Protecting Privacy in Classifiers by Token Manipulation

Anonymous ACL submission

Abstract

Using language models as a remote service entails sending private information to an untrusted provider. In addition, potential eavesdroppers can intercept the messages, thereby exposing the information. In this work, we explore the prospects of avoiding such data exposure at the level of text manipulation. We focus on text classification models, examining various token mapping and contextualized manipulation functions in order to see whether classifier accuracy may be maintained while keeping the original text unrecoverable. We find that although some token mapping functions are easy and straightforward to implement, they heavily influence performance on the downstream task, and via a sophisticated attacker can be reconstructed. In comparison, contextualized manipulation provides an improvement in performance.

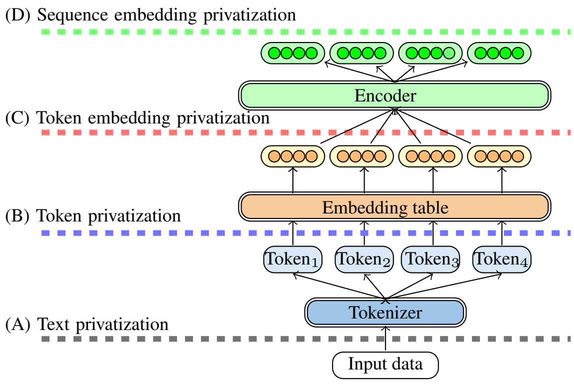

Figure 1: A schematic of the various stages where differential privacy techniques can be applied in an LLM. This work focuses on level (B).

1 Introduction

Large language models (LLMs) have greatly advanced the field of NLP in recent years, exhibiting exceptional proficiency across a wide spectrum of tasks, including dependency parsing (Duong et al., 2015), natural language understanding (Dong et al., 2019), automatic question-answering (OpenAI, 2021; Ouyang et al., 2022), machine translation (Dabre et al., 2020), text classification (Minaee et al., 2021), and many more (Li et al., 2022). However, this success comes with potential privacy risks, as the models process vast amounts of data that might contain personal or sensitive information and may abuse or leak it. For instance, information can be leaked by model inversion (Li et al., 2017), re-identification techniques (Lison et al., 2021; Ben Cheikh Larbi et al., 2023), exploitation of feature memorization within the LLM (Carlini et al., 2021), and more. Offering LLMs as cloud services, such as ChatGPT (Ouyang et al., 2022), might also impose potential threats to privacy if the server exhibits a semi-honest stance, actively seeking to glean more insights from the input than is appropriate or by a possible eavesdropper intercepting the input sent to the server.

In order to safeguard privacy, many privacy-preserving techniques have been proposed, based on the local differential privacy framework (LDP; Arachchige et al., 2019). In this framework, the user applies a differential privacy mechanism, which can be hosted on a local server, and then sends the privatized data to the remote server. This approach doesn't require trust from the remote server, and protects the data against potential eavesdroppers. In general, any privacy mechanism can be applied at one or several components of the LLM pipeline. Figure 1 depicts these components: at the text level (*text privatization*), after the tokenization process (*token privatization*), after the initial embedding lookup (*token embedding privatization*), or after applying several layers of the encoder (*sequence embedding privatization*).

Currently, most privacy-preserving strategies focus on incorporating noise into sequence embedding vectors. The rationale behind this strategy is to minimize the privacy-preserving technique's impact on the downstream task. Specifically, most systems first obtain a sequence embedding representation, either by assuming partial access to the remote model (Zhou et al., 2022; Lyu et al., 2020; Qu et al., 2021) or by using a dedicated model to create these embeddings (Li et al., 2018; Coavoux et al., 2018; Mosallanezhad et al., 2019; Plant et al., 2021; Zhou et al., 2023). Afterwards, random noise is incorporated into the embeddings, thus concealing the original input. However, this approach relies on partial access to the remote model, on the ability to provide input to the remote model in vector form, or on sufficient computational and memory resources on the user's end. These are often not the case. In addition, Kugler et al. (2021) showed that publishing a model's encoder along with the contextualized embeddings allows an adversary to generate data to train a decoder with a high level of reconstruction accuracy, making these approaches highly susceptible to violation of privacy.

We propose **a secure way to use LLMs without assuming access to their parameters**. In our framework, both input and output for the privacy-providing mechanism must be given in a token sequence format, eliminating the need to intervene with the LLM's pre-training procedure or text processing. We focus on applying privacy preservation techniques at the token level, corresponding to layer (B) in Figure 1.

Specifically, we propose two privacy-preserving techniques based on manipulating **the input token sequence**. The first set of techniques relies on naïve rules of token substitution. The second is based on leveraging contextual information to strategically replace tokens, aiming to retain as much actionable information as possible for the classifier to minimize the impact on the performance of the downstream task. We test these techniques both for their impact on the downstream task accuracy and for their resilience against reconstruction attacks. We find that replacing tokens based on simple rules is easy for a knowledgeable attacker to reverse, while manipulating tokens based on contextual information can enhance privacy without sacrificing much of the performance.

2 Lossy Mapping

In order to protect against potential eavesdropping by a middle party, under the assumption that the layers of LLMs are inaccessible to the local device, we start by employing several mapping functions on the tokens of the input text available at the local device. Our initial, naïve mapping functions introduce a random noise component that follows a specific rule: the vocabulary is partitioned into pairs of tokens (u, v), or triplets (u, v, z), and when encountered in an input text to be manipulated, all tokens are mapped to a single representative token of their tuple, without loss of generality u. This strategy produces outputs that are inherently ambiguous, blocking any potential eavesdroppers from recovering the original input text deterministically, given that a many-to-one mapping is not invertible. The only available recourse for an attacker is a statistical strategy, which imposes assumptions on the properties of the input, for example that it was grammatical English text written by a speaker with high proficiency. Indeed, even if an eavesdropper obtains full information of the privacy system, i.e. the partition into token tuples and each tuple's representative token, each mapped sequence of length m still generates a candidate set of 2^m or 3^m possible permutations (depending on tuple size) through which the attacker must search. We will examine the practical implications of this large search space later in the section.

For our stated use case of manipulating text being input into a sequence classifier operating atop an LLM, there are two distinct scenarios depending on when we may apply our manipulation. The first scenario involves applying the manipulation process only during the inference phase of a model trained on regular, unmanipulated text, which we will refer to as the TEST case. This operation mode simulates a query sent by a user to an already-trained model, such as a user interacting with ChatGPT or another model allowing only inference text interaction via user interface or an API. In the second scenario, which we call ALL, we also apply the manipulation during the training phase, protecting sensitive information in the training data, hoping that the inference phase will now leverage the model's ability to handle manipulated input as expected and produce better results. In this scenario the model does not inadvertently learn or memorize the sensitive data during the training process, nor does it spend learning resources on tokens never to be seen during inference, but since it is not always possible to assume its availability, we perform our experiments in both settings.

When protecting the original input data, it is essential for the mapper to have minimal impact on the performance of the downstream task, defining the fundamental trade-off in our study. Therefore,

Dataset	Mapper	TEST	ALL	Unchanged Tokens
SST2	Plain text	94.5%	94.5%	100%
	2-Random	75.0%	85.0%	51.0%
	3-Random	62.0%	80.0%	34.0%
	High-freq	90.0%	91.0%	93.0%
	Low-freq	60.0%	78.0%	7.0%
IMDb	Plain text	95.0%	95.0%	100%
	2-Random	75.0%	90.0%	50.0%
	3-Random	68.0%	85.0%	32.0%
	High-freq	93.0%	94.0%	94.0%
	Low-freq	60.0%	80.0%	6.0%

Table 1: The mapping strategy accuracy on SST2 and IMDb datasets and the percentage of unchanged tokens after applying the mappers to the training and test sets.

Mapper	Text		
Plain Text	**no**	**apparent**	**joy**
2-Random	his	buffers	University
High-freq	no	apparent	joy
Noise(150)	non	evident	joyful
STEN(9, 0.8)	No	evident	joyful
STEN$_p$(9, 1.0)	apparent	No	joyful

Table 2: Examples of the privatized textual sequences obtained with different privacy-preserving techniques.

the selection process for grouping tokens and selecting each tuple's representative token is crucial, as it aims to both minimize the mapping's effect on the downstream task and hinder the attacker's ability to uncover the original text. We consider the following mapping functions:

Purely random mapping the selection of the token pairs tuples from the vocabulary and of each tuple's representative is uniformly random.

High-frequency mapping token pairs are selected based on their frequency of occurrence in a tokenized corpus, such as Wikipedia (Foundation, 2023). This involves pairing a higher-frequency token with a lower-frequency token, with the higher-frequency token being designated as the representative. In our mapper, given a vocabulary of even size V, sorted by descending frequency, each token with rank $1 \leq k \leq \frac{V}{2}$ is paired with the token of rank $k + \frac{V}{2}$. While selecting the high-frequency token as the representative may have a lesser impact on the downstream task, it could potentially weaken the privacy-preserving characteristics, depending on the knowledge possessed by the attacker.

Low-frequency mapping the process is similar to that of the higher-frequency mapper, except that the lower-frequency token is chosen as the representative. Opting for less-frequent tokens as representatives can aid in preserving privacy, but it will likely harm the downstream task.

Due to the simplicity of these mapping strategies, we consider them baselines for further research and developing better, potentially language-aware strategies. In addition, these mapping functions can easily be generalized to larger tuples, expanding the search space even further, but greatly harming downstream task performance as a result of a much more restricted active vocabulary.

2.1 Task Performance

To assess the impact of the baseline models on downstream task performance, we use two datasets for sequence classification: SST2 (Socher et al., 2013) and IMDb (Maas et al., 2011). The base model chosen was RoBERTa (Liu et al., 2019), a state-of-the-art encoder language model known for its strong performance in sequence classification tasks. In Table 1, we present the results of four baselines on the two datasets, compared with the null mapping results labeled "Plain text". Perhaps unsurprisingly, the high-frequency baseline achieved the highest accuracy, most likely due to the fact that retaining high-frequency tokens while removing low-frequency ones results in a relatively small number of tokens altered in the datasets. In both datasets this number is roughly 6%, compared with low-frequency mapping's complement of 94% and with the randomly-selected sets' 50% and 67%, giving a correlative relationship between this number and the performance level: the fewer tokens are altered, the better the model performs. This effect is much more pronounced when only the test set is affected, and the model is dealing not only with loss of information but also with out-of-distribution behavior. In absolute terms, we find it remarkable that this alteration of a non-negligible portion of tokens causes only a 1–2 percentage point reduction in performance for the IMDb dataset and still under 5 points for SST2.

In Table 2, we present an example of the outcome of applying the 2-Random and the High-freq privatization techniques on a random phrase ("no apparent joy") from the SST2 dataset. As expected, the 2-random baseline produces a random sequence of words, whereas the high-frequency mapper leaves the phrase unchanged as the tokens in the original sequence are frequent.

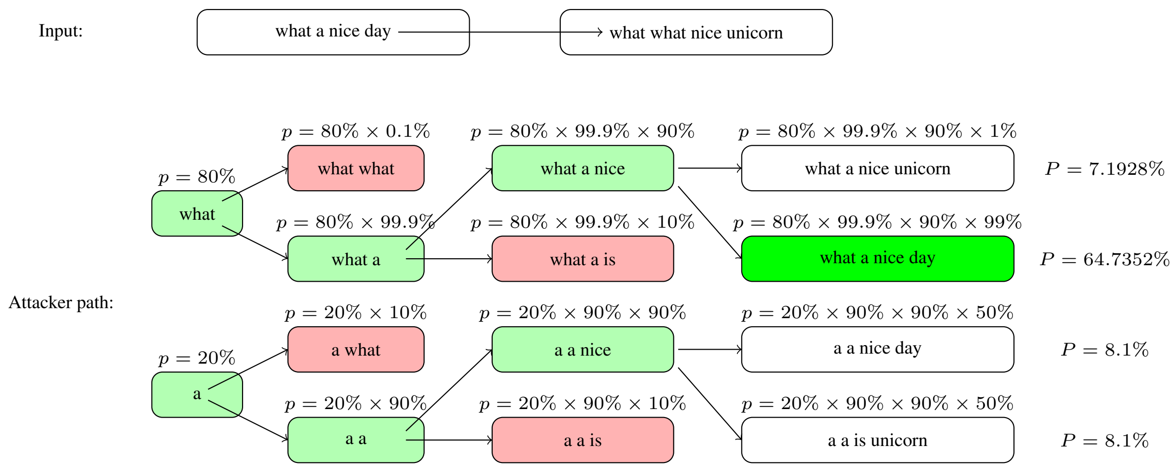

Figure 2: Schematic overview of the proposed heuristic oracle attacking scenario path over trying to reconstruct the sentence "what a nice day" which is remapped to "what what nice unicorn". The red boxes indicate that the probability (presented above the box) of the candidate is low enough to be dropped in the next step, while the green boxes are the candidates that will be expanded in the next step.

2.2 Brute-force Attacker

Although the many-to-one mapping function introduces some form of protection against data leakage, in practice, reconstructing the original text might be relatively straightforward under certain circumstances. In particular, if an "oracle" attacker has access to the token pairings, it can theoretically determine the original text from the pool of 2^m possible permutations by applying a generative LLM such as GPT (Radford et al., 2019) and picking the most probable sequence. However, generating and evaluating all 2^m permutations is impractical even for small values of m due to the computational complexity involved. To mitigate this challenge, alternative approaches, such as employing heuristics or utilizing statistical methods, can be explored to narrow down the potential candidates for the original text.

To cope with this task, we describe a heuristic approach to reducing the search space based on **beam search** (Eisenstein, 2019, §11.3.1) and **nucleus sampling** (Holtzman et al., 2019). In each step of the process, candidates are generated based on the prefixes of tokens that were produced in the previous steps. In the case of token pairs, each prefix sequence is followed by one of two candidate tokens for the next step based on the known (oracle) token pair that the observed representative token belongs to. Unlike conventional beam search, where a fixed number of candidates is retained following each step, we opt for a dynamic approach inspired by nucleus sampling, made possible since the scores for each of the two tokens reflect a generative probabilistic process where the relative probability of each interim token sequence on the beam can be estimated and used for dropping highly unlikely sequence prefixes. This means that the number of candidates remaining on the beam varies at each step, adapting to their likelihood and ensuring flexibility in the selection process. We estimate the likelihood of each candidate prefix using a language model.[1] After all prefixes on the beam have been scored, we remove the least probable candidates such that the total probability of the remaining candidates exceeds a certain threshold π set by computational constraints but maintaining discoverability. Since the probability of a sequence cannot exceed that of its prefix, the process guarantees that complete sequences that are likely are not being discarded before getting the chance to be fully generated. Overall, this process effectively eliminates highly unlikely candidates, dramatically reducing the search space during its application and streamlining the computational efforts.

This process is illustrated in Figure 2. The "oracle" attacker gains access to the remapped words: `(what,a)` $\rightarrow$ `a`, `(nice, is)` $\rightarrow$ `nice`, `(day, unicorn)` $\rightarrow$ `unicorn`. In the first step, two initial candidates (`what` and `a`) are generated based on the first observed token (`what`). Following the described process, each prefix is evaluated via an LLM to determine its probability, for instance, the probability of `what` being the first word is 80% when considering the possible set `{[s] what,`

[1] https://github.com/simonepri/lm-scorer

Dataset	Mapper	MRR (↓)	Pr@5 (↓)	Edit dist (↑)
SST2	2-Random	0.89	0.97	1.32
	3-Random	0.81	0.92	1.35
	High-freq	0.86	0.98	1.33
IMDb	2-Random	0.48	0.59	1.60
	3-Random	0.45	0.53	1.70
	High-freq	0.63	0.72	1.60

Table 3: The three random mappings' capability of preserving privacy against an "oracle" attacker. Edit distance is calculated at the token level.

`[s] a}`. This process is repeated, and the candidates with low probability are removed, such that the total probability of the remaining candidates is above 85%, as indicated by the red boxes. Finally, the probability of the sequence `what a beautiful day` is the highest, thus the "oracle" attacker returns it as the inferred original text. We note that the low-frequency and high-frequency mappers, despite their differences in representative token selection, will demonstrate equivalent safeguarding mechanisms against this attacker since the attacker does not factor in the choice of the representative token and examines all potential candidates in its effort to uncover the original text.

2.3 Resilience Against Reconstruction Attacks

In Table 3, we present the outcomes of the attacker's endeavors to reveal the original text from the three techniques: 2-Random, 3-Random, and High-freq (equivalent to Low-freq for a knowledgeable attacker). We report the mean reciprocal rank (MRR) of the correct sequences, the rate of the actual input sequence ranking among the top 5 predictions (Pr@5), and the token-level edit distance between the produced top prediction and the original sequence. The relative success of the mappers in thwarting the oracle attacks on the IMDb dataset compared to SST2 can be attributed to the average token sequence length ($\bar{m}$), which is 65 and 12, respectively. As sequence length increases, the attacker's task of uncovering the original text becomes more challenging.

Our results indicate that the naïve baselines are overly simplistic and allow an easy and straightforward reconstruction, even within a vast search space (although attacker knowledge of the mapping specifications is required). In cases where performance on the task remains close to that of unmapped text, the recovery price is too high to neglect. Having said that, the computational complexity of applying the naïve baselines is relatively low, and the greatly reduced active vocabulary brings great savings in parameter budgets, which embedding tables often dominate. In a less powerful attack environment, this would make them an efficient choice for preserving privacy on low-resource devices. We expect future work on more principled many-to-one static mappings would be able to improve both task performance and resilience to attackers, while work on attack strategies can present challenges hitherto unseen.

3 STENCIL Privacy Preservation

In the context of protecting privacy within NLP practices, a widely adopted approach for implementing local differential privacy involves introducing a controlled level of *noise* into different components of the model, effectively concealing the original input. These components may include sequence embeddings, token embeddings, or the tokens themselves (Mosallanezhad et al., 2019; Feyisetan et al., 2020; Lyu et al., 2020; Qu et al., 2021; Zhou et al., 2022). However, in essence, the success of models in most NLP tasks is primarily attributed to their effective utilization of contextual information. Moreover, our study focuses on token-level privacy preservation, i.e., we assume that the parameters of the LLMs are inaccessible, making the importance of contextual information more pronounced. Therefore, a fundamental limitation associated with incorporating noise is the exclusion of contextual information when defining the noise. This omission may hinder the potential benefits contextual details can offer for maintaining the performance of the downstream tasks.

Given this limitation, we propose a new privacy preservation technique, which we call STENCIL.[2] With this technique, a mapped token in a sequence "absorbs" information from adjacent tokens to form a new context-aware token, effectively concealing the original token while retaining information beneficial for maintaining task performance.

In order to generate the new contextualized token $t_k \to t'_k$, we first retrieve an embedding vector representation of the neighborhood, of size $n+1$, containing the tokens $t_i, \forall i \in \{k - n/2 \dots k + n/2\}$ using some embedding lookup table $\mathbf{E} \in \mathbb{R}^{V \times d}$,

[2]This term hails from numerical analysis (Spotz, 1995), where it denotes a computation that involves the surrounding values.

which can be trained independently in a preliminary step or obtained from an available model such as the target model itself. We then subject the $n+1$ embedding vector representations to a weighted transformation and incorporate them to form a new "quasi-embedding" vector $\sum_{i=k-n/2}^{k+n/2} f_i \cdot \mathbf{E}[t_i]$. Finally, we return the token t'_k that is closest to the quasi-embedding vector in the embedding space, based on cosine-similarity or euclidean distance computation, as an output. To further enhance privacy, we ensure that the new token is different from the original one. Formally, the process can be defined as follows:

$$t'_k = \underset{t_j \in \mathcal{V}}{\arg\min} \left\| \mathbf{E}[t_j] - \sum_{i=k-\frac{n}{2}}^{k+\frac{n}{2}} f_i \cdot \mathbf{E}[t_i] \right\|, \quad (1)$$

where $\mathcal{V}$ is the vocabulary and f_i is the weighted transformation function of the tokens such that $\sum_{i=k-\frac{n}{2}}^{k+\frac{n}{2}} f_i = 1$.

The level of privacy enhancement and its impact on the downstream task by employing the STENCIL method can be managed by adjusting the window size and the properties of the weighted function f. In our study, we use the gaussian smoothing function as the weighted function. Consequently, the standard deviation, σ, plays a crucial role in the performance and amount of privacy achieved.

In our experiments, we compared our STENCIL mechanism to two other privacy-preserving techniques. The first, technique was proposed by Qu et al. (2021)'s, namely the NOISE mapper. In contrast to our proposed technique, this approach does not consider context but rather incorporates random noise into token embeddings to enhance privacy. Similar to our proposed method, the new token is the closest to the quasi-embedding vector in the embedding space. The random noise is obtained by multiplying a sample from a Gamma distribution $\Gamma(d, 1/\eta)$ and a uniform sample from a unit hypersphere, where η corresponds to the amount of noise introduced to the original token and d is the dimension of the embedding space.

For the second technique, we include Chen et al. (2023)'s CUSTEXT$^+$ privacy-preserving mechanism. The CUSTEXT$^+$ mechanism consists of a mapping procedure and a sampling function. The mapping procedures generate a list of the top K tokens for each token, selecting those with the highest semantic relevance to the original token. Similar to the NOISE and STENCIL mechanisms, semantic relevance is determined by calculating either the cosine similarity or Euclidean distance of the quasi-embedding vectors. Then, each token is remapped to one of the K candidates using an exponential sampling function.

We note that the most time-intensive operation in all mechanisms is searching for the closest token to the perturbed quasi-embedding vector, whereas all other operations are negligible in comparison. Overall, the average computational cost per token is 0.005 seconds on two 16-core 3.2 GHz AMD EPYC 7343 Milan processors.

3.1 STENCIL$^+$ and STENCIL$_p$ Mechanisms

Identifying sensitive words, such as those involved in named entity recognition (e.g., names, addresses, workplaces), is a challenging task typically approached using statistical methods (Liu et al., 2017; Cohn et al., 2019; Poostchi et al., 2018; Friedrich et al., 2019). As a result, our mechanism treats all tokens as sensitive since we cannot reliably distinguish sensitive from non-sensitive tokens. However, since treating stopwords as non-sensitive may pose a low privacy risk (Chen et al., 2023), we propose STENCIL$^+$, which applies the STENCIL mechanism to all words except stopwords, thereby enhancing accuracy while maintaining privacy.

An additional variation of STENCIL, namely STENCIL$_p$, can be obtained by excluding the target token from the computation of the quasi-embedding vector in (1) by setting f_k to zero. This exclusion significantly improves the privacy of each token and diminishes the attacker's ability to reconstruct the original token at the expense of performance.

3.2 Downstream Task Performance

To evaluate the impact of the STENCIL, NOISE and CUSTEXT$^+$ methods on the model performance, we repeat the methodology outlined in §2: we use RoBERTa as the base model; SST2 and IMDb as the datasets; and the two distinct application cases: manipulating tokens on inference data only (TEST), and applying the technique during the training phase as well (ALL). However, as these privacy techniques exhibit a realistic case, we also test it on an encoder-decoder model T5-small (Raffel et al., 2020) on the QNLI task from the GLUE dataset (Wang et al., 2019). As in Raffel et al. (2020), we concatenate the question and its corresponding sentence to form a single sequence that serves as the input, while the target prediction is

Dataset	Mapper	TEST (↑)	ALL (↑)	Pr@5 (↓)
SST2	Plain Text	94.5%	94.5%	-
	NOISE(100)	80.0%	87.8%	70.0%
	NOISE(150)	83.0%	**90.0%**	75.0%
	CUSTEXT$^+$	79.4%	82.5%	70.0%
	STEN(9, 0.8)	83.5%	89.3%	49.0%
	STEN$^+$(9, 0.8)	**85.3%**	89.5%	47.0%
	STEN$_p$(9, 1.0)	84.7%	87.0%	**0.0%**
	STEN$_p^+$(9, 1.0)	85.0%	89.4%	**0.0%**
IMDb	Plain Text	95.0%	95.0%	-
	NOISE(100)	89.0%	92.6%	86.0%
	NOISE(150)	90.0%	93.5%	90.0%
	CUSTEXT$^+$	88.9%	91.1%	90.0%
	STEN(9, 0.8)	90.2%	93.1%	67.0%
	STEN$^+$(9, 0.8)	**92.4%**	**94.0%**	69.0%
	STEN$_p$(9, 1.0)	89.7%	92.4%	**0.0%**
	STEN$_p^+$(9, 1.0)	89.7%	92.4%	**0.0%**
QNLI	Plain Text	88.1%	88.1%	-
	NOISE(100)	80.0%	84.0%	93.0%
	NOISE(150)	81.1%	84.4%	93.0%
	CUSTEXT$^+$	78.5%	81.5%	85.0%
	STEN(9, 0.8)	74.8%	83.1%	54.0%
	STEN$^+$(9, 0.8)	**81.4%**	**84.8%**	52.3%
	STEN$_p$(9, 1.0)	67.9%	82.5%	**0.0%**
	STEN$_p^+$(9, 1.0)	71.4%	83.8%	**0.0%**

Table 4: The best results achieved by the different STENCIL mapper variations, the NOISE mapper and CUSTEXT$^+$ considering the Test and All cases on the SST2, IMDb, and QNLI datasets. Pr@5 represents the average token hit managed by the nearest-neighbor attacker.

either "entailment" or "not_entailment", thus forming a classification task.

We report three distinct manipulations based on STENCIL, STENCIL$^+$, and STENCIL$_p$. The weighting function f_i, for all three variations, is derived from a gaussian smoothing. For the STENCIL and STENCIL$^+$ mechanism, we consider a standard deviation of $\sigma = 0.8$, and the number of adjacent tokens considered is set to nine (four from each side, as well as the target token). The standard deviation we consider for the STENCIL$_p$ approach is $\sigma = 1.0$, with a window width of nine. In all approaches, to preserve model performance, the tokenizer and embedding lookup table used to derive the new tokens were sourced directly from the model being trained. For the NOISE mechanism, we report the two best η values: $\eta = 100, 150$. For the CUSTEXT$^+$ mechanism, the K parameter was set to 20 with the privacy parameter $\epsilon = 3$, which yields the overall best results.

The results are presented in Table 4. The overall best accuracy is achieved by STENCIL and STENCIL$^+$, demonstrating the advantage of utilizing contextual information to achieve privacy and maintain high performance. Nevertheless, in the SST2 dataset, NOISE ($\eta = 150$) achieves the highest accuracy. NOISE with $\eta = 150$ introduces minimal noise, resulting in negligible alterations to the original tokens. However, the NOISE method comes at great cost in discoverability, to be presented in §3.3.

Compared to the sentiment analysis tasks (SST2 and IMDb), the QNLI task presents greater challenges, primarily due to the complex logical connections required for the model to discern entailment between the given sentence and question. Therefore, despite its instance sizes being very similar to those of IMDb (62 vs. 65), the fact that noise-based perturbations disrupt contextual and semantic information leads to a significant decrease in the model's ability to discern the logical connections between the parts of the input. This results in a more pronounced performance degradation compared to the long-sequenced IMDb on the TEST case. In contrast, training the model on the noisy data (the ALL setup) proves effective in overcoming this effect, leading to improved results for T5-small.

In Table 2, we present an example of the outcome of applying STENCIL, STENCIL$_p$, and the NOISE mapper on a random phrase from the SST2 dataset. The NOISE mapper with a value of $\eta = 150$ introduces negligible noise, thus producing a similar sequence to the original one. The STENCIL-based techniques also produce a similar sequence, although STENCIL$_p$ swaps the positions of some tokens as a direct result of excluding the target token from the obfuscation process.

3.3 Nearest-neighbor Reconstruction

An attacker can potentially exploit the fact that these techniques utilize contextualized tokens and the selection of the nearest token as the quasi-embedding vector (Qu et al., 2021). Specifically, given the new perturbed token t', the attacker can obtain the embedding vector representation $\mathbf{E}[t']$. Afterward, the attacker can calculate the cosine similarity between $\mathbf{E}[t']$ and the other embedding vector representations ($\mathbf{E}[t]$ where $t \in \mathcal{V} \setminus \{t'\}$) and statistically determine the original token.

Additionally, since STENCIL incorporates information from its neighboring tokens, the new perturbed token t'_k might resemble the original token of a neighboring token, for instance, t_{k+1}. This allows the attacker to attempt to determine the original token by analyzing and comparing the neighbors' most similar tokens.

To test the resilience of these techniques against token inversion attacks, we implement the described attacker and report whether the original token was found to be one of the nearest five (Pr@5), or its neighbor's nearest five.

The success rate of the attacker for the four techniques is presented in Table 4. While the minor alterations in the original tokens contributed to performance improvement in the NOISE mapper, it is found to be highly vulnerable to simple reconstruction attacks. Taking into account both accuracy and resilience against reconstruction attacks, the STENCIL method demonstrates the best results, with a marginal trade-off in performance. The STENCIL_p demonstrates the best privacy protection against the attacker, highlighting the effectiveness of excluding the target token from the computation of the new token.

4 Conclusion

In this paper, we propose several token manipulation methods to preserve privacy under the assumption that the model parameters are inaccessible. We first introduce four simple mappers that offer distinct advantages compared to existing privacy-preserving techniques. Notably, these mappers operate independently of the LLM and the specific downstream task, resulting in a high degree of versatility. Additionally, their computational complexity is relatively low, making them efficient choices for privacy preservation on local, low-resource devices. However, it is essential to acknowledge that these mappers harm the performance of the downstream tasks and can be easily reconstructed by a knowledgeable attacker.

The second mapper class we propose is based on utilizing contextualized information to maintain performance while obfuscating the original input text. This technique achieves higher privacy measures and has less impact on the downstream task, which makes it more applicable for cases where the downstream task is important. Nevertheless, opting for different weighted functions, such as ones based on a trained model, can further help improve both accuracy and privacy.

An inherent problem with existing privacy-preserving techniques is their inability to maintain linguistic properties such as grammar and readability (as seen in Table 2) that are crucial for the performance of the model. Therefore, an additional avenue we plan to explore is application of these and similar rules in differential privacy techniques. For instance, following the application of random perturbations to an embedding vector, instead of simply returning the nearest token to the perturbed vector, one could consider returning a token with similar syntactic attributes, such as part of speech, or verbs with similar causative meanings or stable subcategorization frames.

Lastly, our experiments were limited to classification tasks in the English language. In future research, we intend to explore the effectiveness of these methods in generative tasks, across languages, and in multilingual settings.

Limitations

We demonstrated the privacy achieved by our methods empirically under one attacking scenario. Further comprehensive testing or mathematical proofs would enhance our understanding of the extent of privacy achieved.

An additional limitation of our proposed mechanism is the unchanged sentence length. This imposes a privacy breach in which an author who prefers writing longer or shorter sentences can be re-identified even when introducing random perturbations. Hence, another avenue in this research is reducing the amount of tokens by introducing, for example, a stride parameter to the STENCIL family of mappers. This parameter will determine how often tokens will be output, thus reducing the amount of tokens.

References

Pathum Chamikara Mahawaga Arachchige, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman. 2019. Local differential privacy for deep learning. *IEEE Internet of Things Journal*, 7(7):5827–5842.

Iyadh Ben Cheikh Larbi, Aljoscha Burchardt, and Roland Roller. 2023. Clinical text anonymization, its influence on downstream NLP tasks and the risk of re-identification. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 105–111, Dubrovnik, Croatia. Association for Computational Linguistics.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar

Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Sai Chen, Fengran Mo, Yanhao Wang, Cen Chen, Jian-Yun Nie, Chengyu Wang, and Jamie Cui. 2023. A customized text sanitization mechanism with differential privacy. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5747–5758, Toronto, Canada. Association for Computational Linguistics.

Maximin Coavoux, Shashi Narayan, and Shay B Cohen. 2018. Privacy-preserving neural representations of text. *arXiv preprint arXiv:1808.09408*.

Ido Cohn, Itay Laish, Genady Beryozkin, Gang Li, Izhak Shafran, Idan Szpektor, Tzvika Hartman, Avinatan Hassidim, and Yossi Matias. 2019. Audio de-identification - a new entity recognition task. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 197–204, Minneapolis, Minnesota. Association for Computational Linguistics.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850, Beijing, China. Association for Computational Linguistics.

Jacob Eisenstein. 2019. *Introduction to natural language processing*. MIT press.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy-and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th international conference on web search and data mining*, pages 178–186.

Wikimedia Foundation. 2023. Wikimedia downloads.

Max Friedrich, Arne Köhn, Gregor Wiedemann, and Chris Biemann. 2019. Adversarial learning of privacy-preserving text representations for de-identification of medical records. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5829–5839, Florence, Italy. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Kai Kugler, Simon Münker, Johannes Höhmann, and Achim Rettinger. 2021. Invbert: Reconstructing text from contextualized word embeddings by inverting the bert pipeline. *arXiv preprint arXiv:2109.10104*.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A systematic investigation of commonsense knowledge in large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11838–11855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yang Li, Sandro Schulze, and Gunter Saake. 2017. Reverse engineering variability from natural language documents: A systematic literature review. In *Proceedings of the 21st International Systems and Software Product Line Conference-Volume A*, pages 133–142.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30, Melbourne, Australia. Association for Computational Linguistics.

Pierre Lison, Ildikó Pilán, David Sanchez, Montserrat Batet, and Lilja Øvrelid. 2021. Anonymisation models for text data: State of the art, challenges and future directions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4188–4203, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. 2017. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics*, 75:S34–S42.

Lingjuan Lyu, Xuanli He, and Yitong Li. 2020. Differentially private representation for NLP: Formal guarantee and an empirical study on privacy and fairness. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2355–2365, Online. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the*

Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40.

Ahmadreza Mosallanezhad, Ghazaleh Beigi, and Huan Liu. 2019. Deep reinforcement learning-based text anonymization against private-attribute inference. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2360–2369, Hong Kong, China. Association for Computational Linguistics.

OpenAI. 2021. Chatgpt. OpenAI Website. Accessed on 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Richard Plant, Dimitra Gkatzia, and Valerio Giuffrida. 2021. CAPE: Context-aware private embeddings for private language learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7970–7978, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hanieh Poostchi, Ehsan Zare Borzeshi, and Massimo Piccardi. 2018. BiLSTM-CRF for Persian named-entity recognition ArmanPersoNERCorpus: the first entity-annotated Persian dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Natural language understanding with privacy-preserving bert. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1488–1497.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

William Frederick Spotz. 1995. *High-order compact finite difference schemes for computational mechanics*. The University of Texas at Austin.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of ICLR*.

Xin Zhou, Jinzhu Lu, Tao Gui, Ruotian Ma, Zichu Fei, Yuran Wang, Yong Ding, Yibo Cheung, Qi Zhang, and Xuanjing Huang. 2022. TextFusion: Privacy-preserving pre-trained model inference via token fusion. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8360–8371, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Yuran Wang, Yong Ding, Yibo Zhang, Qi Zhang, and Xuanjing Huang. 2023. TextObfuscator: Making pre-trained language model a privacy protector via obfuscating word representations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5459–5473, Toronto, Canada. Association for Computational Linguistics.