國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

有限查詢存取下的文本嵌入逆推攻擊

Open-World Document Embedding Inversion Attack Under Limited Query Access

陳韋恩

Wei-En Chen

指導教授: 林守德 博士

Advisor: Shou-De Lin, Ph.D.

中華民國 112 年 7 月

July, 2023

doi:10.6342/NTU202302014

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

有限查詢存取下的文本嵌入逆推攻擊

Open-World Document Embedding Inversion Attack under Limited Query Access

本論文係陳韋恩君（學號 R10922058）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 112 年 7 月 19 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Computer Science and Information Engineering on 19 July 2023 have examined a Master's thesis entitled above presented by CHEN, WEI-EN (student ID: R10922058) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

（指導教授 Advisor）

系主任/所長 Director:

國立臺灣大學碩士學位論文
口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

有限查詢存取下的文本嵌入逆推攻擊

Open-World Document Embedding Inversion Attack under Limited Query Access

本論文係陳韋恩君（學號 R10922058）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 112 年 7 月 19 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department of Computer Science and Information Engineering on 19 July 2023 have examined a Master's thesis entitled above presented by CHEN, WEI-EN (student ID: R10922058) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

(指導教授 Advisor)

eSigned by:
李政德
07/19/2023 @ 04:13 UTC

彭文志

系主任/所長 Director:

誌謝

在這完成論文的時刻，我心中除了有完成碩士學業的喜悅外，也夾雜著濃濃的離愁。一方面是因為即將告別陪伴我度過無數時光的學生身份，另一方面則是因為即將離開 MSLab 而感到不捨。

首先，我要衷心感謝我的指導教授－林守德老師。我非常感激老師當初願意收我進入 MSLab，讓我有機會學習更多關於機器學習的知識並將其應用於實際情境中。感謝老師在這兩年的研究過程中給予我指引和寶貴建議，以及在投稿和 rebuttal 期間提供的許多想法和幫助。同時，我也非常感謝老師給予我在 Appier 實習的機會，讓我更深入了解機器學習在業界的應用，並將所學運用於實務中。這些寶貴的經驗對我未來的職業發展有著巨大的幫助。衷心感謝老師一直以來給予我的幫助和機會。

對於我的 DM 夥伴們，

蔡育哲學長，感謝你一直以來的支持和幫助。每當我遇到瓶頸時，你都會給予我許多建議並分享你的想法，幫助我在研究上有所突破。即使當我想學習其他領域的知識時，你也會提供我過去整理過的資料，讓我能更輕鬆地學習新知識。

劉力仁學長，感謝你在我剛開始碩士生涯時的耐心指導，讓我能更快地上手。也是因為你和育哲學長當初找我加入實驗室，才讓我有機會融入這個大家庭，

doi:10.6342/NTU202302014

讓我在研究之餘也能享受和實驗室夥伴們共度的歡樂時光。

黃昱翔和林泓毅，謝謝你們一直以來和我進行的研究討論，提供我許多想法和你們找到的資料。在最後投稿的期限逼近時，你們幫忙跑了許多實驗數據，讓這篇論文的實驗得以順利完成。

感謝所有 DM 夥伴們一直以來的陪伴，讓我這兩年的研究時光充實而愉快。

此外，我也要感謝其他 MSLab 的夥伴－楊鈞百、孫念恩、黃柏瑋、劉燕芬、張烟郁、郭濬睿、顏廷聿、吳庭維、詹凱傑。感謝大家一直以來的幫助、建議，以及一起聊天和休閒。謝謝你們與我共同度過的這些時光。

最後，我要感謝我的家人，感謝你們一直以來給予我無條件的支持和鼓勵。

doi:10.6342/NTU202302014

摘要

本文提出了一個針對文本嵌入逆推攻擊的解決方法，適用於當查詢數量有限的情境。為了應對這個挑戰，我們提出了一個名為”輔助文檔嵌入攻擊與查詢選擇（SADE）”的模型，旨在適應實際應用的場景。具體而言，我們引入了一種利用外部文檔的方法來克服當私有文檔非常有限的情況，這在實際場景中是一個常見的問題。此外，我們提出了一種新穎的查詢策略，解決了當私有編碼器的查詢數量非常有限的情況。與之前的方法不同的是，我們的方法利用少量的查詢即可實現高度的準確性，而不需要大量的私有文檔和無限制的查詢私有編碼器。整體來說，我們提出的模型和查詢策略證明了我們的方法在文本嵌入逆推攻擊中的有效性和實用性。

關鍵字：嵌入逆推攻擊、文本嵌入、有線查詢、代理模型、深度學習

doi:10.6342/NTU202302014

Abstract

This paper proposes a solution for embedding inversion attack of textual embedding in scenarios where the number of queries is limited. To address this challenge, we propose a model called Surrogate-Assisted Document Embedding Attack with Query Selection (SADE) that is designed to fit practical scenarios. Specifically, we introduce a means to exploit external documents to overcome the challenge of limited private documents, which is a common issue in practice. Additionally, we propose a novel query strategy that addresses the issue of limited queries to the private encoder. Unlike previous works that require a large number of private documents and unlimited query access, our approach utilizes a small number of queries to achieve high retrieval accuracy. Overall, our proposed model and query strategy demonstrate the effectiveness and practicality of our approach for embedding inversion attacks of textual embedding.

Keywords: Embedding Inversion Attack, Document Embedding, Limited Query, Surrogate Model, Deep Learning

doi:10.6342/NTU202302014

Contents

doi:10.6342/NTU202302014

doi:10.6342/NTU202302014

List of Figures

doi:10.6342/NTU202302014

List of Tables

doi:10.6342/NTU202302014

Chapter 1 Introduction

In recent years, representation learning techniques have experienced exponential growth and demonstrated remarkable success in various fields, including natural language processing, computer vision, and graph data analysis. The primary objective of representation learning is to convert structured or unstructured data into dense vectors that retain the input data's essential characteristics. These embeddings include semantic and contextual information, enabling humans to leverage them for downstream tasks. Despite the fact that complex black-box models (such as neural networks) generate the embedding vectors, and the dimensions of the embeddings may lack explicit meanings, the learned embeddings are highly valuable. Embeddings can be shared with third parties to streamline downstream tasks or uploaded to cloud services for further analysis. For instance, the document owner could generate document embedding locally and upload it to an online visualization platform[1]. Another example would be companies that release document embedding systems along with their pre-trained embeddings such as Word2vec [20], GloVe [24]. While releasing embedding vectors is intriguing, the underlying privacy risk is still unclear to the public. Therefore, exploring the privacy and information leakage of embedding vectors becomes the key motivation of this work.

With the increased literature on interpretability, fairness, and privacy, understanding

[1] https://projector.tensorflow.org/

doi:10.6342/NTU202302014

modern machine learning models' privacy concerns has become an essential topic. The privacy issues include revealing information of training data [27], stealing model weights or decision boundaries [23, 33] or inference sensitive information [4, 21] from the learned embedding vectors. In particular, inference sensitive information of input data from embedding (i.e., embedding inversion attack [28]) has become an emerging topic due to the advance of embedding models. In general, the embedding inversion attack exploits the idea that the adversary is given the latent representations(i.e., embeddings) $\mathbf{E}$ of data that contains sensitive or important information. On the other hand, the adversary aims to infer such information of interest from $\mathbf{E}$. Mahendran et al. [17] first proposed to invert image representation to its original image and demonstrate that several layers in CNNs retain photographically accurate information about the image. In the NLP domain, InvBERT [9] demonstrated that contextual representations could be perfectly reverted to the original sentence sequence when the weights of the embedding model are known to the adversary. On the other hand, Song et al. [28] proposed the Multi-Set model to facilitate the black box embedding attack where the embedding model is unknown. In this work, we aim to investigate the privacy risks of document embeddings in the context of inferring sensitive key terms of the document. For instance, we consider the scenario where electronic medical records are encoded into dense embedding vectors by the data owner (e.g., hospital or medical institute) to protect data privacy while releasing these vectors to third parties. However, if these document embeddings are leaked to an adversary, who is able to predict the top-k important terms in the original document with high accuracy, it could result in serious personal information leakage to the patients. To formalize this problem, we assume that the adversary is provided with the private embedding $\mathbf{E}$ as input to the threat model, and the objective is to predict the top-k important terms in the original document

doi:10.6342/NTU202302014

Table 1.1: Comparison of related studies. Note: given model weights of the embedding model (MW), acknowledgment of embedding model type (MT), query access limitation (QA)

	MW		MT		QA	
	Available	Unavailable	Known	Unknown	Limited	Unlimited
InvBERT [9]	✓		✓			✓
Multi-set [28]		✓	✓			✓
Coavoux et al. [4]		✓	✓			✓
Pan et al. [21]		✓	✓			✓
This work		✓		✓	✓	

according to an arbitrary evaluation function defined by the adversary, such as the term frequency-inverse document frequency (TF-IDF) metric, KeyBERT [7], or YAKE [2]. We evaluate the performance of the adversary in terms of the top-k accuracy, where the importance score of each word is ranked in descending order.

Despite the privacy risks revealed by previous literature discussed above, existing attack scenarios usually followed with several assumptions: (i) The adversary can obtain a large number of documents from a private domain, where the private domain denotes the domain of the document(e.g., medical or financial domain) used to generate embeddings $\mathbf{E}$. (ii) The adversary has unlimited query access to the private embedding algorithm [21, 28]. (iii) The adversary was given the type of embedding model (e.g., BERT or GPT) [4, 9]. The attack scenario under such assumptions exhibits a gap to general open-world adversary attack settings. Therefore, in this work, we bridge this gap by addressing the following research questions.

- **Public-to-Private Threat Model Transferability**: Can the adversary build a threat model from the publicly available dataset and perform a transferable attack on embeddings generated with a private dataset?
- **Limited Query Access**: How well can adversaries identify sensitive terms from embeddings with few, limited query access to the private embedding model?

doi:10.6342/NTU202302014

- **Unknown Model Algorithm**: Can sensitive information still be extracted by adversaries even when the type of embedding model is unknown?

To address these questions, we propose the use of domain adaptation techniques to achieve the transferability of threat models between public and private datasets. Additionally, we suggest a query selection strategy that accounts for the limitations of query access and enables adversaries to select the most appropriate documents to query. We also introduce a surrogate model that can learn the distribution of the private embedding model, rendering the private embedding model type irrelevant to the adversary. By addressing these challenges, our work provides practical solutions to mitigate the gap in general open-world adversary attack settings. To compare our approach with previous works [9], [28], [4], and [21], we present a comparison of the limitation settings across these different works in Table 1.1.

Moreover, the follow-up experimental results present compelling evidence of the efficacy of our proposed approach, exemplified by achieving the highest top-5 precision score of 75% and an NDCG score of 72%. Our approach effectively addresses the challenges posed by an unknown private embedding model type and a restricted number of 200 queries imposed on the adversary. Notably, our method outperforms the second-best competitor in terms of precision and NDCG score across most of the experiment settings, thus demonstrating its robustness and superior performance. These results lend strong support for the practical viability and utility of our proposed approach.

To summarize, the main contribution of our work is as follows:

- We introduce a publicly available dataset to leverage domain adaptation, mitigating the domain gap between the private and public datasets. This overcomes the chal-

doi:10.6342/NTU202302014

lenge posed by the limited number of documents available in the private domain and enhances the transferability of the attack.

- Query selection strategy: We propose a query selection strategy to assist adversaries in selecting the most suitable document to query under the situation of limited query access. This approach improves the efficiency of the attack and enhances its effectiveness, and reduces the number of queries required to achieve a high precision score.

- Pre-training of surrogate models: We pre-train the surrogate models to learn the distribution of the private embedding model. In doing so, we eliminate the need for adversaries to know the type of private embedding model used. This enhances the transferability of the attack and makes our method more robust to different types of embedding models.

- Experimental evaluation: We conduct thorough experiments evaluating the effectiveness of our method in a scenario where only 200 queries are allowed and the type of private embedding model is unknown. Our experiments demonstrate that our method outperforms the second-best competitors in terms of both the top-5 precision and NDCG score.

Overall, our approach enables transferable attacks on embeddings generated with private datasets while overcoming the limitations of previous attack scenarios.

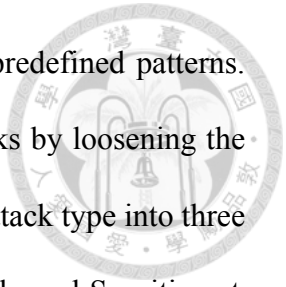
doi:10.6342/NTU202302014

Chapter 2 Related Work

Embedding model for text representation. Document embedding is a technique used to represent text documents as dense vectors in a high-dimensional space, allowing them to be used as inputs for machine learning models. One of the earliest methods for document embedding is the Bag-of-words model[8]. This method represents text as a multiset of its words, disregarding grammar and word order. A more recent approach to document embedding involves constructs document embeddings by performing vector arithmetic on the vectors corresponding to the words in the document. Doc2vec[14], is an attempt to generalize word2vec to work with word sequences. Recently, contextualized word embedding approaches have gained widespread use. Some of the most notable examples include [10], [11], and [15]. The Sentence-BERT [26] further enhances the quality of the sentence embeddings which is achieved by fine-tuning a pre-trained BERT model using a siamese network. By leveraging SBERT, more semantically meaningful sentence embeddings can be derived, leading to better performance in tasks such as semantic search and paraphrase mining.

Privacy in deep representation. In [29], Song et al. demonstrate that supervised models have the potential to retain sensitive information from input data that is not related to the task they were trained on, leading to unintended privacy violations. In the domain of NLP, [21] first to consider privacy concerns related to general-purpose language models.

doi:10.6342/NTU202302014



They particularly emphasize the reconstruction targets that exhibit predefined patterns. After that, [28] expands the taxonomy of embedding inversion attacks by loosening the requirement for a predefined pattern of the target. They classify the attack type into three categories, Membership inference attack, Embedding inversion attack, and Sensitive attribute inference attack. Membership inference attacks[27] aims to determine whether a data point was used in the training process or not. In embedding inversion attack, the adversary tries to reconstruct a set of words from the given document embedding and our work also belongs to this type of attack. In sensitive attribute inference attack[21], the adversary aims to identify sensitive information within the document, such as Citizen ID, authorship, and medical history. However, this information may not be explicitly present in the document embedding. Even so, [5] demonstrated that an adversary with access to the black-box model can still retrieve some sensitive information using a multi-label classifier.

doi:10.6342/NTU202302014

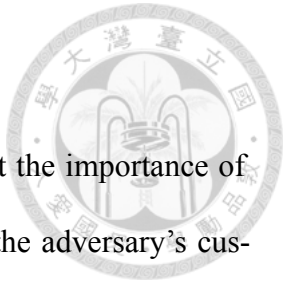
Chapter 3 Problem Definition

In this research, we adopt the attack pipeline described in a prior study [21], in which an adversary can make queries to a private encoder to obtain the document embeddings. To elucidate our approach, we introduce the following notations. We consider a private encoder denoted by PE, which an adversary can access with a limited number of queries N, a set of private embeddings denoted by $\tilde{E}$ that the adversary aims to attack and retrieve, and a set of private documents D_p that belong to the same domain as the documents used to generate the private embeddings by P. We define a query set as $Q = \{(d_n, e_n, y_n); 0 \leq n < N\}$, where $d_n \in D_p$ represents the documents selected by the adversary to query the private encoder P, e_n denotes the corresponding embeddings generated by P, and y_n denotes the label vectors. The label vectors are defined by the adversary and can be generated using techniques such as TF-IDF or the outputs of keyphrase extraction algorithms such as KeyBERT [7] and YAKE [2]. In this work, we refer to the label vectors as ”importance vectors”.

Additionally, we define the external set $E = \{(d'_m, \bar{e}'_m, y'_m), 0 \leq m < M\}$, where d'_m denotes the external documents that belong to the public domain and that the adversary can collect from other NLP benchmark datasets or the Internet, $\bar{e}'_m$ denotes the embeddings of d'_m through the surrogate model, y'_m denotes the importance score vectors of the external documents d'_m, and M represents the number of external documents available to

doi:10.6342/NTU202302014

the adversary.



The importance score vectors $y_n \in \mathbb{R}^V$ and $y'_m \in \mathbb{R}^V$ represent the importance of each word in the document d_n and d_m respectively with respect to the adversary's customized vocabulary. The score for each word can be calculated using various methods, such as term frequency-inverse document frequency (TF-IDF), or using neural network-based approaches like attention mechanisms. Typically, the size of the importance score vector is equal to the size of the customized vocabulary V. The detailed definition of those notations mentioned above is reported in Table 3.1.

Table 3.1: Detailed definition of all notations.

Notation	Definition	Notation	Definition
PE	The private encoder.	d_p	The private document.
S	The surrogate model.	e_p	The embedding of d_p generated from PE.
D_p	The private document sets.	$\bar{e}_p$	The embedding of d_p generated from S.
D_E	The external document sets	$\tilde{e}_u$	The private embedding $\in \tilde{E}$.
$\tilde{E}$	Private document embeddings set.	d_n	The document selected from D_p to query to PE.
P	Number of private documents.	y_n	The importance score vector of d_n.
M	Number of external documents.	e_n	The embedding of d_n generated from PE
U	Number of private emmbeddings $\in \tilde{E}$.	d'_m	The external document.
N	The budget for query access.	$\bar{e}_m$	The embedding of d_m generated from S.
V	Vocabulary size.	y'_m	The importance score vector of d'_m.

doi:10.6342/NTU202302014

Chapter 4 Methodology

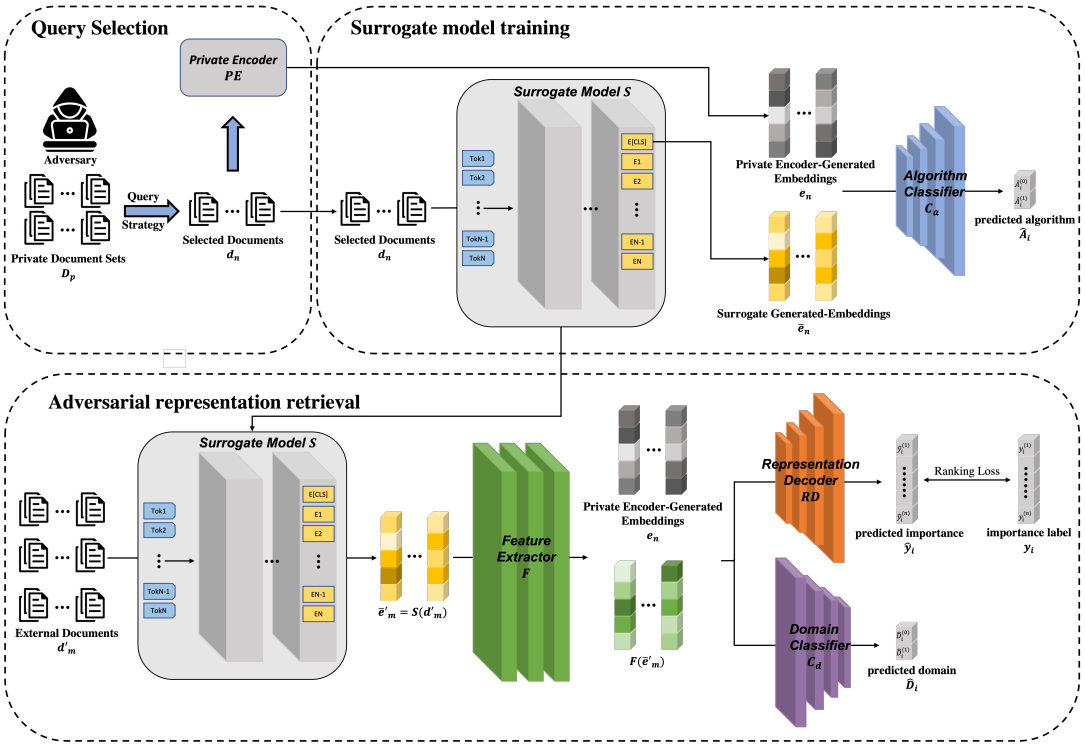


Figure 4.1: Overall structure of SADE.

This section presents our proposed model, Surrogate-Assisted Document Embedding Attack with Query Selection (SADE), which aims to extract sensitive information from private embeddings. The overall architecture of SADE is presented in Figure 4.1, which comprises three major stages: query selection, surrogate model training, and adversarial representation retrieval. Given the limited query access to the embeddings, the query strategy becomes a crucial factor in the query selection stage. In the query selection section, we explore various strategies that the adversary could employ to effectively select appro-

doi:10.6342/NTU202302014

priate documents from the private dataset D_p, while adhering to the constraints imposed by their query budget N. We elaborate on the details in Section 4.0.1 and the experiment results presented in Section 5.0.6.1 demonstrated that could significantly affect the attack performance. However, utilizing the limited queried data to build an attack model could lead to sub-optimal attack performance due to insufficient training data. To mitigate such an issue, we propose to learn a surrogate model S that aims to mimic the behavior of private encoder PE. With the aid of the surrogate model, the adversary to-some-extend obtains unlimited query access where the adversary could collect the publicly available datasets and obtain the corresponding document embedding by feeding to S for gaining additional labeled data. Finally, in the adversarial representation retrieval stage, we train the representation decoder that takes the document embedding as input and outputs the predicted importance of the individual word. Meanwhile, since there exists a distributional discrepancy between embeddings derived from public and private documents, we introduce a domain classifier to ensure the embeddings were mapped to identical embedding space. In the following, we provide a detailed description of the process for SADE.

4.0.1 Query Selection Strategy

Prior to introducing the detailed architecture of SADE, we will outline our query selection strategy, which aims to identify the most appropriate documents d_n from the private document set D_p given the adversary's limited query access to the private encoder PE. As a reminder, the adversary has access to the private document sets D_p, which contain a limited number of documents related to the private domain, and seeks to attack the private embeddings sets $\tilde{E}$ without knowledge of the original documents in order to retrieve sensitive information.

doi:10.6342/NTU202302014

The selection of suitable documents for querying the private encoder PE is a crucial factor that can significantly impact the performance of the representation decoder. The private encoder-generated embeddings e_n are highly dependent on the documents selected for querying, which in turn can affect the quality of the final retrieval results. Thus, it is imperative for the adversary to adopt an effective query selection strategy, especially when the number of queries is limited.

In our study, we propose a novel query selection strategy to address the aforementioned challenge. The conventional approach involves randomly selecting a set of documents d_n from D_p, but this approach has the drawback of generating unstable private encoder-generated embeddings e_n, which can, in turn, affect the performance of the representation decoder. The crux of the query selection strategy is to select documents whose embeddings are similar to the target embeddings while simultaneously ensuring that these documents are diverse enough to be scattered around the target embeddings distribution. To this end, we propose a document selection strategy that involves mapping the surrogate embedding $\bar{e}_p$ to the distribution of the private embeddings $\tilde{e}_u$ in $\tilde{E}$ and calculating the similarity between each transformed embedding and the K-means clustering center c_n.

Initially, the adversary generates surrogate embeddings $\bar{e}_p$ for all documents $d_p \in D_p$ by utilizing the surrogate model S. However, due to the potential non-congruity between the surrogate embeddings $\bar{e}_p$ and the private clustering centers of the private embeddings $\tilde{e}_u$, there exists a risk of low similarity scores for certain pairs $(c_n, \bar{e}_p)$ even when the corresponding original document $\bar{d}_p$ of $\bar{e}_p$ exhibits high similarity with the scattered documents within cluster n. This arises due to the fact that the clustering center c_n and the embeddings $\bar{e}_p$ belong to fundamentally different distributions. In essence, the similarity metric may not adequately reflect the similarity between the original documents when computed

doi:10.6342/NTU202302014

across differing distributions. To mitigate this challenge, it is imperative to ensure that the embeddings derived from disparate encoders are positioned in the same distribution.

To overcome this issue, it is necessary to identify a transformation matrix that can effectively map the surrogate embeddings $\bar{e}_p$ to the distribution of the private embeddings $\tilde{e}_u$. One commonly used approach to find the transformation matrix between two matrices is the Procrustes transformation [18]. In this work, we propose to apply the Procrustes transformation to the surrogate embeddings $\bar{e}_p$ prior to computing the similarity score. The objective is to ensure that similarity scores accurately reflect the similarity between the original document pairs even when the embeddings originate from different encoders with different distributions. To this end, we represent the embeddings as matrices $\mathbf{H} \in \mathbb{R}^{P \times W}$ and $\mathbf{B} \in \mathbb{R}^{U \times W}$, where each row corresponds to an embedding $\bar{e}_p$ and $\tilde{e}_u$ respectively, and W denotes the dimension of the embeddings. Due to the fact that the number of the $\bar{e}_p$ embeddings is less than the number of $\tilde{e}_u$ embeddings, resulting in $\mathbf{H} \in \mathbb{R}^{P \times W}$, we augment the matrix with zero vectors to obtain $\mathbf{A} \in \mathbb{R}^{U \times W}$. Subsequently, we use the Procrustes transformation to compute a matrix $\mathbf{T} \in \mathbb{R}^{W \times W}$ that maps $\mathbf{A}$ to $\mathbf{B}$ and transforms the embeddings into the same distribution space. The resulting transformed matrix is denoted as $\mathbf{A}'$, which satisfies the following equation:

$$\underbrace{\min}_{\mathbf{T}} \|\mathbf{AT} - \mathbf{B}\|_F^2 \tag{4.1}$$

$$\mathbf{A}' = \mathbf{AT} \tag{4.2}$$

where $\|\mathbf{A}\|_F$ denotes the Frobenius norm of matrix $\mathbf{A}$, defined as:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{p=1}^{P} \sum_{w=1}^{W} |a_{pw}|^2} = \sqrt{Tr(\mathbf{A}^\dagger \mathbf{A})} \tag{4.3}$$

doi:10.6342/NTU202302014

Here, a_{pw} denotes the element of matrix **A** with embedding index p at dimension w, and $Tr(A)$ denotes trace of matrix A.

After obtaining the transformed embeddings, we perform K-means clustering on the private embeddings $\tilde{e}_u$ to obtain N clustering centers c_n. For each clustering center c_n, we compute the cosine similarity Sim_{np} between it and all transformed surrogate embeddings a'_p. The similarity score is defined as follows:

$$Sim_{np} = \frac{c_n \cdot a'_p}{|c_n||a'_p|}; \quad 0 \leq n < N \quad 0; \leq p < P \tag{4.4}$$

Here, a'_p denotes the row with index p in matrix $\mathbf{A}'$, which represents the Procrustes-transformed embedding of $\bar{e}_p$.

After transforming the surrogate embeddings and computing the similarity scores, for each clustering center c_n, we select the candidate document d_x with the top one similarity scores with it. These documents are then queried to the private encoder PE to generate their corresponding private encoder-generated embeddings e_n. Finally, we construct the query set Q which consists of the selected documents d_n, the corresponding private encoder-generated embeddings e_n, and their importance score vectors y_n. The equations are denoted as follows:

$$Q = (d_n, e_n, y_n); 0 \leq n < N \tag{4.5}$$

$$d_n = QS(D_p), e_n = PE(d_n), y_n = IS(d_n) \tag{4.6}$$

where QS represents the query selection strategy employed by the adversary and IS denotes the importance score algorithm used, such as term frequency-inverse document frequency (TF-IDF).

doi:10.6342/NTU202302014

The complete process of query selection strategy of SADE is depicted in algorithm 1.

Algorithm 1: Query strategy algorithm

Input: Private documents set D_p, Private document embeddings $\tilde{E}$
Output: Query set Q

Generate embeddings with the surrogate model $\bar{e}_p \leftarrow S(d_p)$
// Procrustes Transformation
Construct the embedding matrix **A** with $\bar{e}_p$
Construct the embedding matrix **B** with $\tilde{e}_u \in \tilde{E}$
$\mathbf{A}' \leftarrow$ Procrustes($\mathbf{A}, \mathbf{B}$)
// K-Means similarity
$E_c \leftarrow$ K-Means($\tilde{E}$) with N Clsuters
for $c_n \in E_c$ **do**
    Compute the similarity between each a'_p
    Get the index x with max similarity
    $e_x \leftarrow P(d_x)$
    $Q_x \leftarrow (d_x, e_x, y_x)$
end
output query set Q

4.0.2 Surrogate Model Training

The first stage of our approach involves training the surrogate model S, which consists of an encoder (e.g., transformer-based encoder, Doc2Vec, or other document embedding encoder algorithms). The surrogate model generates the embeddings $\bar{e}_n$ of the selected documents $d_n \in Q$ and aims to mimic the output embeddings distribution of the private encoder PE via private encoder-generated embeddings e_n. We also introduce an algorithm classifier C_a, which discriminates whether the embeddings e_n and $\bar{e}_n$ are from the private encoder PE or the surrogate model S, respectively.

The primary task for the surrogate model S is to ensure that the distribution of $\bar{e}_n$ is similar to that of $e_n \in Q$, while also confusing C_a by generating embeddings that are difficult to distinguish from e_n. As training progresses, the discrimination from C_a forces the embeddings $\bar{e}_n$ generated by S to gradually align with e_n in the adversarial process.

doi:10.6342/NTU202302014

To achieve this goal, we define an objective function L_{align} as follows:

$$L_{align} = \min_{S} \max_{c_a} V(C_a, S) \tag{4.7}$$

$$V(C_a, S) = \mathbb{E}_{e\sim Q}[\log C_a(e)] + \mathbb{E}_{d\sim Q}[\log (1 - C_a(S(d)))] \tag{4.8}$$

where $\mathbb{E}e \sim Q[logC_a(e)]$ represents the expected logarithmic probability of the algorithm classifier C_a given input embeddings e sampled from the query set Q. Similarly, $\mathbb{E}_{d\sim Q}[log(1-C_a(S(d)))]$ represents the expected logarithmic probability of C_a given surrogate embeddings $\bar{e} = S(d)$ generated and sampled from the surrogate model. The objective function encourages the surrogate model S to generate embeddings that are difficult for the algorithm classifier C_a to classify as either from the private encoder or the surrogate model, effectively confusing C_a.

After completing the surrogate model training stage, we can gather external documents from various NLP benchmarks or directly from parsed documents on the internet. These external documents are denoted as $\{d'_m \in D_E; 0 \leq m < M\}$, where D_E represents the set of collected external documents, and M represents the total number of documents in D_E. The domain of these documents d'_m is considered public, which may or may not be similar to the private domain. The surrogate model S is then used as a document embedding encoder to generate embeddings $\bar{e}'_m$ for external documents d'_m. This creates an external set as follows:

$$E = (d'_m, \bar{e}'_m, y'_m), 0 \leq m < M \tag{4.9}$$

$$d'_m \in D_E, \bar{e}'_m = S(d'_m), y'_m = IS(d'_m) \tag{4.10}$$

The external set will function as a pivot for the subsequent stage of our proposed approach.

doi:10.6342/NTU202302014

4.0.3 Adversarial Representation Retrieval

The purpose of incorporating external documents d'_m and their embeddings $\bar{e}'_m$ is to improve the retrieval performance of the representation decoder, particularly when the adversary only has access to limited private encoder embedding pairs (d_n, e_n). The set of the external set denoted as $E = (d'_m, \bar{e}'_m, y'_m)$, in our work, d'_m is the documents from the WikiText [19]. For each document $d_n \in Q$ and $d'_m \in E$, we construct the retrieval labels or importance score vector y_n and y'_m respectively. Each element y_n^v or y_m^v represents the TF-IDF score of a word in the original corpus d_n or d'_m, with v indicating the index of the word in the customized vocabulary.

It is important to note that the domain of the public documents d'_m may not necessarily be similar to the private domain of d_n. Thus, to transform the embeddings $e'_m \in E$ from the public domain to the private domain, we apply domain adaptation through an adversarial training process. Once the embeddings have undergone domain adaptation, we employ the feature extractor and representation decoder to extract the latent representation for the retrieval task.

Since the adversary selects the documents $d'_m \in E$ from a known domain, while the documents $d_n \in Q$ are from an unknown domain, domain adaptation can be employed to transform the public domain (chosen by the adversary) into the private domain (unknown to the adversary). We adopt an adversarial training approach, similar to the one described in [6], that involves a feature extractor F and a domain classifier C_d. The feature extractor converts the embeddings $\bar{e}'_m$ of the public domain to that e_n of the private domain, while the domain classifier aims to accurately classify the domain of each embedding. The feature extractor also functions as an auxiliary role, ensuring that the transformed embeddings

doi:10.6342/NTU202302014

which are sampled from the public domain are suitable for the representation decoder RD to extract the latent representation. Thus, the objective function of the feature extractor is a combination of the adversarial objectives and representation decoder objectives, and is expressed as follows:

$$L_F = L_{RD} + \alpha \times L_{adv} \tag{4.11}$$

$$L_{adv} = \min_{F} \max_{C_d} V(C_d, F) \tag{4.12}$$

$$V(C_d, F) = \mathbb{E}_{e \sim Q}[\log C_d(e)] + \mathbb{E}_{\bar{e}' \sim E}[\log (1 - C_d(F(\bar{e}')))] \tag{4.13}$$

where L_{RD} denotes the objective functions of the representation decoder RD, and α represents the weight of the adversarial loss term L_{adv}. The term $\mathbb{E}_{e \sim Q}[\log(1 - C_d(e))]$ represents the expected value of the logarithmic probability of the domain classifier C_d when the embeddings $e_n \in Q$ are sampled. The domain classifier tries to minimize this term by correctly classifying the private domain embeddings. Similarly, the term $\mathbb{E}_{\bar{e}' \sim E}[\log 1 - C_d(F(\bar{e}'))]$ represents the expected value of the logarithmic probability of the domain classifier C_d when the embeddings $\bar{e}' \in E$ are sampled and transformed by the feature extractor F to the private domain. In this case, the domain classifier tries to maximize this term by distinguishing between the embeddings in the public and private domains. Simultaneously, the feature extractor tries to minimize it by transforming the embeddings such that they are indistinguishable by the domain classifier. As training progresses, the distribution $\bar{e}'_m \in E$ in the public domain becomes increasingly similar to the distribution of $e_n \in Q$ in the private domain, making it harder for the domain classifier C_d to distinguish between them. Moreover, the embeddings $\bar{e}'_m \in E$ are transformed to retain more relevant and abundant latent semantics, enabling the representation decoder RD to extract sensitive information from the original corpus d'_m. The domain classifier

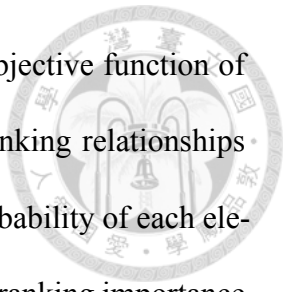
doi:10.6342/NTU202302014

C_d, which is adversarially trained with the feature extractor F, discriminates between the elements $e_n \in Q$ and $\bar{e}'_m \in E$ to determine their domain. This competitive process enables the domain classifier to become more precise at distinguishing domain class while forcing the feature extractor to transform the public domain where $\bar{e}'_m$ lie closer to the private domain where e_n lie, thus reducing the domain gap gradually.

In this work, the representation decoder RD is tasked with extracting the latent semantics of the embeddings e_n and $\bar{e}'_m$, and predicting the importance vector $\hat{y_n}$ and $\hat{y'_m}$, which represents the importance of each word in the original corpus. The adversary can infer sensitive information of d_n and d'_m through the ranking of the elements in the predicted importance vector $\hat{y_n}$ and $\hat{y'_m}$. It's worth noting that the importance score vectors can be quite large - in the order of the customized vocabulary size V, which can be over 4000 in some cases. However, the original documents corresponding to these embeddings are typically quite short, with fewer than 100 words, leading to highly sparse importance score vectors with most elements being zero. This sparsity can cause the representation decoder RD to predict all-zero outputs during training when using mean-squared error (MSE) as the objective function.

Previous studies, such as [28], [21], and [4], have treated this task as a multi-label classification problem and used binary cross-entropy (BCE) as the objective function. While BCE can identify and detect the presence of words in the original documents, it is more practical for the RD to be able to extract and analyze the relative relationships between these detected words. In other words, when given the embeddings, identifying the ranking of the importance of the presence words provides more valuable information for an adversary who wants to extract and attack the sensitive information of the embeddings.

doi:10.6342/NTU202302014



To mitigate the above challenges, we apply ListNet [3] as the objective function of our representation decoder RD, a listwise method that learns the ranking relationships between the target and predicted vectors. By optimizing the top-1 probability of each element in the prediction lists, this approach enables RD to optimize the ranking importance of the words, rather than just detecting their presence.

ListNet, which was introduced in [3], presents a listwise ranking approach for learning the ranking relationship between target and prediction lists. The authors propose a probabilistic model for ranking that is capable of computing the probability of all possible permutations of a list based on the scores of its elements. The ranking objective function is formally defined as follows:

$$L(y_i, \hat{y}_i) = -\sum_{j=1}^{V} P_{y_i}(j) \log(P_{\hat{y}_i}(j)), \tag{4.14}$$

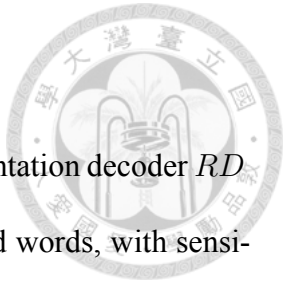
$$P_{y_i}(j) = \frac{\exp(y_i^j)}{\sum_{l=1}^{V} \exp(y_i^l)} \tag{4.15}$$

where V is the size of the customized vocabulary, y_i and $\hat{y}_i$ are the importance score vectors of the ground truth and predicted importance scores, respectively, and $Py_i(j)$ is the probability that the importance score of the word with index j is ranked at the top one position in the importance vector y_i.

The proposed probabilistic method provides a principled way to capture the complex dependencies among the elements in a list and obtain a ranking probability distribution over all possible permutations. By maximizing the likelihood of the observed rankings, ListNet [3] can learn a set of effective ranking features that are specific to the target list and improve the ranking performance. Overall, ListNet [3] offers a promising solution to the challenging task of learning to rank and has demonstrated its effectiveness on various

doi:10.6342/NTU202302014

datasets in information retrieval and natural language processing.



Optimizing the objective function in Eq. 4.14 enables the representation decoder RD to identify and analyze the relative relationships between the detected words, with sensitive words assigned higher importance scores in the original documents. The ranking objective function offers an improvement over previous studies that regarded the task as multi-label classification and adopted binary cross-entropy as the objective function, which only allows the representation decoder to detect the presence of the words without considering their ranking importance. To clarify the entire attack pipeline of SADE, we depict the training algorithm in Algorithm 2. It is worth noting that during the inference stage, only the trained representation decoder RD is used to extract sensitive information and predict the importance score vectors from the private embeddings $\tilde{e}_u$.

doi:10.6342/NTU202302014

Algorithm 2: Training algorithm

Input: Query set Q, External documents set D_E
Output: Trained SADE
// Train surrogate model S
Initialize algorithm classifier C_a with parameters Θ_{C_a}
Initialize surrogate model S with parameters Θ_S
for epoch $\in \{1, 2, \ldots,$ pre-training epochs$\}$ **do**
    $\bar{e}_n \leftarrow S(d_n)$, where $d_n \in Q$
    Update Θ_{C_a} with $\{e_n, \bar{e}_n\}$ using cross entropy loss
    Update Θ_S using adversarial training loss L_{align}
end
// Generate external set E
for $d'_m \in D_E$ **do**
    $\bar{e}'_m \leftarrow \mathrm{S}(d'_m)$
end
Construct the external set E
// Train representation decoder RD
Initialize feature extractor F with parameters Θ_F
Initialize domain classifier C_d with parameters Θ_{C_d}
Initialize representation decoder RD with parameters Θ_{RD}
for epoch $\in \{1, 2, ...,$ training epochs$\}$ **do**
    $\hat{y}_n \leftarrow RD(e_n)$
    $\hat{y}'_m \leftarrow RD(F(\bar{e}'_m))$
    Update Θ_{C_d} with $\{e_n, F(\bar{e}'_m)\}$ using cross entropy loss
    Update Θ_F using adversarial training loss L_{adv}
    Update Θ_{RD} with $\{(y_n, \hat{y}_n), (y'_m, \hat{y}'_m)\}$ using ListNet
end
output trained weight Θ

doi:10.6342/NTU202302014

Chapter 5 Experiments

In this section, we conduct a comprehensive analysis and comparison of our proposed method via a series of experiments. We begin by evaluating the performance of our method under different experimental settings, including various datasets, retrieval targets, private encoder algorithms, query selection strategies, and surrogate model algorithms. Our experiments consist of detailed analyses that demonstrate the consistent superiority of our proposed method over other baseline models. Furthermore, we conduct an ablation study to investigate the impact of different components of SADE on the overall architecture, providing insights into the effectiveness of each component. We conducted a series of experiments to address the following research questions:

- RQ1: To what extent can SADE adapt to diverse corpora?
- RQ2: What is the effectiveness of SADE across different target?
- RQ3: How effectively can SADE adapt to multiple embedding algorithms?
- RQ4: Is gaining knowledge of a private encoder algorithm critical for an adversary?
- RQ5: Which component is responsible for the pivotal role in SADE?

doi:10.6342/NTU202302014

5.0.1 Experimental Settings

Datasets and Embedding Algorithm. We conducted experiments on three widely used datasets: (1) 20 News (20NG)[13], consisting of about 20,000 newsgroup posts on 20 different topics, (2) Ag News[31], containing news articles from over 2000 different news sources, and (3) IMDB [16], comprising 50,000 highly polarized movie reviews. We also utilized (4) WikiText [19], a dataset with a collection of over 100 million tokens extracted from Wikipedia articles, as the external set. In the data preprocessing phase, we removed stopwords and words that appeared less than 0.3% of the time from the sentences and removed words other than nouns and verbs through POS tagging. Finally, we filtered out articles with fewer than 15 words. Table 5.1 displays the statistical data of the datasets after preprocessing, including the number of documents, vocabulary size, and average length.

Table 5.1: Detail information of different datasets.

Dataset	# document	vocab. size	Avg. length
20 News [13]	18,589	4823	86.8
AG News [31]	19,081	1154	18.8
IMDB [16]	48,404	3904	63.7
WikiText [19]	13052	2479	31.9

For the data split, we partitioned the dataset into three subsets: 80% for testing, containing the private embeddings $\tilde{e}_i \in \tilde{E}$ to be attacked by the adversary; 1000 documents reserved for query selection, denoted as the private document set D_p possessed by the adversary, from which only N (the query budget) documents were selected for querying to the private encoder, while the other documents were discarded; and the remaining data were used for validation to determine the optimal hyperparameters. Furthermore, we randomly sampled 10,000 documents from WikiText [19] to create the external set. To

doi:10.6342/NTU202302014

investigate the effect of attacks on different embeddings, we selected seven representative methods, namely SBERT [26], BERT [11], ALBERT [12], ERNIE [32], XLNET [30], GPT2 [25], and Doc2vec [14]. To verify the attack performance, we compared three retrieval targets: TF-IDF, KeyBERT [7], and YAKE [2].

Attack Models. We used five different baseline models to compare with our proposed method.

- MLP: The approach introduced in [28] transforms the given document embeddings to the important score vectors. We use ListNet [3] as the objective function to train this baseline model.
- PLD: Pseudo labeling decoder. A two-layer neural network that trains with a pseudo-labeling technique. We also apply ListNet [3] as its objective function. We introduce the same external dataset E as mentioned in our method to achieve semi-supervised learning.
- DAAM: The Domain Adversarial Attack Model, introduced and following the same setting as in [21].
- RB: A rule-based approach that outputs the average of the label vectors of owned private domain documents as the prediction.
- BERT-GAN: A semi-supervised method that introduces adversarial training to help improve model performance under limited data scenarios. It consists of a BERT as the generator and a two-layer network as the decoder. We replace the original objective function from cross-entropy to ListNet [3] to better fit our task.

Evaluation Criteria. In our study, we employed two widely used evaluation metrics in

doi:10.6342/NTU202302014

machine learning to measure the performance of SADE: Precision@K and Normalized Discounted Cumulative Gain (NDCG)@K.

Precision@k is a standard metric used to evaluate the effectiveness of ranking algorithms by measuring the proportion of relevant words among the top k words in a ranked list. The relevance of a word is typically assessed using a binary label, where a relevant word is assigned a label of 1, and an irrelevant word is assigned a label of 0. The precision@k score is computed by dividing the number of relevant words in the top k by k.

NDCG@K is another widely used metric for evaluating ranking algorithms, which measures the effectiveness of a ranking algorithm by assessing how well it ranks a set of words based on their relevance scores. The NDCG@k score is computed by taking the discounted cumulative gain (DCG) and normalizing it by the ideal discounted cumulative gain (IDCG). DCG is computed by taking the sum of the relevance scores of the top-ranked words, where the relevance scores are typically binary or graded on a scale from 0 to 1, and are discounted based on their position in the ranking, with words at the top being given more weight.

The use of these evaluation metrics in our study allowed us to quantitatively evaluate the performance of our proposed model (SADE) in document embedding retrieval.

Hyperparameters. In our experiment, we fix the decoder as a 2-layer neural network, set the batch size to 32, and the learning rate to 1e-4 during training. In addition, as mentioned in Section 4.2, we set the value of α, which is used to adjust the adversarial loss, to 10 to make the adversarial loss value similar to the ListNet [3] loss. Finally, in the model selection part, we use the validation set to choose the best-performing model among every

doi:10.6342/NTU202302014

epoch and evaluate the score on the testing set.

Default Setting. It is worth noting that for the following experiments, the default setting includes the 20 Newsgroups [13] as the dataset, TF-IDF as the retrieval target, SBERT [26] as the surrogate model and private encoder algorithm, and 200 as the number of allowed queries.

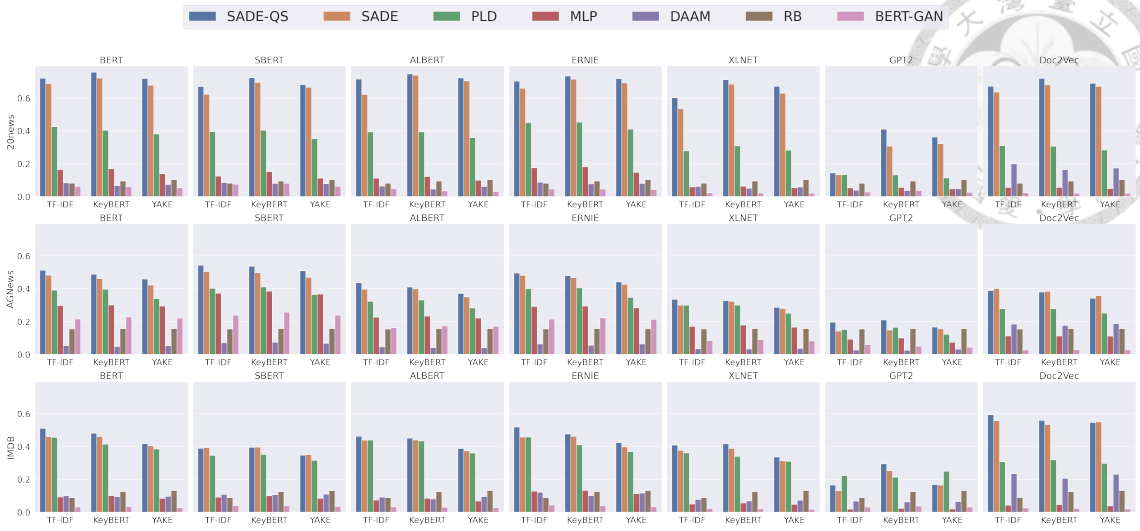
5.0.2 Attack performance analysis

To evaluate the effectiveness of our proposed method, SADE, we experimented with seven distinct encoders and utilized SBERT [26] as the surrogate model to retrieve importance scores for three different targets. Due to the limited number of queries allowed to the private encoder, we only used 200 private encoder-generated embeddings. We then compared the performance of SADE with three baseline models. The obtained Precision@5 and NDCG@5 results are presented in Figure 5.1a and Figure 5.1b respectively.

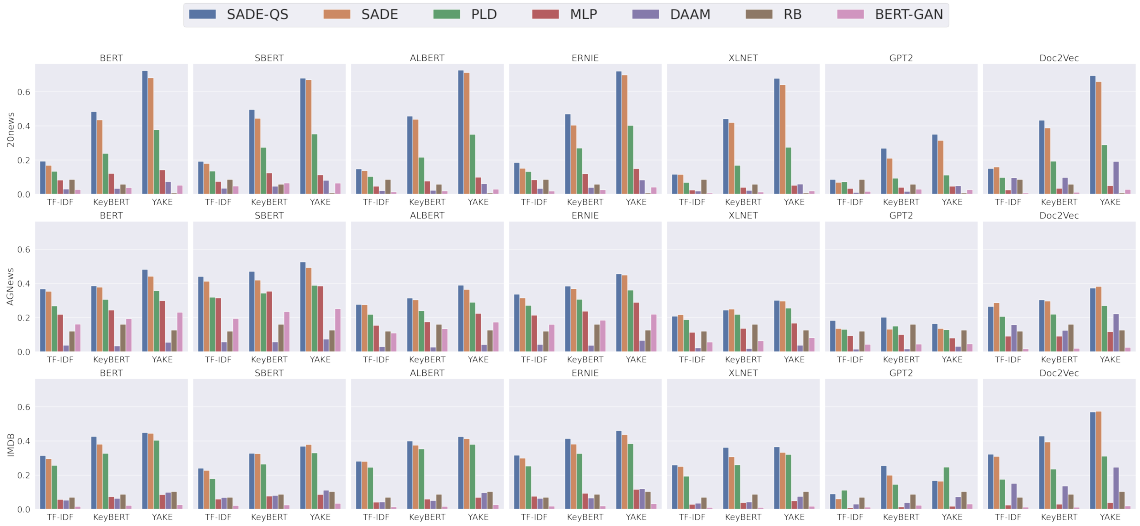
5.0.2.1 Comparing on different threat models (RQ1).

The results of our experiments, presented in Figures 5.1a and Figure 5.1b, demonstrate that SADE outperforms three baseline models in terms of top 5 precision and NDCG scores across all datasets and retrieval targets, even when limited to only 200 queries. Specifically, in the first column of Figure 5.1a and Figure 5.1b, when the private encoder is SBERT [26], SADE achieves top-5 precision and NDCG scores of 67% and 19%, respectively, while the second strongest baseline (PLD) achieves only 39% and 13%. Overall, SADE obtains an average top-5 precision and NDCG score of 48% and 36%, respectively, compared to 33% and 25% for PLD. These results demonstrate the effectiveness and poten-

doi:10.6342/NTU202302014



(a) Attack performance $w.r.t.$ Precision@5.



(b) Attack performance $w.r.t.$ NDCG@5.

Figure 5.1: Comparison of attack performance on different datasets, retrieval targets, and private encoder algorithms using different models. SADE-QS represents SADE with query selection, which selects different document embedding pairs compared to the five baselines. On the other hand, SADE represents SADE with random selection, which chooses the same document embedding pairs as the five baselines.

tial of SADE in the extraction of sensitive information from document embeddings and its superior performance over competing methods. Furthermore, our experiments show that SADE is robust and reliable in terms of embedding inversion attacks, and does not require a large number of queries to achieve effective retrieval of important words. These findings suggest that SADE can be applied to various domains and datasets for the extraction of sensitive information.

doi:10.6342/NTU202302014

5.0.2.2 Comparing on different retrieval targets (RQ2).

In practical scenarios, the importance score may vary based on the adversary's focus on different sensitive information. To address this, we introduced additional types of importance scores by leveraging the output of other automatic keyphrase extraction tasks [22]. Keyphrase extraction is a process that involves automatically extracting representative phrases to summarize the main ideas of a document. To generate importance score vectors for each document, we leverage two different models: KeyBERT [7], which calculates the distance between each n-gram in the context and the document embedding to identify the most relevant phrases, and YAKE [2], an unsupervised model that combines several handcrafted features, including word casing, word position, word frequency, and word relatedness to the context, to derive scores for the words in the context. The experimental results demonstrate that SADE can effectively extract and retrieve sensitive words even when customized importance scores are used, which allows for a more targeted and focused approach to extracting sensitive information.

5.0.2.3 Comparing on different embedding algorithms (RQ3).

We investigated the effectiveness of a proposed embedding attack on several private embedding algorithms using SBERT [26] as the surrogate model. The results reveal that, in general, the attack achieves high precision scores (ranging from 40% to 70%) on top-5 predictions for most private embedding algorithms, with the exception of GPT2 [25]. This finding suggests that the use of GPT2 as a private embedding algorithm significantly improves the robustness of the embeddings against the proposed attack. Consequently, our study recommends the adoption of GPT2 as a private embedding algorithm for individuals

doi:10.6342/NTU202302014

seeking to prevent embedding attacks.

5.0.3 Attack performance varying number of queries.

To investigate the impact of the number of queries on SADE's performance, we conducted an experiment where the number of queries varied from a highly limited scenario of 10 to a more relaxed scenario of 1000. The performance of SADE was compared with other baseline models. The results, as illustrated in Figure 5.2, indicate that SADE and PLD consistently outperform other competitors in terms of top-5 precision, regardless of the number of queries. This outcome is attributed to the use of external documents in both methods, which assist the representation decoder in learning to extract and retrieve keywords from the given embeddings, even with a limited number of private documents. Additionally, the superiority of SADE over PLD suggests the effectiveness of domain adaptation on external documents, enabling the representation decoder to learn effectively from these additional embeddings. Furthermore, when the number of queries is quite limited (only 10 is allowed), the difference in the decline of SADEand PLD also indicates that these additional embeddings, applied through domain adaptation, serve as an auxiliary role for the representation decoder to learn to extract information, particularly in scenarios where only a small number of queries are available.

5.0.4 Attack performance varying number of private documents.

In this experiment, we investigate how the number of private documents affects the attack performance. With a fixed query budget of 100, we vary the number of private domain documents from 1000, 500, and 200, to 100. As shown in Figure 5.3, we observe

doi:10.6342/NTU202302014

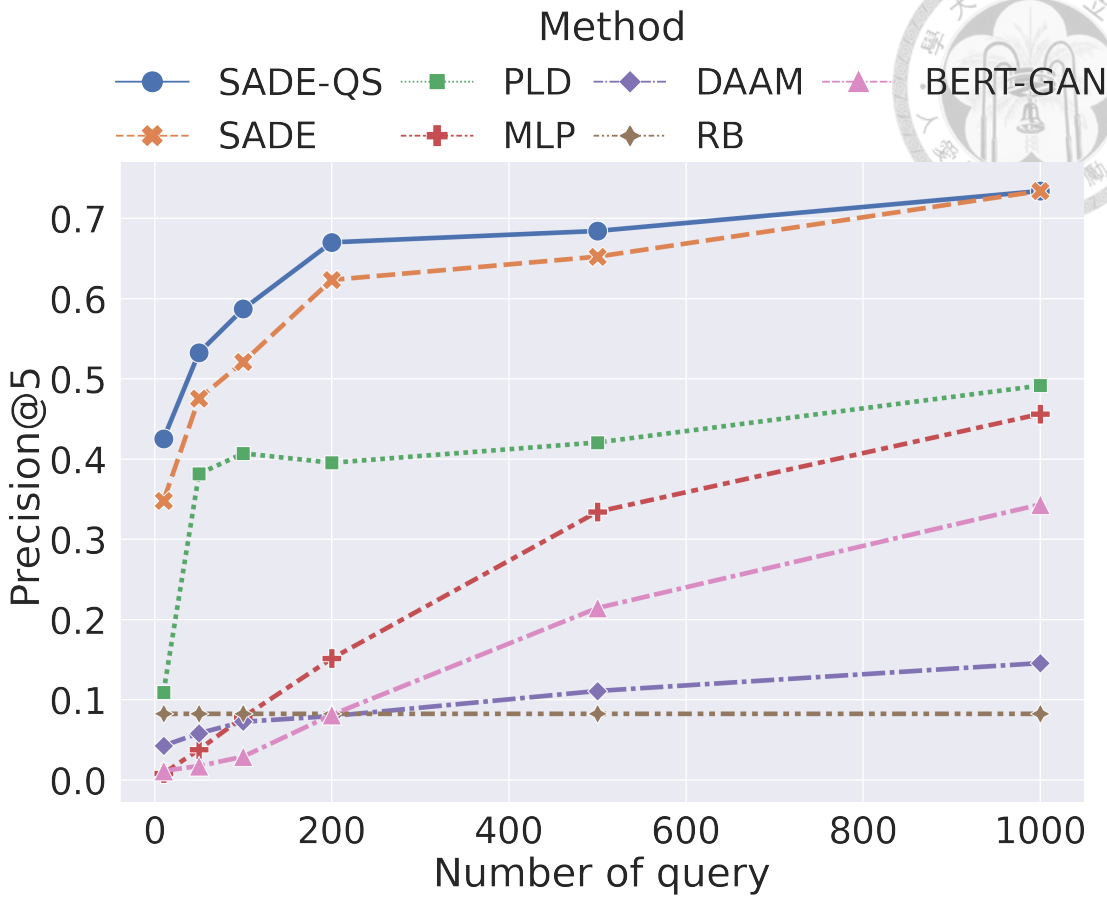


Figure 5.2: Performance varying number of query. SADE-QS represents SADE with query selection, which selects different document embedding pairs compared to the five baselines. On the other hand, SADE represents SADE with random selection, which chooses the same document embedding pairs as the five baselines.

that the number of selection pool (i.e., the private documents) has the most significant impact on SADE-QS. This is because the strategy relies on selecting the most appropriate documents, some candidate documents may be excluded from the selection pool when the number of private domain documents decreases. Notably, when the number of private documents decreases to the query budget of 100, SADE-QS demonstrates similar attack performance to SADE, which adopts random selection.

doi:10.6342/NTU202302014

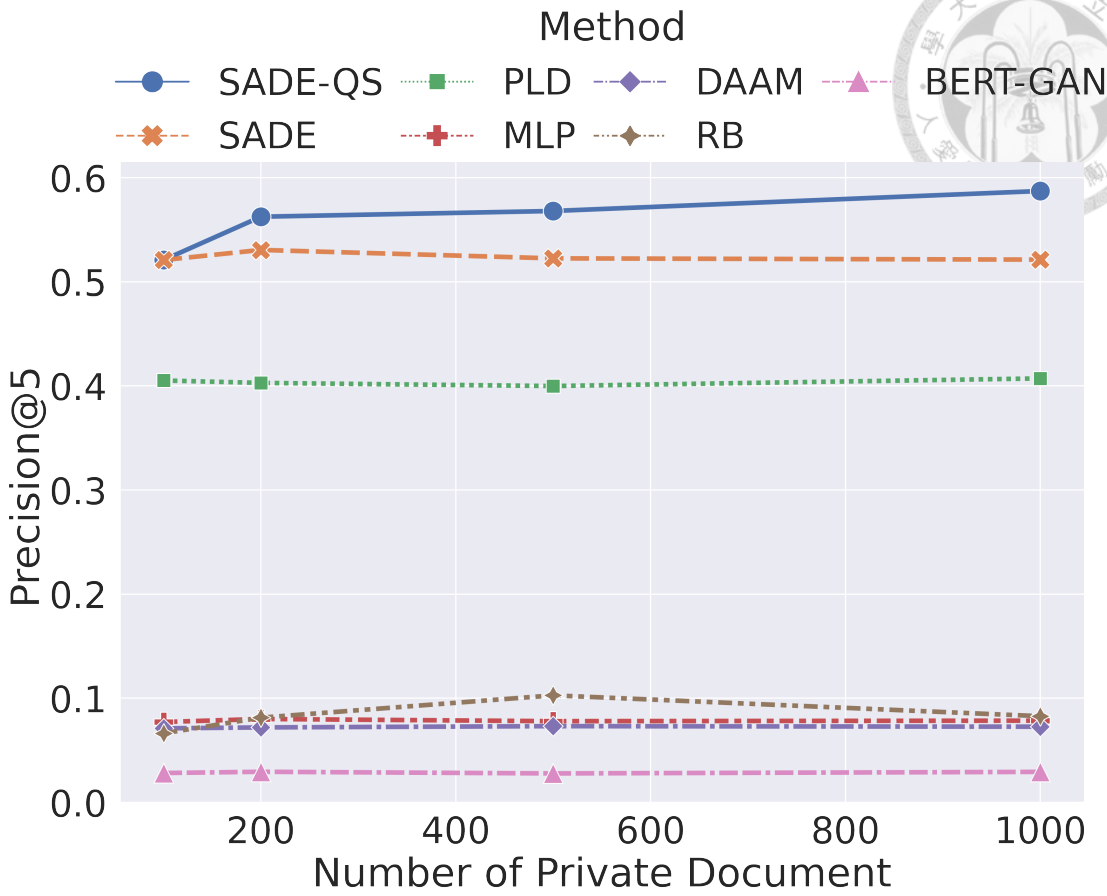


Figure 5.3: Performance varying number of private document. SADE-QS represents SADE with query selection, which selects different document embedding pairs compared to the five baselines. On the other hand, SADE represents SADE with random selection, which chooses the same document embedding pairs as the five baselines.

5.0.5 Effectiveness of surrogate model structure with different embeddings algorithm (RQ4).

This study aims to explore the impact of different encoder algorithms on the performance of the representation decoder. To this end, we have conducted experiments in which we evaluated the top 5 precision scores achieved by utilizing SBERT [26], BERT [11], ALBERT [12], ERNIE [32], XLNET [30], GPT2 [25], or Doc2Vec [14] embeddings as the surrogate model, along with seven distinct private encoder algorithms.

Our findings indicate that the optimal performance of the representation decoder,

doi:10.6342/NTU202302014

as measured by top-5 precision, is not necessarily obtained when the corresponding private encoder algorithm is employed as the surrogate model. This suggests that gaining knowledge of the private encoder algorithm may not be a critical factor for the adversary. Figure 5.4 displays the precision scores for each embedding algorithm and private encoder combination, highlighting the varied performance outcomes observed across the different experimental conditions.

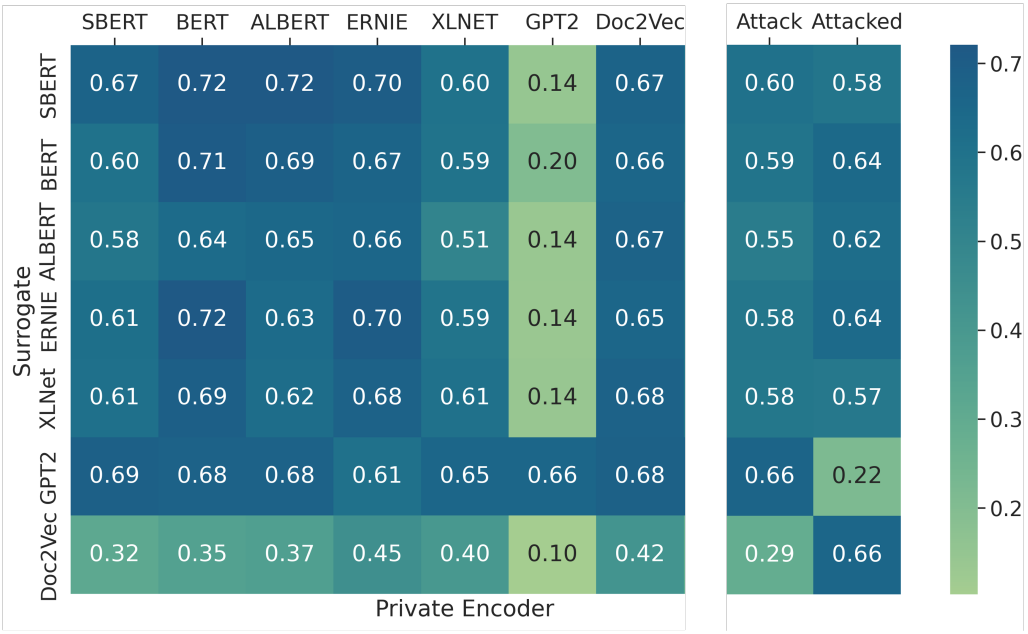
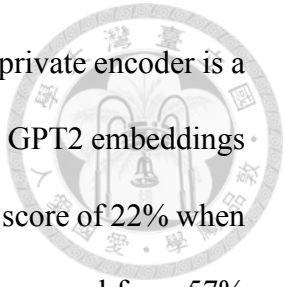


Figure 5.4: Heatmap of different encoder algorithms.

In Figure 5.4, the ”Attack” column shows the average of the top 5 precision scores obtained by using an encoder as a surrogate model to attack all seven encoders as the private encoder (the higher the better). Our results indicate that GPT2 [25] is a more favorable choice for adversaries attempting to launch embedding attacks, with an average top 5 precision score of 66%, compared to Doc2Vec [14], which only achieves an average top 5 precision score of 29%.

The ”Attacked” column indicates the average of the top 5 precision scores obtained by all seven encoders as the surrogate models when using an encoder as the private encoder

doi:10.6342/NTU202302014



(the lower the better). Our findings suggest that using GPT2 [25] as a private encoder is a promising approach for individuals seeking to prevent such attacks, as GPT2 embeddings are more robust to adversarial attacks, with an average top 5 precision score of 22% when subjected to attacks, compared to the other six encoders whose scores ranged from 57% to 66%.

In summary, our study demonstrates that GPT2 [25] is both a more effective surrogate model for embedding attacks and a more robust private encoder compared to other models evaluated.

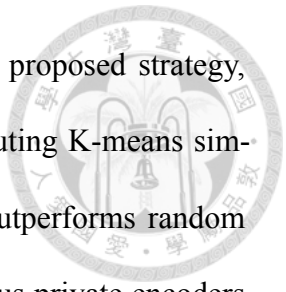
5.0.6 Ablation Study (RQ5).

This section presents experimental analysis to evaluate the impact of each component of SADE on document embedding retrieval. Our experiments are organized into four parts. Firstly, we examine the impact of query selection. Secondly, we investigate the effectiveness of the ranking objective. Thirdly, we examine the effectiveness of pre-training the surrogate model. Fourthly, we analyze the impact of adversarial training during the 2nd stage. Lastly, we verify the combined effect of pre-training and the existence of adversarial training. For comparison purposes, we include the baseline MLP, which can be viewed as a version of SADE that does not pre-train the surrogate model and does not include adversarial training in the 2nd stage.

5.0.6.1 Analysis on different query strategies.

Table 5.2 presents a comprehensive analysis of the impact of query selection on retrieval performance in SADE across various private encoders and datasets. The study

doi:10.6342/NTU202302014



compares two query selection strategies: random selection and the proposed strategy, which involves applying the Procrustes transformation before computing K-means similarity. The experimental results show that the proposed strategy outperforms random selection in terms of precision scores and NDCG scores across various private encoders and datasets, underscoring the significance of query selection strategy in determining the overall retrieval performance of SADE.

These findings are particularly noteworthy in embedding inversion attack, where the extraction of sensitive information from private encoders is the primary objective. The proposed query selection strategy enhances the performance of SADE in extracting sensitive information, thereby improving the overall retrieval performance. Moreover, the generalizability of the proposed query selection strategy is demonstrated by its consistent effectiveness across different private encoders and datasets.

In summary, this study provides empirical evidence of the critical role played by query selection in the retrieval performance of SADE and highlights the effectiveness of the proposed query selection strategy in enhancing the retrieval performance of the model. For more details on different query selection strategies, refer to 7. Additionally, 7 provides insights into the impact of diversity in selected documents on the attack results.

5.0.6.2 Effectiveness of the ranking objective

As previously mentioned, the sparsity of retrieval target-importance score vectors can negatively impact the performance of the representation decoder in terms of retrieval accuracy. To address this issue, we conducted a comparison between mean squared error

doi:10.6342/NTU202302014

Table 5.2: Strategy of query selection.

	Precision@5 / NDCG@5						
20 Newsgroups	SBERT	BERT	ALBERT	ERNIE	XLNET	GPT2	Doc2vec
Random	0.6233	0.6891	0.6222	0.6597	0.5355	0.1325	0.6370
	0.1795	0.1688	0.1376	0.1516	0.1154	0.0689	**0.1597**
P-Kmeans	**0.6701**	**0.7208**	**0.7161**	**0.7032**	**0.6022**	**0.1438**	**0.6723**
	0.1918	**0.1933**	**0.1480**	**0.1851**	**0.1162**	**0.0857**	0.1501
AG News							
Random	0.5039	0.4825	0.3969	0.4808	0.2998	0.1415	**0.4004**
	0.4126	0.3536	0.2755	0.3153	**0.2169**	0.1353	**0.2868**
P-Kmeans	**0.5423**	**0.5124**	**0.4361**	**0.4958**	**0.3355**	**0.1959**	0.3881
	0.4405	**0.3682**	**0.2767**	**0.3370**	0.2081	**0.1828**	0.2645
IMDB							
Random	**0.3935**	0.4604	0.4389	0.4578	0.3773	0.1305	0.5578
	0.2271	0.2954	0.2797	0.2996	0.2501	0.0605	0.3090
P-Kmeans	0.3891	**0.5105**	**0.4622**	**0.5186**	**0.4088**	**0.1652**	**0.5935**
	0.2401	**0.3138**	**0.2812**	**0.3166**	**0.2593**	**0.0898**	**0.3225**

(MSE) and the ranking objective for optimizing SADE. The results, presented in Table 5.3, demonstrate that optimizing the ranking objective yields superior performance, with an improvement of approximately 40% in the top 5 precision score and 5% in the NDCG score compared to using MSE.

These findings support our insight that the sparsity of the retrieval target can negatively affect the performance of the representation decoder when MSE is used as the objective function. Moreover, the results suggest that by adopting ListNet [3] as the objective function, the representation decoder is able to extract and retrieve sensitive words based on their ranking positions.

Table 5.3: Comparison of Objective function.

	k = 5		**k = 10**		**k = 15**	
	Precision	NDCG	Precision	NDCG	Precision	NDCG
MSE	0.2875	0.1499	0.2430	0.1479	0.2190	0.1466
BCE	0.1374	0.1250	0.0956	0.1106	0.0757	0.1029
ListNet	**0.6701**	**0.1918**	**0.5433**	**0.1905**	**0.4526**	**0.1891**

doi:10.6342/NTU202302014

5.0.6.3 Effectiveness of training the surrogate model

Table 5.4 presents our analysis of the impact of training the surrogate model on the performance of SADE. The results demonstrate that training the surrogate model effectively enhances the retrieval performance of SADE. Specifically, the improved accuracy in document embedding retrieval indicates that the closer the surrogate model imitates the private encoder, the smaller the gap between the surrogate-generated embedding and the private encoder-generated embedding. With training the surrogate model, SADE achieves top 5 precision score and NDCG values of 67% and 19%, respectively, both of which outperform those when the process of training the surrogate model is deprecated. Notably, even without training the surrogate model, SADE can still outperform the other baseline MLP. These results underscore the importance of training the surrogate model for SADE to achieve superior retrieval performance.

Table 5.4: Ablation of training the surrogate model.

	k = 5		**k = 10**		**k = 15**	
	Precision	NDCG	Precision	NDCG	Precision	NDCG
MLP	0.1240	0.0742	0.1029	0.0738	0.0922	0.0740
without training	0.6021	0.1479	0.4572	0.1445	0.3717	0.1418
with training	**0.6701**	**0.1918**	**0.5433**	**0.1905**	**0.4526**	**0.1891**

5.0.6.4 Impact of the adversarial training during 2nd stage

We conducted an experiment to compare the retrieval performance of the modified SADE architecture with the original SADE architecture, in which the feature extractor and domain classifier were removed. The results are presented in Table 5.5. The modified SADE architecture achieved a lower top-5 precision score and NDCG score than the original SADE architecture, which indicates that the adversarial training between the feature extractor and the domain classifier significantly improves the retrieval performance.

doi:10.6342/NTU202302014

Thus, we conclude that the adversarial training component during the 2nd stage is indeed essential for transferring the embeddings from the public domain to the private domain while retaining sufficient information for the representation decoder to extract the sensitive information. Therefore, it is indispensable for SADE to achieve superior retrieval performance.

Table 5.5: Ablation of Adversarial training in 2nd-Stage.

	k = 5		**k = 10**		**k = 15**	
	Precision	NDCG	Precision	NDCG	Precision	NDCG
MLP	0.1240	0.0742	0.1029	0.0738	0.0922	0.0740
no adversarial	0.6035	0.1617	0.4846	0.1690	0.3595	0.1715
adversarial	**0.6701**	**0.1918**	**0.5433**	**0.1905**	**0.4526**	**0.1891**

5.0.6.5 Comprehensive Analysis

The results presented in Table 5.6 clearly demonstrate that the performance improvements observed in SADE are not solely attributable to the training of the surrogate model or adversarial training components alone. Rather, it is the combination of both that leads to the most significant enhancements in retrieval performance. Specifically, the configurations lacking either training the surrogate model or adversarial training outperform the MLP baseline but fail to match the performance of the complete SADE architecture that integrates both components. These findings indicate that training the surrogate model and adversarial training during the second stage are complementary and synergistic techniques that work in tandem to enable the representation decoder to extract sensitive information with superior top-5 precision and NDCG scores. Therefore, we conclude that both components are indispensable for SADE to achieve optimal performance.

doi:10.6342/NTU202302014

Table 5.6: Ablation of training the surrogate model and the adversarial training in 2nd-stage.

	k = 5		**k = 10**		**k = 15**	
	Precision	NDCG	Precision	NDCG	Precision	NDCG
MLP	0.1240	0.0742	0.1029	0.0738	0.0922	0.0740
training + no adversarial	0.6035	0.1617	0.4846	0.1690	0.3595	0.1715
no training + adversarial	0.6021	0.1479	0.4572	0.1445	0.3717	0.1418
training + adversarial	**0.6701**	**0.1918**	**0.5433**	**0.1905**	**0.4526**	**0.1891**

doi:10.6342/NTU202302014

Chapter 6 Defense Approach

It is imperative to investigate effective defense strategies to address the potential issue of information leakage through embeddings in the context of adversarial attacks. One commonly utilized approach is to obfuscate the embeddings. In this study, we evaluate four distinct defense techniques for embedding obfuscation while preserving the utility of the embeddings. The ultimate objective is to remove sensitive information while retaining the essential information for downstream tasks, including classification tasks on the 20 Newsgroups dataset. Our empirical findings suggest that achieving this desirable scenario necessitates a trade-off between privacy and utility. Specifically, we evaluate four defense approaches, namely, Laplace noise, Random Projection, PCA, and Autoencoder which are introduced as follows:

6.0.1 Laplace noise

In this defense approach, we leverage a differential privacy approach in our study to address potential information leakage through embeddings. Specifically, we utilize Laplace noise to obfuscate the embeddings while preserving their utility. The approach involves randomly selecting a 1000 document from the private documents set and removing the top frequency of the word from each document multiple times. After each removal,

doi:10.6342/NTU202302014

we generate the embedding for the modified document and compute the distance between the original and modified embeddings. The average distance computed, Δf, is used as the Laplace noise scale. To ensure differential privacy, we add Laplace noise to the embedding with a scale of $\frac{\Delta f}{\varepsilon}$.

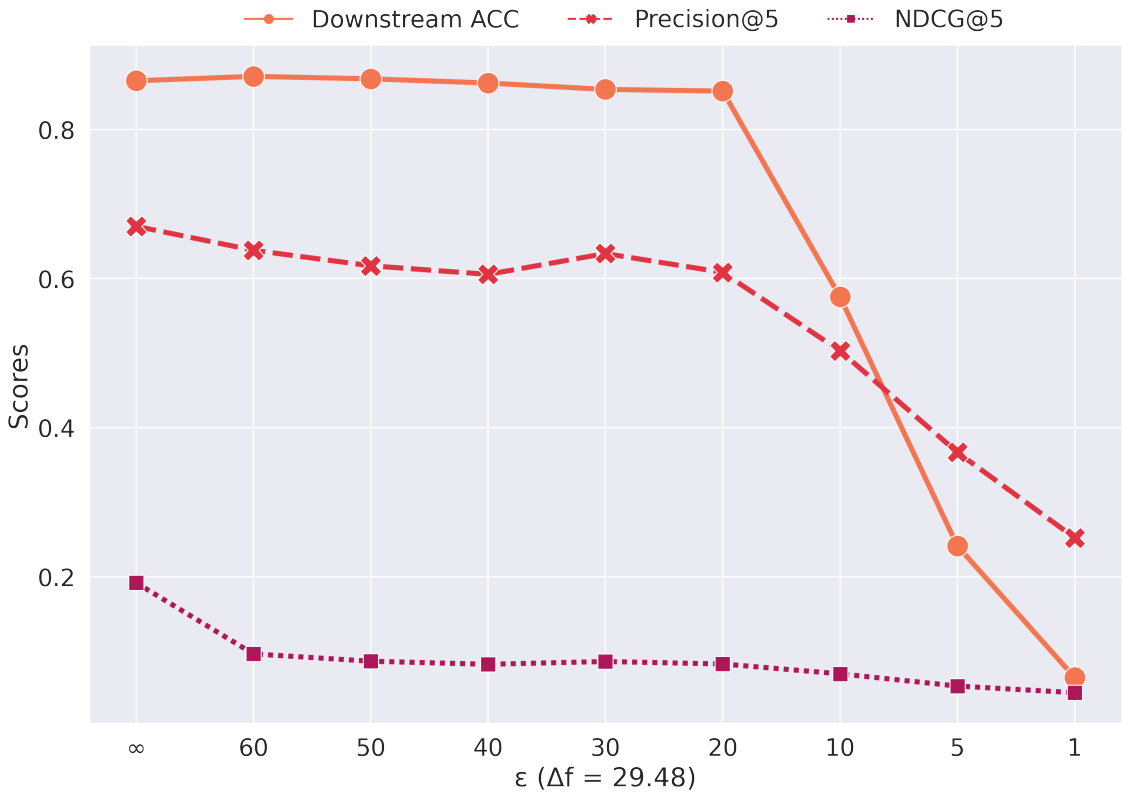


Figure 6.1: Attack and utility performance varying different ε.

6.0.2 Random Projection

The second approach aims to reduce the dimensionality of the original embedding space by projecting it with a randomly generated matrix. The matrix follows the distribution $\mathcal{N}(0, \frac{1}{d})$, where d represents the reduced dimension. The primary objective of this method is to preserve the pairwise distance between any two samples of the original data while obfuscating the embeddings to mitigate potential information leakage. According to [1], this method has shown applicability to text documents.

doi:10.6342/NTU202302014

6.0.3 PCA

The third defense approach we employ is Principal Component Analysis (PCA), a widely used technique to reduce the dimensionality of data while retaining as much information as possible. PCA transforms the original data into a new set of orthogonal variables that account for most of the variance in the data. By reducing the dimensionality of the original embedding space, we aim to obfuscate the embeddings and mitigate potential information leakage.

6.0.4 Autoencoder

The last defense approach is a non-linear dimension reduction method utilizing an autoencoder. An autoencoder is a neural network that learns to reconstruct its input data through an encoder and decoder architecture. We train a two-layer autoencoder with the objective of minimizing the mean squared error (MSE) between the original document embeddings and their reconstructed versions. We use the encoder part of the autoencoder to achieve dimension reduction. This approach is particularly effective when the reduced dimension is relatively small.

Figure 6.2: Attack and utility performance varying different dimension.

The efficacy of Laplace noise as a defense mechanism is substantiated by the results shown in Figure 6.1, which reveal the challenges in simultaneously preserving the utility

doi:10.6342/NTU202302014

of embeddings while mitigating the potential information leakage due to adversarial attacks. Moreover, Figure 6.2 provides a comparative analysis of three distinct dimension reduction-based methods, namely Random Projection, PCA, and Autoencoder, which also validate this finding. Consequently, the findings indicate the existence of a trade-off between privacy and utility when implementing defense approaches for embedding obfuscation.

doi:10.6342/NTU202302014

Chapter 7 Conclusion

In this study, we conducted a comprehensive investigation into the privacy and information leakage of document embeddings in the open-world scenario, characterized by severely limited query access. To address this challenge, we propose a query selection strategy that identifies the most appropriate documents for an adversary to query the private encoder. Moreover, we introduce external documents from the public domain and leverage domain adaptation techniques to reduce the domain gap between the private and public domains.

Our experimental results demonstrate that our proposed approach consistently outperforms other baseline models on various datasets, retrieval targets, and private encoder algorithms. We also conduct a thorough analysis of the impact of different configurations of each component of our approach on attack performance. Finally, we present practical defense approaches to our attack, demonstrating the importance of addressing privacy and information leakage in document embeddings.

doi:10.6342/NTU202302014

References

[1] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 245–250, 2001.

[2] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. Yake! keyword extraction from single documents using multiple local features. Information Sciences, 509:257–289, 2020.

[3] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In Proceedings of the 24th international conference on Machine learning, pages 129–136, 2007.

[4] M. Coavoux, S. Narayan, and S. B. Cohen. Privacy-preserving neural representations of text. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1–10, 2018.

[5] M. Coavoux, S. Narayan, and S. B. Cohen. Privacy-preserving neural representations of text. arXiv preprint arXiv:1808.09408, 2018.

[6] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marc-

doi:10.6342/NTU202302014

hand, and V. Lempitsky. Domain-adversarial training of neural networks. The journal of machine learning research, 17(1):2096–2030, 2016.

[7] M. Grootendorst. Keybert: Minimal keyword extraction with bert., 2020.

[8] Z. S. Harris. Distributional structure. Word, 10(2-3):146–162, 1954.

[9] J. Höhmann, A. Rettinger, and K. Kugler. Invbert: Text reconstruction from contextualized embeddings used for derived text formats of literary works. arXiv preprint arXiv:2109.10104, 2021.

[10] S. Ilić, E. Marrese-Taylor, J. A. Balazs, and Y. Matsuo. Deep contextualized word representations for detecting sarcasm and irony. arXiv preprint arXiv:1809.09795, 2018.

[11] J. D. M.-W. C. Kenton and L. K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT, pages 4171–4186, 2019.

[12] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942, 2019.

[13] K. Lang. Newsweeder: Learning to filter netnews. In in Proceedings of the 12th International Machine Learning Conference (ML95, 1995.

[14] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In International conference on machine learning, pages 1188–1196. PMLR, 2014.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettle-

doi:10.6342/NTU202302014

moyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[16] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. 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, June 2011. Association for Computational Linguistics.

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

[18] F. Meng, M. Richer, A. Tehrani, J. La, T. D. Kim, P. W. Ayers, and F. Heidar-Zadeh. Procrustes: A python library to find transformations that maximize the similarity between matrices. Computer Physics Communications, 276:108334, 2022.

[19] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models, 2016.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[21] X. Pan, M. Zhang, S. Ji, and M. Yang. Privacy risks of general-purpose language models. In 2020 IEEE Symposium on Security and Privacy (SP), pages 1314–1331. IEEE, 2020.

[22] E. Papagiannopoulou and G. Tsoumakas. A review of keyphrase extraction. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(2):e1339, 2020.

doi:10.6342/NTU202302014

[23] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security, pages 506–519, 2017.

[24] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

[26] N. Reimers and I. Gurevych. 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, 2019.

[27] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In 2017 IEEE symposium on security and privacy (SP), pages 3–18. IEEE, 2017.

[28] C. Song and A. Raghunathan. Information leakage in embedding models. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pages 377–390, 2020.

[29] C. Song and V. Shmatikov. Overlearning reveals sensitive attributes. arXiv preprint arXiv:1905.11742, 2019.

[30] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32, 2019.

doi:10.6342/NTU202302014

[31] X. Zhang, J. J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In NIPS, 2015.

[32] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. Ernie: Enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129, 2019.

[33] H. Zheng, Q. Ye, H. Hu, C. Fang, and J. Shi. Bdpl: A boundary differentially private layer against machine learning model extraction attacks. In Computer Security–ESORICS 2019: 24th European Symposium on Research in Computer Security, Luxembourg, September 23–27, 2019, Proceedings, Part I 24, pages 66–83. Springer, 2019.

doi:10.6342/NTU202302014

Appendix A — Dimension Reduction and Transformation

Table A.1: Comparison of different defense strategies.

	k = 5		**k = 10**		**k = 15**	
	Precision	NDCG	Precision	NDCG	Precision	NDCG
SADE	**0.6701**	**0.1918**	**0.5433**	**0.1905**	**0.4526**	**0.1891**
Random projection	0.6321	0.1518	0.5114	0.1526	0.4212	0.1516
PCA	0.6521	0.1623	0.5171	0.1609	0.4282	0.1600
Autoencoder	0.6689	0.1703	0.5336	0.1692	0.4383	0.1677

In this section, we aim to assess the impact of transformation on attack performance in various defense approaches, aside from just dimensional reduction. To achieve this, we perform experiments on the 20news dataset. We ensure that the dimensionality reduction is set to match the dimension of the original embedding. This allows us to isolate the effect of transformation and observe its influence independently.

The result is shown in Table A.1. Our investigation yields two findings. Firstly, we validate that defense transformations have a noticeable influence on the performance of attacks, as evidenced by the observed performance drop in different defense approaches, even when their embedding dimensions match those of the original embeddings. Secondly, we observe that different transformation methods produce distinct effects on attack performance. Notably, we find that more complex dimension reduction methods, which potentially incorporate a larger amount of information, lead to better decoding per-

doi:10.6342/NTU202302014

formance.

doi:10.6342/NTU202302014

Appendix B — Query Strategy Comparison

Table B.2: Comparison of different query strategies.

	k = 5		**k = 10**		**k = 15**	
	Precision	NDCG	Precision	NDCG	Precision	NDCG
Random	0.6233	0.1795	0.5154	0.1783	0.4315	0.1761
Max Sim	0.5353	0.1465	0.4346	0.1497	0.3698	0.1504
Kmeans	0.5640	**0.2008**	0.4924	**0.1978**	0.4255	**0.1956**
P-Kmeans	**0.6701**	0.1918	**0.5433**	0.1905	**0.4526**	0.1891

In this section, we compare the performance of four different query strategies. These strategies are as follows:

- Random: The adversary randomly selects documents to query the private encoder.
- Max Sim: The adversary selects the documents whose surrogate-generated embeddings have the highest similarity as candidates to query the private encoder.
- Kmeans: The adversary selects documents whose surrogate-generated embeddings have the highest similarity to the k-means centroids as candidates to query the private encoder. The key distinction from the strategy mentioned in Section 4.0.1 is the exclusion of the Procrustes transformation before computing the similarity.
- P-Kmeans: The adversary adopts the strategy mentioned in Section 4.0.1.

doi:10.6342/NTU202302014

The results in Table B.2 illustrate that random selection is a strong baseline, providing diverse candidates. However, a strategy solely based on similarity may lack diversity, as similar candidates may originate from the same cluster or group. The lower performance of the K-means strategy compared to P-Kmeans emphasizes the importance of using the Procrustes transformation to align embedding distribution spaces when selecting candidates with high similarity to centroids. These findings underscore the significance of selecting candidates with both high similarity and sufficient diversity in the distribution space while ensuring they belong to the same or similar distribution space for optimal performance.

doi:10.6342/NTU202302014

Appendix C — Impact of Diversity in Selected Documents

Table C.3: Impact of Different Clustering Centers (n) on Attack Performance. The number of clustering centers is 200, while n denotes the selected centers used to compute the cosine similarity between transformed surrogate embeddings. We vary n from 200, 150, 100, 50, 10, 5, to 1. The table presents the corresponding attack performance results for different values of n.

	k = 5		**k = 10**		**k = 15**	
n	Precision	NDCG	Precision	NDCG	Precision	NDCG
1	0.5613	0.1204	0.4396	0.1221	0.3618	0.1229
5	0.5491	0.1297	0.4341	0.1300	0.3636	0.1305
10	0.5843	0.1395	0.4671	0.1383	0.3837	0.1371
50	0.5935	0.1490	0.4778	0.1483	0.3979	0.1471
100	0.6164	0.1760	0.4981	0.1734	0.4157	0.1716
150	0.6578	0.1894	0.5352	0.1892	0.4514	0.1885
200	**0.6701**	**0.1918**	**0.5433**	**0.1905**	**0.4526**	**0.1891**

In this section, we explore the impact of private document diversity on attack performance using the query strategy described in 4.0.1. We vary the number of clustering centers used to compute the cosine similarity between them and the transformed surrogate embeddings. For instance, when there are 200 clustering centers (i.e., the query budget), we consider only 100 centers. As a result, the selected documents are confined to the distribution of these 100 centers, which becomes more concentrated and exhibits less diversity compared to considering all 200 centers.

doi:10.6342/NTU202302014

Table C.3 illustrates that lower attack performance is observed when the number of clustering centers decreases. This suggests that selecting documents from a less diverse distribution leads to weaker attack performance. Conversely, when documents are selected from a broader distribution, higher attack performance is achieved.

doi:10.6342/NTU202302014

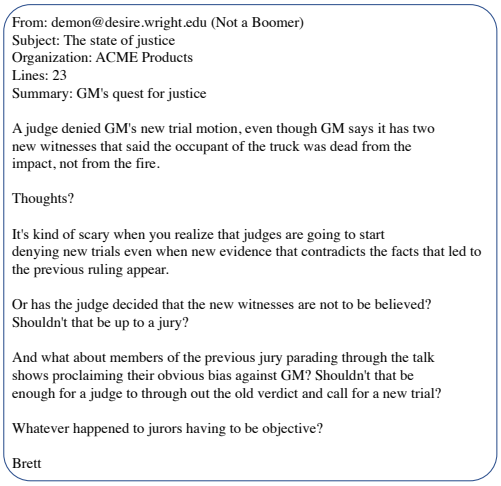
Appendix D — Case Study

Figure D.1 depicts the examples of attack output, with each column representing the prediction of the corresponding attack model. These examples utilize TF-IDF as the retrieval target and SBERT [26] as the private encoder.

Additionally, we investigate the effectiveness of the attack on practical documents with customized retrieval targets. To achieve this, we perform the embedding inversion attack on clinical record documents[1] and customize the retrieval target by setting the vocabulary to include medical keywords. We apply the output score of KeyBERT [7] as the importance score. The attack results on practical clinical records are illustrated in Figure D.2.

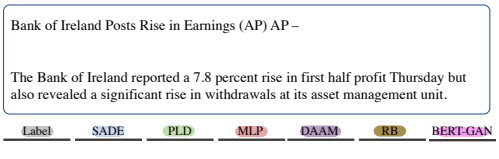
[1] https://huggingface.co/datasets/Elfsong/ClinicalDataset

doi:10.6342/NTU202302014



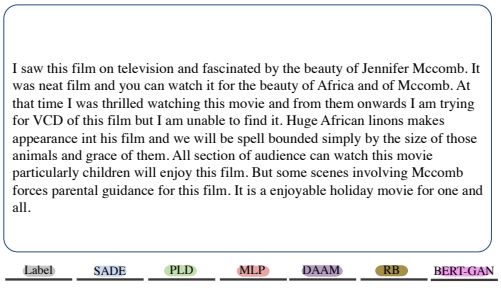
Label	SADE	PLD	MLP	DAAM	RB	BERT-GAN
gm	wright	reply	wins	wright	quadra	per
judge	witnesses	car	cray	contains	milwaukee	democracy
witnesses	trial	anyone	honest	causing	min	upenn
trial	desire	wins	fathers	party	minded	provides
justice	ruling	posting	spending	rate	task	press
ruling	judge	see	honda	coast	ecs	headed
denying	truck	accident	lance	mind	economy	useless
bias	denying	dave	mt	life	ece	guide
brett	demon	opinions	tactics	passing	minimal	mailing
wright	motion	everyone	reserved	winds	eastern	eternal

(a) The attack results on the 20News.



Label	SADE	PLD	MLP	DAAM	RB	BERT-GAN
ireland	irland	ireland	rates	district	retail	wing
rise	bank	rates	demand	rugby	defeat	de
bank	rise	effort	effort	death	decline	argued
revealed	management	charged	rise	bank	decided	back
posts	percent	calls	rose	power	know	line
management	profit	countries	profit	introduced	known	decision
unit	rose	rise	dollar	play	debut	published
earning	ap	going	ireland	nation	striker	freedom
reported	thursday	come	point	reason	debate	fun
profit	deal	blair	costs	causing	death	rates

(b) The attack results on the AG News.

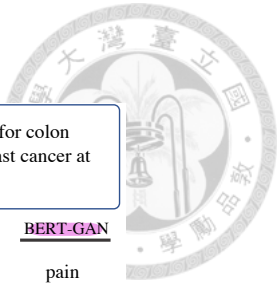


Label	SADE	PLD	MLP	DAAM	RB	BERT-GAN
beauty	film	saw	shock	home	grade	emotion
film	movie	movie	boyfriend	parliament	station	psychological
spell	love	mind	storyline	moved	grab	edit
holiday	story	get	question	jennifer	status	anyway
size	acting	scene	saw	residents	gothic	gameplay
section	made	film	role	house	gory	gripping
jennifer	jennifer	get	fashion	colonel	gorgeous	station
forces	recommend	please	honesty	move	staying	interview
animals	camera	lot	please	thomas	stays	young
grace	work	go	tea	see	goofy	president

(c) The attack results on the IMDB.

Figure D.1: Case Study on three different corpora

doi:10.6342/NTU202302014



Positive for heart disease, hypertension and cerebrovascular accidents. Family history is positive for colon cancer affecting her father and a brother. The patient has a daughter who was diagnosed with breast cancer at age 40.

Label	SADE	PLD	MLP	DAAM	RB	BERT-GAN
hypertension	disease	headache	spondylosis	diabetes	pulmonary	pain
cancer	cancer	symptoms	treatment	weight	allergy	headaches
accidents	family	cancer	surgery	tobacco	specialist	symptoms
heart	heart	medication	female	cancer	photophobia	disease
diagnosed	hypertension	surgery	hypertension	sister	appendectomy	male
daughter	dementia	diabetes	risk	heart	liver	blood
disease	patient	history	xylocaine	lenalidomide	smoked	vomiting
patient	incontinence	hypertension	benefits	died	mammogram	potassium
colon	history	heart	corpectomy	complications	aspirin	incontinence
family	prostate	brain	ankle	alcohol	glass	nausea

(a)

His father died from breast cancer. He also had diabetes. He has a strong family history of diabetes. His mother is 89. He has a sister with diabetes. He is unaware of any family members with neurological disorders.

Label	SADE	PLD	MLP	DAAM	RB	BERT-GAN
diabetes	diabetes	cancer	symptoms	diabetes	pulmonary	disease
family	family	patient	diabetes	foot	allergy	history
died	disease	symptoms	patient	orthopedics	specialist	dementia
disorders	cancer	diabetes	weight	cellulitis	photophobia	parents
cancer	acupuncture	deformity	time	patient	appendectomy	anemia
breast	heart	tenderness	cholesterol	hospital	liver	constipation
history	mother	vomiting	reflux	itching	smoked	diabetes
members	breast	female	glucose	extremity	mammogram	vomiting
sister	siblings	surgery	home	debridement	aspirin	told
mother	sister	disfigurement	bronchitis	pain	glass	smoke

(b)

She has had a hysterectomy, salpingoophorectomy, appendectomy, tonsillectomy, two carpal tunnel releases. She also has had a parathyroidectomy but still has had some borderline elevated calcium. Also, hypertension, hyperlipidemia, as well as diabetes. She also has osteoporosis.

Label	SADE	PLD	MLP	DAAM	RB	BERT-GAN
tonsillectomy	parathyroidectomy	drug	symptoms	diabetes	pulmonary	patient
releases	hypertension	diabetes	hypertension	ms	allergy	weeks
parathyroidectomy	osteoporosis	hypertension	female	surgery	specialist	medications
calcium	headache	family	dysarthria	symptoms	photophobia	home
borderline	asthma	heart	weakness	patient	appendectomy	presents
osteoporosis	diabetes	parathyroidectomy	pain	ambulation	liver	history
tunnel	tonsillectomy	tunnel	edema	radiculopathy	smoked	chills
hyperlipidemia	disease	cancer	arm	cuff	mammogram	today
diabetes	symptoms	history	presented	time	aspirin	headache
hypertension	thromboembolism	lung	cva	dysfunction	glass	removed

(c)

Figure D.2: Case Study on clinical records

doi:10.6342/NTU202302014