EasyChair Preprint

№ 14540

Effectiveness of Privacy-Preserving Algorithms for Large Language Models: a Benchmark Analysis

Jinglin Sun, Basem Suleiman and Imdad Ullah

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

August 26, 2024

Effectiveness of Privacy-Preserving Algorithms for Large Language Models: A Benchmark Analysis

Jinglin Sun
School of Computer Science
University of Sydney
jsun2697@uni.sydney.edu.au

Basem Suleiman
School of Computer Science and Engineering
The University of New South Wales
b.suleiman@unsw.edu.au

Imdad Ullah
School of Computer Science
University of Sydney
imdad.ullah@sydney.edu.au

***Abstract*—Recently, several privacy-preserving algorithms for NLP have emerged. These algorithms can be suitable for LLMs as they can protect both training and query data. However, there is no benchmark exists to guide the evaluation of these algorithms when applied to LLMs. This paper presents a benchmark framework for evaluating the effectiveness of privacy-preserving algorithms applied to training and query data for fine-tuning LLMs under various scenarios. The proposed benchmark is designed to be transferable, enabling researchers to assess other privacy-preserving algorithms and LLMs. The benchmark focuses on assessing the privacy-preserving algorithms on training and query data when fine-tuning LLMs in various scenarios. We evaluated the SANTEXT+ algorithm on the open-source Llama2-7b LLM using a sensitive medical transcription dataset. Results demonstrate the algorithm's effectiveness while highlighting the importance of considering specific situations when determining algorithm parameters. This work aims to facilitate the development and evaluation of effective privacy-preserving algorithms for LLMs, contributing to the creation of trusted LLMs that mitigate concerns regarding the misuse of sensitive information.**

***Index Terms*—large language models, privacy-preserving algorithms, differential privacy, benchmarks**

I. INTRODUCTION

As large language models (LLMs) become increasingly integrated into various industries, concerns over privacy and data security have grown. The research community has thoroughly investigated the privacy concerns related to LLMs. Numerous studies have demonstrated that LLMs can memorize certain portions of their training data [1] [2] [3] [4]. Furthermore, multiple works have proposed and demonstrated various attacks designed to extract the training data from trained LLMs [5] [6]. The successful execution of these attacks has emphasised the urgent requirement to tackle the privacy issues linked to LLMs.

An increasing number of papers have proposed word-level privacy-preserving algorithms for NLP tasks [7] [8] [9] [10] [11] [12]. These algorithms are also well-suited for LLMs as they can protect both training and query data, thereby addressing the privacy leakage problem of LLMs. However, researchers have not yet applied these algorithms to LLMs or provided a systematic benchmark to guide the evaluation process of different privacy-preserving algorithms. Meisenbacher et al. [13] proposed a benchmark for evaluating word-level algorithms on LSTM models, but not on LLMs.To address this need, we propose a comprehensive benchmark for evaluating privacy-preserving algorithms for LLMs, focusing on those with robust protection mechanisms, such as differential privacy and its variants, which have been recognized for their effectiveness. We exclude techniques such as anonymization [14] or de-identification [15] because LLMs have demonstrated that they have powerful inferential capabilities that could make these methods insufficient for safeguarding private information.

During the benchmark procedure, after selecting the Large Language Model (LLM) and privacy-preserving methods, we implement a parameter tuning stage to identify the best-optimised settings. After applying the algorithms to the data, we evaluate the utility of the algorithms by measuring the performance of the LLM on downstream tasks. Moreover, in order to measure the level of privacy protection provided by the privacy-preserving algorithm, we employ the canary insertion attack, a widely acknowledged method for evaluating the exposure of sensitive information.

Our benchmark encompasses three distinct scenarios: (1) training with privacy-preserving algorithms, (2) testing with privacy-preserving algorithms, and (3) both training and testing with privacy-preserving algorithms. We evaluate the performance of one privacy-preserving algorithm, namely SANTEXT+ [7], which is a variant of differential privacy, on the widely-used open-source Llama2-7b model using one sensitive medical transcription dataset.

The main contributions of this paper are as follows:

1. Introducing a comprehensive benchmark for assessing the privacy-preserving algorithms when applied to training and test data for fine-tuning LLMs under various scenarios, including model selection, algorithm selection, parameter tuning and final evaluation.

2. Constructing a set of quantitative metrics to measure the level of privacy protection and the utility of different algorithms, providing a standardized approach for evaluation.

3. Fine-tuning the Llama2-7b on the medical transcription dataset, which we carefully pre-processed using our innovative data cleaning approach. We also customised prompts for this dataset to fine-tune and test the model.

4. Conducting a detailed evaluation using the proposed benchmark to evaluate the performance of SANTEXT+.

Section II reviews prior work, contextualizing and critiquing relevant studies. Section III details our methodology, including the proposed benchmark and its use. Section IV presents

the experimental setup and results. Section V will offer a discussion of our findings, and Section VI will conclude the article, summarizing our contributions and exploring potential future directions for our research.

II. Related Work

Large language models (LLMs), such as the Generative Pre-trained Transformers (GPTs) [16], are becoming increasingly prevalent, with current LLMs capable of producing text with little or no specific fine-tuning, known as 'few-shot' or 'zero-shot' performance [17]. This enables the fine-tuning of LLMs with small and less complex datasets for specific tasks. Llama2, developed by Touvron et al. [18] at Meta AI, is a collection of open-source pre-trained and fine-tuned LLMs. The pre-trained Llama2(70 B) model outperforms all other open-source models and is comparable to or better than the PaLM(540B) model [19] across all benchmarks, despite a significant performance gap between Llama2(70 B) and GPT-4 [20]. Llama2 is the best choice for this work, as it is the best open-source pre-trained LLM that challenges even some closed-source LLMs.

Differential privacy (DP) [21] is a rigorous mathematical definition of privacy that guarantees the behaviour of an algorithm hardly changes when a single individual is added to or removed from the dataset, providing protection against the disclosure of individual-level information [22]. Abadi et al. [23] proposed **DP-SGD**, integrating state-of-the-art machine learning methods with differential privacy to train neural networks within a modest privacy budget. Their work inspired researchers to enhance the algorithm and apply it to Natural Language Processing (NLP) and large language models (LLMs) [24] [25] [26] [27]. However, searching for parameters in DP learning is difficult due to the lengthy training period and high sensitivity to various parameters [25]. Consequently, our research focuses on implementing privacy-preserving algorithms directly on the text data prior to incorporating it into the fine-tuning and testing stages of LLMs.

Implementing privacy-preserving algorithms on unstructured textual data is a challenging task in data security. Yue et al. [7] introduced a novel local DP notion called UMLDP, considering both privacy and utility for text sanitization. The proposed token-wise sanitization methods with UMLDP, **SANTEXT+**, is constructed based on a variant of the exponential mechanism (EM), using "native" text tokens as both input and output spaces to avoid the "curse of dimensionality." However, the efficacy of SANTEXT+ on LLMs remains uncertain, which our research aims to explore.

Carlini et al. [28] introduce a quantitative metric called **"exposure"** to assess a model's tendency to reveal sensitive information from private training data. This metric can empirically evaluate the model's potential for unintentionally memorizing unique in the training data. The exposure formula is calculated as follows:

$$\begin{aligned}\text{Exposure} = \log_2(\text{Total number of guesses}) \\ - \log_2(\text{Rank of canary})\end{aligned} \tag{1}$$

The exposure of a canary is determined by its rank, which is based on the empirical model perplexity of all possible canary sequences. Exposure values range from 0 to $\log_2(\text{Total number of guesses})$, with the most likely canary achieving the maximum and the least likely canary receiving the minimum. In this article, we employ the exposure metric to evaluate the effectiveness of privacy-preserving algorithms in protecting sensitive information.

III. Methodology

A. Benchmark Process

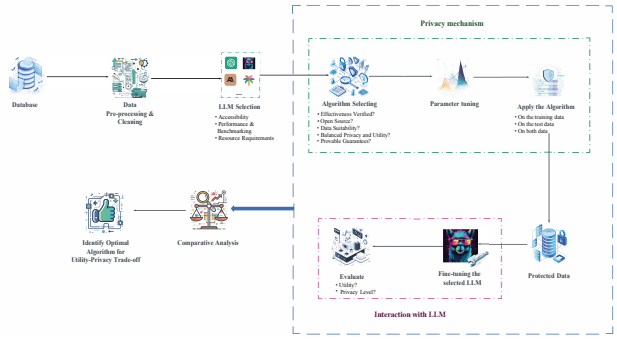


Fig. 1: Benchmark Process Overview

In this section, we present our benchmark process for evaluating privacy-preserving algorithms for Large Language Models (LLMs), as illustrated in Figure 1. The process commences with a dataset intended for fine-tuning and testing LLMs for a specific task, followed by data cleaning and pre-processing. Subsequently, an appropriate LLM is selected based on predefined criteria. This selection process considers three key factors:

- **Accessibility**: We assess whether the LLM is open-source, allowing for code review and modification. For proprietary models, we evaluate API access and support for deployment and fine-tuning.
- **Performance and Benchmarking**: We examine the LLM's performance on established benchmarks designed for LLM evaluation, its comparison with state-of-the-art models, and its efficacy in the relevant domain of the organization.
- **Resource Requirements**: We consider whether the existing infrastructure can support efficient fine-tuning and deployment of the LLM, and whether the computational costs fall within budget limitations.

LLMs that satisfy these criteria proceed to the subsequent steps in the benchmark process. The next stage involves selecting a suitable privacy-preserving algorithm for the specific task. Five key considerations guide this selection:

- Empirical validation of the algorithm's effectiveness for NLP tasks
- Public accessibility of the algorithm's source code
- Alignment of the algorithm with the specific characteristics and structure of the selected data

- Robustness of the algorithm in protecting sensitive information while maintaining acceptable utility for intended NLP tasks
- Provision of verifiable privacy assurances by the algorithm

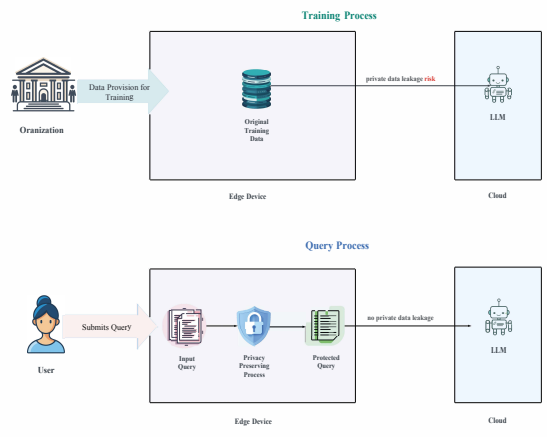
Algorithms meeting these criteria undergo parameter tuning (Section III-C) to identify optimal parameters for application to textual data. The selected algorithm, with its optimized parameters, is then applied to the prepared dataset under three scenarios: training data, test data, and both. The resulting protected data enters the LLM interaction phase, where the LLM is fine-tuned and tested according to these scenarios. The test stage comprises both utility and privacy evaluation, which will be discussed further in Section III-D. The completion of this stage marks the end of one iteration, evaluating a single algorithm with the selected LLM.

The privacy mechanism and LLM interaction stages can be iterated to evaluate different algorithms on the selected LLM. Upon testing all algorithms, a comparative analysis is conducted to identify the optimal algorithm for the chosen LLM. Due to resource and time constraints, our current study is limited to the use of a single LLM (Llama2-7b) for the experiments. However, in real-world scenarios, the selection of different LLMs could be incorporated into the iterative process to identify the optimal combination of LLM and privacy-preserving algorithm tailored to an organization's specific needs.

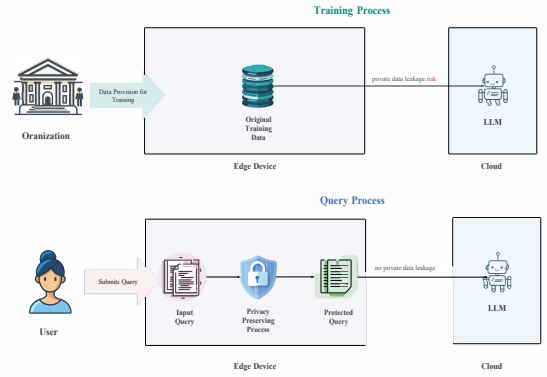
B. Real-World Scenario Simulation

As previously mentioned, three distinct scenarios require our consideration. Figure 2 illustrates this process. This figure describes the experimental setup in our work, but in real-world applications, the process of fine-tuning may not always take place on the cloud. If the organization has sufficient resources and the target LLM is open-source, fine-tuning can be conducted locally.

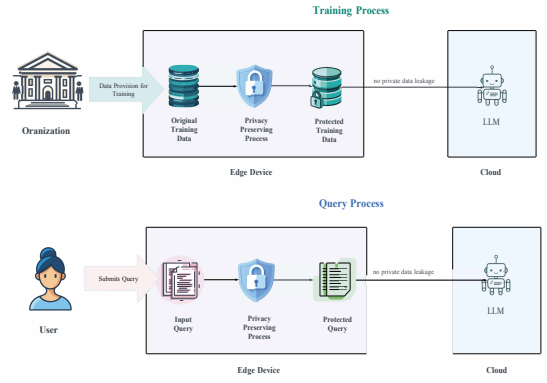
- **Scenario One: Privacy-Enhanced Training Data with Unprotected User Queries**
  In this scenario, organizations possess sensitive training data that needs protection during the fine-tuning of large language models. The data may encompass personal information or proprietary knowledge. To mitigate privacy risks, organizations employ privacy-preserving algorithms to the training data before model fine-tuning. However, user queries, which serve as test data, are considered non-sensitive and do not need additional privacy enhancements.
- **Scenario Two: Unprotected Training Data with Privacy-Enhanced User Queries**
  In this scenario, the training data utilized for fine-tuning large language models is non-sensitive, comprising public datasets or general knowledge that does not require privacy protection. Organizations can fine-tune the model without applying privacy-preserving techniques to the training data. However, user queries, which constitute the test data, are considered sensitive as they may contain personal information. To safeguard user privacy, these inputs undergo privacy protection algorithms before querying the fine-tuned model.
- **Scenario Three: Privacy-Protected Framework for Both Training Data and User Queries**
  In this scenario, both the training data and user queries are considered sensitive, necessitating a comprehensive



(a) Scenario One



(b) Scenario Two



(c) Scenario Three

Fig. 2: Three real-world Scenarios

approach to privacy protection. Organizations handling such sensitive data must ensure that privacy is maintained throughout the entire pipeline, from model fine-tuning to user interaction. To achieve this, privacy-preserving algorithms are applied to the training data and user queries before interacting with the large language model.

These scenarios demonstrate the diverse privacy requirements encountered in real-world applications of large language models. Organisations need to evaluate the sensitivity of training and query data in order to determine the necessary privacy protection.

C. Parameter Tuning for Privacy-preserving algorithms

Performing parameter tuning is essential to identify the optimized parameters for an individual algorithm when applied to Large Language Models (LLMs). This process is universal and can be extended to any privacy-preserving algorithm when employing our proposed benchmark. Researchers testing other types of algorithms should modify the details accordingly, such as changing the ϵ value to the parameter that controls the privacy level of their specific algorithms.

The process begins by selecting a privacy-preserving algorithm and exploring its parameters to identify parameter $\mathcal{P}$, which affects the utility and degree of privacy protection, except ϵ. Parameter $\mathcal{P}$ is modified within a reasonable range and the algorithm is applied with different values of $\mathcal{P}$ to the training and testing data while keeping ϵ fixed. The algorithm is subsequently trained and assessed using different p values to ascertain the utility under various $\mathcal{P}$ settings.

Next, canary data (a piece of privacy information) is inserted into the training data, with an insertion frequency of approximately 10% of the total training data. The LLM is trained using this modified training data with different values of parameter $\mathcal{P}$ while keeping ϵ fixed for the algorithm. The exposure calculation formula [28] is employed to assess whether the algorithm effectively protects privacy information by determining if the calculated exposure is sufficiently large to expose the canary data.

Finally, the values of parameter $\mathcal{P}$ that prevent the exposure of canary data when the algorithm is applied to the LLM are identified. Among these "safe" parameters that do not compromise privacy, the value of $\mathcal{P}$ that achieves the highest utility is selected. This approach ensures that the selected parameter $\mathcal{P}$ strikes an optimal balance between privacy protection and model performance.

D. Evaluation Process

The evaluation process is conducted following parameter tuning and is applicable to all variants of differential privacy algorithms. Researchers evaluating other types of algorithms may need to modify the process to better suit their specific algorithms by replacing the ϵ with their own privacy budget or privacy control values.

During the evaluation phase, the algorithm with the optimized parameter $\mathcal{P}$ and varying ϵ is applied to the training and testing datasets. The LLM is then trained and tested on different datasets according to the three scenarios, assessing the algorithm's utility.

To evaluate the algorithm's ability to protect privacy information, canary data is inserted into the training dataset, with an insertion frequency of approximately 10% of the total training data. The algorithm with a fixed parameter $\mathcal{P}$ but varying ϵ is applied to this modified training data, which is then used to train the LLM. The exposure of the canary data is calculated to determine whether the algorithm, with increasing ϵ, can still effectively protect privacy information. This analysis helps understand which algorithms can be applied under different ϵ budgets while maintaining privacy protection.

If the exposure exceeds the threshold at a certain ϵ, it indicates that the algorithm cannot protect privacy information at that ϵ level. In such cases, we will modify the optimized parameter $\mathcal{P}$ previously identified and try to find a parameter value $\mathcal{P}$ that prevents exposure. We will then re-calculate the algorithm's utility with the new parameter at that specific ϵ. This approach ensures that we can determine the correct utility of an algorithm without compromising privacy.

IV. Experiments and Results

A. Medical Transcription Dataset

For this study, we selected the Medical Transcription Dataset [29], which is scrapped from MTSamples, a repository of sample medical transcriptions across various specialties. This dataset was chosen to simulate real-world scenarios in fine-tuning large language models on sensitive data. The primary objective of our task is to infer medical specialties from given transcription keywords.

Initial examination revealed significant noise in the dataset. To address this and ensure data quality, we implemented the following pre-processing steps:

- **Step 1**: Medical specialties with fewer than 50 samples were removed.
- **Step 2**: Overlapped specialties were carefully identified and removed.
- **Step 3**: The 20 most common keywords within each specialty were identified. For samples containing more than 20 keywords, the most common keywords were retained, and other keywords were randomly added until the sample reached 20 keywords. Samples lacking any of the most common keywords and having more than 20 keywords were removed. This step effectively reduces noise.

Following the pre-processing stage, the medical transcription dataset was reduced to 1,532 rows with 9 medical specialties. Figure 3 displays the distribution of medical specialities in this improved dataset. To address the imbalanced data, we applied over-sampling and under-sampling techniques. However, resampling yielded even lower accuracy than the original dataset. Thus, we will use the unbalanced dataset for our experiments in this work.

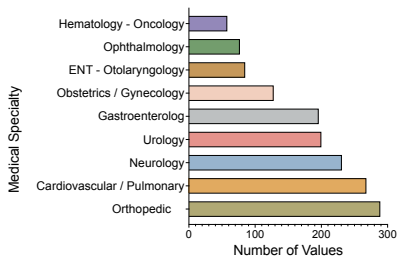


Fig. 3: Distribution of Different Medical Specialties

B. Experimental Setup

All experiments were conducted using Google Colab with an A100 GPU. We employed the Llama2-7b model from Meta AI, a 7-billion-parameter open-source large language model directly loadable from the Hugging Face. For the experiments, we set the batch size to 4, the learning rate to $4e-4$, the weight decay to 0.1, and the warm-up rate to 0.03 during fine-tuning. For the inference stage, we set Top_k to 10 and Max_new_token to 15. To efficiently fine-tune Llama2-7b, we used Low-Rank Adaptation (LoRA) [30], a lightweight training technique that reduces trainable parameters.

LoRA freezes pre-trained model weights and injects trainable rank decomposition matrices into each Transformer layer. This significantly decreases the number of trainable parameters for downstream tasks, enabling faster training, improved memory efficiency, and smaller model weights (typically a few hundred MBs), making it easier to store and share.

C. Parameter Tuning Practice

To select the optimized parameter for SANTEXT+, we conducted experiments using the Medical Transcription dataset, with ϵ fixed at 3 and GloVe-300d used for the SANTEXT+ algorithm.

Utility evaluation followed these steps:

1) The Medical Transcription dataset was divided into 80% for training and 20% for testing.
2) Privacy-preserving algorithms with different parameter values were applied to both training and testing data.
3) The Llama2-7b model was trained on the 80% training data and evaluated on the 20% testing data, following the three outlined situations.

To assess the algorithms' ability to protect sensitive information, we employed the canary insertion attack:

1) The sensitive information "patient name: elsa" was inserted as canary data into 10% of the Medical Transcription dataset records.
2) The Llama2-7b model was trained on the modified dataset using privacy-preserving algorithms with different parameter values.
3) Canary data exposure was calculated to determine the algorithm's effectiveness in preserving privacy:

$$\text{Exposure} = \log_2(26^4) - \log_2(\text{Rank of canary})$$

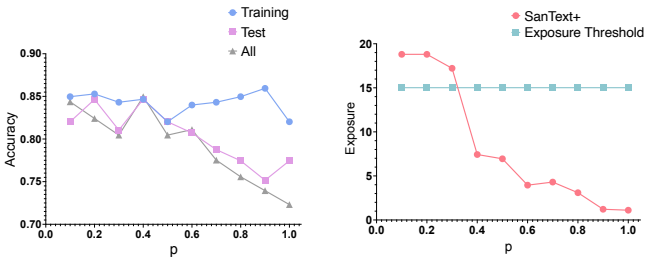
4) The exposure threshold was set as $\log_2(26^4) - \log_2(10)$, based on the topk value of 10 for Llama2 inferencing phase.

To begin, we aim to identify the parameter that influences both the utility and the degree of privacy protection of SANTEXT+. Yue et al. [7] discussed the impact of the parameter p, which modifies the probability of non-sensitive words being sanitized, on the utility of the SANTEXT+ algorithm. They observed that as p increases, the accuracy decreases.

Based on the assumption that the algorithm considers frequently appearing words as non-sensitive, we hypothesize that in contexts with a high frequency of canary insertions, the value of p will significantly impact the exposure of sensitive information. When p is small, fewer non-sensitive words will be sanitized, leading to greater exposure. Conversely, when p is large, more non-sensitive words will be sanitized, resulting in reduced exposure.

To test our hypothesis and validate the findings of Yue et al. [7], we conducted experiments following the previously discussed steps. We adjusted the p value from 0.1 to 1.0 under a fixed ϵ value of 3. Our experimental results, shown in Figure 4, indicate that the lines for "test" and "all" generally trend downward as p increases, despite some fluctuations. These fluctuations (e.g., at $p = 0.1$, the accuracy is not particularly high, but at $p = 0.2$, the accuracy increases significantly, and at $p = 0.3$, the accuracy drops again) might be attributed to the instability of the Llama2-7b model, which may not generate consistent answers when given the same input. Additionally, the "training" line only fluctuates without showing a clear decreasing trend. To summarize, our results corroborate the observations made by Yue et al. [7], demonstrating that as the p value increases, the overall utility decreases, with the exception of the "training", which remains unchanged.

Moreover, we have verified our hypothesis, showing that as the p value increases, the exposure decreases. Notably, we discovered that when p is lower than 0.4, the exposure exceeds the acceptable threshold, indicating a risk of sensitive information being exposed. Consequently, we selected 0.4 as the optimal parameter value for SANTEXT+.



(a) Accuracy with different p values (b) Exposure with different p values

Fig. 4: Optimized Parameter Selection for SANTEXT+

D. Evaluation Experiments

Having determined the optimized parameters for SANTEXT+, we evaluate its performance using its best parameter. Experiments are conducted on the Medical Transcription dataset, following the steps outlined in Section III-D. The training and testing situations are shown in Figure 5. We vary the privacy budget ϵ from 1 to 10 to observe how the algorithm behaves under different privacy constraints. This range of ϵ values provides insights into the algorithm's ability to maintain performance as privacy requirements become more stringent.

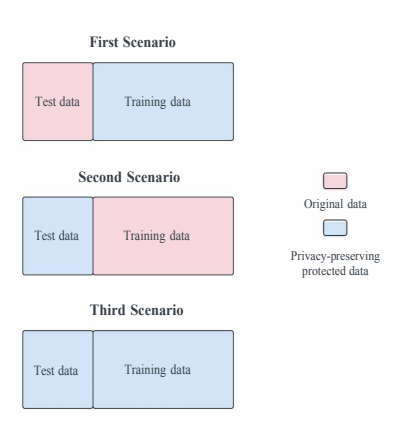
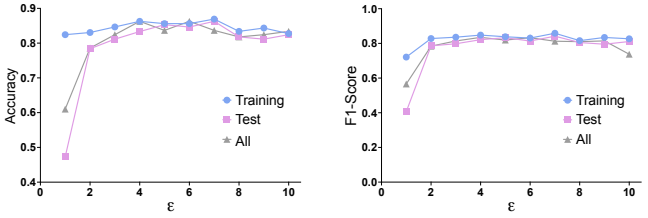


Fig. 5: Three Processes of Experiments

1) Utility Results: As shown in Figure 6 and Table I, the SANTEXT+ algorithm performs quite well when applied only to the training data. However, it exhibits poor utility when applied to both training and test data with ϵ equal to 1. The test data's utility is the worst, with an accuracy of only 0.4728 and an F1-score of 0.4065. These results suggest that SANTEXT+ may not be a suitable choice for protecting query data when strict privacy requirements, such as $\epsilon = 1$, are in place. Moreover, when epsilon $= 1$, the F1-score is significantly lower than the accuracy. This is because the precision and recall values for the Hematology-Oncology specialty are very low, even equal to 0 when applied to test data. As other medical specialties perform much better than this specialty, the accuracy is higher than the F1-score. However, when epsilon ≥ 2, the precision increases significantly while recall increases slightly. This increases the F1-score, making it comparable to the accuracy.

TABLE I: SANTEXT+ in Different Situations(Medical Transcription Dataset)

		Epsilon									
Data	**Metrics**	1	2	3	4	5	6	7	8	9	10
Training	**Accuracy**	0.82	0.83	0.85	0.86	0.86	0.86	0.87	0.83	0.84	0.83
	F1-score	0.72	0.83	0.84	0.85	0.84	0.83	0.86	0.82	0.83	0.83
Testing	**Accuracy**	0.47	0.78	0.81	0.83	0.85	0.85	0.86	0.82	0.81	0.82
	F1-score	0.41	0.79	0.80	0.82	0.83	0.81	0.84	0.80	0.80	0.81
All	**Accuracy**	0.61	0.79	0.82	0.86	0.84	0.86	0.84	0.82	0.82	0.83
	F1-score	0.57	0.78	0.81	0.83	0.82	0.83	0.81	0.81	0.81	0.74
Original	**Accuracy**					0.88					
	F1-score					0.86					



(a) Accuracy with different ϵ (b) F1-Score with different ϵ

Fig. 6: Utility for SANTEXT+

2) Exposure Results: The exposure of SANTEXT+ with different ϵ values is shown in Figure 7. It could be seen that the SANTEXT+ becomes vulnerable to exposure when the ϵ value equals or exceeds 9. Consequently, SANTEXT+. is not a recommended choice when the privacy budget is substantial.

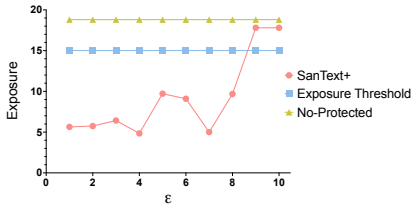


Fig. 7: Exposure with different ϵ

Furthermore, our analysis reveals that SANTEXT+ exhibits significant vulnerability, demonstrating a propensity to expose sensitive information when the p value is low or when ϵ is high. To further explore its vulnerability, we investigate the potential impact of canary data insertion frequency on the exposure risk when p is low ($p = 0.1$ and $\epsilon = 3$). Figure 8 indicates that exposure increases with the increase in insertion frequency. Although SANTEXT+ reaches the highest level of exposure($\log_2(26^4)$) later than the unprotected baseline and effectively protects sensitive information when the insertion frequency is low, its protective capabilities decrease when faced with high insertion frequencies. This observation highlights that careful consideration is important when employing the SANTEXT+. Firstly, conducting a comprehensive analysis of the target dataset is crucial to estimating the expected frequency of occurrence of sensitive information. Armed with the knowledge of the anticipated repetition of sensitive data, practitioners and researchers must carefully consider the choice of the p value and the ϵ for the SANTEXT+ algorithm.

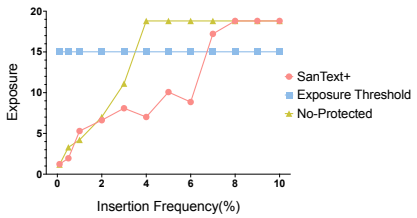


Fig. 8: Exposure with Different Insertion Frequency: SANTEXT+ is set with $p = 0.1$, $\epsilon = 3$.

To prevent privacy leakage at ϵ values of 9 and 10, we adjusted the parameter p from 0.4 to 0.5 of the SANTEXT+. As shown in Figure 9, this modification ensures that the exposure of the "modified SANTEXT+ " remains below the established threshold at these ϵ levels.

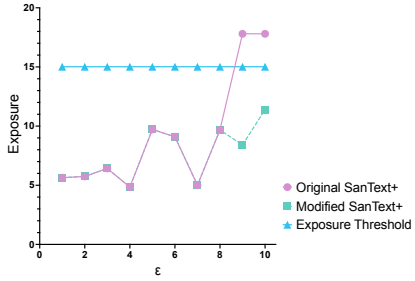
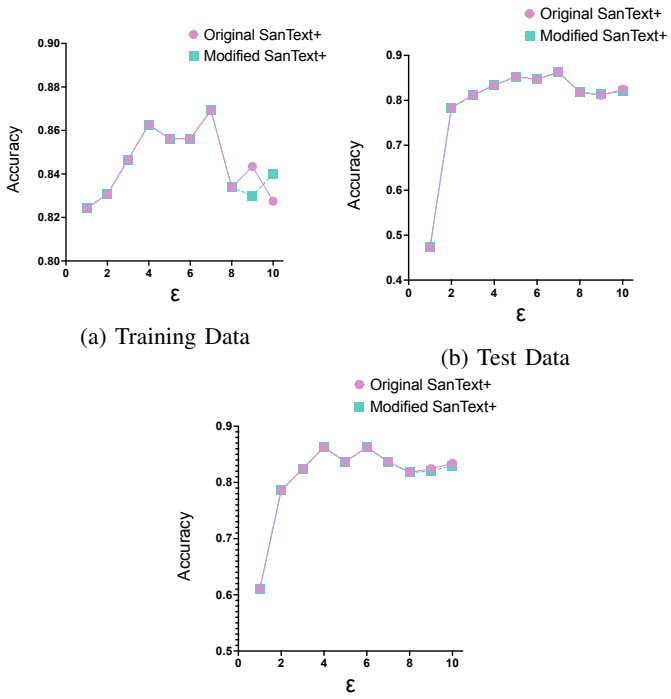


Fig. 9: Exposure of Modified SANTEXT+

We further assessed the utility of the "modified SANTEXT+" on the Medical Transcription Dataset. The findings shown in Figure 10 suggest that the accuracy remains relatively unchanged for the medical transcription dataset. Consequently, adjusting the parameter p to 0.5 is appropriate, as it ensures that data privacy is maintained without compromising utility.



(a) Training Data

(b) Test Data

(c) All Data

Fig. 10: Utility of Modified SANTEXT+ on Medical Transcription Dataset

V. DISCUSSION

The results demonstrate that SANTEXT+ can be an effective algorithm when applied to LLMs. However, researchers must carefully calibrate parameters to achieve an optimal balance between utility and privacy for various real-world scenarios. In comparison to similar work by Meisenbacher et al. [13], our benchmark incorporates parameter tuning, addressing a limitation in their approach that potentially led to unfair comparisons of algorithms without optimized settings. Furthermore, while their study relied solely on data-level privacy metrics without model linkage, we employ the exposure test to assess whether models retain the ability to memorize protected fine-tuning data.

Our findings indicate that the performance of the evaluated algorithms tends to decline when applied to test data (query data) or both test and training data. This observation may be attributed to the fact that existing algorithms are predominantly designed for training datasets. Consequently, the development of privacy-preserving algorithms adapted to query data is crucial for achieving a more favourable utility-privacy trade-off in real-world applications.

It is important to note that SANTEXT+ was originally designed for BERT (Bidirectional Encoder Representations from Transformers). BERT [31] considers both preceding and subsequent context for each word, utilizing a "masked language model" (MLM) pre-training objective, where some tokens are randomly masked, and the model predicts the original vocabulary based on the context. In contrast, LLMs typically employ Transformer-based decoder architectures with unidirectional architectures and are primarily used for text generation tasks. Given these architectural differences, existing privacy-preserving algorithms may require customization to optimize their performance with LLMs. While our experiments focused on classification tasks, users generally expect LLMs to comprehend natural language queries and instructions, enabling diverse functionalities. Therefore, developing or enhancing algorithms to enable LLMs to better "understand" protected data is essential to meet user expectations.

However, our benchmark has several limitations. Although we test for the exposure of canary data, low exposure does not guarantee protection against all potential attacks. Developing a broader range of attack scenarios would enhance the credibility of the privacy guarantee evaluation. Additionally, a quantitative metric for assessing the protection of query data by privacy-preserving algorithms is needed.

Furthermore, our benchmark currently encompasses only one dataset, limiting its ability to evaluate algorithms across diverse tasks and data structures comprehensively. Expanding the benchmark to include a wider array of datasets, tasks, and privacy-preserving algorithms would enhance its comprehensiveness and persuasiveness. Finally, evaluating the benchmark's effectiveness across multiple LLMs would further demonstrate its applicability and robustness.

VI. CONCLUSION

This paper presents a comprehensive benchmark for evaluating privacy-preserving algorithms for LLMs in various scenarios, considering three distinct privacy leakage scenarios

that organizations or individuals may encounter during fine-tuning and querying. We evaluate the SANTEXT+ privacy-preserving algorithm on the Llama2-7b model using a sensitive medical transcription dataset, presenting the utility and privacy levels achieved.

In our further work, we will extend our benchmark by evaluating a broader range of privacy-preserving algorithms and diverse datasets with varying sizes, data structures, and characteristics. This will provide a more comprehensive assessment of the algorithms' performance and applicability across different scenarios.

For future research in this field, we suggest the following areas of focus: incorporating additional LLMs into the benchmark to evaluate the algorithms' effectiveness when applied to different models; developing privacy-preserving algorithms specifically adapted to query data; designing more sophisticated attack scenarios to enhance the credibility of privacy guarantee evaluations; and establishing quantitative metrics for assessing the protection of query data by privacy-preserving.

REFERENCES

[1] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, "Quantifying memorization across neural language models," in *The Eleventh International Conference on Learning Representations*, 2023.

[2] S. Biderman, U. S. Prashanth, L. Sutawika, H. Schoelkopf, Q. Anthony, S. Purohit, and E. Raff, "Emergent and predictable memorization in large language models," 2023.

[3] K. Tirumala, A. H. Markosyan, L. Zettlemoyer, and A. Aghajanyan, "Memorization without overfitting: Analyzing the training dynamics of large language models," 2022.

[4] 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," 2024.

[5] 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.

[6] M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace, F. Tramèr, and K. Lee, "Scalable extraction of training data from (production) language models," 2023.

[7] X. Yue, M. Du, T. Wang, Y. Li, H. Sun, and S. S. M. Chow, "Differential privacy for text analytics via natural text sanitization," 2021.

[8] S. Chen, F. Mo, Y. Wang, C. Chen, J.-Y. Nie, C. Wang, and J. Cui, "A customized text sanitization mechanism with differential privacy," pp. 5747–5758, Jul. 2023. [Online]. Available: https://aclanthology.org/2023.findings-acl.355

[9] X. Zekun, A. Abhinav, F. Oluwaseyi, and T. Nathanael, "A differentially private text perturbation method using a regularized mahalanobis metric," 2020.

[10] R. S. Carvalho, T. Vasiloudis, and O. Feyisetan, "Tem: High utility metric differential privacy on text," 2021.

[11] M. Du, X. Yue, S. S. Chow, and H. Sun, "Sanitizing sentence embeddings (and labels) for local differential privacy," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2349–2359.

[12] O. Feyisetan, B. Balle, T. Drake, and T. Diethe, "Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations," in *Proceedings of the 13th international conference on web search and data mining*, 2020, pp. 178–186.

[13] S. Meisenbacher, N. Nandakumar, A. Klymenko, and F. Matthes, "A comparative analysis of word-level metric differential privacy: Benchmarking the privacy-utility trade-off," 2024.

[14] C. Yu, L. Tingxin, L. Huiming, and Y. Yang, "Hide and seek (has): A lightweight framework for prompt privacy protection," 2023.

[15] D. Ajinkya, B. Saumya, and S. Anantha, "Life of pii – a pii obfuscation transformer," 2023.

[16] G. Yenduri and et al., "Gpt (generative pre-trained transformer)— a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions," *IEEE Access*, vol. 12, pp. 54 608–54 649, 2024.

[17] T. B. Brown and et al., "Language models are few-shot learners," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020, pp. 159:1–159:25.

[18] H. Touvron, L. Martin, and et al., "Llama 2: Open foundation and fine-tuned chat models," 2023.

[19] R. Anil, A. M. Dai, and et al., "Palm 2 technical report," 2023.

[20] OpenAI, J. Achiam, and et al., "Gpt-4 technical report," 2024.

[21] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2006, pp. 1–12.

[22] Harvard Privacy Tools Project, "Differential privacy," 2024, [Online; accessed 22-April-2024]. [Online]. Available: https://privacytools.seas.harvard.edu/differential-privacy

[23] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 308–318. [Online]. Available: https://doi.org/10.1145/2976749.2978318

[24] D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz, S. Yekhanin, and H. Zhang, "Differentially private fine-tuning of language models," 2022.

[25] X. Li, F. Tramèr, P. Liang, and T. Hashimoto, "Large language models can be strong differentially private learners," 2022.

[26] R. Behnia, M. R. Ebrahimi, J. Pacheco, and B. Padmanabhan, "Ew-tune: A framework for privately fine-tuning large language models with differential privacy," in *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, Nov. 2022.

[27] W. Shi, R. Shea, S. Chen, C. Zhang, R. Jia, and Z. Yu, "Just fine-tune twice: Selective differential privacy for large language models," 2022.

[28] N. Carlini, C. Liu, U. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *Proceedings of the 28th USENIX Conference on Security Symposium*, ser. SEC'19. USA: USENIX Association, 2019, pp. 267–284.

[29] T. Boyle, "Medical transcriptions," 2018, kaggle, https://www.kaggle.com/datasets/tboyle10/medicaltranscriptions [Accessed: 2024-07-14].

[30] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.