Just Rewrite It Again: A Post-Processing Method for Enhanced Semantic Similarity and Privacy Preservation of Differentially Private Rewritten Text

Stephen Meisenbacher
Technical University of Munich
School of Computation, Information and Technology
Department of Computer Science
Garching, Germany
stephen.meisenbacher@tum.de

Florian Matthes
Technical University of Munich
School of Computation, Information and Technology
Department of Computer Science
Garching, Germany
matthes@tum.de

ABSTRACT

The study of Differential Privacy (DP) in Natural Language Processing often views the task of text privatization as a *rewriting* task, in which sensitive input texts are rewritten to hide explicit or implicit private information. In order to evaluate the privacy-preserving capabilities of a DP text rewriting mechanism, *empirical privacy* tests are frequently employed. In these tests, an adversary is modeled, who aims to infer sensitive information (e.g., gender) about the author behind a (privatized) text. Looking to improve the empirical protections provided by DP rewriting methods, we propose a simple post-processing method based on the goal of aligning rewritten texts with their original counterparts, where DP rewritten texts are rewritten *again.* Our results show that such an approach not only produces outputs that are more semantically reminiscent of the original inputs, but also texts which score on average better in empirical privacy evaluations. Therefore, our approach raises the bar for DP rewriting methods in their empirical privacy evaluations, providing an extra layer of protection against malicious adversaries.

CCS CONCEPTS

• **Security and privacy** → **Domain-specific security and privacy architectures**; • **Computing methodologies** → **Natural language processing**.

KEYWORDS

Data Privacy, Differential Privacy, Natural Language Processing

ACM Reference Format:
Stephen Meisenbacher and Florian Matthes. 2024. Just Rewrite It Again: A Post-Processing Method for Enhanced Semantic Similarity and Privacy Preservation of Differentially Private Rewritten Text. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30-August 2, 2024, Vienna, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3664476.3669926

1 INTRODUCTION

The proliferation of Large Language Models (LLMs) in recent years has given rise to discussions of data privacy in Natural Language Processing (NLP), particularly as the need for high-quality user-generated text becomes increasingly important to fuel the training of such models [3]. While LLMs have demonstrated very impressive capabilities across the spectrum of NLP tasks, the privacy risks inherent in the requirement for massive amounts of training data have motivated the study of privacy-preserving NLP [20, 30]. This need is made even more salient when considering the sheer amounts of data being passed to hosted LLMs as prompts [9].

In response to these concerns, a stream of research within the NLP community studies the integration of Differential Privacy (DP) [8] into NLP workflows. As a mathematically grounded blueprint for achieving privacy in data processing scenarios, DP offers a promising solution, yet its direct incorporation into the textual domain does not come without challenges [11, 20, 24]. Nevertheless, many innovative solutions have been proposed in recent works [17], among these the idea of *differentially private text rewriting*.

In a DP text rewriting scenario, an input text is transformed under DP guarantees with the help of a *rewriting mechanism*, which ideally outputs a privatized text that is semantically similar, yet obfuscated from the original [24]. These mechanisms often operate at the *local* level, where users rewrite their data before releasing it to some aggregator. Different rewriting mechanisms operate at various syntactic levels, such as at the word level [4, 12], sentence level [26], or document level [19]. In any scenario, the level at which a DP rewriting mechanism operates leads to the DP guarantee that is provided: for example, given a DP privatized *sentence* in the local setting, this sentence is indistinguishable from all other sentences within some bound, governed by the DP privacy parameter ε.

An important part of designing DP rewriting mechanisms is the evaluation of its privacy-preserving capabilities. In many recent works, this evaluation takes the form of *empirical privacy* tests, where the DP privatized texts are shown to reduce the ability of an attacker to perform some adversarial inference task, as compared to the non-privatized (unrewritten) baseline. In modeling such adversaries, two basic archetypes have been predominantly used in the literature, namely the *static* and *adaptive* attackers [24, 34, 37].

As shown by empirical privacy evaluations in the literature, it is often the case that the *adaptive* attacker proves to be a considerably more difficult challenge for DP rewriting mechanisms [34], as this attacker is able to mimic the rewriting process and thereby train

ARES 2024, July 30-August 2, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3669926

a more accurate adversarial model. At the same time, achieving better results in empirical privacy evaluations often necessitates lower (stricter) ε values, which in turn can lead to loss of utility, i.e., semantic similarity to the original texts. This highlights a major challenge of DP text rewriting, that is, finding the balance between privacy and utility [19, 28].

In this work, we aim to address the challenge presented by the strong capabilities of adversaries modeled in the literature, particularly with the adaptive attacker. To accomplish this goal, we aim to leverage an important property of DP, the *post-processing* principle, to propose a method in which users can enhance the privacy preservation of their DP rewritten texts. As such, we pose the following research question:

> *How can users in the local differentially private text rewriting scenario leverage language models to enhance both the empirical privacy and semantic similarity of their rewritten texts?*

To answer this question, we design, formulate, and evaluate a post-processing method that essentially rewrites the DP rewritten text *again*, with the goal of increasing its privacy while also aligning its semantics to the original text. In evaluating our proposed method, we observe that this extra post-processing step provides clear and significant privacy gains, while also often resulting in higher semantic similarity to the original text counterparts.

We make three contributions to the field of DP text rewriting:

1. To the best of the authors' knowledge, this is the first work to leverage the post-processing property of DP to improve the privacy and quality of DP rewritten texts.
2. We present a mechanism-agnostic method which demonstrates strong capabilities to enhance the privacy preservation and semantic similarity of DP rewritten texts.
3. We add to the body of knowledge on DP text rewriting evaluation by highlighting the usefulness of a post-processing step in enhancing the abilities of existing DP mechanisms.

2 FOUNDATIONS

In order to ground our work in previous literature, we now walk through key foundational concepts, which become important in motivating our proposed method.

2.1 Differentially Private Text Rewriting

The goal of differentially private text rewriting is to rewrite a sensitive input text in a manner that satisfies Differential Privacy (DP) [8]. Specifically, a mechanism $\mathcal{M}$ with privacy parameter ε satisfies DP if for any two *adjacent* inputs x and y, and $\varepsilon > 0$:

$$\frac{Pr[\mathcal{M}(x) = z]}{Pr[\mathcal{M}(y) = z]} \leq e^{\varepsilon} \tag{1}$$

In essence, the inequality in Equation 1 necessitates a certain level of *indistinguishability* between the outputs of two neighboring inputs. The notion of *adjacent* or *neighboring* is dataset-specific, but must be defined in order to satisfy the original notion of DP as defined in Equation 1. This level is governed by the ε parameter: a higher ε requires less indistinguishability, and vice versa.

The primary challenge with DP text rewriting comes with the design of the underlying *mechanism* that performs the rewriting [17, 20]. In light of the considerations required by DP, important design decisions include the definition of what *any two text inputs* means, often referred to as *adjacency*. In the literature, adjacency is often either defined on the token-/word-level or the sentence-/document-level. Both approaches have advantages and drawbacks: word-level approaches suffer from lack of contextualization [24], while offering the ability for tighter privacy guarantees. Mechanisms operating on entire documents can better preserve syntactic and semantic coherence, but document representations in large dimensions often necessitate high levels of noise to achieve the DP guarantee, due to their large *sensitivity* [19].

In this work, we focus on both approaches, namely ones that provide either word- or document-level DP guarantees. We are thereby able to evaluate the effectiveness of our proposed evaluation on both streams of DP text rewriting research. The exact mechanisms we utilize will be introduced in Section 4, but firstly, two important notions associated with DP text rewriting are introduced.

Local Differential Privacy. DP rewriting mechanisms often leverage the *local* notion of Differential Privacy. As opposed to *global* or *central* DP, Local DP (LDP) places the utilization of DP mechanisms at the user level [7]. In other words, the transformation (rewriting) of text data is performed by the user locally before releasing the output to some central aggregator. The definition of ε-LDP is as follows, for some mechanism π, *any* inputs x and x', and $\varepsilon > 0$,

$$\frac{Pr[\pi(x) = z]}{Pr[\pi(x') = z]} \leq e^{\varepsilon} \tag{2}$$

As one may observe, the important difference that comes with LDP is the requirement for *any two* inputs from the user to satisfy Equation 2. Although LDP allows for privacy protection to be ensured already at the user level, the major drawback comes with this strict adjacency requirement, thereby often leading to higher amounts of additive noise needed, including in the text rewriting scenario [19]. In this work, we seek to improve this trade-off, whereby our proposed method offers better utility than the raw rewritten outputs of DP text rewriting mechanisms.

Post-processing. An important property of Differential Privacy that is leveraged in this work is the notion of *post-processing*. In particular, mechanisms that satisfy DP (i.e., satisfy Equation 1) are robust against post-processing, defined as any arbitrary operation or computation performed on top of the output of a DP mechanism is safe. Specifically, given a DP mechanism $\mathcal{M}$, some deterministic or randomized function g, and input x, $g(\mathcal{M}(x))$ upholds the guarantee provided by $\mathcal{M}$ [8].

This important property states that the output of any function g is still resistant against adversaries, despite these adversaries possessing some auxiliary information. From an information theory perspective, this is due to the fact that regardless of what *auxiliary* knowledge may be possessed by an adversary, lacking knowledge of the private database (here, texts) makes it impossible to compute the function of the outputs of $\mathcal{M}$ [8]. As stated by Near and Abuah [29], DP mechanisms often leverage this useful DP property to improve the accuracy of DP outputs. As such, we aim to perform post-processing on DP rewritten texts with the goal of enhancing the utility and usability of these texts.

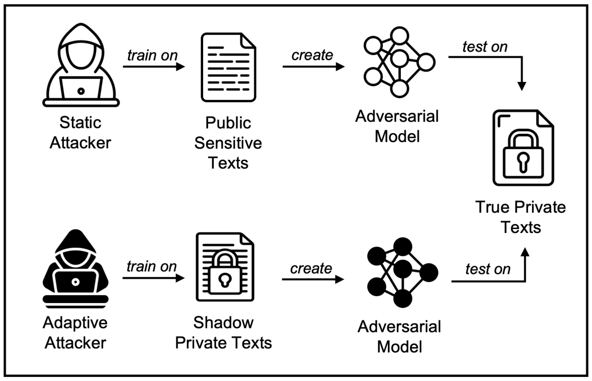

Figure 1: The *Static* and *Adaptive* Attackers. The adaptive attacker, with knowledge of the rewriting mechanism, generates "shadow" texts by rewriting publicly available texts from the same domain as the target "true" private texts.

2.2 Empirical Privacy Evaluation

The evaluation of DP text rewriting mechanisms offer takes the form of two-part experiments, resting the *utility* and *privacy* preservation of a proposed mechanism. In particular, a good rewriting mechanism not only protects the privacy of a text, but only rewrites the original text in a way that preserves its downstream utility, i.e., its ability to be useful for training models in some defined task [24].

With this, the question becomes how to evaluate the "privacy preservation" of a rewriting mechanism. In many recent works, such an evaluation takes the form of *empirical privacy* experiments, where the demonstration of a mechanism's privacy-preserving capabilities is performed empirically [17, 24, 34]. At a high level, this is typically done by showing that the output texts (i.e., post-rewriting) reduce the adversarial advantage of an attacker seeking to misuse the text for some nefarious purpose. Prominent examples, which we employ later in this work, include inferring the gender or authorship of the writer behind a given text. As such, a good DP rewriting mechanism should produce texts that reduce an attacker's ability to perform such inferences accurately.

To perform these empirical privacy experiments, two types of attackers have been modeled by the recent literature: the *static* and *adaptive* attackers [24]. Both types of attackers work towards the ultimate goal of accurately performing inferences (for a sensitive attribute such as gender) given some corpus of DP-rewritten texts:

- **Static attacker**: the static attacker has access to the DP rewritten texts, but has no knowledge of the mechanism used to perform the rewriting. The attacker does, however, possess knowledge of the *domain* of the original data, and furthermore, has access to a public dataset of (unrewritten) texts associated with the target sensitive attribute. Using this public data, the static attacker trains a model to predict the target attribute given an input text, and uses this model to infer the attribute of each text in the DP rewritten corpus.
- **Adaptive attacker**: the adaptive attacker possesses all the knowledge that the static attacker does. In addition, the adaptive attacker is stronger in the sense that the exact mechanism (including ε) is known. Using this knowledge, which includes the ability to run the mechanism, the adaptive attacker uses the public data to create a DP rewritten version of this data. The attacker then trains a model on the *rewritten* data to predict the target attribute, i.e., to predict the sensitive attribute of the original DP rewritten texts.

Both the static and adaptive attacker setups are illustrated in Figure 1. As shown unanimously by recent works [17, 24, 34], the adaptive attacker proves to be the stronger adversary. This can be explained by the fact that the distribution on which the adversarial model is trained more closely matches that of the target texts.

In this work, we use both of these attacker types to underline our empirical privacy evaluations. Specifically, we introduce a post-processing method which aims to decrease the adversarial advantage of both attackers, in the way that our post-processing method provides an extra layer of protection on top of DP rewritten texts. This method is introduced in Section 3.

2.3 Text2Text Generation

Recent advances in language models have demonstrated the impressive abilities of these models to generate highly coherent and plausible texts given an input prompt [42]. Many (large) language models also show very strong performance when *fine-tuned* for some specific downstream task, which necessitates only further training data upon which a base model can be improved.

A prominent paradigm of language model fine-tuning comes in the form of *Text2Text Generation* [23], which describes the case where there is a full-text input and full-text output. Common tasks employing such as setup include Machine Translation, Text Summarization, Text Simplification, and Paraphrasing, among others. Given *parallel* datasets, modern LMs can be fine-tuned efficiently, where the base model is then made an "expert" by being given domain- and/or task-specific training data.

In this work, we leverage the Text2Text fine-tuning setup to envision a post-processing step to "rewrite" texts "again", that is effectively to improve the empirical privacy gains of DP text rewriting by once again rewriting these texts. The definition and requirements of such a method are now introduced in the following.

3 A POST-PROCESSING METHOD FOR DP REWRITTEN TEXTS

In the following, we outline in detail our proposed post-processing method, which can be broken down into two tracks. We present these methods as a way to improve both the semantic similarity and privacy preservation offered by DP rewritten texts. The process flow of these two tracks is illustrated in Figure 2.

3.1 Preliminaries

We assume a user wishing to use some DP rewriting mechanism in the LDP setting. Concretely, a user will rewrite his or her textual dataset before releasing it to some data aggregator or central analyst. To do this, the user leverages a DP mechanism $\mathcal{M}$ with privacy budget ε, where this budget is chosen for the entire text dataset to be rewritten. Additionally, we assume that the user has access to large-scale public text corpora, as well as the ability and resources to fine-tune LLMs on such data in a Text2Text Generation setup.