

Article

PrivRewrite: Differentially Private Text Rewriting Under Black-Box Access with Refined Sensitivity Guarantees

Jongwook Kim 

Department of Computer Science, Sangmyung University, Seoul 03016, Republic of Korea; jkim@smu.ac.kr

Abstract

Text data is indispensable for modern machine learning and natural language processing but often contains sensitive information that must be protected before sharing or release. Differential privacy (DP) provides rigorous guarantees for privacy preservation, yet applying DP to text rewriting poses unique challenges. Existing approaches frequently assume white-box access to large language models (LLMs), relying on internal signals such as logits or gradients. These assumptions limit practicality, since real-world users typically interact with LLMs only through black-box APIs. We introduce PrivRewrite, a framework for differentially private text rewriting that operates entirely under black-box access. PrivRewrite constructs a diverse pool of candidate rewrites through randomized prompting and pruning and then employs the exponential mechanism to select a single release with end-to-end ϵ -DP. A key contribution is our refined sensitivity analysis of the utility function, which yields tighter bounds than naive estimates and thereby strengthens the accuracy guarantees of the exponential mechanism. The framework requires no fine-tuning, internal model access, or local inference, making it lightweight and deployable in practical API-based settings. Experimental results on benchmark datasets demonstrate that PrivRewrite achieves strong privacy–utility trade-offs, producing fluent and semantically faithful outputs while upholding formal privacy guarantees. These results highlight the feasibility of black-box DP text rewriting and show how refined sensitivity analysis can further improve utility under strict privacy constraints.

Keywords: differentialprivacy; text rewriting; black-box large language models; privacy-preserving natural language processing



Academic Editor: Andrea Prati

Received: 19 September 2025

Revised: 5 November 2025

Accepted: 7 November 2025

Published: 10 November 2025

Citation: Kim, J. PrivRewrite: Differentially Private Text Rewriting Under Black-Box Access with Refined Sensitivity Guarantees. *Appl. Sci.* **2025**, *15*, 11930. <https://doi.org/10.3390/app152211930>

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid advancement of machine learning and deep learning technologies, the demand for large-scale datasets has steadily increased. In particular, free-text data are widely used in natural language processing tasks, making them an important resource for model training and evaluation. However, in sensitive domains such as healthcare, law, and finance, text data often contain private and sensitive information [1–5]. Thus, sharing such data, even for legitimate research or application development, raises serious privacy concerns and may violate legal regulations such as HIPAA [6] or GDPR [7]. This creates a fundamental challenge: how to make use of valuable textual data while preserving individual privacy.

To address the privacy risks associated with text data sharing, differential privacy (DP) [8–10] has been explored as a formal mechanism to prevent the original input text from revealing sensitive content. Existing approaches have applied DP at the token level,

for example by injecting noise into word embeddings or substituting tokens with randomly selected alternatives according to DP-calibrated probabilities [11–13]. While these methods provide theoretical privacy guarantees, they often yield ungrammatical or semantically distorted outputs, which limits their practical utility. Sentence-level approaches apply DP to entire sequences, aiming to preserve both meaning and fluency while enforcing privacy constraints [14]. Nevertheless, these methods can still suffer from degraded semantic accuracy under stricter privacy budgets, and their utility often varies considerably across different downstream tasks.

With the widespread adoption of large language models (LLMs) [15,16], increasing efforts have focused on combining LLMs with DP to sanitize sensitive text data. Recent approaches typically leverage the generative or rewriting capabilities of LLMs to produce paraphrased outputs that satisfy formal DP constraints. These methods generally assume white-box access to the underlying model, including visibility into internal components such as logits or attention weights [17–19]. Such access enables fine-grained control over the generation process and facilitates DP mechanisms that depend on token-level scores. Although effective in controlled environments, these approaches require running and managing the model locally, which can be impractical in resource-constrained or deployment-sensitive scenarios.

While white-box methods provide fine-grained control over text generation, they rely on access to internal model components such as logits or attention weights. This typically requires running the model locally, which is often infeasible in practical scenarios. In contrast, most real-world users interact with LLMs through commercial APIs that only support input–output access, without exposing internal computations [20–22]. Under this constraint, it is important to design DP mechanisms that function entirely in a black-box setting, without depending on internal access, fine-tuning, or local inference. Such approaches would enable wider adoption of DP-based text rewriting, particularly in resource-limited or tightly regulated environments.

Thus, in this paper, we propose PrivRewrite, a text rewriting framework that provides formal DP guarantees using only black-box access to LLMs. The method requires no internal model access, fine-tuning, or local inference, making it lightweight and practical for real-world deployment. PrivRewrite consists of two key components: (i) sanitized candidate generation via an LLM and (ii) a DP selection mechanism applied to the candidate outputs. This design achieves end-to-end ϵ -DP while preserving semantic fidelity and fluency, even in scenarios where only API-based LLM access is available.

Unlike prior methods that assume internal access to LLMs, PrivRewrite demonstrates that formal DP can be achieved entirely under black-box access, enabling practical deployment in API-based and enterprise environments. This design aligns with the dominant deployment paradigm of contemporary LLM applications, where models are accessed through hosted APIs rather than locally operated white-box systems. By removing the need for access to model internals such as logits, gradients, or parameters, PrivRewrite extends formal DP protection to realistic and compliance-sensitive settings where white-box approaches cannot be applied. The contributions of this paper can be summarized as follows:

- We propose PrivRewrite, a novel method for DP text rewriting that achieves formal ϵ -DP guarantees using only black-box access to LLMs. The approach is lightweight, practical, and avoids reliance on model internals, fine-tuning, or local inference.
- We formally define the selection utility for the exponential mechanism in the text-rewriting setting and derive its global sensitivity. Our analysis yields a tighter sensitivity bound than naive estimates, enabling improved utility at the same privacy level.

- We conduct extensive experiments to assess the privacy–utility trade-off of PrivRewrite. Results demonstrate that our method achieves high semantic quality while maintaining formal DP guarantees and remains effective even in constrained API-only access scenarios.

The remainder of this paper is organized as follows. Section 2 reviews related work, and Section 3 provides necessary background and formally defines the problem addressed in this paper. Section 4 details the proposed PrivRewrite framework. Section 5 evaluates the proposed approach through experiments conducted on real-world datasets, and Section 6 presents the conclusions of the paper.

2. Related Work

Early efforts to apply DP to text focused on perturbing individual words using word embeddings. Feyisetan et al. [11] introduce a mechanism based on $d\chi$ -privacy, where each word is mapped into a vector space using a pre-trained embedding model, followed by nearest-neighbor decoding. Xu et al. [12] propose a differentially private text perturbation method based on a regularized Mahalanobis metric, which injects elliptical noise into word embeddings to improve privacy guarantees. Carvalho et al. [13] propose the truncated exponential mechanism, a general metric-DP method that privatizes words by selecting from nearby candidates using the exponential mechanism. Zhou et al. [23] propose TextObfuscator, a method that obfuscates word embeddings via random cluster-based perturbation to protect privacy. Yue et al. [24] propose SANTEXT and SANTEXT+, token-wise local DP mechanisms that sanitize raw text into readable output using embedding-based distance metrics. Chen et al. [25] extend this line of work by introducing CusText, a token-level sanitization method that avoids metric assumptions and instead privatizes each token using a customized candidate set and the exponential mechanism.

In contrast to word-level approaches, sentence-level DP methods aim to rewrite entire sentences to preserve global coherence. Meehan et al. [14] propose SentDP, a sentence-level local DP framework for document embeddings, which guarantees that any sentence in a document can be substituted without significantly changing the embedding. Bollegala et al. [26] propose CMAG, a metric DP mechanism for sentence embeddings that adapts Mahalanobis noise to local embedding geometry for improved privacy-utility tradeoff.

Recently, several methods have leveraged LLMs for sentence-level text rewriting under DP. Mattern et al. [17] propose a sentence-level anonymization approach based on paraphrasing with fine-tuned transformer models, offering formal DP guarantees while addressing the semantic and syntactic limitations of word-level methods. Utpala et al. [18] propose DP-Prompt, a sentence-level privacy mechanism that applies local DP using zero-shot prompting with LLMs. Meisenbacher et al. [19] propose DP-MLM, a sentence-level differentially private text rewriting method that leverages white-box access to masked language models to perform token-wise privatization via temperature-controlled sampling under a formal exponential mechanism. In contrast to these methods, our approach is fully black-box and operates solely through input-output interaction with LLM APIs, without requiring access to model internals or gradient information.

Another line of work applying DP in these days focuses on protecting private information in LLM prompting workflows, including both the user prompts and the data used to generate them. InferDPT [20] introduces a black-box framework that ensures DP by perturbing user prompts before submitting them to LLM APIs, aiming to prevent sensitive information leakage during inference. DP-GTR [27] introduces a local DP framework that protects sensitive prompts submitted to LLMs by combining group text rewriting and in-context learning to balance privacy and utility. EmojiPrompt [28] obfuscates sensitive prompts via symbolic transformations (e.g., emojis) using generative LLMs. While not

formally DP, it adopts DP-inspired constraints to mitigate inference-time leakage. Split-and-Denoise [29] applies local DP to protect sensitive prompts by perturbing their token embeddings before sending them to the LLM. A local denoising model is then used to recover utility, achieving a privacy-utility tradeoff in black-box LLM inference. Cape [30] introduces a context-aware prompt perturbation mechanism using local DP, which enhances semantic coherence by combining token embedding distance and contextual logits in its utility function. To mitigate the long-tail sampling problem over large vocabularies, Cape adopts a bucketized exponential mechanism. DP-OPT [31] focuses on protecting sensitive training data used for prompt tuning. It uses a DP ensemble method to generate discrete prompts entirely on the client side, which are then deployed to black-box LLMs for inference. While DP-OPT does not obfuscate the prompt itself, it ensures that no private data can be reconstructed from the prompt, thus preventing leakage during prompt tuning.

Beyond DP, recent studies have extensively explored security and trust issues in LLMs. Zhou et al. [32] provide a comprehensive survey of backdoor threats in LLMs, categorizing attacks across the pre-training, fine-tuning, and inference phases, and summarizing existing defenses and evaluation methodologies. Wang et al. [33] review LLM-assisted program analysis, outlining static, dynamic, and hybrid pipelines for tasks such as vulnerability detection, malware analysis, and verification. Jaffal et al. [34] survey cybersecurity applications of LLMs, highlighting vulnerabilities such as data poisoning, backdoors, and prompt injection, along with emerging defense strategies. Choi et al. [35] analyze whether ChatGPT-style code transformations can evade authorship attribution and demonstrate that a feature-based method retains strong attribution accuracy. Lin and Mohaisen [36] conduct a comparative evaluation of multiple LLM families for vulnerability detection, examining how model size, quantization, and context window affect performance across programming languages. Alghamdi and Mohaisen [37] employ BERT-based models to assess the transparency of AR/VR application privacy policies. Lin and Mohaisen [38] evaluate LLM robustness in vulnerability detection, showing that performance strongly depends on tokenized input length and context window configuration.

3. Preliminary and Problem Definition

In this section, we present the necessary preliminaries, formally define the problem, and describe the threat model and assumptions considered in this work.

3.1. Preliminary

DP is a rigorous framework ensuring that the output of a randomized algorithm does not significantly change when a single individual's data is modified. As a result, an adversary, even with arbitrary external knowledge, cannot confidently determine whether a specific individual's data was included in the computation [8]. A randomized mechanism $\mathcal{A}$ satisfies ϵ -DP if, for any two neighboring inputs x and x' , and any subset $S \subseteq \mathcal{X}$, the following condition holds:

$$\Pr[\mathcal{A}(x) \in S] \leq e^\epsilon \cdot \Pr[\mathcal{A}(x') \in S].$$

Here, neighboring inputs refer to input datasets x and x' that differ in at most one individual's data record. In the context of text rewriting, this typically means that the two inputs differ by a single sentence, document, or token, depending on the granularity of the privacy guarantee. This ensures that the presence or absence of a single data point has a limited influence on the output, thereby protecting individual privacy.

The parameter ϵ , known as the privacy budget, quantifies the strength of the privacy guarantee. A smaller value of ϵ implies stronger privacy, as it forces the output distributions under x and x' to be nearly indistinguishable. However, this typically requires injecting

more randomness, which can reduce output utility. On the other hands, a larger ε permits weaker privacy, but allows the mechanism to preserve more useful information with less noise. Thus, selecting ε involves a trade-off between privacy and utility.

The exponential mechanism is a general-purpose DP mechanism that selects an output from a discrete set based on a utility function, while preserving ε -DP. It is particularly useful when the output space is non-numeric and standard noise-addition methods, such as the Laplace or Gaussian mechanisms, are not applicable.

Let $\mathcal{C}$ be a finite set of possible outputs, and let $u(x, c)$ be a utility function that evaluates the quality of each candidate $c \in \mathcal{C}$ with respect to the input x . The exponential mechanism selects an output c with probability proportional to:

$$\Pr[\mathcal{A}(x) = c] \propto \exp\left(\frac{\varepsilon \cdot u(x, c)}{2\Delta u}\right),$$

where Δu is the sensitivity of the utility function, defined as:

$$\Delta u = \max_{c \in \mathcal{C}} \max_{x, x'} |u(x, c) - u(x', c)|,$$

for all neighboring inputs x and x' . As long as the sensitivity Δu is bounded, the exponential mechanism ensures ε -DP. It is particularly suited for tasks like text rewriting, where the goal is to select a high-utility output from a set of candidates generated by a language model.

3.2. Problem Definition

In this paper, we consider the task of privatized text rewriting. Given a sensitive input sequence $x = (x_1, x_2, \dots, x_T)$ consisting of T tokens, our objective is to design a randomized mechanism $\mathcal{M}$ that outputs a rewritten text $\tilde{x}$ while meeting three requirements:

- **Utility:** $\tilde{x}$ should preserve the semantic content of x and remain fluent and coherent.
- **Privacy:** $\mathcal{M}$ must satisfy ε -DP with respect to x .
- **Practicality:** The mechanism should operate with only black-box access to a language model, without relying on internal components such as logits, gradients, or model weights.

To formalize the privacy requirement, we first specify the notion of neighboring inputs. We adopt a token-level definition: two sequences $x = (x_1, \dots, x_T)$ and $x' = (x'_1, \dots, x'_T)$ are neighbors, written $x \sim x'$, if they differ in at most one token position, i.e.,

$$d_H(x, x') = |\{j \in \{1, \dots, T\} : x_j \neq x'_j\}| \leq 1.$$

With this definition of neighbors, a randomized mechanism $\mathcal{M}$ satisfies ε -DP if, for all $x \sim x'$ and all measurable subsets S of outputs,

$$\Pr[\mathcal{M}(x) \in S] \leq e^\varepsilon \Pr[\mathcal{M}(x') \in S].$$

In summary, the problem studied in this paper is to design a mechanism $\mathcal{M}$ that maps each sensitive text x to a privatized rewrite $\tilde{x}$, ensuring semantic fidelity, ε -DP at the token level, and deployability under black-box LLM access.

3.3. Threat Model and Assumptions

We consider two adversaries in the privatized text rewriting setting. The first is the provider-side adversary A_{prov} , representing the external LLM provider that receives all queries through the API. Since each input $x = (x_1, \dots, x_T)$ is first processed by a local ε_1 -DP sanitizer, the provider does not see the raw text but only a sanitized version that already

satisfies DP. The second is the output-side adversary A_{out} , which observes only the final rewrite $\tilde{x} = \mathcal{M}(x)$. This output is produced through an additional ε_2 -DP selection step, ensuring that even with arbitrary auxiliary information, A_{out} cannot reliably distinguish neighboring inputs.

This model reflects realistic deployment conditions. Major LLM providers offer enterprise configurations with contractual assurances that submitted prompts are not stored or reused for training. Although such assurances are operational rather than formal guarantees, our mechanism provides provable protection: the provider sees only locally privatized queries, while the external observer sees only the final DP-protected output.

4. Proposed Method

In this section, we present *PrivRewrite*, a black-box text rewriting mechanism that satisfies ε -DP. The proposed method converts each input sentence into a single privatized rewrite through two phases, illustrated in Figure 1.

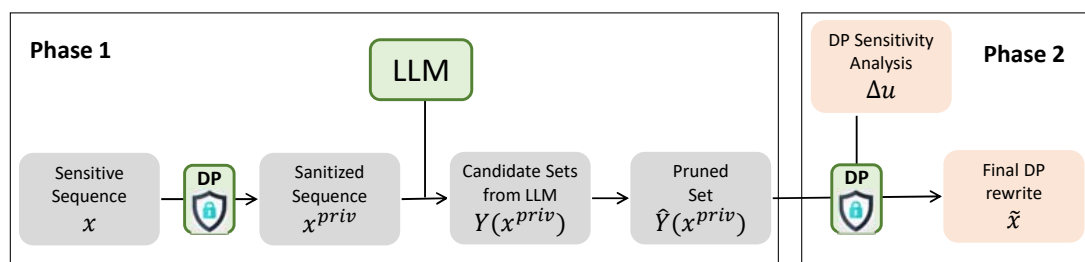


Figure 1. Overview of the PrivRewrite process with two phases: sanitized candidate generation using a black-box LLM, and differentially private selection with a tight sensitivity bound.

- Phase 1 (Sanitized candidate generation): Given an input sequence, we construct a privatized view using a per-token DP sanitizer. The sanitized text is then submitted to the black-box LLM, which produces a set of k rewrite candidates. To reduce redundancy, near-duplicate candidates are pruned based on candidate-to-candidate similarity.
- Phase 2 (Differentially private selection): From the candidate set, we choose a single rewrite using the exponential mechanism. This mechanism samples according to a bounded similarity score with respect to the input, so that the final output preserves meaning while incorporating randomized noise for privacy.

In the following subsections, we describe each phase in detail.

4.1. Phase 1 (Sanitized Candidate Generation)

Let $\mathcal{V}$ denote a fixed vocabulary and $x = (x_1, \dots, x_T) \in \mathcal{V}^T$ an input sequence. Phase 1 constructs a pruned candidate multiset in three steps: (i) sanitize x into a privatized view x^{priv} , (ii) generate candidate rewrites from a black-box LLM using x^{priv} , and (iii) remove near-duplicates based solely on candidate-to-candidate similarity.

4.1.1. Token-Level Sanitization

Given an input sequence $x = (x_1, \dots, x_T) \in \mathcal{V}^T$, we generate its privatized counterpart $x^{\text{priv}} = (x_1^{\text{priv}}, \dots, x_T^{\text{priv}}) \in \mathcal{V}^T$ using the token-level exponential mechanism, a standard local-DP approach widely applied to text rewriting. For each position $i \in [T]$, we define a bounded utility function $u(t; x_i) \in [0, 1]$ that measures the suitability of replacing the original token x_i with $t \in \mathcal{V}$ (for example, a clipped cosine similarity between public unit-norm embeddings). Because $u(\cdot; x_i)$ is bounded, its global sensitivity is 1.

The mechanism samples the privatized token according to

$$\Pr[x_i^{\text{priv}} = t \mid x_i] = \frac{\exp(\frac{\epsilon_1}{2} u(t; x_i))}{\sum_{t' \in \mathcal{V}} \exp(\frac{\epsilon_1}{2} u(t'; x_i))}.$$

Since the adjacency relation $x \sim x'$ is defined by a single-token difference and the mechanism operates independently across positions, the sanitizer $x \mapsto x^{\text{priv}}$ satisfies ϵ_1 -DP under parallel composition.

4.1.2. Candidate Generation Using LLM

In the second step, we query an LLM with the sanitized input x^{priv} to obtain multiple rewrite candidates. Formally, let $\text{LLM}_k(\cdot)$ denote a black-box API that generates k text completions in response to a given prompt. We denote the resulting candidate multiset as

$$Y(x^{\text{priv}}) = \{y^{(1)}, \dots, y^{(k)}\} = \text{LLM}_k(x^{\text{priv}}).$$

Since the query depends only on x^{priv} , this step is a randomized post-processing of the privatized sequence and does not affect the privacy guarantee.

The motivation for generating k candidates, rather than directly releasing x^{priv} , is to improve utility. Although x^{priv} already satisfies ϵ_1 -DP, it may deviate substantially from the semantics of the original input due to per-token noise. This loss of semantic fidelity is a well-known limitation of token- and word-level approaches [11–13,23]. By using x^{priv} only as a prompt to the LLM, we can leverage the model's generative capacity to produce fluent and coherent rewrites. Sampling multiple candidates increases the probability that at least one candidate aligns well with the intended meaning of x . In Phase 2, a differentially private selection mechanism chooses a single output from $Y(x^{\text{priv}})$, ensuring that the final release preserves DP while improving semantic fidelity compared to directly publishing x^{priv} .

4.1.3. Near-Duplicate Pruning

LLMs are known to produce highly similar or even near-duplicate outputs across different generations. As a result, the candidate set $Y(x^{\text{priv}})$ may contain substantial redundancy. To mitigate this, we apply a pruning step that relies only on pairwise similarity among candidates. Let $\psi : \text{Text} \rightarrow \mathbb{R}^d$ be a fixed sentence encoder with $\|\psi(y)\|_2 \leq 1$ for all y , and define the cosine-based similarity

$$s(y, y') = \frac{1}{2} \left(1 + \langle \psi(y), \psi(y') \rangle \right) \in [0, 1].$$

Here, $\langle \psi(y), \psi(y') \rangle$ denotes the inner product of the normalized embeddings, which equals their cosine similarity in $[-1, 1]$. The transformation $\frac{1}{2}(1 + \cdot)$ simply shifts and rescales this range to $[0, 1]$.

Given a similarity threshold $\tau_{\text{dup}} \in (0, 1)$ and a fixed public order π over $\{1, \dots, k\}$, we prune candidates using the following greedy procedure:

- Initialize $\hat{Y} \leftarrow \emptyset$.
- For each j in order π , add $y^{(j)}$ to $\hat{Y}$ if $s(y^{(j)}, y) < \tau_{\text{dup}}$ for every $y \in \hat{Y}$.

By construction, every pair of retained candidates is guaranteed to satisfy

$$s(y, y') < \tau_{\text{dup}} \quad \text{for all } y \neq y' \in \hat{Y},$$

which ensures that $\hat{Y}$ forms a set of mutually dissimilar rewrites. Equivalently, the procedure selects a maximal τ_{dup} -separated subset of $Y(x^{\text{priv}})$ under the similarity measure s . The parameter τ_{dup} directly controls the degree of enforced diversity: larger values impose

stricter separation and result in fewer but more distinct candidates, while smaller values allow more candidates at the cost of potential redundancy.

4.2. Phase 2 (Differentially Private Selection)

Given the pruned candidate multiset $\hat{Y} = \{y^{(1)}, \dots, y^{(m)}\}$ obtained in Phase 1, the goal of Phase 2 is to select a single rewrite that balances semantic fidelity with formal privacy protection. We achieve this using the exponential mechanism, which favors candidates that are semantically closer to the input while ensuring that the selection procedure itself satisfies DP.

4.2.1. Utility Function

To apply the exponential mechanism, we first need to specify a utility function that measures the semantic alignment between the original sequence x and any candidate $y \in \hat{Y}$, and then establish its global sensitivity. Let $v : \mathcal{V} \rightarrow \mathbb{R}^d$ be a token embedding map satisfying $\|v(t)\|_2 \leq 1$ for all $t \in \mathcal{V}$. For a sequence $x = (x_1, \dots, x_T)$, we define its average embedding as

$$\phi(x) := \frac{1}{T} \sum_{i=1}^T v(x_i) \in \mathbb{R}^d.$$

Since the Euclidean norm is convex, the average of vectors with norm at most one also has norm at most one. Hence, $\|\phi(x)\|_2 \leq 1$ for all sequences x .

Given the sensitive input x and a candidate $y \in \hat{Y}$, we define the utility function

$$u_x(y) := \text{clip}_{[0,1]} \left(\frac{1}{2} + \frac{1}{\rho} \langle \phi(x), \phi(y) \rangle \right),$$

where $\rho > 0$ is a smoothing parameter. The term $\langle \phi(x), \phi(y) \rangle$ is the Euclidean inner product between the normalized average embeddings of x and y , which lies in $[-1, 1]$ and coincides with their cosine similarity. The operator $\text{clip}_{[0,1]}$ truncates its argument into the unit interval, ensuring that $u_x(y)$ remains bounded in $[0, 1]$. The parameter ρ controls the trade-off between expressiveness and sensitivity: larger values yield a flatter utility landscape with lower sensitivity, while smaller values make the utility more responsive to semantic differences at the cost of higher sensitivity.

Given the utility function $u_x(\cdot)$, the next step is to quantify its global sensitivity, which determines the noise level used by the exponential mechanism. While the trivial bound $\Delta u = 1$ follows from $u_x(\cdot) \in [0, 1]$, it is overly conservative and would inject unnecessary noise, harming utility. Instead, we bound how much $u_x(y)$ can change when x and x' differ in a single token, yielding the following result.

Lemma 1 (Global sensitivity of u_x). *Under the neighboring relation $x \sim x'$ (differing in at most one token),*

$$\Delta u := \sup_{x \sim x'} \sup_{y \in \hat{Y}} |u_x(y) - u_{x'}(y)| \leq \frac{2}{T\rho}.$$

Here, $\sup$ denotes the supremum, i.e., the worst-case value over all neighboring inputs and all candidates.

Proof. Let x, x' be neighboring sequences that differ only at position j . By definition, their average embeddings are

$$\phi(x) = \frac{1}{T} \sum_{i=1}^T v(x_i), \quad \phi(x') = \frac{1}{T} \sum_{i=1}^T v(x'_i).$$

Since all terms cancel except at index j , the difference reduces to

$$\phi(x) - \phi(x') = \frac{1}{T} (v(x_j) - v(x'_j)).$$

By taking the Euclidean norm and applying the triangle inequality, we obtain

$$\|\phi(x) - \phi(x')\|_2 = \frac{1}{T} \|v(x_j) - v(x'_j)\|_2 \leq \frac{1}{T} (\|v(x_j)\|_2 + \|v(x'_j)\|_2) \leq \frac{2}{T}.$$

For any fixed $y \in \hat{Y}$, define

$$s := \langle \phi(x), \phi(y) \rangle, \quad s' := \langle \phi(x'), \phi(y) \rangle.$$

Then, by Cauchy–Schwarz,

$$\begin{aligned} |s - s'| &= |\langle \phi(x), \phi(y) \rangle - \langle \phi(x'), \phi(y) \rangle| \\ &= |\langle \phi(x) - \phi(x'), \phi(y) \rangle| \leq \|\phi(x) - \phi(x')\|_2 \|\phi(y)\|_2 \leq \frac{2}{T}. \end{aligned}$$

Define $h(t) := \text{clip}_{[0,1]}(\frac{1}{2} + \frac{1}{\rho}t)$ with $\rho > 0$. Write $f(t) = \frac{1}{2} + \frac{1}{\rho}t$ and $g(z) = \text{clip}_{[0,1]}(z)$; then $h = g \circ f$. The map f is $(1/\rho)$ -Lipschitz and the projection g is 1-Lipschitz, hence h is $(1/\rho)$ -Lipschitz:

$$|h(a) - h(b)| \leq \frac{1}{\rho} |a - b| \quad \text{for all } a, b \in \mathbb{R}.$$

Applying this with $a = s, b = s'$ yields

$$|u_x(y) - u_{x'}(y)| = |h(s) - h(s')| \leq \frac{1}{\rho} |s - s'| \leq \frac{1}{\rho} \cdot \frac{2}{T} = \frac{2}{T\rho}.$$

Taking the supremum over all neighboring $x \sim x'$ and all $y \in \hat{Y}$ gives $\Delta u \leq 2/(T\rho)$. This completes the proof. $\square$

The trivial bound $\Delta u \leq 1$ holds because $u_x(y) \in [0, 1]$. However, Lemma 1 provides the sharper estimate $\Delta u \leq \frac{2}{T\rho}$. Thus, in general, we obtain

$$\Delta u \leq \min\left\{1, \frac{2}{T\rho}\right\}.$$

In typical settings with $T > 2$ and $\rho \geq 2$ (avoiding clipping), this simplifies to $\Delta u \leq \frac{1}{T} < 1$. Thus, the exponential mechanism can operate with a tighter sensitivity bound, leading to stronger concentration on high-utility candidates and improved semantic fidelity at the same privacy budget.

4.2.2. Exponential-Mechanism Selection

Given the pruned candidate set $\hat{Y} = \{y^{(1)}, \dots, y^{(m)}\}$, the final output is selected using the exponential mechanism with privacy budget ε_2 . By Lemma 1, the global sensitivity of the utility function is bounded as

$$\Delta u = \min\left\{1, \frac{2}{T\rho}\right\}.$$

The exponential mechanism defines a probability distribution over $\hat{Y}$ that favors candidates with higher utility while ensuring ε_2 -DP. Specifically, the probability of selecting candidate $y \in \hat{Y}$ is

$$\Pr[\tilde{x} = y \mid x, \hat{Y}] = \frac{\exp(\frac{\varepsilon_2}{2\Delta u} u_x(y))}{\sum_{y' \in \hat{Y}} \exp(\frac{\varepsilon_2}{2\Delta u} u_x(y'))}.$$

The released privatized rewrite $\tilde{x}$ is then drawn according to this distribution. By construction, the mechanism satisfies ε_2 -DP with respect to the input x , and the tighter bound on Δu directly reduces the amount of randomness required, thereby improving the fidelity of the selected output.

4.3. Privacy Guarantee and Utility Analysis

This subsection provides the theoretical analysis of the proposed mechanism, addressing both its DP guarantee and the utility of the selection step.

4.3.1. Privacy Guarantee

The overall privacy follows directly from the composition of the two phases.

Theorem 1 (End-to-End Privacy). *Let ε_1 and ε_2 be the privacy budgets used in Phase 1 and Phase 2, respectively. Under the token-level neighboring relation, the mechanism that outputs the final rewrite $\tilde{x}$ satisfies $(\varepsilon_1 + \varepsilon_2)$ -DP.*

Proof. Phase 1 (sanitization). Each token x_i is privatized independently using an ε_1 -DP mechanism. Under token-level adjacency, parallel composition implies that the entire sanitized sequence x^{priv} is ε_1 -DP.

Phase 2 (selection). Conditioned on the pruned candidate set $\hat{Y}$, the exponential mechanism with privacy budget ε_2 and sensitivity bound Δu (Lemma 1) satisfies ε_2 -DP with respect to the original input x .

Composition. The final output $\tilde{x}$ is obtained by applying Phase 2 to the output of Phase 1. By the sequential composition theorem, the overall mechanism is $(\varepsilon_1 + \varepsilon_2)$ -DP. $\square$

4.3.2. Utility Analysis and Fallback Under Post-Processing

The exponential mechanism employed in Phase 2 provides a standard utility guarantee. With probability at least $1 - \delta$, the selected candidate $\tilde{x} \in \hat{Y}$ achieves utility close to that of the best element in $\hat{Y}$ [39]:

$$u_x(\tilde{x}) \geq \max_{y \in \hat{Y}} u_x(y) - \frac{2\Delta u}{\varepsilon_2} \left(\ln m + \ln \frac{1}{\delta} \right), \quad m = |\hat{Y}|.$$

Here, $u_x(\tilde{x})$ denotes the utility of the selected candidate with respect to the input x , and $\max_{y \in \hat{Y}} u_x(y)$ is the maximum achievable utility among all candidates in the pruned set $\hat{Y}$. The parameter m is the size of the candidate set, and δ is a confidence parameter that quantifies the probability of failure. In other words, with probability at least $1 - \delta$, the selected output is nearly as good as the best available candidate.

Lemma 1 shows that the sensitivity decreases as $\Delta u \leq 2/(T\rho)$, so the additive error term in the exponential-mechanism bound, $\frac{2\Delta u}{\varepsilon_2} (\ln m + \ln(1/\delta))$, scales as $O(1/T)$ for fixed ε_2 and m . Pruning further decreases m , which in turn reduces the $\ln m$ penalty in the bound. Together, these factors strengthen the utility of Phase 2 without altering the DP guarantee. However, Phase 1 may become more difficult as T increases, since maintaining semantic fidelity becomes harder for longer inputs. Token-level perturbations accumulate in the sanitized prompt, reducing the likelihood that a fixed candidate budget k yields a high-fidelity rewrite. Hence, while larger T theoretically improves the privacy–utility trade-off in Phase 2, the overall performance may vary due to these Phase 1 effects rather than the DP mechanism itself.

In deployment, some inputs may be heavily corrupted, resulting in incoherent or low-quality candidate rewrites. To maintain stable system behavior in such cases, the framework can incorporate simple fallback procedures that operate entirely as post-processing. When

all generated candidates fail to produce a coherent rewrite, the system can either return the sanitized input generated in Phase 1, which already satisfies ϵ_1 -DP, or abstain from releasing a rewrite. Because these operations occur after the differentially private mechanism, they preserve the end-to-end $(\epsilon_1 + \epsilon_2)$ -DP guarantee while preventing the release of meaningless outputs in practice.

5. Experiments

In this section, we evaluate the proposed scheme using real datasets. We first describe the experimental setup and then present a discussion of the results.

5.1. Experimental Setup

Datasets: We evaluate the performance of the proposed method using real-world datasets to demonstrate its practical applicability. We use the MedQuAD dataset [40], which contains consumer health questions paired with authoritative answers from trusted sources, and randomly sample 500 question–answer pairs for evaluation. This dataset allows us to assess performance on specialized text. We also use the IMDB Movie Reviews dataset [41], a sentiment analysis corpus consisting of movie reviews labeled as positive or negative, from which we randomly sample 1000 reviews for evaluation. This dataset represents opinionated, more stylistically varied text, which helps evaluate the robustness of our method across different writing styles.

Baselines: We evaluate the performance of our method against the following alternatives:

- WordPerturb (WP) [11]: A word-level privatization approach based on d_χ -privacy, a metric-based relaxation of DP. Each word embedding is perturbed with calibrated noise in the vector space and then mapped back to the nearest vocabulary word. This provides metric-DP guarantees while aiming to preserve semantic meaning.
- Exponential mechanism-based text rewriting (EM): An approach that applies the exponential mechanism [39] to rewrite each token independently. This corresponds to directly releasing the token-level sanitized text produced in Phase 1.
- PrivRewrite with naive sensitivity (PrivRewrite-Naive): A variant of our method where the global sensitivity in Phase 2 is conservatively set to $\Delta u = 1$. This baseline isolates the effect of our proposed tight sensitivity analysis.
- PrivRewrite: The proposed approach introduced in this paper, which combines token-level sanitization (Phase 1) with tight-sensitivity exponential selection (Phase 2).

We emphasize that EM, PrivRewrite-Naive, and PrivRewrite all satisfy standard ϵ -DP, whereas WP is based on the relaxed notion of d_χ -privacy. Therefore, the numeric privacy parameter used by WP is not directly comparable to ϵ under our token-level adjacency. In this study, WP is included only as a metric-DP reference to illustrate relative utility trends under a weaker privacy notion. All quantitative performance statements and comparisons are made among the ϵ -DP methods (EM, PrivRewrite-Naive, and PrivRewrite), while WP is discussed qualitatively for contextual reference.

Evaluation metrics and implementation: Given an input sequence x and its privatized rewrite $\tilde{x}$, we assess utility using two complementary measures of semantic preservation. First, we compute cosine similarity between Sentence-BERT embeddings of x and $\tilde{x}$ [42]; we refer to this measure as SBERT-Cos for brevity. Formally, let $f(\cdot)$ denote the Sentence-BERT encoder. Then, SBERT-Cos is defined as

$$\text{SBERT-Cos}(x, \tilde{x}) = \frac{\langle f(x), f(\tilde{x}) \rangle}{\|f(x)\| \|f(\tilde{x})\|}.$$

Second, we report BERTScore [43], a widely used metric for evaluating semantic overlap based on contextualized token embeddings from pretrained transformers. Together, these metrics capture both coarse-grained text similarity and fine-grained semantic alignment.

For implementation, we use the Gemini-2.0-Flash model accessed via API to generate candidate rewrites. The model is configured with temperature = 0.75 and a maximum generation length of 6000 tokens. The privacy budget ϵ is varied over $\{0.25, 0.5, 1.0, 2.0, 4.0\}$ and is evenly split between Phase 1 and Phase 2, i.e., $\epsilon_1 = \epsilon_2 = \epsilon/2$. In Phase 1, the number of candidates generated by the LLM is varied in the range $[10, 50]$, with a default value of $k = 50$ when not specified explicitly. The threshold for near-duplicate pruning τ_{dup} is varied in the range $[0.6, 1.0]$, with a default value of $\tau_{\text{dup}} = 0.8$. In Phase 2, the scaling parameter ρ is varied in the range $[1.0, 8.0]$, with a default value of $\rho = 2$ when not specified explicitly.

5.2. Experimental Results

Before evaluating the final released privatized rewrite $\tilde{x}$, we first assess the utility of Phase 1 under varying privacy budgets ϵ . To this end, we measure two types of semantic similarity. First, we compute the average SBERT-Cos between each input x and its privatized view x^{priv} , which quantifies the direct utility loss introduced by the token-level DP sanitizer. Second, we compute the average SBERT-Cos between x and all candidates in $Y(x^{\text{priv}})$, which reflects how much semantic fidelity is recovered through LLM-based candidate generation.

As shown in Table 1, the similarities between x and x^{priv} are significantly lower, but the candidate scores remain substantially higher and relatively stable across different ϵ values. This gap highlights the recovery ability of the LLM. Even when the input is heavily perturbed, candidate generation restores much of the lost semantic content. Maintaining strong utility at this stage is crucial, as it allows Phase 2 to focus on differentially private selection without being constrained by low-quality candidates.

Table 1. Average SBERT-Cos under varying privacy budgets ϵ . We report similarity between the input x and its privatized view x^{priv} , as well as between x and the Phase 1 candidates $Y(x^{\text{priv}})$.

Dataset	x vs. x^{priv}					x vs. Phase 1 Candidates				
	$\epsilon = 0.25$	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 2.0$	$\epsilon = 4.0$	$\epsilon = 0.25$	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 2.0$	$\epsilon = 4.0$
MedQuAD	0.708	0.725	0.737	0.736	0.746	0.798	0.801	0.802	0.806	0.809
IMDB	0.679	0.689	0.696	0.713	0.722	0.772	0.778	0.788	0.792	0.798

Figure 2 compares the utility of the final privatized rewrites across privacy budgets $\epsilon \in \{0.25, 0.5, 1.0, 2.0, 4.0\}$, using both SBERT-Cos and BERTScore on MedQuAD and IMDB. These two measures capture complementary aspects of semantic preservation, with SBERT-Cos reflecting overall sentence-level similarity and BERTScore providing a more fine-grained token-level alignment.

Across both datasets and metrics, PrivRewrite consistently achieves the strongest utility. Even under small privacy budgets where the injected noise is most severe, the privatized rewrites produced by PrivRewrite remain semantically close to the original inputs. This shows that the combination of token-level sanitization in Phase 1 and tight-sensitivity exponential selection in Phase 2 effectively balances the trade-off between privacy and utility. Importantly, the LLM plays a key role in this pipeline. Although Phase 1 inevitably reduces semantic fidelity by injecting noise, the candidate generation step leverages the LLM's expressive capacity to recover much of the lost utility, as we showed in the results of Table 1. The tight sensitivity calibration in Phase 2 then ensures that the exponential mechanism can reliably favor these high-quality candidates, assigning higher probability

to semantically faithful outputs without exceeding the privacy budget. This explains the steady improvement over the naive baseline.

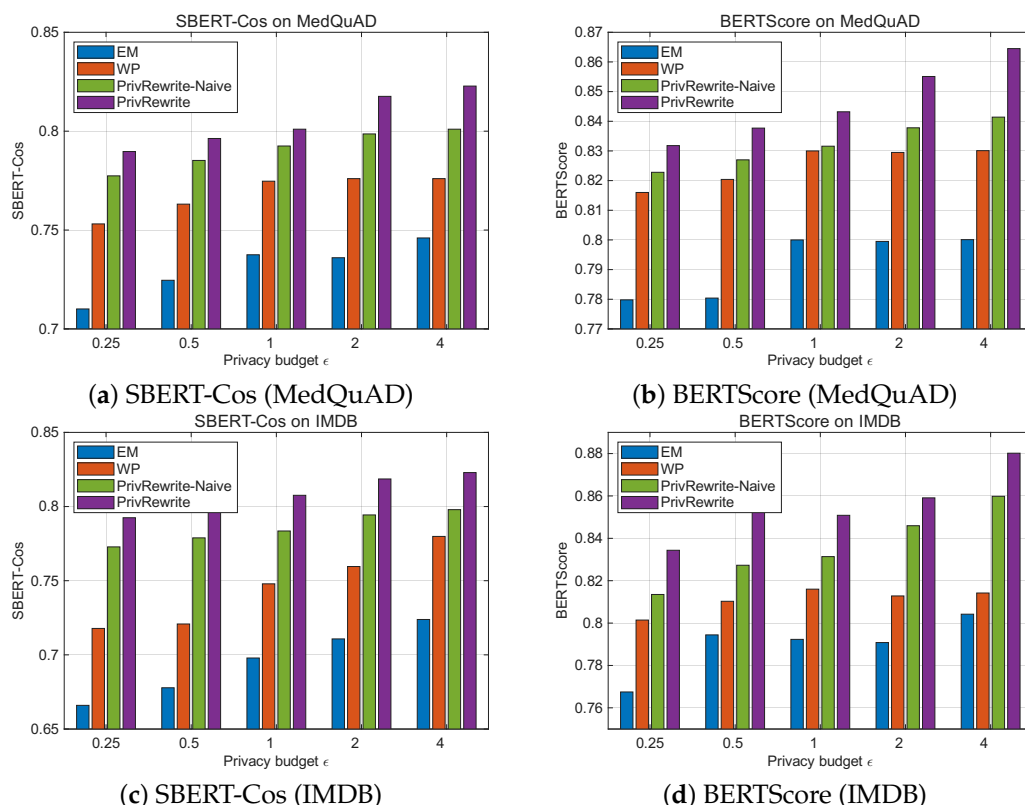


Figure 2. Average SBERT-Cos and BERTScore between input sequences and their final privatized rewrites under varying privacy budgets ϵ . WP is based on metric DP rather than standard ϵ -DP and is included only as a reference for weaker privacy guarantees.

PrivRewrite-Naive follows similar trends but does not match the performance of PrivRewrite. By fixing the global sensitivity conservatively at $\Delta u = 1$, it avoids privacy risks but sacrifices semantic quality. The performance gap between the two variants illustrates the value of our proposed sensitivity analysis. Rather than adopting a worst-case bound, calibrating the exponential mechanism with a tighter estimate leads to measurable gains across all privacy budgets.

WP, which perturbs word embeddings under d_χ -privacy, provides moderate utility. As a metric-DP method, its numeric privacy parameter is not directly comparable to ϵ under our framework, and its results are included only for contextual reference. WP performs better than the direct exponential-mechanism baseline but remains below both PrivRewrite variants. While metric-DP perturbation can retain some semantic content, it lacks the structured candidate-selection process of PrivRewrite, whose two-phase design enables LLM-generated candidates to recover much of the utility lost in the initial sanitization step. Finally, the exponential mechanism alone shows the weakest performance. This implies that directly perturbing tokens without LLM-based rewriting produces outputs that are differentially private but poorly aligned with the original input.

A comparison between SBERT-Cos and BERTScore reveals that the absolute values of BERTScore are higher, reflecting its sensitivity to fine-grained contextual token overlaps. Nevertheless, both metrics produce consistent trends: utility improves steadily as ϵ increases, all methods benefit from relaxed privacy constraints, and PrivRewrite maintains a clear advantage across settings. Together, these findings confirm that our framework adapts robustly to DP levels while preserving strong semantic fidelity.

Figure 3 shows the effect of varying ρ on utility scores with privacy budget $\varepsilon = 2.0$ on the MedQuAD dataset. We note that although the results for $\varepsilon = 2.0$ are presented here, similar trends were consistently observed across other privacy budgets. EM, WP, and PrivRewrite-Naive are plotted as reference baselines. These methods are unaffected by ρ since their mechanisms do not depend on this parameter. In contrast, PrivRewrite varies with ρ , reflecting its role in the Phase 2 utility function. The results reveal a non-monotonic pattern. Performance improves when moving from small to moderate values of ρ , reaching its peak around $\rho = 2$, but then gradually saturates and shows a slight decline for larger values. This behavior matches the theoretical insight that increasing ρ tightens the sensitivity bound and initially enhances selection quality, but excessive smoothing flattens the utility landscape and reduces the mechanism's ability to discriminate among candidates. Both SBERT-Cos and BERTScore exhibit this trend, with BERTScore reporting slightly higher absolute values due to its token-level granularity. The consistency across metrics confirms that PrivRewrite achieves the best utility at moderate values of ρ while remaining robust across a broad range.

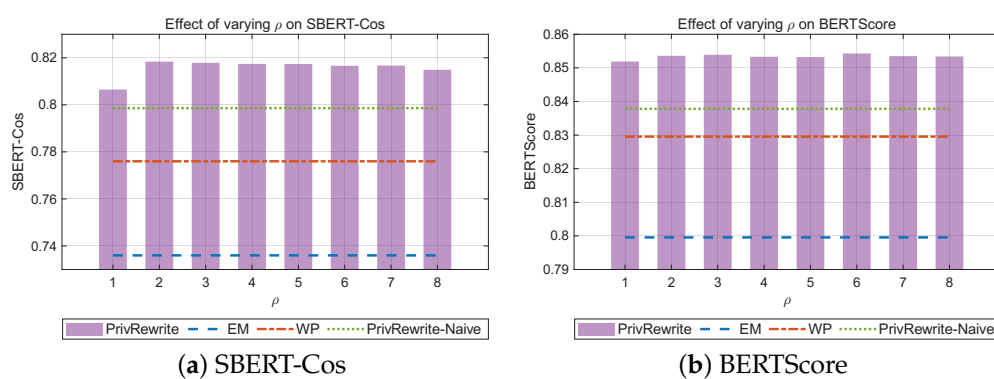


Figure 3. Effect of varying ρ on utility scores with privacy budget $\varepsilon = 2.0$ on the MedQuAD dataset. WP is based on metric DP rather than standard ε -DP and is included only as a reference for weaker privacy guarantees.

Figure 4 illustrates the impact of the near-duplicate pruning threshold τ_{dup} on utility for $\varepsilon = 2.0$ using the MedQuAD dataset. EM and WP are included as reference baselines since they are unaffected by τ_{dup} . The case $\tau_{\text{dup}} = 1.0$ corresponds to no pruning. As τ_{dup} decreases from 1.0 to approximately 0.7–0.8, the utility improves, indicating that removing highly similar candidates enhances diversity and provides the exponential-mechanism selector with a more informative candidate pool. When τ_{dup} becomes too small, however, performance declines because excessive pruning limits coverage and removes semantically relevant alternatives. This behavior reflects the inherent trade-off between candidate diversity and completeness: moderate pruning reduces redundancy and improves Phase 2 selection, whereas overly aggressive pruning constrains the mechanism's ability to identify high-utility rewrites.

Figure 5 presents the effect of different allocations of the total privacy budget ε between Phase 1 and Phase 2, evaluated using the MedQuAD dataset. In this experiment, ε is fixed to 2.0, and the allocation ratios ($\varepsilon_1 : \varepsilon_2$) are varied over 1:9, 3:7, 5:5, 7:3, and 9:1. EM and WP are plotted as horizontal reference baselines since their performance is unaffected by the split. For the two-phase methods, PrivRewrite and PrivRewrite-Naive, performance remains stable across moderate variations but reaches its best value under a balanced split. When the allocation of the privacy budget becomes highly unbalanced, the overall utility declines. This occurs because an insufficient budget in Phase 1 limits the generation of semantically meaningful rewrite candidates, while an inadequate budget in Phase 2 reduces the ability of the exponential mechanism to reliably select the highest-utility candidate.

These results indicate that an even allocation of the privacy budget across the two phases provides the most stable performance.

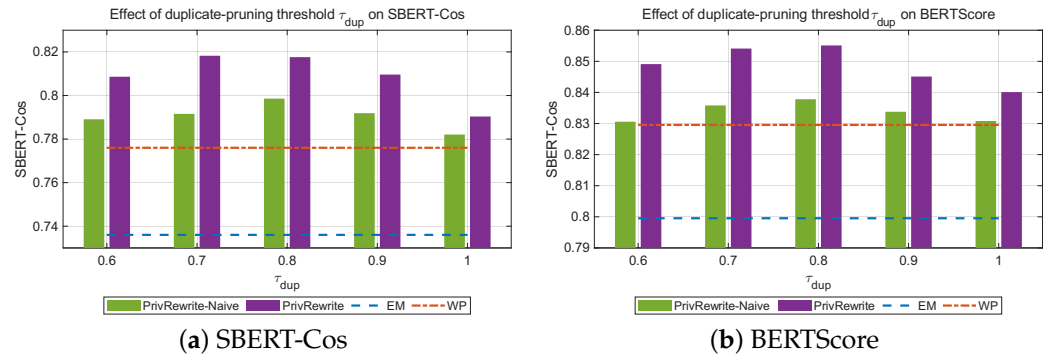


Figure 4. Effect of varying the duplicate pruning threshold τ_{dup} on utility scores for $\epsilon = 2.0$ using the MedQuAD dataset. The case $\tau_{dup} = 1$ corresponds to no duplicate pruning.

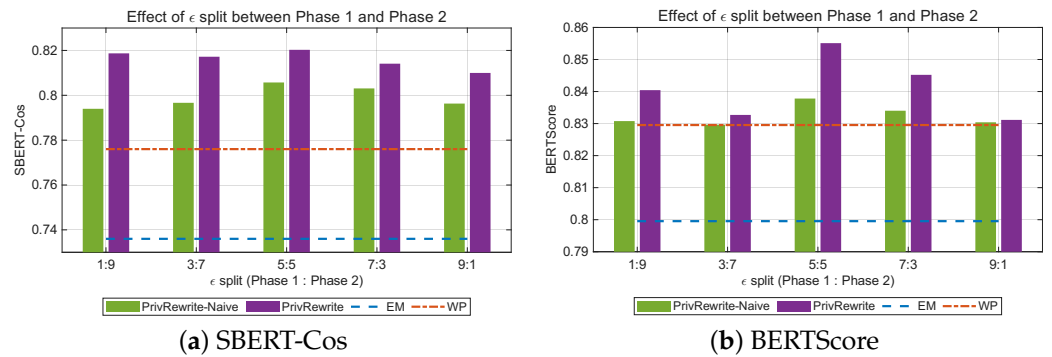


Figure 5. Impact of different splits of the total privacy budget ϵ between Phase 1 and Phase 2 on the MedQuAD dataset.

Figure 6 illustrates the impact of the candidate size k on (a) SBERT-Cos and (b) the runtime required to generate k candidates using the LLM on the MedQuAD dataset. In this experiment, the total privacy budget is fixed at $\epsilon = 2.0$, while k varies from 10 to 50. EM and WP are included as horizontal reference baselines since their performance remains constant regardless of k . As shown in Figure 6a, increasing k consistently enhances the performance of PrivRewrite and PrivRewrite-Naive. This improvement arises because a larger candidate pool allows the selection of rewrites that better preserve the semantics of the original text. However, as depicted in Figure 6b, the runtime also increases with k , which is expected since generating more candidates entails greater computational and communication overhead for the LLM.

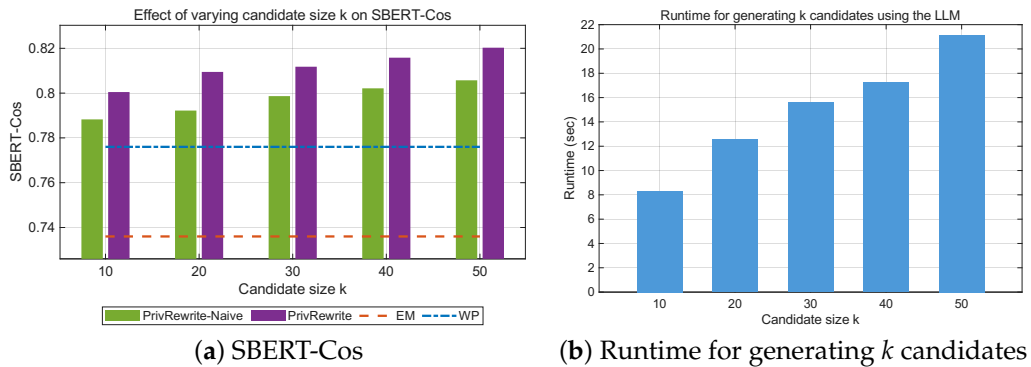


Figure 6. Effect of candidate size k on (a) SBERT-Cos and (b) runtime for generating k candidates using the LLM.

Even with a small candidate size (e.g., $k = 10$), PrivRewrite already outperforms both EM and WP, demonstrating the robustness of the proposed approach with respect to k . Moreover, since PrivRewrite targets offline text publishing scenarios for privacy-preserving rewriting, the increased runtime with larger k does not impact real-time performance. In practice, the choice of k can be adjusted according to deployment needs: smaller k values are suitable for lightweight or latency-sensitive scenarios, whereas larger k values are recommended for applications requiring higher semantic quality.

Table 2 presents example outputs from the ϵ -DP approaches (EM, PrivRewrite-Naive, and PrivRewrite) on MedQuAD. These examples are included for illustration purposes only. EM often produces broken or semantically inconsistent sentences, whereas PrivRewrite-Naive improves readability but sometimes introduces additional details that are not present in the original text. In contrast, PrivRewrite maintains a more stable tone, preserves source meaning more faithfully, and produces fluent, coherent sentences. These examples show that PrivRewrite achieves clearer and more faithful outputs than both EM and PrivRewrite-Naive.

Table 2. Representative output excerpts from EM, PrivRewrite-Naive, and PrivRewrite on MedQuAD.

Sample 1	
Method	Text
Original	Pilomatrixoma is a benign (non-cancerous) skin tumor of the hair follicle (structure in the skin that makes hair). They tend to develop in the head and neck area and are usually not associated with any other signs and symptoms (isolated).
EM	Pilomatrixoma is a impervious (basis-carcinoma) plastic nerve of the wear pineal (includes in the yellowish that always bones). They get to establishing in the over and neck situated and are usually not possible with any no real and infection (remain).
PrivRewrite-Naive	Pilomatrixoma is a benign skin tumor that originates from hair follicle cells and often contains calcifications, giving it a firm texture. These tumors commonly appear in the head and neck region, typically presenting as painless, slow-growing masses without signs of inflammation.
PrivRewrite	Pilomatrixoma is a benign skin tumor, sometimes confused with more concerning conditions. These growths tend to occur on the head and neck and typically present without pain or signs of infection.
Sample 2	
Original	Autoimmune hepatitis is a chronicor long lastingdisease in which the body's immune system attacks the normal components, or cells, of the liver and causes inflammation and liver damage. The immune system normally protects people from infection by identifying and destroying bacteria, viruses, and other potentially harmful foreign substances.
EM	Autoimmune hepatitis is a chronicor in which the similar's inhibiting provide arson the life aircraft, or bacterial, of the virus and chronic rashes and treat threatening. The protects provide if protect wanted from malignancy by diagnosing and tearing micro-organisms, viruses, and known cause deleterious foreigners toxin.
PrivRewrite-Naive	Autoimmune hepatitis is a chronic inflammatory liver disease where the body's defense system mistakenly attacks the liver. This can cause prolonged liver damage, potentially leading to severe health risks and even life-threatening complications if left untreated.
PrivRewrite	Autoimmune hepatitis is a chronic inflammatory condition affecting the liver, where the body's immune system mistakenly attacks its own liver cells. This can result in ongoing inflammation and damage to the liver, potentially leading to severe health problems and even life-threatening complications if left untreated.

6. Conclusions

In this paper, we introduced PrivRewrite, a two-phase mechanism for differentially private text rewriting. The first phase privatizes the input through token-level sanitization, and the second phase applies the exponential mechanism with a tight sensitivity bound to select among candidates. A key novelty of PrivRewrite is the integration of an LLM in a black-box fashion together with a tightly bounded exponential mechanism. The LLM receives a privatized input and generates diverse candidate rewrites without requiring access to its internal parameters or training data. PrivRewrite then applies the exponential

mechanism with the sharper sensitivity bound, which reduces unnecessary noise and allows the selection process to favor semantically faithful candidates. Experimental results on MedQuAD and IMDB demonstrated that PrivRewrite consistently outperforms existing baselines. By combining local sanitization with LLM-based generation and carefully calibrated DP selection, PrivRewrite provides a practical approach for privatized text rewriting with rigorous DP guarantees and robust utility.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF-2023R1A2C1004919).

Data Availability Statement: The original data presented in the study are openly available at <https://huggingface.co/datasets/lavita/MedQuAD/> for MedQuAD, and <https://huggingface.co/datasets/stanfordnlp/imdb/> for IMDB (accessed on 6 November 2025).

Conflicts of Interest: The author declare no conflicts of interest.

References

1. Song, S.; Kim, J. Adapting Geo-Indistinguishability for Privacy-Preserving Collection of Medical Microdata. *Electronics* **2023**, *12*, 2793. [CrossRef]
2. Saura, J.R.; Ribeiro-Soriano, D.; Palacios-Marques, D. From user-generated data to data-driven innovation: A research agenda to understand user privacy in digital markets. *Int. J. Inf. Manag.* **2021**, *60*, 102331. [CrossRef]
3. Kim, J.W.; Lim, J.H.; Moon, S.M.; Jang, B. Collecting health lifelog data from smartwatch users in a privacy-preserving manner. *IEEE Trans. Consum. Electron.* **2019**, *65*, 369–378. [CrossRef]
4. Dash, S.; Shakyawar, S.K.; Sharma, M.; Kaushik, S. Big data in healthcare: Management, analysis and future prospects. *J. Big Data* **2019**, *6*, 1–25. [CrossRef]
5. Li, M.; Liu, J.; Yang, Y. Automated Identification of Sensitive Financial Data Based on the Topic Analysis. *Future Internet* **2024**, *16*, 55. [CrossRef]
6. Health Insurance Portability and Accountability Act. Available online: <https://www.hhs.gov/hipaa/index.html> (accessed on 18 April 2025).
7. General Data Protection Regulation. Available online: <https://gdpr-info.eu/> (accessed on 18 April 2025).
8. Dwork, C. Differential privacy. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Venice, Italy, 10–14 July 2006; pp. 1–12.
9. Erlingsson, U.; Pihur, V.; Korolova, A. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 1054–1067.
10. Wang, T.; Blocki, J.; Li, N.; Jha, S. Locally differentially private protocols for frequency estimation. In Proceedings of the USENIX Conference on Security Symposium, Berkeley, CA, USA, 16–18 August 2017.
11. Feyisetan, O.; Balle, B.; Drake, T.; Diethe, T. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In Proceedings of the International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 178–186.
12. Xu, Z.; Aggarwal, A.; Feyisetan, O.; Teissier, N. A differentially private text perturbation method using regularized Mahalanobis metric. In Proceedings of the Second Workshop on Privacy in NLP, Online, 20 November 2020; pp. 7–17.
13. Carvalho, R.S.; Vasiloudis, T.; Feyisetan, O. TEM: High utility metric differential privacy on text. *arXiv* **2021**, arXiv:2107.07928. [CrossRef]
14. Meehan, C.; Mrini, K.; Chaudhuri, K. Sentence-level privacy for document embeddings. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 3367–3380.
15. Li, X.; Wang, S.; Zeng, S.; Wu, Y.; Yang, Y. A survey on LLM-based multi-agent systems: Workflow, infrastructure, and challenges. *Vicinagearth* **2024**, *1*, 9. [CrossRef]
16. Zhou, H.; Hu, C.; Yuan, Y.; Cui, Y.; Jin, Y.; Chen, C. Large Language Model (LLM) for Telecommunications: A Comprehensive Survey on Principles, Key Techniques, and Opportunities. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 1955–2005. [CrossRef]
17. Mattern, J.; Weggenmann, B.; Kerschbaum, F. The Limits of Word Level Differential Privacy. In *Findings of the Association for Computational Linguistics: NAACL 2022*; Association for Computational Linguistics: Singapore, 2022; pp. 867–881.
18. Utpala, S.; Hooker, S.; Chen, P.-Y. Locally Differentially Private Document Generation Using Zero Shot Prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*; Association for Computational Linguistics: Singapore, 2023; pp. 8442–8457.

19. Meisenbacher, S.; Chevli, M.; Vladika, J.; Matthes, F. DP-MLM: Differentially Private Text Rewriting Using Masked Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*; Association for Computational Linguistics: Singapore, 2024; pp. 9314–9328.
20. Tong, M.; Chen, K.; Zhang, J.; Qi, Y.; Zhang, W.; Yu, N.; Zhang, T.; Zhang, Z. InferDPT: Privacy-Preserving Inference for Black-box Large Language Model. *arXiv* **2023**, arXiv:2310.12214. [[CrossRef](#)]
21. Miglani, V.; Yang, A.; Markosyan, A.; Garcia-Olano, D.; Kokhlikyan, N. Using Captum to Explain Generative Language Models. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software*, Singapore, 6 December 2023; pp. 165–173.
22. Chang, Y.; Cao, B.; Wang, Y.; Chen, J.; Lin, L. XPrompt: Explaining Large Language Model’s Generation via Joint Prompt Attribution. *arXiv* **2024**, arXiv:2405.20404. [[CrossRef](#)]
23. Zhou, X.; Lu, Y.; Ma, R.; Gui, T.; Wang, Y.; Ding, Y.; Zhang, Y.; Zhang, Q.; Huang, X. TextObfuscator: Making Pre-trained Language Model a Privacy Protector via Obfuscating Word Representations. In *Findings of the Association for Computational Linguistics: ACL 2023*; Association for Computational Linguistics: Singapore, 2023; pp. 5459–5473.
24. Yue, X.; Du, M.; Wang, T.; Li, Y.; Sun, H.; Chow, S.S.M. Differential Privacy for Text Analytics via Natural Text Sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*; Association for Computational Linguistics: Singapore, 2021; pp. 3853–3866.
25. Chen, H.; Mo, F.; Wang, Y.; Chen, C.; Nie, J.-Y.; Wang, C.; Cui, J. A Customized Text Sanitization Mechanism with Differential Privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*; Association for Computational Linguistics: Singapore, 2023; pp. 4606–4621.
26. Bollegala, D.; Otake, S.; Machide, T.; Kawarabayashi, K. A Metric Differential Privacy Mechanism for Sentence Embeddings. *ACM Trans. Priv. Secur.* **2025**, *28*, 1–34. [[CrossRef](#)]
27. Li, M.; Fan, H.; Fu, S.; Ding, J.; Feng, Y. DP-GTR: Differentially Private Prompt Protection via Group Text Rewriting. *arXiv* **2025**, arXiv:2503.04990. [[CrossRef](#)]
28. Lin, S.; Hua, W.; Wang, Z.; Jin, M.; Fan, L.; Zhang, Y. EmojiPrompt: Generative Prompt Obfuscation for Privacy-Preserving Communication with Cloud-based LLMs. *arXiv* **2025**, arXiv:2402.05868. [[CrossRef](#)]
29. Mai, P.; Yan, R.; Huang, Z.; Yang, Y.; Pang, Y. Split-and-Denoise: Protect Large Language Model Inference with Local Differential Privacy. *arXiv* **2024**, arXiv:2310.09130. [[CrossRef](#)]
30. Wu, H.; Dai, W.; Wang, L.; Yan, Q. Cape: Context-Aware Prompt Perturbation Mechanism with Differential Privacy. *arXiv* **2025**, arXiv:2505.05922. [[CrossRef](#)]
31. Hong, J.; Wang, J.T.; Zhang, C.; Li, Z.; Li, B.; Wang, Z. DP-OPT: Make Large Language Model Your Privacy-Preserving Prompt Engineer. *arXiv* **2024**, arXiv:2312.03724. [[CrossRef](#)]
32. Zhou, Y.; Ni, T.; Lee, W.-B.; Zhao, Q. A Survey on Backdoor Threats in Large Language Models (LLMs): Attacks, Defenses, and Evaluation Methods. *Trans. Artif. Intell.* **2025**, *1*, 28–58. [[CrossRef](#)]
33. Wang, J.; Ni, T.; Lee, W.-B.; Zhao, Q. A Contemporary Survey of Large Language Model Assisted Program Analysis. *Trans. Artif. Intell.* **2025**, *1*, 105–129. [[CrossRef](#)]
34. Jaffal, N.O.; Alkhanafseh, M.; Mohaisen, D. Large Language Models in Cybersecurity: A Survey of Applications, Vulnerabilities, and Defense Techniques. *AI* **2025**, *6*, 216. [[CrossRef](#)]
35. Choi, S.; Alkinoon, A.; Alghuried, A.; Alghamdi, A.; Mohaisen, D. Attributing ChatGPT-Transformed Synthetic Code. In *Proceedings of the IEEE International Conference on Distributed Computing Systems*, Glasgow, Scotland, 20–23 July 2025; pp. 89–99.
36. Lin, J.; Mohaisen, D. From Large to Mammoth: A Comparative Evaluation of Large Language Models in Vulnerability Detection. In *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA, 24–28 February 2025.
37. Alghamdi, A.; Mohaisen, D. Through the Looking Glass: LLM-Based Analysis of AR/VR Android Applications Privacy Policies. In *Proceedings of the International Conference on Machine Learning and Applications*, Vienna, Austria, 21–27 July 2024; pp. 534–539.
38. Lin, J.; Mohaisen, D. Evaluating Large Language Models in Vulnerability Detection Under Variable Context Windows. In *Proceedings of the International Conference on Machine Learning and Applications*, Vienna, Austria, 21–27 July 2024; pp. 1131–1134.
39. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [[CrossRef](#)]
40. Ben Abacha, A.; Demner-Fushman, D. A Question-Entailment Approach to Question Answering. *Bmc Bioinform.* **2019**, *20*, 511. [[CrossRef](#)] [[PubMed](#)]
41. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, OR, USA, 19–24 June 2011.

42. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence embeddings using Siamese BERT networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
43. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating text generation with BERT. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.