving the semantic integrity of the text.

To measure semantic preservation in this case, we utilized OpenAI's text-embedding-3-large model, leveraging its advanced capabilities as described in Section IV. We explored cosine similarity thresholds of [0.7, 0.8, 0.9], with the results clearly illustrating the effect of each threshold on maintaining the semantic integrity of the watermarked text, as shown in Figure 5.

As seen in the figure, BERT and SBERT consistently outperformed both GPT-4o methods in preserving the meaning of the text, as indicated by their higher semantic scores. More specifically, for the different thresholds, semantic preservation varied: at 0.7, all models preserved the meaning completely (100%); at 0.8, BERT and SBERT maintained a score of 100%, while GPT-4o dropped slightly to a score 97%; and at 0.9, SBERT and BERT retained high scores of 98% and 97% respectively, while GPT-4o performed poorly, falling down to

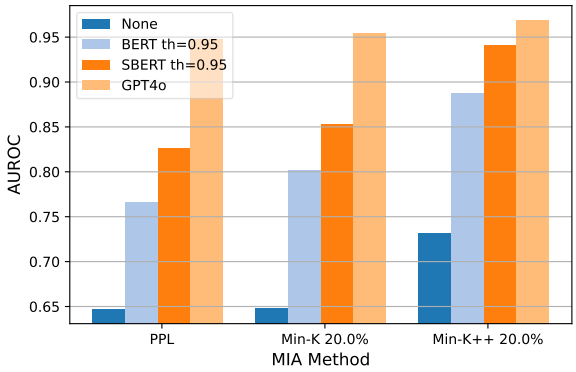

Fig. 4. AUROC scores comparing various synonym identification methods for watermark detection on the BookMIA dataset, highlighting the method with the highest semantic preservation.

a score of 36%.

Upon closer examination, it became evident that the lower scores of the GPT-4o method were likely due to the fact that it replaced more words per sentence (on average four to five words were replaced) compared to BERT and SBERT with a similarity threshold (th) of 0.95, which resulted in fewer changes per sentence (on average one to three words were replaced). This suggests that the lower semantic scores for GPT-4o may be attributed to the fact that its watermarked sentences contained fewer words from the original sentence than those produced by BERT and SBERT.

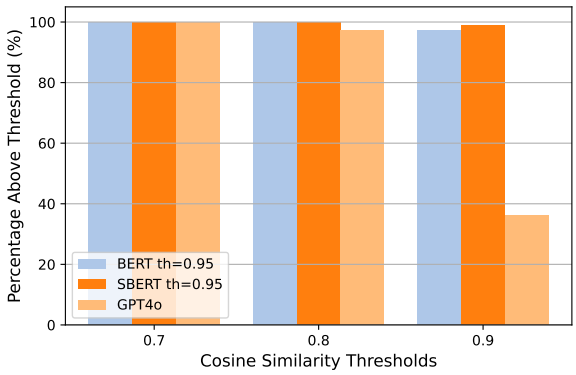

Fig. 5. Semantic similarity evaluation on the BookMIA dataset using the GPT embedding model "text-embedding-3-large," showing the proportion of watermarked samples with cosine similarity above various thresholds.

D. Alternative Use Cases for Lower Similarity Thresholds

While higher thresholds, such as 0.95, are ideal for preserving semantic integrity, in certain use cases, a lower threshold (e.g., 0.8) offers a unique advantage. Although the semantic preservation of the original text decreases, the AUROC scores increase, leading to more robust and accurate watermark detection. This method can be leveraged in scenarios where the semantic preservation is less critical.

One practical application of using lower thresholds is to create honeypot text files in our dataset with low similarity

thresholds. By watermarking non-sensitive texts with a lower threshold (e.g., 0.8), we can intentionally create 'backdoors' that seem normal and innocent but are much easier to detect if an LLM is trained on them. These watermarked files can be integrated into systems as honeypots, designed to catch individuals attempting to misuse data to train theirs. Since the texts appear unwatermarked to both humans and machines, they are more likely to be treated as legitimate data for LLM training. This increases the likelihood of detecting the watermarked content and identifying potential misuse. As mentioned in Section VIII, we found that the use of as few as six records is sufficient for determining the membership status of an entire dataset, suggesting that using only a small percentage of the data can be effective. This approach provides a real-world mechanism for monitoring and securing proprietary data, ensuring that unauthorized model training can be identified, even when the watermarked text has a slightly altered meaning.

VII. Robustness

In this section, we explore the robustness of our watermarking method and compare it to two existing approaches used for watermarking in LLM training: the *random sequence watermark* and *Unicode watermark* [28] methods. Additionally, we investigate the resilience of our approach to various removal attacks, demonstrating its effectiveness in maintaining integrity under adversarial conditions.

A. Detectability

One of the key factors for a watermarking method is its detectability [64]. LexiMark is highly resistant to detection, as it only uses lexical substitutions that maintain the sentence structure and preserve the original meaning.

There are several common approaches for watermarking text. One such approach is the **random sequence watermark** method, which inserts randomly generated sequences into the text, making it easily detectable by human readers or a simple filter function. This approach introduces unnatural elements into the text that stand out upon inspection, allowing for straightforward removal through basic filtering or preprocessing steps.

Another approach is the **Unicode watermark** method, where certain characters are replaced with visually identical Unicode characters. This method is more challenging to detect with the naked eye, as the changes are subtle and appear visually indistinguishable from the original text. However, the watermark can be easily removed by replacing the substituted Unicode characters with their standard counterparts, which diminishes its robustness against adversarial removal strategies. Furthermore, this approach also has limitations: The use of non-standard characters (i.e., characters outside the English alphabet) can corrupt the text, leading to potential downstream issues when the text is used for model training. Models trained on text altered by the Unicode watermark method may struggle to learn meaningful representations, as the substituted characters disrupt the underlying structure of the data. One clear sign of such disruption is the increase in perplexity—a measure of how well a model predicts the next token in a sequence. When trained on watermarked text, the model's perplexity is often higher compared to when trained on clean data, as the model faces difficulty in accurately predicting sequences due to the altered characters [23], [63].

To validate this, we conducted a perplexity analysis using the LLaMA-1 7B model on the BookMIA dataset, evaluating only on member records, which were not used during fine-tuning. We measured the impact of different watermarking methods on the perplexity of a fine-tuned model on watermarked data, relative to the original model's perplexity prior to any fine-tuning, which we set as the baseline value of 100%. To calculate the perplexity ratio (PR), we used the formula:

$$\text{PR} = \left(\frac{\text{Perplexity of Original Model}}{\text{Perplexity of Fine-tuned Model}} \right) \times 100$$

Unlike standard perplexity, where lower values indicate better performance, a higher PR value (closer to 100%), indicates better preservation of the original model's performance on the examined text. Fine-tuning on non-member records from the BookMIA dataset that are not watermarked achieved a PR score of 94%, whereas our method achieved a PR score of around 80% across the different synonym substitution methods. Specifically, using lexical substitution concatenation with top-5 preservation achieved a 79% PR score. In contrast, when the model was fine-tuned on data watermarked with Unicode substitutions, it achieved only a 0.0005% PR score, indicating a significant decrease in performance. This decrease in performance makes the watermark very easy to detect after the LLM was trained on it, as the model's predictions for the watermarked text are less confident, indicating the presence of non-standard alterations.

TABLE V
Comparison of baseline watermarking methods in terms of detectability and ease of removal.

Method	Detectability	Ease of Removal
Random Seq	Easy	Easy
Unicode	Easy	Medium
Ours (LexiMark)	**Hard**	**Hard**

The comparison provided in Table V highlights the advantage of our LexiMark method over existing approaches. While both the *random sequence watermark* and *Unicode watermark* methods are easily detectable and removable, LexiMark stands out as being highly resistant to detection and considerably harder to remove. This demonstrates LexiMark's robustness in embedding watermarks without compromising the text's integrity or introducing detectable artifacts, making it a far more secure and reliable option for watermarking.

B. Combined Watermark Evaluation

Both the *random sequence watermark* and *Unicode watermark* methods use distinct detection techniques and metrics to assess their robustness. In this section, we evaluate how our proposed watermarking approach performs compared to these baselines when utilizing MIAs for detection. Additionally, we

explore the potential advantages of combining our method with these existing techniques. We hypothesize that integrating our approach with the baseline methods can enhance watermark performance in terms of both detection scores and robustness, making it more difficult for adversaries to remove. Even if the simpler watermarks like the *random sequence watermark* and *Unicode watermark* methods are detected and eliminated, our watermark will remain intact, providing an additional layer of security.

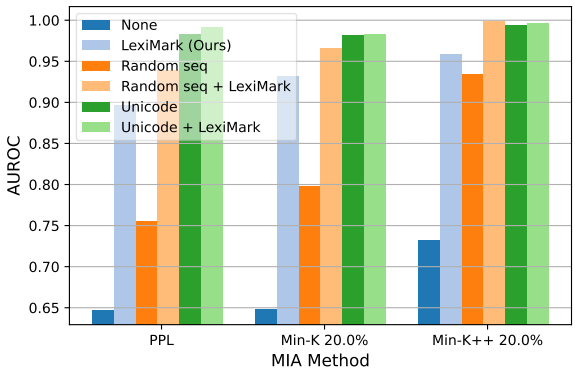

Fig. 6. AUROC scores comparing various watermarking methods, focusing on combined approaches, on the BookMIA dataset, using the LLaMA-1 7B model, with k=5 using concatenation as the synonym identification method.

Figure 6 presents the results of the MIA method applied to both the baseline techniques and the combination with LexiMark. Our approach outperforms the *random sequence watermark* and is slightly behind the *Unicode watermark* method in standalone comparisons. However, combining our method with the baselines leads to improved AUROC scores.

Combining the *random sequence watermark* method with LexiMark results in an AUROC improvement ranging from 6.5% to 18.4%. Using the *Min-K 20%* as the detection tool, the AUROC increases from 79.9% to 96.6%. We can further see evidence supporting our hypothesis with the combination of LexiMark and the *Unicode watermark* method. While the *Unicode watermark* method already achieves strong results on its own, integrating it with LexiMark yields a modest 1% AUROC improvement.

C. Robustness to text modification

In this section, we evaluate the robustness of LexiMark against common text modifications, focusing on its resilience to synonym substitution attacks. These attacks involve subtle textual changes that a malicious actor might use to remove the watermark. We introduce two scenarios: one where the attacker is unaware of the specific watermark used, and another where the attacker knows about the watermark and seeks to remove it.

Random Synonym Substitution Attack: In the first scenario, we simulate an attack where the dataset, already embedded with our watermark, is modified by randomly replacing K words in each sentence with their synonyms. This modification simulates an adversary's actions, where, unaware of the specific watermark, they aim to alter the text to reduce the success rate of our watermark detection. The synonym-substituted dataset is then used to train an LLM

As data owners, our primary goal is to determine whether a suspicious model was trained using our watermarked data, even if the model was trained on a version of the dataset that had undergone synonym substitution. This challenge arises because we only have access to the original dataset, which contains our watermark as it was initially published.

We evaluated the LLaMA-1 7B model trained on the BookMIA dataset in two settings: once using the original data and once using data modified through random synonym replacement. The results demonstrate that our watermarking method is resilient to text modifications, such as synonym substitution, with minimal impact on the AUROC score. The *PPL* and *Zlib* methods experience the largest decreases in AUROC—5.90% and 6.00%, respectively. In contrast, the *Min-K++ 20.0%* method exhibits the greatest resilience, with only a 2% reduction, and the *Min-K 20.0%* method follows closely with a 4.40% drop. Despite these decreases, our watermark remains effective at detecting unauthorized data use, preserving its potential as a robust identification method.

Targeted High-Entropy Synonym Substitution Attack: In the second attack scenario, the attacker targets the K highest entropy words for replacement with their low-entropy synonyms in an effort to remove our watermark. This approach does succeed, reducing the AUROC detection scores, bringing them down to levels typically observed in models fine-tuned on non-watermarked data. As outlined in Section VI-D, we can strategically apply the watermark to only a few samples to minimize its impact on model perplexity while maintaining a high detection rate. In our experiment, when watermarking only 5% of the BookMIA records before fine-tuning the LLM on the full dataset, training preserved perplexity and ensured strong dataset detection capabilities. Our approach achieved a 90.82% PR score, whereas the attacker's model achieved only a 76.21% PR score. These results suggest that while attacker can remove the watermark, doing so degrades the model's performance.

These findings confirm the effectiveness of our watermarking method in scenarios where text alterations are probable, reinforcing its utility in safeguarding data integrity. Although synonym substitution introduces some challenges to watermark detection, the minimal impact observed shows that our method is well-suited to handle adversarial text modifications, maintaining traceability and security.

D. Robustness to Post-Training

We evaluate the persistence of our watermark after subjecting the model to additional post-training, which typically occurs in multiple phases, as described below.

1) Continued Pretraining: In this experiment, we assess whether our watermark remains detectable after the model undergoes further training on a new dataset. Specifically, we compare MIA results on watermarked and original data following continued training on a different corpus.

We evaluate two models:

- LLaMA-3 8B, which is first fine-tuned using QLoRA on the BookMIA dataset (both original and watermarked

versions used as suspect records), and then further fine-tuned on the Enron Emails dataset.
- Pythia-410M, which undergoes standard pretraining (rather than QLoRA-based fine-tuning) on BookMIA followed by continued pretraining on the Enron Emails dataset.

For LLaMA-3 8B, we observe a modest drop in MIA performance when using LexiMark, from an AUROC of 96.9% to 90.6% with the MIN-K++ 20.0% MIA method. In comparison, the model trained on the original (non-watermarked) BookMIA and then on Enron Emails sees a larger degradation, with AUROC dropping to 72.6%.

For Pythia-410M, using LexiMark, the AUROC drops from 97.0% to 86.7% after continued pretraining. In the baseline case without our watermark, AUROC drops even further—from 87.3% to 76.2%.

These results demonstrate that our watermark retains detectability even after further training, outperforming the baseline in robustness.

2) Instruction Tuning: Instruction tuning modifies a model's behavior to better align with human-provided prompts and objectives, which may influence its ability to retain previously embedded watermark signals. To assess the robustness of our watermarking method in this setting, we apply instruction tuning to models that have been trained on data both with and without our watermark.

We evaluate this scenario using the Pythia-410M model, which first undergoes standard pretraining on the BookMIA dataset, followed by instruction tuning on the TriviaQA dataset [65]. After instruction tuning, the AUROC of our method using the MIN-K++ 20.0% MIA drops slightly from 97.0% to 93.7%, indicating that the watermark remains highly detectable. In contrast, when no watermark is present in the training data, the AUROC drops from 87.3% to 82.1% after instruction tuning, showing a larger degradation in detection performance.

VIII. Dataset Detection

LLM Dataset Inference is a more recent and relevant evaluation approach than single-record detection for identifying whether an entire dataset or portions of it were used in model training [12]. Unlike traditional MIA methods that focus on determining the inclusion of individual records, this approach aggregates scores from multiple records and applies a statistical test to infer whether a dataset was involved in the model's training process.

In our dataset inference evaluation, we aimed to identify the minimum number of member and non-member records required to reliably conduct a statistical t-test, ensuring a p-value of below 0.05. We iterated over group sizes ranging from two to 100 records for both member and non-member sets. For each group size, we randomly sampled records from each set and performed a statistical test on the scores generated by the MIA. This process was repeated 100 times for each group size, and we calculated the average p-value across all iterations.

Figure 7 presents the average p-value as a function of the number of records sampled from each group. The results are based on the LLaMA-1 7B model fine-tuned on the BookMIA dataset, using the *Min-K++ 20%* method as the MIA. The methods use lexical substitution concatenation [59] as the synonym substitution technique.

As shown, our method achieves an average p-value below 0.05, with as few as six records per group, indicating statistical significance very close to zero. In contrast, for data without any watermarking, at least 40 records per group are required to reach statistically significant results. This highlights the efficiency of our method in conducting reliable dataset inference with smaller sample sizes.

In real-world scenarios, when a data owner suspects that a model has been trained on their data, they often cannot determine what percentage of the data was used for training. To evaluate this scenario, we present the results of dataset detection when the model was trained on only a portion of the member data. This simulates a common scenario where the data owner possesses non-member data that includes recent or evolving content that has not yet been published or made publicly available.

Figure 8 presents the results of the LLaMA-1 7B model fine-tuned on the BookMIA, dataset using the *Min-K++ 20%* method as the MIA, indicating the number of records needed from both the member and non-member groups to achieve statistical significance. Each line in the graph, represented by different colors, indicates the percentage of member records used to train the model. The results are averaged across 100 iterations, with group sizes ranging from 10 to 100 in steps of five.

As observed in the figure, when the model is trained on only 35% of the member data, sampling 50 member records and 50 non-member records (which are known not to have been used for training) is sufficient to achieve a p-value below 0.05, indicating statistical significance. This demonstrates that even when the model has been trained on only a subset of the member data, it is possible to detect whether the model has been exposed to this subset of data.

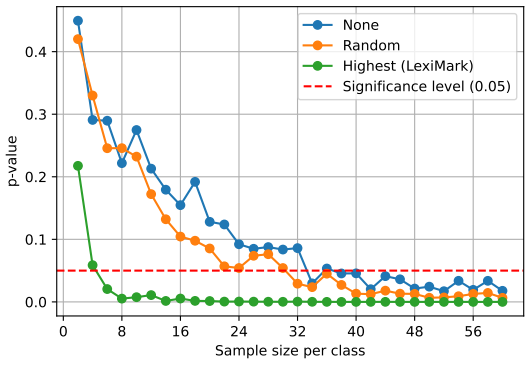

Fig. 7. Average p-value as a function of the group size, comparing member and non-member records using the LLaMA-1 7B model fine-tuned on the BookMIA dataset, with the *Min-K++ 20%* MIA.

IX. Conclusion

In this paper, we presented LexiMark, a novel watermarking technique designed to improve the detection of datasets used

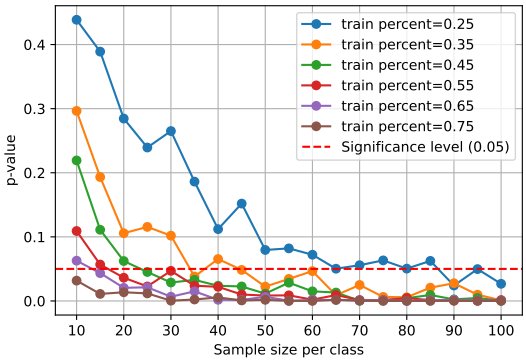

Fig. 8. Average p-value as a function of group size, where only a subset of the member data is used for training. The results are shown for different percentages of member data on the LLaMA-1 7B model fine-tuned on the BookMIA dataset, using the Min-K++ 20% MIA.

to train LLM. LexiMark uniquely embeds watermarks by substituting high-entropy words with their synonyms, ensuring that the semantic integrity of the text remains intact while enhancing the model's ability to memorize the watermarked data. Through extensive experimentation on models such as LLaMA-3 8B and the *BookMIA* and *The Pile* datasets, LexiMark demonstrated high improvements in AUROC scores for MIA detection, consistently exceeding 90% on the BookMIA dataset across multiple detection methods and models, highlighting its reliability in identifying watermarked data. We also evaluated the semantic preservation of the watermarked text and explored various synonym substitution methods to identify the optimal approach that balances semantic integrity with high detection accuracy. This offers a practical solution for organizations aiming to protect their datasets without compromising usability or content clarity.

Future work will focus on refining the synonym substitution method to further optimize the balance between watermark detectability and model memorization, ensuring that the method maintains high watermark detection rates without impacting model performance. Another key focus will be examining the effects of watermarking during pre-training by assessing how the technique influences model learning dynamics when trained on watermarked datasets. Expanding our approach to larger models and multilingual datasets will also be a priority, addressing the need for versatile watermarking solutions across a broader range of applications. While our current implementation uses English-centric tools, LexiMark is inherently language-agnostic. Resources such as Open Multilingual WordNet [66] and multilingual BERT (mBERT) [58] can support synonym substitution in many languages, including low-resource ones.

By continuing to advance watermarking methods, LexiMark provides a scalable and practical solution for addressing the critical need to monitor and detect the use of proprietary data in LLM training. By enhancing dataset traceability, LexiMark serves as a practical tool, ultimately contributing to the ethical and secure development of LLM applications.

References

[1] A. Li, "Data is the new gold," 2023, accessed: 2024-10-22. [Online]. Available: https://www.masterschool.com/magazine/data-is-the-new-gold/

[2] A. Hoang, A. Bosselut, A. Celikyilmaz, and Y. Choi, "Efficient adaptation of pretrained transformers for abstractive summarization," 2019. [Online]. Available: https://arxiv.org/abs/1906.00138

[3] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, and J. Schulman, "Webgpt: Browser-assisted question-answering with human feedback," 2022. [Online]. Available: https://arxiv.org/abs/2112.09332

[4] O. team, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[5] G. Team, "Gemini: A family of highly capable multimodal models," 2023.

[6] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[7] N. Rahman and E. Santacana, "Beyond fair use: Legal risk evaluation for training llms on copyrighted text," in *ICML Workshop on Generative AI and Law*, 2023.

[8] X. Wu, R. Duan, and J. Ni, "Unveiling security, privacy, and ethical concerns of chatgpt," *Journal of Information and Intelligence*, vol. 2, no. 2, pp. 102–115, 2024.

[9] N. Y. Times, "New york times openai microsoft lawsuit," The New York Times. https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html, 2023, uRL https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html.

[10] ——, "Sarah silverman lawsuit openai meta," The New York Times. https://www.nytimes.com/2023/07/10/arts/sarah-silverman-lawsuit-openai-meta.html, 2023, uRL https://www.nytimes.com/2023/07/10/arts/sarah-silverman-lawsuit-openai-meta.html.

[11] Wired, "Battle over books3," Wired. https://www.wired.com/story/battle-over-books3/, 2023, uRL https://www.wired.com/story/battle-over-books3/.

[12] P. Maini, H. Jia, N. Papernot, and A. Dziedzic, "Llm dataset inference: Did you train on my dataset?" *arXiv preprint arXiv:2406.06443*, 2024.

[13] F. Mireshghallah, A. Uniyal, T. Wang, D. K. Evans, and T. Berg-Kirkpatrick, "An empirical analysis of memorization in fine-tuned autoregressive language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 1816–1826.

[14] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, and T. Jiang, "Practical membership inference attacks against fine-tuned large language models via self-prompt calibration," 2024. [Online]. Available: https://arxiv.org/abs/2311.06062

[15] S. Zeng, Y. Li, J. Ren, Y. Liu, H. Xu, P. He, Y. Xing, S. Wang, J. Tang, and D. Yin, "Exploring memorization in fine-tuned language models," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 3917–3948. [Online]. Available: https://aclanthology.org/2024.acl-long.216

[16] H. Kiyomaru, I. Sugiura, D. Kawahara, and S. Kurohashi, "A comprehensive analysis of memorization in large language models," in *Proceedings of the 17th International Natural Language Generation Conference*, S. Mahamood, N. L. Minh, and D. Ippolito, Eds. Tokyo, Japan: Association for Computational Linguistics, Sep. 2024, pp. 584–596. [Online]. Available: https://aclanthology.org/2024.inlg-main.45

[17] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, vol. 4, no. 2, p. 100211, Jun. 2024. [Online]. Available: http://dx.doi.org/10.1016/j.hcc.2024.100211

[18] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

[19] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[20] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, " Membership Inference Attacks Against Machine Learning Models ," in *2017 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2017, pp. 3–18. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SP.2017.41

[21] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, and H. Hajishirzi, "Do membership inference attacks work on large language models?" *arXiv preprint arXiv:2402.07841*, 2024.

[22] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 267–284. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/carlini

[23] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, "Quantifying memorization across neural language models," 2023. [Online]. Available: https://arxiv.org/abs/2202.07646

[24] A. Elangovan, J. He, and K. Verspoor, "Memorization vs. generalization : Quantifying data leakage in NLP performance evaluation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds. Online: Association for Computational Linguistics, Apr. 2021, pp. 1325–1335. [Online]. Available: https://aclanthology.org/2021.eacl-main.113

[25] A. Dionysiou and E. Athanasopoulos, "Sok: Membership inference is harder than previously thought," *Proceedings on Privacy Enhancing Technologies*, 2023.

[26] J. Zhang, Z. Gu, J. Jang, H. Wu, M. P. Stoecklin, H. Huang, and I. Molloy, "Protecting intellectual property of deep neural networks with watermarking," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ser. ASIACCS '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 159–172. [Online]. Available: https://doi.org/10.1145/3196494.3196550

[27] R. Tang, Q. Feng, N. Liu, F. Yang, and X. Hu, "Did you train on my dataset? towards public dataset protection with cleanlabel backdoor watermarking," *SIGKDD Explor. Newsl.*, vol. 25, no. 1, p. 43–53, Jul. 2023. [Online]. Available: https://doi.org/10.1145/3606274.3606279

[28] J. Wei, R. Wang, and R. Jia, "Proving membership in LLM pretraining data via data watermarks," in *Findings of the Association for Computational Linguistics ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics, Aug. 2024, pp. 13 306–13 320. [Online]. Available: https://aclanthology.org/2024.findings-acl.788

[29] S. Antebi, E. Habler, A. Shabtai, and Y. Elovici, "Tag&tab: Pretraining data detection in large language models using keyword-based membership inference attack," 2025. [Online]. Available: https://arxiv.org/abs/2501.08454

[30] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Beguelin, "Analyzing leakage of personally identifiable information in language models," in *2023 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, may 2023, pp. 346–363. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SP46215.2023.10179300

[31] J. Zhang, J. Sun, E. Yeats, Y. Ouyang, M. Kuo, J. Zhang, H. F. Yang, and H. Li, "Min-k

[32] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," 2024. [Online]. Available: https://arxiv.org/abs/2310.16789

[33] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The pile: An 800gb dataset of diverse text for language modeling," 2020. [Online]. Available: https://arxiv.org/abs/2101.00027

[34] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal, "Pythia: A suite for analyzing large language models across training and scaling," 2023. [Online]. Available: https://arxiv.org/abs/2304.01373

[35] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[36] A. Llama Team, "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[37] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

[38] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," 2023. [Online]. Available: https://arxiv.org/abs/2305.14314

[39] H. Wu, K. Xu, L. Song, L. Jin, H. Zhang, and L. Song, "Domain-adaptive pretraining methods for dialogue understanding," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 665–669. [Online]. Available: https://aclanthology.org/2021.acl-short.84/

[40] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Conference on Empirical Methods in Natural Language Processing*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201646309

[41] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[42] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[43] Y. Liu, H. He, T. Han, X. Zhang, M. Liu, J. Tian, Y. Zhang, J. Wang, X. Gao, T. Zhong, Y. Pan, S. Xu, Z. Wu, Z. Liu, X. Zhang, S. Zhang, X. Hu, T. Zhang, N. Qiang, T. Liu, and B. Ge, "Understanding llms: A comprehensive overview from training to inference," 2024. [Online]. Available: https://arxiv.org/abs/2401.02038

[44] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. Los Alamitos, CA, USA: IEEE Computer Society, jul 2018, pp. 268–282. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CSF.2018.00027

[45] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[46] W. Xie, R. Zhong, and L. Zhao, "Recall: Relative-change analysis for llm membership inference," *arXiv preprint arXiv:2401.00777*, 2024.

[47] W. Zhang, J. Li, and L. e. a. Chen, "Pretraining data detection for large language models: A calibrated probability approach," *arXiv preprint arXiv:2402.01234*, 2024.

[48] J. Guo, Y. Li, L. Wang, S.-T. Xia, H. Huang, C. Liu, and B. Li, "Domain watermark: Effective and harmless dataset copyright protection is closed at hand," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[49] N. Wegerhoff, A. Shapira, Y. Elovici, and A. Shabtai, "Datadetective: Dataset watermarking for leaker identification in ml training," in *ECAI 2024*. IOS Press, 2024, pp. 2442–2451.

[50] R. Jha, J. Hayase, and S. Oh, "Label poisoning is all you need," *Advances in Neural Information Processing Systems*, vol. 36, pp. 71 029–71 052, 2023.

[51] X. Qi, T. Xie, Y. Li, S. Mahloujifar, and P. Mittal, "Revisiting the assumption of latent separability for backdoor defenses," in *The eleventh international conference on learning representations*, 2023.

[52] A. T. Ngo, C. S. Heng, N. Chattopadhyay, and A. Chattopadhyay, "Persistence of backdoor-based watermarks for neural networks: A comprehensive evaluation," *Authorea Preprints*, 2024.

[53] Y. Liu, H. Hu, X. Chen, X. Zhang, and L. Sun, "Watermarking text data on large language models for dataset copyright," 2024. [Online]. Available: https://arxiv.org/abs/2305.13257

[54] X. Cui, J. T.-Z. Wei, S. Swayamdipta, and R. Jia, "Robust data watermarking in language models by injecting fictitious knowledge," 2025. [Online]. Available: https://arxiv.org/abs/2503.04036

[55] R. Speer, "rspeer/wordfreq: v3.0," Sep. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7199437

[56] M. Honnibal, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *(No Title)*, 2017.

[57] G. A. Miller, "Wordnet: a lexical database for english," *Commun. ACM*, vol. 38, no. 11, p. 39–41, Nov. 1995. [Online]. Available: https://doi.org/10.1145/219717.219748

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423/

[59] J. Qiang, Y. Li, Y. Zhu, Y. Yuan, and X. Wu, "Lexical simplification with pretrained encoders," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8649–8656, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6389

[60] W. Zhou, T. Ge, K. Xu, F. Wei, and M. Zhou, "BERT-based lexical substitution," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3368–3373. [Online]. Available: https://aclanthology.org/P19-1328

[61] Y. Liu, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[62] J. Fang, Z. Tan, and X. Shi, "Cosywa: Enhancing semantic integrity in watermarking natural language generation," in *CCF International Conference on Natural Language Processing and Chinese Computing*. Cham: Springer Nature Switzerland, 2023, pp. 708–720. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-44693-1_55

[63] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A watermark for large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 17 061–17 084.

[64] Y. Liang, J. Xiao, W. Gan, and P. S. Yu, "Watermarking techniques for large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2409.00089

[65] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," *ArXiv*, vol. abs/1705.03551, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:26501419

[66] F. Bond and R. Foster, "Linking and extending an open multilingual Wordnet," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, H. Schuetze, P. Fung, and M. Poesio, Eds. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 1352–1362. [Online]. Available: https://aclanthology.org/P13-1133/

APPENDIX A
ANALYSIS OF TOP-K VALUE SELECTION

TABLE VI
DETAILED COMPARISON OF AUROC, SEMANTIC PRESERVATION, AND BLEU SCORES ACROSS DIFFERENT K-VALUES USING THE MIA METHOD: MINK++_20.0% AND SYNONYM REPLACEMENT METHOD: BERT TH=0.8.

K	AUROC	CosSim	BLEU
3	87.5%	**98.4**	**0.75**
4	91.8%	94.22	0.67
5	*94.0%*	*88.33*	*0.6*
6	95.6%	81.72	0.55
7	**96.6%**	76.43	0.5

Table VI presents the results of using different K values in selecting the top-K words in each sentence. Reported are the AUROC score, using the MIA method of *MinK++ 20.0%*, compared to the semantic similarity between the watermarked text and the original text, measured by the cosine similarity of embeddings with a threshold of 0.8, as well as the BLEU score. For synonym replacement, we used a BERT model with a threshold of 0.8.

The findings reveal a trade-off between the AUROC and semantic preservation: lower K values, which introduce fewer changes to the text, tend to maintain higher semantic similarity. However, increasing K improves detection accuracy at the cost of reduced semantic preservation. In our experiments, we selected K=5 maintain a balance between detection accuracy and semantic similarity. While higher K values provide only marginal improvements in the AUROC, they highly impact semantic integrity, making K=5 a suitable compromise.

APPENDIX B
EFFICIENCY OF SYNONYM RETRIEVAL METHODS

A potential concern with our approach is the computational overhead introduced by embedding-based synonym generation, particularly when using large language models such as BERT. To assess the practical implications, we benchmarked several synonym retrieval methods on the Enron Email dataset. The methods evaluated include SBERT, context-based BERT, two variants of our LexSub method (concatenation and dropout), and a lightweight alternative based on WordNet. The average runtime per email and total processing time are summarized in Table VII.

TABLE VII
AVERAGE RUNTIME PER EMAIL FOR DIFFERENT SYNONYM RETRIEVAL METHODS. EXPERIMENTS CONDUCTED ON AN NVIDIA RTX 6000 GPU.

Method	Average Time per Email (sec)
WordNet	0.6211
SBERT	2.0153
Concatenation	2.1300
Context BERT	2.7154
Dropout	4.8400

As shown, WordNet offers a highly efficient option that does not rely on model inference, making it suitable for large-scale or resource-constrained applications. SBERT and the LexSub concatenation method also exhibit relatively low latency, averaging around 2 seconds per document. While the LexSub variant with dropout introduces more computational overhead, it remains practical for real-world deployment.

These findings demonstrate that LexiMark supports efficient and flexible deployment. Depending on the available computational resources and desired fidelity of contextual understanding, users can choose between fast dictionary-based methods or more sophisticated neural approaches.