An Inversion Attack Against Obfuscated Embedding Matrix in Language Model Inference

Yu Lin, Qizhi Zhang, Quanwei Cai
Jue Hong, Ye Wu, Huiqi Liu, Bing Duan
Bytedance
{linyu.09, zhangqizhi.zqz, caiquanwei, tanzhuo.107
wuye.2020, liuhuiqi.7, duanbing.0}@bytedance.com

Abstract

With the rapidly-growing deployment of large language model (LLM) inference services, privacy concerns have arisen regarding to the user input data. Recent studies are exploring transforming user inputs to obfuscated embedded vectors, so that the data will not be eavesdropped by service provides. However, in this paper we show that again, without a solid and deliberate security design and analysis, such embedded vector obfuscation failed to protect users' privacy. We demonstrate the conclusion via conducting a novel inversion attack called Element-wise Differential Nearest Neighbor (EDNN) on the glide-reflection proposed in (Mishra et al., 2024), and the result showed that the original user input text can be 100% recovered from the obfuscated embedded vectors. We further analyze security requirements on embedding obfuscation and present several remedies to our proposed attack.

1 Introduction

Inference services of language models are now gaining popularity, with a considerable number of language models having been deployed on the cloud server. However, users might concern about the privacy of their data when requesting inference service, that is, their data would be eavesdropped by malicious service providers.To address this problem, recent research has turned to adopting obfuscation techniques on the embedding matrix, ensuring that user inputs cannot be recovered from the obfuscated embeddings by service providers. Embedding obfuscation becomes appealing since the obfuscated embeddings can be directly forwarded to inference process as efficient as plaintext embeddings, leading to practical potential for real applications compared with secure multi-party computation (MPC) and homomorphic encryption (HE). For example, the state-of-the-art work in (Mishra et al., 2024) leverages glide-reflection for embedding obfuscation combined with the user-side key-based hashing, to claim a private and secure inference solution.

Nonetheless, recent studies show that a malicious server can indeed reconstruct user data through embedding inversion attacks (EIA) (Qu et al., 2021; Kugler et al., 2021). Consequently, without formal security analysis, concerns persist regarding the potential existence of novel EIAs capable of extracting user information from these embedding obfuscation methods. In this paper, we analyze the security of the glide-reflection methodology used in (Mishra et al., 2024), ultimately uncovering its vulnerability. We innovatively design an Element-wise Differential Nearest Neighbor (EDNN) attack to effectively break the security of the glide-reflection. Our experimental outcomes conclusively demonstrate that the EDNN entirely recovers 100% of the user data tokens which ostensibly secured by the glide-reflection. Subsequently, we present an insight on why the naive linear-transformation based obfuscation, like glide-reflection, fails to safeguard user data. We further discuss the security requirements of embedding obfuscation and demonstrate that the deliberate security design is necessary. We also introduce several possible defenses against EIA base on our analysis.

2 Obfuscation schema based on glide-reflection

In this section, we describe the system and threat model of obfuscation schema in (Mishra et al., 2024). Then we give the formal description of the schema and explain its vulnerability.

2.1 System and threat model

The system model is composed of two entities. A *data user* desires to request inference service using its private data. A *model server* is deployed with a fine-tuned language model to offer inference

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 2100–2104

service.

The schema adopts a typical threat model for secure language model inference as used in recent works (Zhou et al., 2022, 2023). The model server might be compromised and act as an adversary, aiming to obtain the user's private data. It possesses the knowledge of the pretrained model and is capable of executing embedding inversion attacks, as demonstrated in (Kugler et al., 2021; Qu et al., 2021), on the embedding matrix of the fine-tuned model. Consequently, this enables the direct recovery of the plaintext tokens and the reconstruction of the user's sensitive data.

2.2 Description of schema

In the schema, the user will apply key-based encryption on tokens of vocabulary and utilize glide-reflection to obfuscate embedding matrix of pretrained model. The encrypted tokens and obfuscated embeddings will be shuffled simultaneously so as the server cannot recognize user data according to token orders in inference process.

In the token encryption step, every token v in the vocabulary $\mathcal{V}$ of pretrained model will be encrypted by Blake (Aumasson et al., 2013) with a user specific key to generate an encrypted vocabulary.

In the embedding obfuscation, suppose that the embedding matrix $E_{d\times M}$ contains M embedding vectors $\{\boldsymbol{e}_i | i \leq M\}, \boldsymbol{e}_i \in \mathbb{R}^d$ where d is the dimension of embeddings. The glide-reflection applied on $\boldsymbol{e}_i$ can be formalized as:

$$\boldsymbol{e}'_i = \boldsymbol{e}_i - 2 \cdot \frac{\boldsymbol{e}_i \cdot \boldsymbol{l}_i}{\boldsymbol{l}_i \cdot \boldsymbol{l}_i} \cdot \boldsymbol{l}_i + \boldsymbol{t}_i, \tag{1}$$

where $\boldsymbol{l}_i = \vec{1} \cdot a_i$ and $\boldsymbol{t}_i = \vec{1} \cdot b_i$ are two vectors constructed by two random values a_i, b_i uniformly sampled from $[0, 1)$[1].

During runtime usage, the user is able to tokenize its input data and request inference service with encrypted tokens. Then the server can use the encrypted vocabulary and the obfuscated embedding matrix to complete inference process.

2.3 Vulnerability

The security of the schema lies on the glide-reflection, while we discover that it cannot hide the differential information within each embedding vector. Specifically, recall that in the equation 2, each element of $\boldsymbol{l}_i$ has the identical value, as well as $\boldsymbol{t}_i$, leading to $\boldsymbol{e}'_i[k_1] - \boldsymbol{e}'_i[k_2] = \boldsymbol{e}_i[k_1] - \boldsymbol{e}_i[k_2]$ for any $k_1, k_2 \in [1, d]$. Therefore, we can construct an attack to draw the relationship between the obfuscated embeddings and original embeddings by checking element-wise difference of embeddings.

[1]This can be found in the implementation of the schema: https://github.com/abhijitmishra/sentinellm-aaai2024. We also test the case that each element of $\boldsymbol{l}_i, \boldsymbol{t}_i$ is random, but the model accuracy will decrease to nearly 50%.

3 Proposed attack: EDNN

The authors of (Mishra et al., 2024) have evaluated the security of glide-reflection against the nearest neighbor (NN) attack, and the accuracy of token recovery for which turns out to be negligible. By extending NN, we propose an efficient inversion attack called Element-wise Differential Nearest Neighbor (EDNN) to break the glide-reflection. The EDNN selects the closest token from pretrained embeddings as the real token by utilizing the difference of vector elements for neighbor retrieval. Therefore, it is effect on the glide-reflection which does not change the element differences within embedding vector.

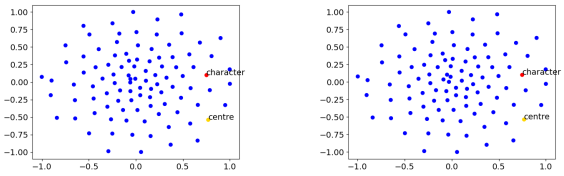
Algorithm 1 EDNN

Input: A obfuscated and fine-tuned embedding matrix $\widetilde{E}_{d\times M} = \{\widetilde{\boldsymbol{e}}_i\}_{i\leq M}$, a pretrained embedding matrix $E_{d\times M} = \{\boldsymbol{e}_i\}_{i\leq M}$, a pretrained vocabulary $\mathcal{V}$

Output: A recovered vocabulary $\mathcal{V}_R$

Initialize distance matrix $D_{M\times M} = \{0\}_{M\times M}$ and output vocabulary $\mathcal{V}_R = \{\}$.
for $i, j \leftarrow 1$ to M **do**
    $D[i][j] = \|(\widetilde{\boldsymbol{e}}_i - lshift(\widetilde{\boldsymbol{e}}_i)) - (\boldsymbol{e}_j - lshift(\boldsymbol{e}_j))\|$
for $i \leftarrow 1$ to M **do**
    $k = argmin(D[i])$
    $\mathcal{V}_R \leftarrow \mathcal{V}_R \bigcup \{\mathcal{V}[k]\}$.
return $\mathcal{V}_R$.

We present the details of EDNN in Algorithm 1, where $\widetilde{E}_{d\times M}$ is the obfuscated embedding matrix after fine-tuning. The algorithm will output a vocabulary $\mathcal{V}_R$ which stores the recovered tokens corresponded to the obfuscated embeddings $\widetilde{E}_{d\times M}$. To compute the element difference inside embedding vector, the algorithm use $lshft(\cdot)$ function to cyclically shift the vector to the left by one position and calculate element-wise subtraction. The algorithm will evaluate the distance between every pairs of plaintext and encrypted tokens. Then for each encrypted token, the algorithm is able to output the plaintext token with the minimum distance



(a) (b)

Figure 1: 2D plot of 100 embeddings from: (a) element-wise difference of original pretrained model, (b) element-wise difference of transformed model after 10 iterations of glide reflection.

in the embedding space as its substitute.

To explain the attack effect of EDNN, we fit the embeddings of 100 tokens into 2D plot by T-distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten and Hinton, 2008) and scale them into $(-1, 1)$ as shown in Figure 1. By comparing the results of Figure 2 and 1b, we can observe that the element-wise differences inner each embedding vector from the original model and the transformed model are the same.

4 Experiment

Experimental details. We encrypted the model according to (Mishra et al., 2024) and fine-tune the model with specific task. Then we evaluate EDNN on the fine-tuned model to recover its encrypted tokens.

Datasets and models. We use the same setting as (Mishra et al., 2024) and conduct experiments on datasets including the General Language Understanding Evaluation (GLUE) benchmark dataset (Wang et al., 2018), the CoNLL2003 Named Entity Recognition Dataset (Sang and De Meulder, 2003) and the XNLI dataset (Conneau et al., 2018) . We use BERT, RoBERTa, and mBERT models from Huggingface[2].

Element-wise differential comparison. For each embedding obfuscated by glide-reflection, we first evaluate the distance of element-wise differential to its corresponding original embedding and other nearest embedding. The results in Fig. 2 shows that after fine-tuning, the element-wise differentials between each embedding and irrelevant embeddings exhibit a three-order-of-magnitude discrepancy compared to its original embedding, facilitating the EDNN to capture the correspondence between obfuscated embeddings and their original counterparts.

[2]https://huggingface.co

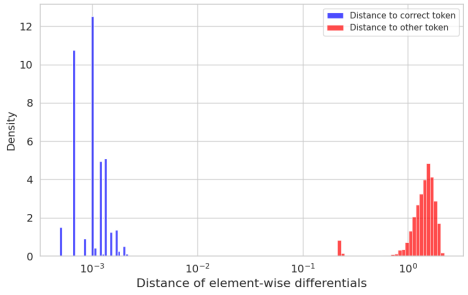


Figure 2: Distance of element-wise differentials under GLUE SST2 dataset. The figure records the density of distance between every encrypted token and its corresponding plain token (blue) or other nearest token (red).

Table 1: Token recovery accuracy of obfuscated model after 10 times glide-reflection

Model	Task	Recovery acc
BERT	GLUE all tasks	100%
RoBERTa	GLUE all tasks	100%
BERT	CoNLL2003 NER	100%
RoBERTa	CoNLL2003 NER	100%
BERTMultilingual	XNLI In-language	100%
BERTMultilingual	XNLI Zero-shot	100%

Attack accuracy. In Table 1, we test the token recovery accuracy of EDNN under different iterations of glide-reflection marked by $nglide$. The results show that EDNN is able to recover all obfuscated tokens even after fine-tuning the model on GLUE, CONLL2003 and XNLI datasets.

5 Analysis and possible defenses

In this section, we analyze the security requirements for the embedding obfuscation and propose security requirements for embedding obfuscation.

5.1 Security analysis

In embedding obfuscation, a secret transformation will be performed on the embedding matrix $E_{d\times M} = \{\boldsymbol{e}_i \in \mathbb{R}^{(d)} | i \leq M\}$ of the pretrained model. In order to maintain the model accuracy, a linear transformation is usually used to ensure that the model can still adapt to the transformed embeddings through fine-tuning. We can formalize the transformation by:

$$\begin{array}{cccc} \phi: & \mathbb{R}^{(n)} & \longrightarrow & \mathbb{R}^{(n)} \\ & \alpha & \mapsto & W\alpha + \vec{b} \end{array} \tag{2}$$

where $W \in M_{d,d}(\mathbb{R})$ and $b \in \mathbb{R}^{(n)}$ are random matrix and vector from some secret distribution. For example, for the transformation in the equation 2, we have $W = I_{d,d} - \frac{2}{d}E_{d,d}$ and $\vec{b} = b\vec{1}$, where $I_{d,d}$

is the diagonal matrix with all diagonal elements 1, $E_{d,d}$ is the matrix with all elements 1, b is a random number in $[0, 1)$.

The server cannot directly obtain α from $\phi(\alpha)$ since it is not given W, b. Nevertheless, considering that the server is aware of the embedding matrix of the pre-trained model, as described in the threat model, it can carry out EIA if the transformed matrix fails to adequately obfuscate the information related to the original matrix. Subsequently, we propose the following two security requirements for the transformation.

Fixed-point nonexistence. There should not exist a probabilistic polynomial time (PPT) adversary who is able to get an invariant of the transformer ϕ, even without the total knowledge of W and b.

Suppose there is a linear invariant map

$$\begin{array}{rccc} \psi: & \mathbb{R}^{(d)} & \longrightarrow & \mathbb{R}^{(f)} \\ & \alpha & \mapsto & A\alpha \end{array} \tag{3}$$

where $A \in M_{f,d}(\mathbb{R})$. Then we have$\psi \circ \phi = \phi$. It induce a linear system about A such that

$$\begin{cases} AW - A = 0 \\ A\vec{b} = 0. \end{cases} \tag{4}$$

The linear system should not have trivial solution. Otherwise, the adversary is able to decide whether an obfuscated embedding $\boldsymbol{e}'$ and an original embedding $\boldsymbol{e}$ are related by checking whether $\psi(\boldsymbol{e}) = \psi(\boldsymbol{e}')$ holds even if shuffling is performed on the obfuscated embedding matrix.

The glide-reflection is unable to securely obfuscate embeddings, as it fails to satisfy the necessary security requirement. Recall that the vulnerability of glide-reflection lies in the fact that it does not change the difference between any two elements within each embedding vector. Without knowing the specific $\boldsymbol{l}_i$ and $\boldsymbol{t}_i$ used in the equation 2, the adversary can still construct the following matrix $A_{d\times d} =$ to meet with the equation 4: $A = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -1 & 0 & 0 & \cdots & 1 \end{pmatrix}$.

(k, ϵ)-anonymity. While the above requirement prevent accurate matching between the transformed and original embeddings, the adversary can still guess the plaintext token of an transformed embedding according to the distance between obfuscated and original embeddings. Therefore, it is necessary to use sufficient noise to obfuscate embeddings.

Compared with traditional differential privacy, (k, ϵ)-anonymity proposed in (Holohan et al., 2017) might be more suitable for embedding obfuscation by combining the k-anonymity and ϵ-differential privacy. Rather than directly applying noise on every embedding with differential privacy (Du et al., 2023; Yue et al., 2021), (k, ϵ)-anonymity only requires embeddings to be indistinguishable within a subset of size k. Therefore, it can keep semantic information of the obfuscated embeddings as much as possible while ensuring the security of obfuscation.

Formally, if a transformation $\mathcal{P}(\cdot)$ satisfies (k, ϵ)-anonymity on embedding matrix E, there should exist a subset $F_i \subset E, |F_i| \geq k$ for any $\boldsymbol{e}_i, \boldsymbol{e}_j \in F_i$ and any subset S of the outputs of $\mathcal{P}$ such that

$$Pr[\mathcal{P}(\boldsymbol{e}_i) \in S] \leq e^{\epsilon} Pr[\mathcal{P}(\boldsymbol{e}_j) \in S], \tag{5}$$

where $\epsilon \geq 0$ is the privacy parameter. The equation 5 represents each embedding should be indistinguishable within a subset of size k. This indistinguishability discretely holds in each subset while keeping sufficient distances between different subsets.

5.2 Defenses

To mitigate such security risks inherent in the paradigm, there exist two potential defense strategies. One approach involves the application of differential privacy, wherein random noise is added to the embeddings (Yue et al., 2021; Du et al., 2023; Shen et al., 2023). However, ensuring both privacy and model accuracy concurrently poses a significant challenge to design elaborate noise mechanisms.

Alternatively, leveraging cryptographic tools such as homomorphic encryption (Cheon et al., 2017) offers another avenue of defense. In this method, the embedding matrix is encrypted using homomorphic encryption techniques. To minimize computational overhead, the server can request decryption after processing several layers, allowing subsequent layers to be processed in plaintext.

6 Conclusion

In this paper, we investigate the vulnerability of the glide-reflection used for embedding obfuscation. We devise an innovate embedding inversion attack to break the security of the glide-reflection. Furthermore, we conduct a comprehensive analysis and introduce two essential security requirements for embedding obfuscation. We explore various techniques that can be leveraged to enhance the security of embedding obfuscation.

Limitations

We have summarized two limitations of this paper. (1) The EDNN attack method proposed in the paper is to illustrate that the embedding obfuscation scheme based on glide-reflection is not secure, but we have not tested the effectiveness of the EDNN attack against other embedding obfuscation schemes. (2) We present two security requirements for embedding obfuscation and we believe they are necessary for protecting user data. However, we have not proposed a concrete scheme to verify the sufficiency of the aforementioned security requirements.

References

Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, and Christian Winnerlein. 2013. Blake2: simpler, smaller, fast as md5. In *Applied Cryptography and Network Security: 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings 11*, pages 119–135. Springer.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. 2017. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*, pages 409–437. Springer.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

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.

Naoise Holohan, Spiros Antonatos, Stefano Braghin, and Pol Mac Aonghusa. 2017. (k, ϵ)-anonymity: k-anonymity with ϵ-differential privacy. *CoRR*, abs/1710.01615.

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*.

Abhijit Mishra, Mingda Li, and Soham Deo. 2024. Sentinellms: Encrypted input adaptation and fine-tuning of language models for private and secure inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21403–21411.

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.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Xicong Shen, Yang Liu, Huiqi Liu, Jue Hong, Bing Duan, Zirui Huang, Yunlong Mao, Ye Wu, and Di Wu. 2023. A split-and-privatize framework for large language model fine-tuning. *CoRR*, abs/2312.15603.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. 2021. Differential privacy for text analytics via natural text sanitization. *arXiv preprint arXiv:2106.01221*.

Xin Zhou, Jinzhu Lu, Tao Gui, Ruotian Ma, Zichu Fei, Yuran Wang, Yong Ding, Yibo Cheung, Qi Zhang, and Xuan-Jing 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.

Xin Zhou, Yi Lu, Ruotian Ma, Tao Gui, Qi Zhang, and Xuan-Jing Huang. 2023. Textmixer: Mixing multiple inputs for privacy-preserving inference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3749–3762.