Transferable Embedding Inversion Attack: Uncovering Privacy Risks in Text Embeddings without Model Queries

Yu-Hsiang Huang
National Taiwan University
r11922053@csie.ntu.edu.tw

Yu-Che Tsai
National Taiwan University
f09922081@csie.ntu.edu.tw

Hsiang Hsiao
National Taiwan University
r12946003@ntu.edu.tw

Hong-Yi Lin
National Taiwan University
b09902100@csie.ntu.edu.tw

Shou-De Lin
National Taiwan University
sdlin@csie.ntu.edu.tw

Abstract

This study investigates the privacy risks associated with text embeddings, focusing on the scenario where attackers cannot access the original embedding model. Contrary to previous research requiring direct model access, we explore a more realistic threat model by developing a transfer attack method. This approach uses a surrogate model to mimic the victim model's behavior, allowing the attacker to infer sensitive information from text embeddings without direct access. Our experiments across various embedding models and a clinical dataset demonstrate that our transfer attack significantly outperforms traditional methods, revealing the potential privacy vulnerabilities in embedding technologies and emphasizing the need for enhanced security measures.

1 Introduction

Text embeddings serve as universal representations of textual data, which can be utilized as features for various downstream tasks. Recent developments in text embedding models (Ni et al., 2022a; Reimers and Gurevych, 2019) have significantly streamlined the process of generating embeddings. Additionally, systems that employ large language models (LLMs) often incorporate a vector database of text embeddings to store and infuse domain specific knowledge or auxiliary data. Retrieval-augmented generation (RAG) (Lewis et al., 2020) is a typical example that enhances LLMs' knowledge by incorporating retrieved documents into the model's prompt. This has led to a growing adoption of vector database services like Chroma[1] and Faiss (Johnson et al., 2019), known for their efficient and scalable embedding searches. In these databases, only the text embeddings are shared with third-party services, not the actual text, leading these platforms to claim that storing embeddings is safe and encouraging the upload of private data.

Despite the bright sight of text embeddings and vectors, a natural question arises: Does sending text embedding to an online service really expose zero privacy risk given that the original text might contain sensitive information? To answer this question, researchers started to investigate the *embedding inversion attack*, which aims to reconstruct the input data from its embedding. In the image domain, prior works (Mahendran and Vedaldi, 2015; Dosovitskiy and Brox, 2016) on computer vision demonstrated that it is possible to reproduce the input image from their visual embeddings. In the text domain, the pioneering work (Song and Raghunathan, 2020) attempts to infer a bag of words from embeddings. Along similar lines, the following research (Morris et al., 2023) further reveals that an adversary can recover 92% of a 32-token text input given embeddings from a T5-based pre-trained transformer.

Although existing works (Song and Raghunathan, 2020; Morris et al., 2023; Li et al., 2023) have studied the privacy risks of text embeddings, the observed threats essentially rely on a strong assumption, which is that the adversary has query access to the embedding model. By querying the embedding model extensively, the adversary can either iteratively edit the input text such that the text is as close as possible to a given embedding or obtain a large amount of paired data to reverse-engineer the embedding model accordingly. Here, we argue that such privileged knowledge might not always be available in real-world scenarios. For instance, consider the data leakage of an online vector database where only a small number of documents and their associated text embeddings were exposed to the adversary. In that case, the adversary was passively offered a small number of query pairs, while querying the embedding model is not allowed. Inspired by this, this work particularly focuses on the privacy risk without assuming the accessibility of the original embedding model for

[1] https://docs.trychroma.com/

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4193–4205
August 11-16, 2024 ©2024 Association for Computational Linguistics

querying; instead, only a small portion of paired document-embedding data is available.

We consider the problem of a black-box attack, where the target victim model is entirely hidden from the attacker. In this setting, standard white-box attacks (Kugler et al., 2021) or even query-based black-box attacks (Li et al., 2023; Morris et al., 2023) become ineffective. As the victim model becomes invisible to the adversary, we present an alternative solution to attack the victim model through a transfer attack. The transferability property of an attack is satisfied when an attack developed for a particular machine learning model (i.e., a surrogate model) is also effective against the target model. Specifically, our transfer attack aims to achieve two goals: 1) **Encoder stealing** attempts to learn a *surrogate model* to steal the victim model only through their returned representations. If the surrogate model successfully replicates the victim model, the adversary gains query access to some extent. 2) **Threat model transferability** enables the adversary to build a threat model by attacking the surrogate model and hopes the threat model can also successfully fool the victim black-box model.[2]

To achieve the first goal, an off-the-shelf text embedding model (e.g., GTR-T5 (Ni et al., 2022b)) followed by a MLP-based adapter is used as the surrogate model. The surrogate model is then optimized with our proposed consistency regularization loss to mimic the behavior of the victim model. To achieve the second goal, we use the adversarial training to mitigate the embedding discrepancy between the surrogate and victim models and thus improve the attack transferability.

To validate the effectiveness of our attack, we perform extensive experiments on 3 popular embedding models, including Sentence-BERT (Reimers and Gurevych, 2019), Sentence-T5 (Ni et al., 2022a), and OpenAI text embedding. Experimental results show that the transfer attack can be 40%-50% more effective than the standard attack approach. The key factors for stealing the victim model are discussed in Sec. 6. To study the privacy risk on a specific threat domain, we conduct a case study on the MIMIC-III clinical note dataset. Results demonstrated in Sec. 7 show that our transfer attack can identify sensitive attributes (e.g., age, sex, disease) with 80%-99% accuracy.

2 Preliminary

2.1 Embedding inversion attack

Given a sequence of text tokens $x \in V^n$, the text encoder $\phi : V^n \rightarrow \mathbb{R}^d$ will map the text x into a fixed-length vector $\phi(x) \in \mathbb{R}^d$ which is the text embedding. An embedding inversion attack is a specific type of embedding attack that aims to reconstruct the original text x from its text embedding $\phi(x)$. Specifically, the attacker seeks to find a function f to approximate the inversion function of ϕ as:

$$f(\phi(x)) \approx \phi^{-1}(\phi(x)) = x. \tag{1}$$

According to the attack target, the embedding inversion attack can be categorized into: (i) token-level inversion (Pan et al., 2020; Song and Raghunathan, 2020) and (ii) sentence-level inversion (Li et al., 2023; Morris et al., 2023). As inverting the whole sentence could potentially reveal more privacy risks, we focus on recovering the whole sentence from its text embedding in this work.

Base attack model. To reconstruct the original text sequence $x = w_0 w_1 ... w_u$ from its corresponding text embedding $\phi(x)$, a recent work (Li et al., 2023) proposed the attack model as a generative task. This involves minimizing the standard language model loss with teacher forcing (Williams and Zipser, 1989). This loss function is defined as:

$$\mathcal{L}_{LM} = -\sum_{i=1}^{u} \log(Pr(w_i|\phi(x), w_0, \ldots, w_{i-1}) \tag{2}$$

2.2 Transferable embedding inversion attack

Motivation. In practical scenarios, attackers might access text embeddings without the ability to query the generating model directly. For instance, a data breach might expose embeddings from a health chatbot containing encoded patient information or from a job recommendation platform with details on resumes and job listings. These situations demonstrate the risk of sensitive data exposure even when direct interaction with the embedding model is not possible and motivate our research into developing methods to study privacy under such constrained conditions.

Attacker's goals: The attacker aims to achieve the following two goals:

- ***Goal 1 (Stealing Text Encoder)***: The attacker seeks to find a surrogate encoder $\hat{\phi}$ to steal the

[2] Code is publicly available at `https://github.com/coffree0123/TEIA`

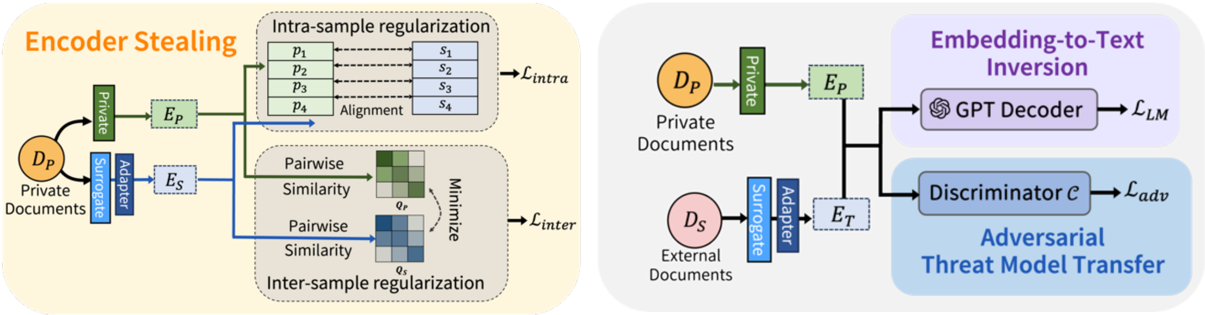

Figure 1: Model architecture of the transferable embedding inversion attack.

anonymous embedding model ϕ. In particular, we expect an optimal surrogate model that satisfies:

$$\hat{\phi}(x) \approx \phi(x);\ \forall x \in \mathcal{X}, \tag{3}$$

where $\hat{\phi}$ gives the similar output embedding as ϕ and $\mathcal{X}$ denotes the domain of an input text x.

- ***Goal 2 (Threat Model Transferability)***: Given the surrogate model $\hat{\phi}$, the attacker constructs the surrogate dataset $D_S = \{(x, \hat{\phi}(x))\}$ which consists of pairs of documents and their text embeddings. The threat model $\mathcal{T} : \hat{\phi}(x) \rightarrow x$ is then built to attack $\hat{\phi}$ using D_S. Finally, $\mathcal{T}$ is used to perform a transfer attack on ϕ.

Attacker's background knowledge. To clarify the scope of the attacker's background knowledge, we make the following assumptions:

- ***Assumption 1 (Anonymous Embedding Model)***: Our attack follows the realm of black-box attack, where the attacker is not aware of the model weights or the architecture of ϕ. Unlike prior works that presume query access to ϕ, we further eliminate such knowledge and make ϕ completely hidden from the attacker.

- ***Assumption 2 (Leaked Dataset)***: We assume a dataset $D_L = \{(x, \phi(x))\}$ is exposed to the attacker due to potential data leakage from an online vector database or embedding services. In a practical sense, we consider D_L to be a small dataset.

3 Methodology

As illustrated in Figure 1, our transfer attack pipeline consists of three major components. First, the encoder stealing aims to learn a surrogate model to mimic the behavior of the victim model ϕ. This goal is achieved by optimizing the surrogate model with our intra- and inter-consistency regularization losses. Second, we adopt adversarial training to make the surrogate embedding indistinguishable from the private embedding and improve attack transferability. Finally, embedding-to-text leverages a GPT-based decoder to invert embeddings to their original text sequence.

3.1 Encoder Stealing with a Surrogate Model

The surrogate model. The primary objective of the surrogate model $\hat{\phi}$ is to steal the black-box embedding model ϕ through the leaked dataset D_L. The surrogate model consists of two components: a surrogate encoder and an adapter. The surrogate encoder is a pre-trained text embedding model used to generate the initial embedding of input text x. A simple linear transformation is used as the adapter to convert the initial embedding such that the resulting representation could be aligned with $\phi(x)$. Adding an adapter behind the surrogate encoder has two advantages: (1) We do not need to fine-tune the surrogate encoder during training; only the adapter's model weight needs to be adjusted. (2) The adapter can solve the issue of the output dimension of the surrogate encoder being inconsistent with $\phi(x)$.

Optimizing the surrogate model with consistency regularization. To achieve Goal 1, we design two types of regularization terms to enforce $\hat{\phi}$ acts similarly to ϕ. Given a batch of N samples from D_L, we have $\mathbf{E}_P = \phi(x)$ and $\mathbf{E}_S = \hat{\phi}(x)$, where $\mathbf{E}_P \in \mathbb{R}^{N \times d}$ and $\mathbf{E}_S \in \mathbb{R}^{N \times d}$ denote the private and surrogate embedding matrix, respectively. Inspired by the concept of stealing image encoder (Liu et al., 2022), the intra-consistency regularization aims to minimize the distance between the $\mathbf{E}_P$ and $\mathbf{E}_S$, which is described using the following loss:

$$\mathcal{L}_{intra}(\mathbf{E}_P, \mathbf{E}_S) = MSE(\mathbf{E}_P, \mathbf{E}_S). \tag{4}$$

Here, we use the mean squared error to measure the distance between two matrices. $\mathcal{L}_{intra}$ is small if $\hat{\phi}$ and ϕ produce similar feature vectors for an input.

However, simply optimizing $\mathcal{L}_{intra}$ only considers point-wise information and ignores pairwise semantic information between documents. Therefore, we designed an additional inter-consistency regularization term to enable our surrogate model to simultaneously preserve the relative semantic relationship between documents. Specifically, we first calculate the in-batch pairwise cosine similarity matrix $\mathbf{Q}_P \in \mathbb{R}^{N \times N}$ from the private embedding $\mathbf{E}_P$ as:

$$\mathbf{Q}_P = \tilde{\mathbf{Q}}_P \tilde{\mathbf{Q}}_P^\top;\ \tilde{\mathbf{Q}}_{P[i,:]} = \mathbf{E}_{P[i,:]} / \|\mathbf{E}_{P[i,:]}\|_2. \quad (5)$$

Similarly, the pairwise similarity $\mathbf{Q}_S$ could be obtained from $\mathbf{E}_S$ using E.q. 5. Finally, we define the similarity-preserving regularization loss as:

$$\mathcal{L}_{inter}(\mathbf{Q}_P, \mathbf{Q}_S) = \frac{1}{N^2} \|\mathbf{Q}_P - \mathbf{Q}_S\|_F^2, \quad (6)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix. We let $\mathcal{L}_{surrogate} = \mathcal{L}_{intra} + \mathcal{L}_{inter}$ be the objective loss function for stealing the text encoder with $\hat{\phi}$.

3.2 Adversarial Threat Model Transferability

To achieve Goal 2, the attacker leverages an external corpus and utilizes the surrogate model to create the surrogate dataset $D_S = \{(x, \hat{\phi}(x)\}$. As the GPT decoder is primarily trained on D_S using the surrogate-generated embeddings, a key obstacle here is the difference in representation between the surrogate and private embeddings, which can hinder the effectiveness of the attack when applied to the latter. To overcome this, we employ adversarial training techniques (Ganin et al., 2016). This method involves training a discriminator $\mathcal{C}$ to distinguish between the surrogate embedding $\mathbf{E}_T$ and private embedding $\mathbf{E}_P$ while simultaneously optimizing $\hat{\phi}$ to generate embeddings that the discriminator cannot differentiate. Note that here we denote $\mathbf{E}_T$ as the surrogate embeddings generated from external documents. Formally, the adversarial training is described as:

$$\begin{aligned} \mathcal{L}_{adv} = \min_{\hat{\phi}} \max_{\mathcal{C}}\ & \mathbb{E}_{e_p \sim \mathbf{E}_P}[\log \mathcal{C}(e_p)] + \\ & \mathbb{E}_{e_t \sim \mathbf{E}_T}[\log(1 - \mathcal{C}(e_t))], \end{aligned} \quad (7)$$

where $\log \mathcal{C}(e_p)$ and $\log \mathcal{C}(e_t)$ represent the expected value of the logarithmic probability of the domain classifier $\mathcal{C}$. During the training phase, we utilize a sequential training strategy, alternating focus between the discriminator and the surrogate encoder.

3.3 Training Pipeline

In every training iteration, we sample a batch data from both D_L and D_S. The leaked dataset is used for encoder stealing and embedding-to-text training. Considering the leaked dataset D_L could be small, we also apply data augmentation to create more examples. The analysis of data augmentation is studied in Appendix D. On the other hand, the surrogate dataset D_S is used for adversarial and embedding-to-text training. Finally, we jointly optimize the surrogate model and the base attack model mentioned in Sec. 2.1 with the following objective function: $\mathcal{L}_{final} = \mathcal{L}_{LM} + \mathcal{L}_{surrogate} + \mathcal{L}_{adv}$.

4 Experiment Setup

Victim Embedding Models. To assess the embedding inversion attack, we utilize three victim models acting as our blackbox encoders: `text-embeddings-ada-002` from OpenAI, SBERT (Reimers and Gurevych, 2019), and ST5 (Ni et al., 2022a). These encoder models remain frozen, with their pre-trained weights employed to generate private embeddings from input text. All encoder models, except for OpenAI, are accessible via Hugging Face.

Datasets. Three datasets are used to evaluate the attack performance. Qnli (Alishahi et al., 2019) is structured around question-answer pairs and collected from Wikipedia articles. IMDB (Maas et al., 2011) comprises movie reviews. AG News (Zhang et al., 2015) includes a diverse collection of news articles. We randomly sample 8000 documents from each dataset to form the leaked dataset D_L. The statistics for these datasets are detailed in Appendix A.

Source Domain of External Dataset. Depending on the data domain of the external dataset, the attack scenario can be categorized into in-domain and out-of-domain text reconstruction. Under the in-domain (out-of-domain) attack setting, we assume the external dataset has the same (different) data domain as the leaked dataset. By default, we employ data from the same domain as the external dataset.

Competing Method. To compare the inversion performance, we employ a generative embedding

Table 1: Comparison of same domain embedding inversion performance between direct and transfer attack. The evaluation is done on QNLI, IMDB, and AGNEWS datasets with embedding models including OpenAI `text-embeddings-ada-002`, SBERT (Reimers and Gurevych, 2019) and ST5 (Ni et al., 2022a). Higher scores are better for all metrics except PPL.

Dataset / Method	OpenAI				SBERT				ST5			
	RougeL	PPL	Cos	LLM-Eval	RougeL	PPL	Cos	LLM-Eval	RougeL	PPL	Cos	LLM-Eval
QNLI												
Direct Attack	0.1433	40.822	0.2797	0.2984	0.1264	27.127	0.3257	0.3194	0.1463	42.911	0.2226	0.2755
Transfer Attack	**0.2226**	**10.242**	**0.4772**	**0.4402**	**0.1934**	**11.633**	**0.4886**	**0.4280**	**0.1985**	**11.808**	**0.4121**	**0.3963**
Improv. (%)	55.3%	74.9%	70.6%	47.5%	53.0%	57.1%	50.0%	34.0%	35.6%	72.4%	85.1%	43.8%
IMDB												
Direct Attack	0.1133	20.549	0.2692	0.3818	0.1137	34.805	0.2891	0.3923	0.1103	24.939	0.2678	0.3909
Transfer Attack	**0.1991**	**12.953**	**0.4297**	**0.4528**	**0.1689**	**14.505**	**0.4467**	**0.4475**	**0.1571**	**14.839**	**0.3866**	**0.4295**
Improv. (%)	75.7%	36.9%	59.6%	18.6%	48.5%	58.3%	54.5%	14.0%	42.4%	40.4%	44.3%	9.8%
AGNEWS												
Direct Attack	0.0612	66.383	0.1162	0.2979	0.0538	286.16	0.1317	0.2742	0.0578	74.085	0.0980	0.2905
Transfer Attack	**0.1271**	**31.159**	**0.4301**	**0.4057**	**0.1067**	**36.793**	**0.4110**	**0.3839**	**0.1042**	**40.809**	**0.3697**	**0.3706**
Improv. (%)	107.0%	53.0%	270%	36.1%	98.3%	87.1%	212.0%	40.0%	80.2%	44.9%	277.2%	27.5%

inversion attack approach (Li et al., 2023) that utilizes the leaked dataset D_L and trains the threat model by optimizing E.q. 2. Here, this method is referred to as "Direct Attack" to distinguish it from our transfer attack strategy. For a fair comparison, we use the same dialogGPT model (Zhang et al., 2020) as the decoder for both direct and transfer attacks.

Evaluation Metrics. We use the following four metrics to evaluate the text reconstruction attack performance. **RougeL** (Lin, 2004) is used to measure the accuracy and overlap between ground truth and reconstructed text based on n-grams. **Perplexity** (Baker, 1977) is used to evaluate the performance of language models by measuring how well they predict a given sequence of words. **Embedding similarity (Cos)**: To evaluate the semantic similarity in latent space, we utilize Sentence-BERT (Reimers and Gurevych, 2019) to compute the cosine similarity between the ground truth sentences' embedding and the embedding of the generated sentences. **LLM-Eval** (Lin and Chen, 2023): We use ChatGPT to provide a score ranging from 0 to 1 to evaluate the relevance between prediction and ground truth. More details can be found in Appendix F.

5 Attack Result

5.1 In-domain Text Reconstruction

Table 1 compares the attack performance between direct and transfer attacks on different datasets and victim embedding models. The result shows a significant improvement with more than 40% in both RougeL and embedding similarity scores when comparing transfer attack to direct attack. It is worth noting that the major difference between direct and transfer attacks is the additional surrogate dataset D_S to enhance the performance of the attack model. Therefore, the improved result indicates a successful transfer of the surrogate model. To better understand the effectiveness of the surrogate model and how well it steals, a detailed discussion can be found in Sec. 6.

Table 2: Comparison of out-of-domain embedding inversion performance between direct and transfer attack.

Dataset / Method	RougeL	PPL	Cos	LLM-Eval
QNLI				
Direct Attack	0.1264	27.127	0.3257	0.3194
Transfer Attack	**0.1800**	**20.515**	**0.4445**	**0.3899**
Improv. (%)	42.4%	24.3%	36.5%	22.1%
IMDB				
Direct Attack	0.1137	34.805	0.2891	**0.3923**
Transfer Attack	**0.1685**	**27.819**	**0.4333**	0.3747
Improv. (%)	48.1%	20.1%	49.8%	-4.4%
AGNEWS				
Direct Attack	0.0538	286.16	0.1317	0.2742
Transfer Attack	**0.0984**	**103.40**	**0.3589**	**0.3497**
Improv. (%)	82.9%	63.8%	172.5%	27.5%

5.2 Out-of-domain Text Reconstruction

To more comprehensively evaluate the capabilities of our methodology, we extended our evaluation of the transfer attack by incorporating an out-of-domain dataset, PersonaChat, as the external dataset, and present the result using SBERT as the victim embedding model in Table 2. We have the following findings. First, we found that utilizing an out-of-domain dataset is still helpful in improving attack performance. As shown in Ta-

Table 3: Ablation study on the QNLI dataset. Rows shaded in grey represent results obtained using the Direct Attack method, while rows shaded in blue indicate the use of attack methods employing only surrogate models without additional training techniques.

# D_L	Surrogate	Adv.	Consist Reg.	RougeL	Cos	LLM-Eval
500	✗	✗	✗	0.0617	0.0609	0.2436
	✓	✗	✗	0.1001	0.1310	0.2443
	✓	✓	✗	0.1192	0.1664	0.2550
	✓	✗	✓	0.1251	0.1801	**0.2686**
	✓	✓	✓	**0.1372**	**0.2031**	0.2663
8000	✗	✗	✗	0.1264	0.3257	0.3194
	✓	✗	✗	0.1701	0.4072	0.3598
	✓	✓	✗	0.1909	0.4742	0.4161
	✓	✗	✓	**0.1982**	**0.4902**	0.4266
	✓	✓	✓	0.1934	0.4886	**0.4280**

ble 7, transfer attack outperforms direct attack by roughly 20%-40% in QNLI and IMDB datasets. Second, we notice that attacking with an out-of-domain dataset can achieve similar performance as the in-domain dataset. Specifically, the relative performance drop when utilizing an out-of-domain dataset is only 9.9%, 3.1%, and 14.5% in embedding similarity and even lower in RougeL. This indicates that knowledge of the source domain is not always necessary for the attacker. Due to the page limit, the full attack result is presented in Appendix C.

6 Discussion

6.1 Ablation Study

Table 3 is presented to study the effectiveness of each component. Here we consider three primary components: surrogate model, adversarial training, and consistency regularization. Note that when all components are eliminated, our method becomes the direct attack method. Since the surrogate model is the key component of our transfer attack, we also highlight the performance of utilizing the surrogate model without any adjustment in blue. We first notice that using a surrogate model without training still improves the performance compared to direct attack, which indicates including additional training data could be helpful to some extent. Moreover, the surrogate model becomes better when either training objective is involved. For instance, the embedding similarity increases from 40% to 47% when including adversarial training when $|D_L|$ is 8000. Utilizing consistency regularization could further boost the embedding similarity from 40% to 49%. Finally, the full model with all components could usually achieve the best or second-best performance, although we see diminishing returns as $|D_L|$ increases.

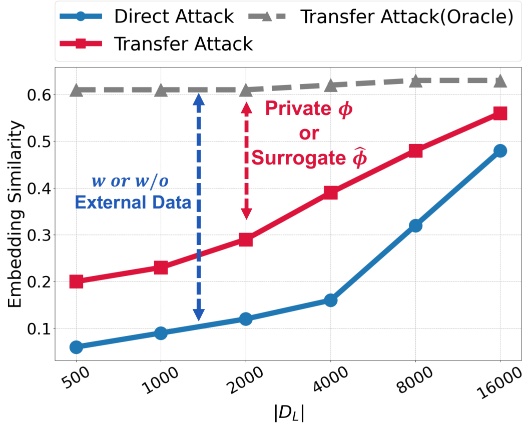

Figure 2: Comparison of attack performance on QNLI dataset $w.r.t.$ the amount of leaked dataset D_L.

6.2 Size of the Leaked Dataset

To understand the effectiveness of the surrogate model, we vary the number of leaked datasets to answer the following research questions.

How well does the surrogate model steal? Although we have seen an improvement in transfer attack over direct attack, it still remains unclear to what extent the surrogate model steals the victim model ϕ. Therefore, we implement the transfer attack(oracle) method which replaces the surrogate embedding with the actual embedding from the victim model. The result is presented in Figure 2. Comparing direct attack and transfer attack(oracle), it is evident that utilizing external data could enhance the performance and thus a good surrogate model becomes essential for a successful attack. Generally, a more leaked dataset makes the surrogate model steal better and reaches its upper bound (i.e., the oracle model) when $|D_L|$ is 16000. Under our default setting where $|D_L|$ is 8000, the transfer attack achieves a score of 0.48, which is roughly 77% of the upper limit's efficiency. Moreover, we also notice that the transfer attack is still effective when $|D_L|$ is small compared to a direct attack.

How much data is the surrogate model required to be effective? To understand when the surrogate model can perform a successful steal $w.r.t.$ the amount of leaked data, we calculate the surrogate stealing rate by the ratio of attack performance with the transfer attack and the oracle model. In Figure 3, the stealing rate across different datasets shows a similar trend. In general, the stealing rate achieves approximately 50% when $|D_L|$ is 2000 and exceeds 70% when $|D_L|$ is 8000. These results indicate our surrogate can effectively mimic the

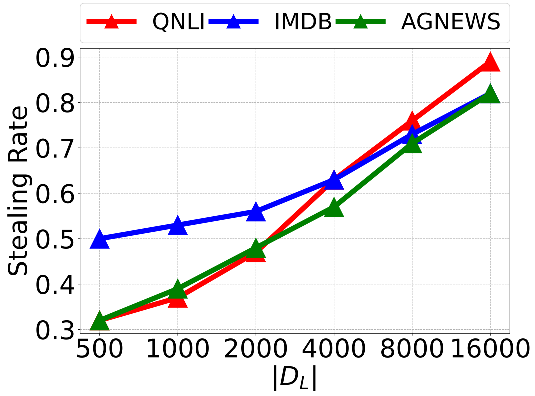

Figure 3: Stealing rate of the surrogate model compared to oracle model by varying the size of D_L.

black box encoder with sufficient leaked data and reveal the privacy associated with the leaked data.

6.3 Choice of a Surrogate Embedding Model

In this section, we explore how much the selection of a surrogate encoder affects the attack performance. Specifically, we use different surrogate encoders to attack embeddings generated with different victim encoders. The result is illustrated in Figure 4. Next, we discuss the result in two aspects. **1) Is it necessary to know the target victim encoder?** As the surrogate model is intended to replicate the behavior of the victim model, we seek to determine if employing an identical encoder enhances attack performance. The result of using an identical encoder can be found in the diagonal part of Figure 4. Comparing the diagonal and non-diagonal parts, we see that using identical encoder attacks slightly better than those with different encoders in the case of OpenAI and SBERT. Moreover, when using ST5 as the victim model, selecting OpenAI or SBERT can even attack better than ST5. This observation suggests that with our method, prior knowledge of the specific victim encoder is not required for a successful attack. **2) Is our attack sensitive to the choice of the surrogate encoder?** In general, Figure 4 suggests that the attack performance does not vary too much when fixing a victim model. Specifically, the largest performance difference is 1.79% for OpenAI, 1.63% for SBERT, and 0.82% for ST5. The result indicates that our attack pipeline is insensitive to the selection of the surrogate encoder.

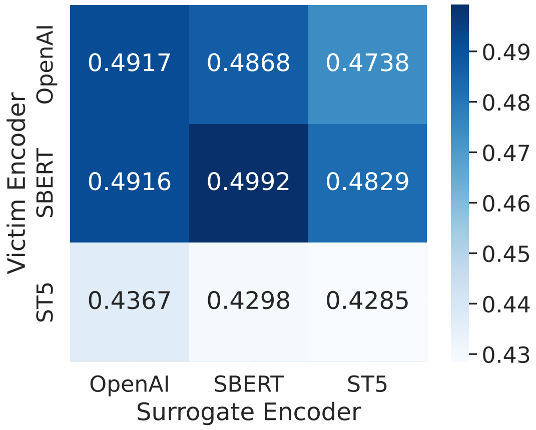

Figure 4: Attack performance by varying different victim and surrogate encoder. Here we use the embedding similarity metric to denote the attack performance.

7 Case Study

7.1 Embedding inversion on MIMIC dataset.

To demonstrate the privacy risks in a specific threat domain, we conduct a case study on MIMIC-III clinical notes (Johnson et al., 2018). The MIMIC-III dataset is a de-identified electronic health record database comprising comprehensive clinical data from intensive care units. Each note is truncated to its first sentence to remove redundant information. To be more realistic, 4K documents are sampled as the leaked dataset D_L. Following the out-of-domain attack setting, we choose PersonaChat as the external dataset.

To show the effectiveness of our transfer attack, we present the inverted result in Table 4 using SBERT as the victim embedding model. For visualization, we highlight the named entities in each sentence as an indicator of sensitive attributes. Comparing the inverted sentences, it is apparent that the transfer attack can almost recover the whole ground truth text, and the sensitive attributes are identified with high accuracy. However, the direct attack performs poorly due to the limited amount of leaked dataset.

7.2 Recovery Rate on Named Entities

To better understand how well can the transfer attack recover sensitive information from embeddings, we evaluate the named entity recover rate (NERR) and present the result in Table 5. Specifically, we use the biomedical NER model (Raza et al., 2022) to extract named entities for each clinical note. The result exhibits that a transfer attack is able to recover 98% of age and 99% of sex. In par-

Table 4: Case study on the MIMIC-III dataset. We highlight the named entities (e.g., age, sex, disease, symptom and medical history) extracted by the biomedical NER model (Raza et al., 2022) for visualization.

Attack Methods	Sentences
Example 1	
Ground truth	59 year-old male with a history of cardiomyopathy ef 45-50% with pcm/icd who presented due to sob.
Transfer Attack	59 year-old male with a past of cardiomyopathy ef 45-50% with pcm/icd who presented due to sob.
Direct Attack	this is a 64 year old male with known mitral regurgitation since.
Example 2	
Ground truth	this is a 78 year-old female with a history of ild who presents with altered mental status.
Transfer Attack	This is a 78 year-old woman with a history of ild who presents with different mental status.
Direct Attack	this is an 80-year-old female with a history of tracheobronchomalacia, copd, who presents with abdominal pain.
Example 3	
Ground truth	this 73 year old white male has known aortic stenosis which has progressed with increasing dyspnea.
Transfer Attack	This 73 year old white male has identified aortic stenosis which has progressed with worsening dyspnea.
Direct Attack	67 year old male with history of aortic stenosis followed by serial echocardiograms.

Table 5: Embedding inversion performance evaluated with named entity recovery rate on MIMIC dataset.

Attack Methods	Age	Sex	Disease	Symptom	History
Transfer Attack	98.84%	99.47%	79.07%	79.45%	65.36%
Direct Attack	7.79%	94.73%	19.35%	22.22%	17.49%

ticular, the transfer attack also achieves reasonable accuracy on disease, symptoms, and patient history and outperforms direct attack with a significant improvement. In summary, we found that the transfer attack can indeed reveal more privacy risks than a standard attack method.

8 Related Work

Inversion attacks on embeddings. Embedding inversion attacks have been explored across computer vision (Bordes et al., 2022; Dosovitskiy and Brox, 2016; Teterwak et al., 2021) and NLP (Pan et al., 2020) domains with significant implications for privacy. Typically, these inversion attacks make assumptions about the attacker's access to the victim model and evaluate the associated privacy risks. White-box scenarios assume attacker access to the full model weights, this enables the attack to approximate the inverse function with nearly 100% recover rate of text sequences (Kugler et al., 2021). Existing black-box attacks (Pan et al., 2020; Li et al., 2023) assume an attacker has no knowledge of the underlying model itself, and can only interact with models with the query access. A recent work (Morris et al., 2023) demonstrated an iterative recovery process that can reconstruct 92% of a 32-token text. Existing embedding inversion research largely depends on querying the victim model, yet the unexplored potential of query-free attacks presents a valuable opportunity for the community.

Stealing attacks on ML models. Many works in model stealing focus on stealing classifiers. In general, these methods steal the exact model parameters or functionality of target classifiers by querying them. For instance, prior works studied stealing ML models (e.g., decision tree or neural networks) deployed on cloud services. In a similar line, a few recent research (Cong et al., 2022; Liu et al., 2022) proposed an attack to steal a pre-trained encoder. By stealing the encoder, the attacker can obtain similar functionality on downstream tasks. For instance, StolenEncoder (Liu et al., 2022) demonstrated the effectiveness of stealing powerful encoders (e.g., CLIP (Radford et al., 2021) by OpenAI) with a ResNet-34 model. Similarly, there are several research (Naseh et al., 2023; Zanella-Beguelin et al., 2021) on stealing language models. Specifically, attacks on BERT-based APIs (Krishna et al., 2020; He et al., 2021) show that attackers can steal effectively via querying it without knowing the training data of the target language model. Different from the prior stealing attacks which steal encoders for downstream applications, our work leverages the stolen encoder to facilitate transfer attacks on the target encoder.

9 Conclusion

In this work, we study the privacy risks associated with text embeddings, especially under constraints where attackers lack direct query access to the embedding models. Through the development of a transfer attack method, we demonstrated the feasibility of inferring sensitive information from embeddings without needing to interact with the original model. Our extensive experiments across various embedding models and a detailed case study on a clinical dataset underline the effectiveness of our approach. As the use of text embeddings continues to grow in a wide range of applications, our work serves as a crucial step toward understanding and mitigating potential privacy threats.

10 Limitations

The primary limitation of our attack methodology is its ineffectiveness in accurately reconstructing longer sentences. Notably, when evaluating the effectiveness of our transfer attack on the AG News dataset, as depicted in Table 1, we observe that the RougeL scores are 5% to 7% lower than those achieved on the IMDB dataset, regardless of the black box embedding algorithm employed. Additionally, the model demonstrates the highest Perplexity (PPL) score on the AG News dataset in comparison to others, indicating a notable instability in model predictions. Table 6 further highlights that AG News has the longest average sentence length among the datasets examined. This trend suggests that our transfer attack approach encounters difficulties in the effective reconstruction of longer sentences.

11 Acknowledgement

This material is based upon work supported by National Science and Technology Council, ROC under grant number 111-2221-E-002 -146 -MY3 and 112-2634-F-002 -005

References

Afra Alishahi, Grzegorz Chrupała, and Tal Linzen. 2019. Analyzing and interpreting neural networks for nlp: A report on the first blackboxnlp workshop. *Natural Language Engineering*, 25(4):543–557.

F. Jelinek; R. L. Mercer; L. R. Bahl; J. K. Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Florian Bordes, Randall Balestriero, and Pascal Vincent. 2022. High fidelity visualization of what your self-supervised representation knows about. *Transactions on Machine Learning Research*.

Tianshuo Cong, Xinlei He, and Yang Zhang. 2022. Sslguard: A watermarking scheme for self-supervised learning pre-trained encoders. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 579–593.

Alexey Dosovitskiy and Thomas Brox. 2016. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35.

Xuanli He, Lingjuan Lyu, Lichao Sun, and Qiongkai Xu. 2021. Model extraction and adversarial transferability, your bert is vulnerable! In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2006–2012.

Alistair E W Johnson, David J Stone, Leo A Celi, and Tom J Pollard. 2018. The mimic code repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association*, 25(1):32–39.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2020. Thieves on sesame street! model extraction of bert-based apis. In *International Conference on Learning Representations*.

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*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Haoran Li, Mingshi Xu, and Yangqiu Song. 2023. Sentence embedding leaks more information than you expect: Generative embedding inversion attack to recover the whole sentence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14022–14040.

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

Yen-Ting Lin and Yun-Nung Chen. 2023. LLM-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 47–58, Toronto, Canada. Association for Computational Linguistics.

Yupei Liu, Jinyuan Jia, Hongbin Liu, and Neil Zhenqiang Gong. 2022. Stolenencoder: stealing pre-trained encoders in self-supervised learning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2115–2128.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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.

Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M Rush. 2023. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460.

Ali Naseh, Kalpesh Krishna, Mohit Iyyer, and Amir Houmansadr. 2023. Stealing the decoding algorithms of language models. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1835–1849.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022a. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, et al. 2022b. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Shaina Raza, Deepak John Reji, Femi Shajan, and Syed Raza Bashir. 2022. Large-scale application of named entity recognition to biomedicine and epidemiology. *PLOS Digital Health*, 1(12):e0000152.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 377–390.

Piotr Teterwak, Chiyuan Zhang, Dilip Krishnan, and Michael C Mozer. 2021. Understanding invariance via feedforward inversion of discriminatively trained classifiers. In *International Conference on Machine Learning*, pages 10225–10235. PMLR.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Santiago Zanella-Beguelin, Shruti Tople, Andrew Paverd, and Boris Köpf. 2021. Grey-box extraction of natural language models. In *International Conference on Machine Learning*, pages 12278–12286. PMLR.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

Table 6: Statistics of datasets.

Statistic Type	QNLI	IMDB	AGNEWS	PersonaChat
Task	NLI	Sentiment	Classification	Dialog
Domain	Wikipedia	Reviews	News	Chit-chat
Avg. sent length	18.25	21.14	28.09	10.12
Unique words	799519	1031895	1372484	1639738

A Detailed Dataset Statistics

Table 6 compares four datasets—QNLI, IMDB, AGNEWS, and PersonaChat—across various metrics. QNLI, focusing on Natural Language Inference from Wikipedia, has an average sentence length of 18.25 words and 799,519 unique words. IMDB, for sentiment analysis from movie reviews, has an average sentence length of 21.14 words and 1,031,895 unique words. AGNEWS, aimed at news classification, features the longest sentences on average (28.09 words) and 1,372,484 unique words. PersonaChat, designed for dialog in chit-chat, has the shortest sentences (10.12 words) but the largest vocabulary, with 1,639,738 unique words. This summary showcases the datasets' diversity in application, domain, and linguistic characteristics.

B Hyperparameters

We utilized pretrained DialoGPT-small (Zhang et al., 2020) as our specified attack model, while employing GTR-base (Ni et al., 2022b) as our surrogate encoder. For optimization, we employed the AdamW (Loshchilov and Hutter, 2018) optimizer with a learning rate of 3×10^{-5} alongside warmup and linear decay, using a batch size of 16.Under these conditions, our model undergoes training for approximately 15 hours.

C Full Out-of-Domain Experiment

The detailed results for different domain experiment are presented in Table 7. In the majority of scenarios, our approach surpasses the baseline method across several evaluation metrics, with the exception of perplexity. The data indicates an improvement exceeding 40% in embedding similarity scores and a 30% enhancement in RougeL scores.

D Comparison of Augmentation Strategies

Upon reviewing the results presented in Table 8, it is evident that Large Language Model-based (LLMDA) approaches exhibit superior performance, consistently ranking either as the best or second-best across all metrics. Notwithstanding, notable observations merit attention: RougeL and Cosine Similarity metrics for the Swap approach surpass those of LLM. This discrepancy can be attributed to RougeL and Cosine Similarity placing lesser emphasis on the sequential order of generated sentences. Additionally, the Swap method avoids introducing out-of-vocabulary words, a characteristic of potential significance, whereas LLM may generate words not present in the original dataset. These considerations contribute to the observed outcome wherein Swap outperforms LLM with respect to RougeL and Cosine Similarity metrics. However, a nuanced examination of specific sentences generated by LLM and Swap reveals the discernible superiority of sentences produced by LLM.

E Prompts for LLM Data Augmentation

We offer the following prompt for LLM Data augmentation, with the constraint of 2 words explicitly within the prompt.

Prompt template:

Table 7: Comparison of different domain embedding inversion performance between direct and transfer attack. The evaluation are done on QNLI, IMDB and AGNEWS datasets with embedding models including: OpenAI `text-embeddings-ada-002`, SBERT (Reimers and Gurevych, 2019) and ST5 (Ni et al., 2022a).

Dataset / Method	OpenAI				SBERT				ST5			
	RougeL	PPL	Cos	LLM-Eval	RougeL	PPL	Cos	LLM-Eval	RougeL	PPL	Cos	LLM-Eval
QNLI												
Direct Attack	0.1433	40.822	0.2797	0.2984	0.1264	27.127	0.3257	0.3194	0.1463	42.911	0.2226	0.2755
Transfer Attack	**0.2071**	**18.692**	**0.4253**	**0.3987**	**0.1800**	**20.515**	**0.4445**	**0.3899**	**0.1931**	**19.829**	**0.3946**	**0.3825**
Improv. (%)	44.5%	54.2%	52.0%	33.6%	42.4%	24.3%	36.5%	22.1%	31.9%	53.8%	77.2%	38.8%
IMDB												
Direct Attack	0.1133	**20.549**	0.2692	0.3818	0.1137	34.805	0.2891	**0.3923**	0.1103	**24.939**	0.2678	0.3909
Transfer Attack	**0.1808**	25.756	**0.4157**	**0.4504**	**0.1685**	**27.819**	**0.4333**	0.3747	**0.1563**	30.311	**0.3792**	**0.4398**
Improv. (%)	59.6%	-25.3%	54.4%	17.9%	48.1%	20.1%	49.8%	-4.4%	41.7%	-21.5%	41.6%	12.5%
AGNEWS												
Direct Attack	0.0612	**66.383**	0.1162	0.2979	0.0538	286.16	0.1317	0.2742	0.0578	**74.085**	0.0980	0.2905
Transfer Attack	**0.1066**	101.04	**0.3655**	**0.3618**	**0.0984**	**103.40**	**0.3589**	**0.3497**	**0.0938**	128.26	**0.3256**	**0.3460**
Improv. (%)	74.1%	-52.2%	214.5%	21.4%	82.9%	63.8%	172.5%	27.5%	62.2%	-73.1%	232.2%	19.1%

Table 8: Comparison of embedding inversion performance between different data augmentation approaches. We bold the best performance and underline the second-best performance in the table.

Method	RougeL	PPL	Cos	LLM-Eval
LLM	0.1782	**18.062**	0.4496	0.3976
Swap	**0.1932**	35.587	**0.5488**	0.3764
Delete	0.1579	19.944	0.3950	**0.4150**
Replace	0.1727	24.991	0.4120	0.3393
Insert	0.1138	18.644	0.3225	0.2387
w/o Aug.	0.1490	23.237	0.3419	0.3270

Please rewrite the original sentence with synonyms within 2 words.
Please output 5 different new sentences.
Please simply modify the original sentence without changing more than 2 words.

Example:
Original sentence:
{ORIGINAL SENTENCE}
New sentence:
{NEW SENTENCE 1}
{NEW SENTENCE 2}
{NEW SENTENCE 3}
{NEW SENTENCE 4}
{NEW SENTENCE 5}

Original sentence:
{INPUT SENTENCE}
New sentence:

F Details of LLM Evaluation

We assess the outcome using a large language model to closely emulate human assessment. Our evaluation metric aims to gauge semantic similarity, fluency, and coherence between the prediction and the ground truth sentence. Below is the prompt template utilized for this purpose.

Input prompt:

Output a number between 0 and 1 describing the semantic similiarity, fluent ,and coherent between the following two sentences: please output the answer without any explaination.
{pred sentence}
{ground truth sentence}

G More Case Study

Table 9 presents another case study conducted on the QNLI dataset, utilizing SBERT as the target embedding model. To aid visualization, we highlight the informative words within the ground truth sentences. Where inverted sentences contain sensitive named entities with analogous meanings, we have applied corresponding color highlights. This outcome underscores the effectiveness of transfer attacks in accurately recovering informative words, whereas direct attacks often result in erroneous accompanying information in the majority of cases.

Table 9: Case study on QNLI dataset. In the ground truth sentence, place is represented by red, time by purple, other noun by blue, verb by orange, and adjective by green.

Attack Methods	Sentence
Example 1	
Ground truth	Who founded the city of London?
Transfer Attack	Who founded the city of London?
Direct Attack	Which county in the Anglo-Saxon Empire?
Example 2	
Ground truth	What is the largest bird?
Transfer Attack	What is the most largest bird?
Direct Attack	How many animals inhabit the Tuna beak are various plankton Empire?
Example 3	
Ground truth	What was Nigeria's population in 2009?
Transfer Attack	What was Nigeria's population in 2011?
Direct Attack	What was the total number of people in the Middle East who had Internet before 2010?
Example 4	
Ground truth	What Air Force base is in Tucson?
Transfer Attack	What military airport is in Tucson?
Direct Attack	What is the total land area of the Army base on the Eisenhower Parkway?
Example 5	
Ground truth	Who established the Tibetan law code?
Transfer Attack	Who implemented the Tibetan Penal Code?
Direct Attack	How was the Tibetan Buddhists' policy on the TB inconsistent with secular practices?