A General Pseudonymization Framework for Cloud-Based LLMs: Replacing Privacy Information in Controlled Text Generation

Anonymous ACL submission

Abstract

An increasing number of companies have begun providing services that leverage cloud-based large language models (LLMs), such as ChatGPT. However, this development raises substantial privacy concerns, as users' prompts are transmitted to and processed by the model providers. Among the various privacy protection methods for LLMs, those implemented during the pre-training and fine-tuning phrases fail to mitigate the privacy risks associated with the remote use of cloud-based LLMs by users. On the other hand, methods applied during the inference phrase are primarily effective in scenarios where the LLM's inference does not rely on privacy-sensitive information. In this paper, we outline the process of remote user interaction with LLMs and, for the first time, propose a detailed definition of a general pseudonymization framework applicable to cloud-based LLMs. Building upon the framework, we have designed various pseudonymization methods and further propose a method that achieves pseudonymization through a controllable text generation process. The experimental results demonstrate that the proposed framework strikes an optimal balance between privacy protection and utility. The code for our method is available to the public at `https://github.com/Mebymeby/Pseudonymization-Framework`.

1 Introduction

Large Language Models (LLMs) have demonstrated considerable promise in advancing the field of artificial intelligence, showcasing remarkable capabilities in instruction following and excelling across a wide range of tasks, including writing, coding, and other text-based activities (Bubeck et al., 2023; Touvron et al., 2023; OpenAI et al., 2024). Consequently, an increasing number of companies have begun providing cloud-based LLM services, such as ChatGPT[1]. However, the widespread use of cloud-based LLM services has raised substantial privacy concerns: the transmission and storage of user data on cloud infrastructures pose significant risks of data breaches and unauthorized access to private information, as illustrated in Figure 1.

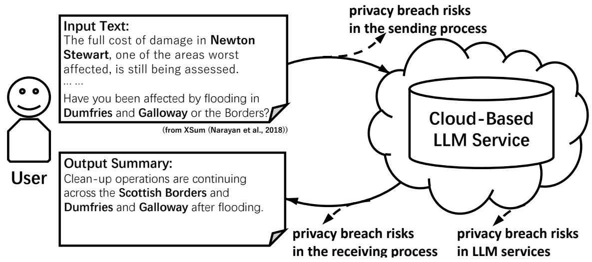

Figure 1: Potential privacy breach risks in using cloud-based LLM services

Current privacy-preserving techniques for cloud-deployed LLMs either prevent untrustworthy customers from accessing privacy-sensitive information in pre-trained datasets (Carlini et al., 2019; Pan et al., 2020; Brown et al., 2022), or safeguard users' pre-training and fine-tuning datasets from untrustworthy cloud service providers (Chi et al., 2018; Jegorova et al., 2022). However, these methods face significant challenges in addressing the unique issues arising from remote access to cloud-based LLMs. On the other hand, researchers have developed various strategies to ensure privacy security during the inference phrase, including Multi-Party Computation (Goldreich, 1998), homomorphic encryption (Acar et al., 2018), differential privacy in inference (Majmudar et al., 2022). However, these methods are not suitable for scenarios in which the cloud-based LLM's inference relies on privacy-sensitive information.

The data pseudonymization technique, which ensures privacy protection by appropriately replacing privacy-sensitive information, has since attracted the attention of researchers. (Kan et al., 2023; Chen et al., 2023; Lin et al., 2024) However, research on applying pseudonymization techniques

[1] `https://chatgpt.com/`

during the inference phase for privacy protection remains limited. Currently, a detailed definition of a pseudonymization framework for the inference phase of cloud-based LLMs is lacking. For example, Yermilov et al. (2023) divides pseudonymization into two parts: recognizing and replacing privacy entities. However, Chen et al. (2023) argues that pseudonymization should consist of two stages: concealing privacy entities for anonymization and restoring them for de-anonymization. We argue that these methods integrate certain steps of the pseudonymization process and, therefore, cannot be regarded as a general pseudonymization framework.

In this paper, we outline the process of remote user interaction with LLMs and, for the first time, propose a detailed definition of a general pseudonymization framework applicable to cloud-based LLMs. We define the pseudonymization framework as comprising three components: the detection of privacy-sensitive information, the generation of replacement terms, and the replacement of privacy information to achieve pseudonymization. We further propose a pseudonymization method based on a controllable text generation process, ensuring that the replaced text preserves maximal semantic correctness after replacement. Furthermore, to evaluate the practical effectiveness of the proposed framework in real-world LLM services, we specifically assessed its performance in text generation tasks, including summarization, question answering, text generation, and machine translation, in addition to classification tasks. The experimental results indicate that the proposed framework achieves an optimal balance between privacy protection and utility.

To summarize, our contributions are as follows:

(1) We propose a general pseudonymization framework applicable to cloud-based LLMs.

(2) We propose a pseudonymization method leveraging a controllable text generation process to preserve the semantic integrity of the replaced text.

(3) We evaluate the proposed framework across various text generation tasks and demonstrate that it achieves the optimal balance between privacy and performance.

2 Related Works

Privacy protection for large language models (LLMs) can be categorized according to the phase in which it is implemented: during the pre-training and fine-tuning phases, and during the inference phase (Yan et al., 2024). Privacy protection during the pre-training and fine-tuning phases of LLMs is essential for safeguarding sensitive data while preserving model effectiveness. Techniques such as differential privacy (Li et al., 2021; Wu et al., 2022; Xu et al., 2024), data cleaning (Bai et al., 2022; Kandpal et al., 2022), and federated learning (Yu et al., 2023; Xu et al., 2024; Zhang et al., 2024a) can be utilized to mitigate privacy risks during these phases. As previously discussed, these methods primarily aim to protect the privacy of information within LLMs. However, they do not fully address the privacy concerns associated with remote access to LLM services. Additionally, privacy protection measures implemented by model providers may not completely alleviate users' concerns regarding the potential misuse of their private data by these providers.

On the other hand, the issue of privacy leakage during the inference phase of LLMs has garnered significant attention. To address this issue, researchers have developed numerous strategies to ensure privacy security during the inference phase. These include encryption-based privacy protection approaches such as Multi-Party Computation (Goldreich, 1998; Dong et al., 2022), homomorphic encryption (Acar et al., 2018; Hao et al., 2022; Lu et al., 2023), and differential privacy in inference (Dwork, 2006, 2008; Majmudar et al., 2022). For example, Huang et al. (2022) proposed a specialized encoding method, Cheetah, which encodes vectors and matrices into homomorphic encryption polynomials. However, these homomorphic encryption methods are challenging to apply to cloud-based black-box LLMs, as they require access to the model's internal structures. Additionally, Du et al. (2023) introduced DP-Forward, which applies differential privacy during inference by perturbing embedding matrices in the forward pass of language models. However, these differential privacy approaches are mainly effective when the LLM's decision-making does not rely on sensitive information, which differs from the focus of our research.

In addition to the aforementioned methods, pseudonymization techniques focus on safeguarding the privacy of the prompt by identifying and removing privacy-sensitive information. For example, Kan et al. (2023) and Chen et al. (2023) proposed anonymizing sensitive terms before inputting

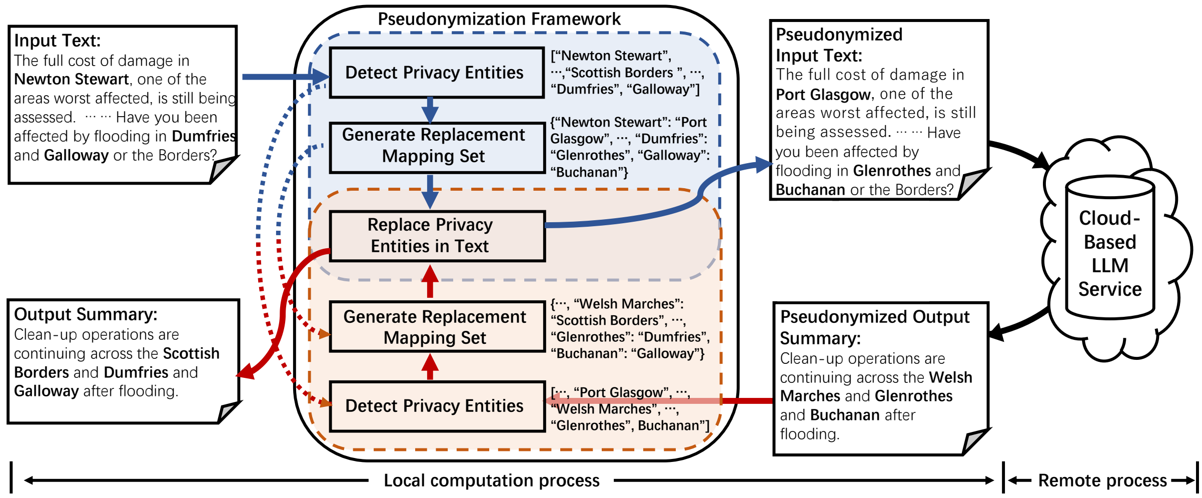

Figure 2: Overview of pseudonymization framework for cloud-based LLMs

them into the LLM and restoring them after the output. Lin et al. (2024) proposed a pseudonymization method to safeguard user privacy by converting user input from natural language into a sequence of emojis. Zhang et al. (2024b) introduced a mixed-scale model collaboration approach that combines the strengths of a large cloud-based model with a smaller, locally deployed model. However, there is currently no general definition of a pseudonymization framework for the inference phase of cloud-based LLMs. Additionally, these methods have primarily been tested on classification tasks, which differ from the core task of text generation in LLMs. Therefore, their results may not fully capture their effectiveness in text generation.

3 Pesudonymization Framework

As shown in Figure 2, a privacy-preserving cloud-based LLM access process consists of two steps: pseudonymizing the privacy information in the input text, as indicated by the blue arrow, and restoring the privacy information in the output results, as indicated by the red arrow. It is clear that the pseudonymization and restoration processes are logically identical, involving the detection of information to be replaced (e.g., privacy entities or entities to be restored), the generation of replacement candidates for detected entities, and the execution of the replacement process. Furthermore, the detection and candidate generation in the restoration process can refer to the results of the pseudonymization process, while the replacement operation itself is identical to that in the pseudonymization process. Therefore, we propose that a general pseudonymization framework should include only the three components of detection, generation and replacement. In the following sections, we will provide a detailed definition of the tasks for each component and discuss several viable approaches for each stage.

3.1 Detecting Privacy Information

Given a user's input X, which may contain multiple pieces of private information, we denote these pieces as $P = \{p^j_{A_i} | p^j_{A_i} \in X, 1 \leq i \leq n, 1 \leq j \leq N_i\}$. Here, A_i represents the i-th privacy attribute (e.g., name, location), and each $p^j_{A_i}$ represents the j-th instance of private information related to the attribute A_i. The total number of private information entries related to A_i is denoted as N_i. The goal of the privacy information detection method is to collect $P' = \{p'^j_{A_i} | p'^j_{A_i} \in X, 1 \leq i \leq n, 1 \leq j \leq N_i\}$, where P' represents the collection of detected private information. To maximize security, P' should closely approximate P, ensuring that all relevant private information is correctly identified while minimizing the risk of missing any sensitive data. The three detection methods employed in our experiments are described as follows.

NER-based Detection uses an off-the-shelf NER system to identify spans of named entities that correspond to privacy information categories. In this work, we utilize the publicly available BERT model, bert-large-cased-finetuned-conll03-english [2]. We refer to this method as $\mathrm{DET_{NER}}$.

Prompt-based Detection employs a locally deployed, small-scale instruction-tuned LLM to identify named entities. We denote this method as $\mathrm{DET_{prompt}}$.

Seq2Seq Detection is developed by fine-tuning

[2] https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-english

Input	John Edward Bates, formerly of Spalding, is now living in London.
Output (mark)	<ENT>John Edward Bates</ENT>, formerly of <ENT>Spalding</ENT>, is now living in <ENT>London</ENT>.
Output (replace)	<ENT>, formerly of <ENT>, is now living in <ENT>.

Table 1: Example output of Seq2Seq detection with entity marking and replacement

a small-scale base LLM on a parallel corpus of pseudonymized texts generated using the NER-based detection method. This method generates sentences that maintain consistency with the input text while marking or replacing privacy entities with designated tags, as illustrated in Table 1. We denote the two Seq2Seq detection variants as $\mathrm{DET}_{\mathrm{tag\_mark}}$ and $\mathrm{DET}_{\mathrm{tag\_rep}}$.

3.2 Generating Replacement Candidates

Based on the detected privacy entities P', the next step is to generate candidate entities Q that do not contain any privacy information to replace P'. Specifically, the goal of generation is to obtain a replacement mapping set $\mathcal{P} = \{(p'^j_{A_i}, q^j_{A_i}) | p'^j_{A_i} \in X, 1 \leq i \leq n, 1 \leq j \leq N_i\}$, where $q^j_{A_i}$ represents the generated candidate for $p'^j_{A_i}$. To ensure that the meaning of the original sentence remains intact after replacement, the replaced entities should generally share certain common characteristics (e.g., gender and language for names) with the original entities. Building on the aforementioned requirement, the semantics of $p'^j_{A_i}$ and $q^j_{A_i}$ should be as distinct as possible, ensuring that privacy information cannot be easily inferred from $q^j_{A_i}$. The two candidate generation methods employed in our experiments are described as follows.

Random Sampling utilizes the entities identified in Section 3.1 as a candidate set. From this set, an entity belonging to the same category as the privacy entities to be replaced is randomly selected as the replacement candidate. We denote this method as $\mathrm{GEN}_{\mathrm{rand}}$.

Prompt-based Generation employs a locally deployed, small-scale instruction-tuned LLM to generate replacement candidates for the privacy entities. We denote this method as $\mathrm{GEN}_{\mathrm{Prompt}}$.

3.3 Replace Privacy Entities

Given the input text X and the replacement mapping set $\mathcal{P}$ obtained from the previous sections, the next step is to replace the entity $p'^j_{A_i}$ in X with the corresponding replacement entity $q^j_{A_i}$. The resulting text after replacement is denoted as X'. To ensure that the meaning of the original text is preserved after the replacement, the remaining content in the text, aside from the replaced entities, should be appropriately adjusted. In other words, the goal of privacy entity replacement is to ensure that X' retains as much semantic correctness as possible. X' is then processed through a prompt template function and input into cloud-based LLMs, generating the output Y'. As mentioned earlier, for Y', there is no need to perform privacy entity detection and replacement candidate generation. Instead, the restoration process of Y' involves directly replacing $q^j_{A_i}$ in Y' with $p'^j_{A_i}$, similar to the replacement process in X, resulting in the final output Y. The three entity replacement methods employed in our experiments are described as follows.

Direct Replacement refers to the process of directly replacing $p'^j_{A_i}$ with $q^j_{A_i}$ without modifying other parts of the text X. This method is denoted as $\mathrm{REP}_{\mathrm{direct}}$. As previously mentioned, this approach may introduce semantic errors.

Prompt-based Replacement employs a locally deployed, small-scale instruction-tuned LLM to perform the replacement of entity names. We denote this method as $\mathrm{REP}_{\mathrm{prompt}}$.

Replacement through Text Generation executes replacement during a controllable text generation process to ensure the semantic correctness of the text after replacement. When the detected privacy entity term $p'^j_{A_i}$ is encountered during the text generation process, it is replaced by the corresponding entity $q^j_{A_i}$, and the generation of the subsequent token proceeds accordingly. The specific technical details of this method will be discussed in Section 4. We denote this method as $\mathrm{REP}_{\mathrm{gen}}$.

4 Pesudonymization Through Controllable Text Generation

We propose a pseudonymization replacement method based on a controllable text generation process, ensuring that the replaced text preserves maximum semantic correctness. In this section, we provide a detailed explanation of the method's process.

Given $X = (x_1, x_2, \ldots, x_L)$, the generation process of the LLM can be formulated as a sequential prediction of the next token, expressed as

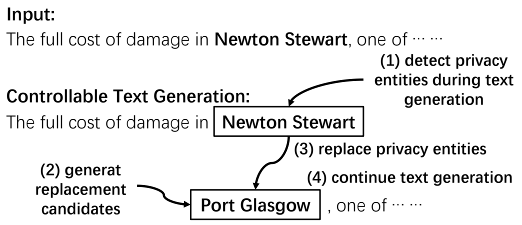

Figure 3: Workflow of pesudonymization through controllable text generation

follows:

$$\hat{y}_i = \operatorname{argmax} P(y_i \mid g(X), \hat{y}_1, \ldots, \hat{y}_{i-1})$$

Here, $g(X)$ represents the prompt text generated from X using a predefined prompt template, and $\hat{y}_i$ (where $1 \leq i \leq N$) denotes the predicted token at the i-th time step. As illustrated in Figure 3, during the pseudonymization process, the majority of the output text $\hat{Y} = (\hat{y}_1, \ldots, \hat{y}_N)$ remains identical to the input text X, except for a small portion where privacy entities is replaced.

Note that when using NER-based or prompt-based detection methods to identify privacy entities, we first employ a model capable of generating text identical to the input. During the text generation process, we compare each generated token $\hat{y}_i$ with elements in P' to determine whether $\hat{y}_i$ corresponds to a privacy entity. Therefore, depending on the privacy entity detection method used, $\hat{y}_i$ can take the following forms:

(1) $\hat{y}_i = x_i$, where $x_i \notin P'$. Here, P' represents a set of identified privacy entities collected using NER-based or prompt-based detection methods, as described in Section 3.1.

(2) $\hat{y}_i = x_i$, where $x_i \in P'$. In this case, x_i is recognized as a privacy entity by the NER-based or prompt-based detection methods.

(3) $\hat{y}_i = x_i$ when utilizing the Seq2Seq detection method described in Section 3.1.

(4) $\hat{y}_i =$ <ENT>x_i</ENT> or $\hat{y}_i =$ <ENT>. In this case, x_i is recognized as a privacy entity by the Seq2Seq detection method.

Next, for privacy entity x_i in cases (2) or (4), we generate the replacement candidate x'_i corresponding to x_i, based on the method described in Section 3.2. Then, we set $\hat{y}'_i = x'_i$. As shown in Figure 3, $\hat{y}'_i$ will be incorporated into the above formula, and the prediction for the output at the $(i+1)$-th time step will proceed as follows:

$$\hat{y}_{i+1} = \operatorname{argmax} P(y_{i+1} | \, g(X), \hat{y}_1, \ldots, \hat{y}_{i-1}, \hat{y}'_i)$$

This process continues until the entire sequence has been generated.

The main contribution of this method lies in its ability to decouple the end-to-end pseudonymization text generation process[3] into the three distinct stages described in Section 3. Additionally, it achieves better pseudonymization results by integrating different methods. By performing pseudonymization through the controllable text generation process, this approach ensures comprehensive coverage of privacy information detection and the correctness of replacement candidate generation by integrating various detection and generation methods. Furthermore, this approach leverages the strengths of LLMs and Seq2Seq generation processes, maximizing the semantic correctness of the text after replacement.

5 Experiment

5.1 Experiment Settings

Datasets We conduct experiments on several publicly available real-world datasets across various NLP tasks, including SQuAD 2.0 (Rajpurkar et al., 2016) for question answering, XSum (Narayan et al., 2018), CNN/Dailymail (See et al., 2017), and SAMSum (Gliwa et al., 2019) for summarization, GLUE (MNLI) (Williams et al., 2017; Wang et al., 2019) for natural language inference, and WMT14 (de-en) (Bojar et al., 2014) for machine translation. For experimental efficiency, we randomly sampled 1,000 samples from the test sets of each dataset to serve as the test set. In this study, we focus our analysis on three primary categories of named entities: person, location, and organization.

Evaluation Metrics For different datasets, we will use distinct performance evaluation metrics. For SQuAD 2.0, we use the F1 score and Exact Match (EM) (Rajpurkar et al., 2018) as the evaluation metrics. For XSum, CNN/Dailymail, and SAMSum, we use ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004) as the evaluation metrics. For GLUE (MNLI), we use the accuracy score as the evaluation metric. For WMT14 (de-en), we use

[3] In our preliminary experimental results, methods for pseudonymization through end-to-end text generation, such as those proposed by Yermilov et al. (2023) and Chen et al. (2023), yielded catastrophic results when trained with a limited amount of training data.

methods	SQuAD 2.0	XSum	CNN/ Dailymail	SAMSum	GLUE (MNLI)	WMT14 (de-en)
Qwen2.5-14B -Instruct	F1 = 79.1 EM = 75.5	ROUGE-1/2/L = 25.4/7.0/17.8	ROUGE-1/2/L = 30.8/10.2/20.5	ROUGE-1/2/L = 41.9/15.8/32.8	ACC = 84.3	BLEU = 12.2
Qwen2.5-1.5B -Instruct	F1 = 58.6 EM = 55.4	ROUGE-1/2/L = 18.9/3.8/13.2	ROUGE-1/2/L = 23.7/7.8/16.5	ROUGE-1/2/L = 36.4/13.0/28.5	ACC = 69.9	BLEU = 8.0
$\mathrm{DET}_{\mathrm{NER}}$ $+\mathrm{GEN}_{\mathrm{rand}}$ $+\mathrm{REP}_{\mathrm{direct}}$	F1 = **76.6** EM = **73.0**	ROUGE-1/2/L = 22.5/4.5/15.3	ROUGE-1/2/L = 28.3/8.7/18.9	ROUGE-1/2/L = **41.0/15.2/32.1**	ACC = 81.6	BLEU = 9.9
$\mathrm{DET}_{\mathrm{NER}}$ $+\mathrm{GEN}_{\mathrm{prompt}}$ $+\mathrm{REP}_{\mathrm{direct}}$	F1 = 75.7 EM = 71.2	ROUGE-1/2/L = **23.0**/4.9/15.8	ROUGE-1/2/L = 28.8/8.7/19.2	ROUGE-1/2/L = 40.7/**15.2**/31.9	ACC = **83.0**	BLEU = 9.5
$\mathrm{DET}_{\mathrm{prompt}}$ $+\mathrm{GEN}_{\mathrm{prompt}}$ $+\mathrm{REP}_{\mathrm{prompt}}$	F1 = 74.8 EM = 70.9	ROUGE-1/2/L = 22.9/**5.7**/**15.9**	ROUGE-1/2/L = 24.4/7.1/16.3	ROUGE-1/2/L = 32.3/11.3/25.5	ACC = 80.0	BLEU = 9.2
$\mathrm{DET}_{\mathrm{NER}}$ $+\mathrm{GEN}_{\mathrm{rand}}$ $+\mathrm{REP}_{\mathrm{gen}}$	F1 = 66.5 EM = 61.7	ROUGE-1/2/L = 19.0/3.6/13.1	ROUGE-1/2/L = 23.0/6.1/15.6	ROUGE-1/2/L = 34.7/12.0/27.1	ACC = 78.2	BLEU = 10.1
$\mathrm{DET}_{\mathrm{NER}}$ $+\mathrm{GEN}_{\mathrm{prompt}}$ $+\mathrm{REP}_{\mathrm{gen}}$	F1 = 67.9 EM = 62.8	ROUGE-1/2/L = 19.6/3.8/13.6	ROUGE-1/2/L = 24.1/6.6/16.1	ROUGE-1/2/L = 34.3/11.6/26.7	ACC = 81.6	BLEU = **10.5**
$\mathrm{DET}_{\mathrm{tag\_mask}}$ $+\mathrm{GEN}_{\mathrm{prompt}}$ $+\mathrm{REP}_{\mathrm{gen}}$	F1 = 74.1 EM = 70.6	ROUGE-1/2/L = 21.9/4.7/15.2	ROUGE-1/2/L = **29.7/9.7/20.1**	ROUGE-1/2/L = 40.8/15.0/31.7	ACC = 80.8	BLEU = 6.9
$\mathrm{DET}_{\mathrm{tag\_rep}}$ $+\mathrm{GEN}_{\mathrm{prompt}}$ $+\mathrm{REP}_{\mathrm{gen}}$	F1 = 71.3 EM = 66.8	ROUGE-1/2/L = 20.5/3.8/14.0	ROUGE-1/2/L = 19.8/5.0/13.8	ROUGE-1/2/L = 40.4/14.9/31.5	ACC = 81.6	BLEU = 8.0

Table 2: Performance of various pseudonymization methods across different NLP tasks and datasets. The bolded parts in the table represent **the best results excluding the large-scale LLM**.

the BLEU-4 (Papineni et al., 2002) score as the evaluation metric. In addition to these performance evaluation metrics, we also calculate the distance between the original text X and the replaced text X', defined as $1 - s(X, X')$, to assess the effectiveness of the pseudonymization method. Here, $s(X, X')$ represents the cosine similarity between the sentence embedding vectors of X and X', both of which are computed using a pretrained model, All-Mpnet-Base-V2 [4].

Baseline Methods We designed two baseline methods and compared the pseudonymization method described in this paper with these baselines: (1) directly using a cloud-based LLM (simulated using a locally deployed large-scale LLM) to perform experimental NLP tasks, and (2) directly using a local small-scale instruction-tuned LLM to perform experimental NLP tasks.

Implementation Details For the efficiency of the experiments, we locally deployed the Qwen2.5-14B-Instruct[5] as the large-scale LLM to simulate the cloud-based LLMs. We used the Qwen2.5-1.5B-Instruct[6] as the local small-scale instruction-tuned LLM for the prompt-based detection, generation, and replacement methods. As described in Section 4, we then fine-tuned the Qwen2.5-1.5B model[7] to output either a repetition of the input text or the results of the Seq2Seq detection method for executing the replacement approach through controllable text generation. A total of 20,000 sam-

[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[5] https://huggingface.co/Qwen/Qwen2.5-14B-Instruct

[6] https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct

[7] https://huggingface.co/Qwen/Qwen2.5-1.5B

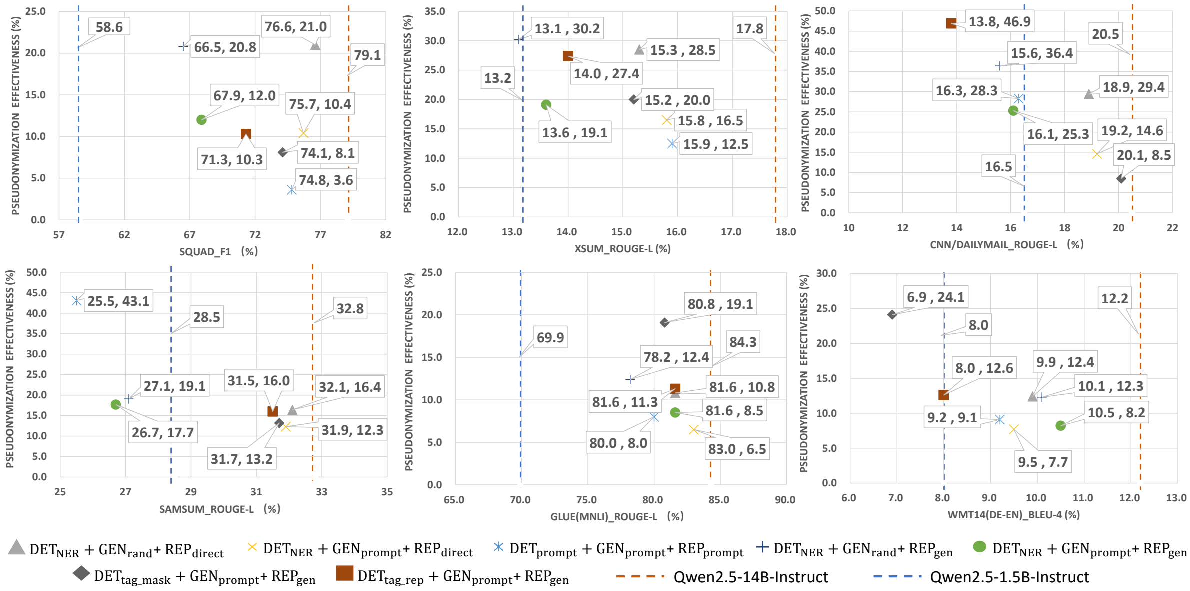

Figure 4: Performance metrics and pseudonymization effectiveness of various methods across different datasets

ples were randomly selected from the training sets of each dataset. Following the procedure outlined in Table 1, these samples were preprocessed and subsequently used as fine-tuning data. We fine-tuned the Qwen2.5-1.5B model for 3 epochs using a learning rate of 1.0e-4.

5.2 Main Result

Notably, each component of the proposed pseudonymization framework is decoupled, allowing the methods described in Section 3 to be freely combined. We evaluate the majority of possible method combinations and present the results of several representative approaches, comparing them against the baselines. The results are shown in Table 2. It is evident that across various NLP tasks and datasets, pseudonymization methods based on the proposed framework achieve results comparable to those of the large-scale LLM baseline. Specifically, these methods achieve over 95% of the large-scale LLM baseline's performance on SQuAD 2.0, CNN/DailyMail, SAMSum, and GLUE (MNLI), over 90% on XSum, and approximately 85% on WMT14 (de-en). Across all datasets, the proposed methods significantly outperform the small-scale LLM baseline. It is important to note that, in real-world scenarios, the parameter scale of cloud-based models is expected to be significantly larger than that of the locally deployed large-scale LLM baseline. This further highlights the necessity of the pseudonymization framework proposed in this paper for enabling the secure remote use of cloud-based large-scale LLMs.

We further compared the key performance metrics and pseudonymization effectiveness of each method across different NLP tasks and datasets, with the results visualized in Figure 4. An interesting finding is that, in tasks like QA and summarization, which are less reliant on the semantic details of the text, the combination of $\mathrm{DET_{NER}} + \mathrm{GEN_{rand}} + \mathrm{REP_{direct}}$ achieves the best overall results in both performance metrics and pseudonymization effectiveness. However, in tasks like MNLI and MT, where text details significantly impact the results, the combination of $\mathrm{DET_{NER}}+\mathrm{GEN_{rand}}+\mathrm{REP_{gen}}$ and $\mathrm{DET_{tag_mask}}+\mathrm{GEN_{prompt}} + \mathrm{REP_{gen}}$ consistently yields the best overall performance.

Table 3 presents an example of the correct output generated by the proposed method. In this example, entities in the premise and hypothesis texts, such as "Vosges" and "Rhine Valley", were replaced with other entities, like "Eifel Mountain" and "Danube River Basin", using the combination of $\mathrm{DET_{NER}} + \mathrm{GEN_{rand}} + \mathrm{REP_{gen}}$. This effectively protects the potential privacy information contained within those entities. Meanwhile, when the pseudonymized text was processed by a large-scale LLM, it generated the correct inference, whereas the small-scale model failed to do so.

5.3 Discussion

We further evaluated the effectiveness of various methods in achieving the stage-specific objectives throughout the different stages of the proposed pseudonymization framework.

<table>
<tr><td>Premise</td><td colspan="2">The vineyards hug the gentle slopes between the Vosges and the Rhine Valley along a single narrow 120-km (75-mile) strip that stretches from Marlenheim, just west of Strasbourg, down to Thann, outside Mulhouse.</td></tr>
<tr><td>Hypothesis</td><td colspan="2">The slopes between the Vosges and Rhine Valley are the only place appropriate for vineyards.</td></tr>
<tr><td>Answer</td><td colspan="2">neutral</td></tr>
<tr><td>Large-scale LLM</td><td colspan="2">neutral (correct)</td></tr>
<tr><td>small-scale LLM</td><td colspan="2">contradiction (incorrect)</td></tr>
<tr><td rowspan="3">$\mathbf{DET_{NER}}$
$\mathbf{+GEN_{rand}}$
$\mathbf{+REP_{gen}}$</td><td>Premise:</td><td>The vineyards hug the gentle slopes between the Eifel Mountains and the Danube River Basin along a single narrow 120-km (75-mile) strip that stretches from Marsden, just west of Erlangen, down to Thompson, outside Lyon City.</td></tr>
<tr><td>Hypothesis:</td><td>The slopes between the Eifel Mountains and Danube River Basin are the only place appropriate for vineyards.</td></tr>
<tr><td>Answer:</td><td>neutral (correct)</td></tr>
</table>

Table 3: Example of correct output by the proposed method on GLUE (MNLI) dataset compared to baselines

	NER	prompt	tag_mask	tag_rep
PRR	**65.7**	47.9	33.5	43.1

(a)

	rand	prompt
PPS	**74.9**	45.2

(b)

	direct	prompt	gen
SCS	20.9	19.7	**19.2**

(c)

Table 4: (a) Privacy Removal Rate (PRR) for each detection method. (b) Privacy Preservation Score (PRS) for each generation method. (c) Semantic Correctness Score (SCS) for replacement method.

First, we calculate the Privacy Removal Rate (PRR) for each privacy entity detection method using the formula $\mathrm{PRR} = \frac{\mathrm{card}(P' \cap P)}{\mathrm{card}(P)} \times 100(\%)$, where $\mathrm{card}(\cdot)$ denotes the cardinality of the corresponding set. The results are shown in Table 4 (a). Notably, the NER-based detection method yielded the highest PRR.

We compute the Privacy Preservation Score (PPS) for each replacement candidate generation method as the average distance between $p'^{j}_{A_i}$ and $q^{j}_{A_i}$, following the formula $\mathrm{PPS} = \mathrm{avg}(1 - s(p'^{j}_{A_i}, q^{j}_{A_i})) \times 100(\%)$. It is evident that a higher PPS score indicates greater difficulty in inferring the privacy entity from the replacement entity, thereby offering better protection for privacy information. The results are presented in Table 4 (b). Notably, the random sampling generation method achieved the highest PPS.

We compute the Semantic Correctness Score (SCS) to assess the effectiveness of each entity replacement method by measuring the perplexity of X' using Qwen2.5-1.4B-Instruct. The SCS is calculated as $\mathrm{SCS} = \mathrm{avg}(\mathrm{loss}(f(x'_{<i}), x'_i))$ ($x'_i \in X'$), where $f(\cdot)$ represents the next-token prediction function, and $\mathrm{loss}(\cdot)$ denotes the loss function of the language model. A lower SCS indicates that X' better aligns with the probability distribution of the language model, thereby exhibiting higher semantic correctness. The results are presented in Table 4 (c). Notably, replacement through controllable text generation achieved the lowest SCS.

6 Conclusion

In this paper, we outline the process of remote user interaction with LLMs and propose a comprehensive definition of a pseudonymization framework applicable to cloud-based LLMs. We believe that this framework provides a universally applicable approach to the text pseudonymization process and can serve as a guide for future research in this area. Additionally, we introduce a pseudonymization method based on a controllable text generation process, which ensures that the replaced text maintains maximal semantic correctness. Experimental results demonstrate that the proposed framework strikes an optimal balance between privacy protection and utility.

Limitations

The primary limitation of this work is that the pseudonymization process is implemented through three relatively independent processing stages rather than an end-to-end machine learning approach. However, even end-to-end pseudonymization methods must inherently incorporate the three stages outlined in this paper: detection, generation, and replacement. Given that these stages have distinct problem definitions and task objectives, integrating them into a unified end-to-end framework presents a significant challenge. Addressing this challenge will be a key focus of our future research.

In addition, we utilized straightforward methods to accomplish the objectives of each stage, such as NER and prompt-based approaches. However, the primary contribution of this work lies in proposing a general pseudonymization framework. Within this framework, incorporating more advanced methods at each stage is expected to enhance overall performance.

For the sake of experimental efficiency, this work employs the same entity replacement method in both the restoration and pseudonymization processes. However, in practical applications, different replacement methods could be utilized for these two processes, potentially enhancing the overall effectiveness of the approach.

Although this work has validated the effectiveness of the proposed framework and methods on multiple NLP tasks across different datasets, certain tasks, such as text continuation, remain unexplored. Text continuation presents unique challenges for pseudonymization and restoration, as it may generate entities not present in the input text. Future work will include experiments to address this aspect.

References

Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51(4):1–35.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*, pages 2280–2292.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284.

Yu Chen, Tingxin Li, Huiming Liu, and Yang Yu. 2023. Hide and seek (has): A lightweight framework for prompt privacy protection. *arXiv preprint arXiv:2309.03057*.

Jianfeng Chi, Emmanuel Owusu, Xuwang Yin, Tong Yu, William Chan, et al. 2018. Privacy partitioning: Protecting user data during the deep learning inference phase. *arXiv preprint arXiv:1812.02863*.

Caiqin Dong, Jian Weng, Jia-Nan Liu, Yue Zhang, Yao Tong, Anjia Yang, Yudan Cheng, and Shun Hu. 2022. Fusion: Efficient and secure inference resilient to malicious servers. *arXiv preprint arXiv:2205.03040*.

Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. 2023. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 2665–2679.

Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.

Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79.

Oded Goldreich. 1998. Secure multi-party computation. *Manuscript. Preliminary version*, 78(110):1–108.

Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. 2022. Iron: Private inference on transformers. *Advances in neural information processing systems*, 35:15718–15731.

Zhicong Huang, Wen-jie Lu, Cheng Hong, and Jiansheng Ding. 2022. Cheetah: Lean and fast secure {Two-Party} deep neural network inference. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 809–826.

Marija Jegorova, Chaitanya Kaul, Charlie Mayor, Alison Q O'Neil, Alexander Weir, et al. 2022. Survey: Leakage and privacy at inference time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):9090–9108.

Zhigang Kan, Linbo Qiao, Hao Yu, Liwen Peng, Yifu Gao, and Dongsheng Li. 2023. Protecting user privacy in remote conversational systems: A privacy-preserving framework based on text sanitization. *arXiv preprint arXiv:2306.08223*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Guo Lin, Wenyue Hua, and Yongfeng Zhang. 2024. Emojicrypt: Prompt encryption for secure communication with large language models. *arXiv preprint arXiv:2402.05868*.

Wen-jie Lu, Zhicong Huang, Zhen Gu, Jingyu Li, Jian Liu, Cheng Hong, Kui Ren, Tao Wei, and WenGuang Chen. 2023. Bumblebee: Secure two-party inference framework for large transformers. *Cryptology ePrint Archive*.

Jimit Majmudar, Christophe Dupuy, Charith Peris, Sami Smaili, Rahul Gupta, and Richard Zemel. 2022. Differentially private decoding in large language models. *arXiv preprint arXiv:2205.13621*.

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

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. 2020. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1314–1331. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pranav Rajpurkar, Jian Zhang, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL 2018*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, et al. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. In the Proceedings of ICLR.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Xinwei Wu, Li Gong, and Deyi Xiong. 2022. Adaptive differential privacy for language model training. In *Proceedings of the First Workshop on Federated Learning for Natural Language Processing (FL4NLP 2022)*, pages 21–26.

Mengwei Xu, Dongqi Cai, Yaozong Wu, Xiang Li, and Shangguang Wang. 2024. Fwdllm: Efficient fedllm using forward gradient. *arXiv. Available at: hjp://arxiv. org/abs/2308.13894 (Accessed: 11 March 2024)*.

Biwei Yan, Kun Li, Minghui Xu, Yueyan Dong, Yue Zhang, Zhaochun Ren, and Xiuzhen Cheng. 2024. On protecting the data privacy of large language models (llms): A survey. *arXiv preprint arXiv:2403.05156*.

Oleksandr Yermilov, Vipul Raheja, and Artem Chernodub. 2023. Privacy-and utility-preserving nlp with anonymized data: A case study of pseudonymization. *arXiv preprint arXiv:2306.05561*.

Sixing Yu, J Pablo Muñoz, and Ali Jannesari. 2023. Federated foundation models: Privacy-preserving and collaborative learning for large models. *arXiv preprint arXiv:2305.11414*.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. 2024a. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6915–6919. IEEE.

Kaiyan Zhang, Jianyu Wang, Ermo Hua, Biqing Qi, Ning Ding, and Bowen Zhou. 2024b. Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following. *arXiv preprint arXiv:2403.03129*.