

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/353191026>

# Dataset Reconstruction Attack against Language Models

Conference Paper · July 2021

CITATIONS

0

READS

78

2 authors, including:



[Rrubaa Panchendrarajan](#)

Sri Lanka Institute of Information Technology

10 PUBLICATIONS 66 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Eatery: A Multi-Aspect Restaurant Rating System [View project](#)



Emotion Analysis in Microblogs related to Social Event [View project](#)

# Dataset Reconstruction Attack against Language Models

Rrubaa Panchendrarajan<sup>1,2</sup>, Suman Bhoi<sup>1,2</sup>

<sup>1</sup>*School of Computing, National University of Singapore*

<sup>2</sup>*Both authors have equal contribution.*

## Abstract

With the advances of deep learning techniques in Natural Language Processing, the last few years have witnessed releases of powerful language models such as BERT and GPT-2. However, applying these general-purpose language models to domain-specific applications requires further fine-tuning using domain-specific private data. Since private data is mostly confidential, information that can be extracted by an adversary with access to the models can lead to serious privacy risks. The majority of privacy attacks on language models infer either targeted information or a few instances from the training dataset. However, inferring the whole training dataset has not been explored in depth which poses far greater risks than disclosure of some instances or partial information of the training data. In this work, we propose a novel data reconstruction attack that also infers the informative words present in the private dataset. Experiment results show that an adversary with black-box query access to a fine-tuned language model can infer the informative words with an accuracy of about 75% and can reconstruct nearly 46.67% of the sentences in the private dataset.

## Keywords

Language Models, Dataset Reconstruction Attack, Information Leakage

## 1. Introduction

Language models (LMs) are fundamental to many natural language processing tasks, which assign probability values to a sequence of words, allowing one to make probabilistic predictions of the next word given preceding ones. The possible set of words known to the model is referred to as *vocabulary* of the model. Recent LMs such as BERT [1], BioBERT [2] and GPT-2 [3], are trained on massive text corpora generally collected from internet and are widely used in various downstream applications such as language translation [4], question answering [5] and search engines [6]. These high-capacity models are available publicly and are further fine-tuned on smaller private data to adapt to domain-specific applications, without requiring expensive re-training [7]. These private data are generally confidential and often contain sensitive information. For example, the models fine-tuned using hospital data for deployment in the medical domain may contain patients' demographic particulars as well as treatment history. If such information is extracted by an adversarial user of the model, then it would pose a severe privacy threat. Therefore, it is inevitable to analyze the privacy risks of these models.

---


*AIofAI'21: 1st Workshop on Adverse Impacts and Collateral Effects of Artificial Intelligence Technologies, Montreal, CA*

✉ [rrubaa@comp.nus.edu.sg](mailto:rrubaa@comp.nus.edu.sg) (R. Panchendrarajan); [sumanbhoi@u.nus.edu](mailto:sumanbhoi@u.nus.edu) (S. Bhoi)

🆔 0000-0002-1403-2236 (R. Panchendrarajan); 0000-0003-0460-9182 (S. Bhoi)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

Membership inference attack (MIA) introduced in [8] is one of the earlier lines of research in the field of privacy risk analysis of machine learning models. The attack infers whether a record is part of the data used to train a model. Any attack can be categorized into two scenarios based on the knowledge of the adversary. One is *white-box* scenario, where the model structure and parameters are known to the adversary. The other one is *black-box* scenario in which the adversary can only query the model and observe the output [9, 10]. Following [8], the MIA have been studied on generative models [11, 12] and LMs [13, 14, 15].

Recent works on privacy analysis of LMs focus on extracting information about the training data. [16, 17, 18] analyze the disclosure of targeted information such as rare sequences [16], sensitive terms [17] and patient information [18]. The works in [19, 20] perform quantitative analysis to determine whether a generated sentence is a member of the training data. Also, a common assumption made during this inference is that the attacker knows the vocabulary of the private dataset. We formulate our attack without this assumption since it might not hold true in many real-world applications (e.g. auto-completion in smartphones). Although these attacks infer information about the training data, inferring the whole training data has higher privacy risks and has not been explored in depth.

In this paper, we design a novel attack to reconstruct the entire private dataset used to fine-tune a LM while inferring top informative words. To the best of our knowledge, this is the first work that attempts reconstruction of the entire training data and informative words extraction. Our attack is executed in a black-box setting, so the adversary is not aware of any information about the updates of the LMs, which we refer to as *generic* and *fine-tuned* LMs. Similar to [21], we base our attack on the hypothesis that observing and comparing the output from two versions of the same model, trained on public and private datasets respectively, leaks information about the private dataset. Three attack scenarios are explored in this work with varying degrees of black-box access. This is based on whether the attacker is able to only observe the generated words or their probabilities as well. We use GPT-2 [3] as the LM which is trained and fine-tuned on datasets from the medical domain for the execution of the attack. Comprehensive experiments and analysis show that our attack is able to infer top informative words with an accuracy of about 75% and can reconstruct nearly 46.7% of the sentences in the dataset. Experiments includes a case-study which analyzes the informative sentences in the private dataset and their implications for data owners.

## 2. Related Works

Recent studies have demonstrated that machine learning models are vulnerable to several privacy attacks as they remember information about the training data. Membership inference attack (MIA) introduced by [8] shows that given black-box access to a classifier model, the confidence in model prediction can reveal whether a record belongs to the training data. [8] uses shadow training technique to train an attack model to distinguish between a member vs non-member of the training data. Later, MIA has been studied on generative models [11, 12] and LMs [13, 14, 15].

Prior works have attempted to analyze leakage of sensitive information about the training data of LMs. [16] performs a quantitative analysis on the risk of unintended disclosure of rare

sequence of words in the training data. [17] performs a similar analysis of the exposure of single sensitive words on publicly available pre-trained LMs. [22] quantifies the toxic generation of pre-trained LMs and illustrates the necessity to reconsider the content used in LM pre-training to avoid toxicity in natural language generation. [18] designs an attack specific to clinical records to infer information of target individuals.

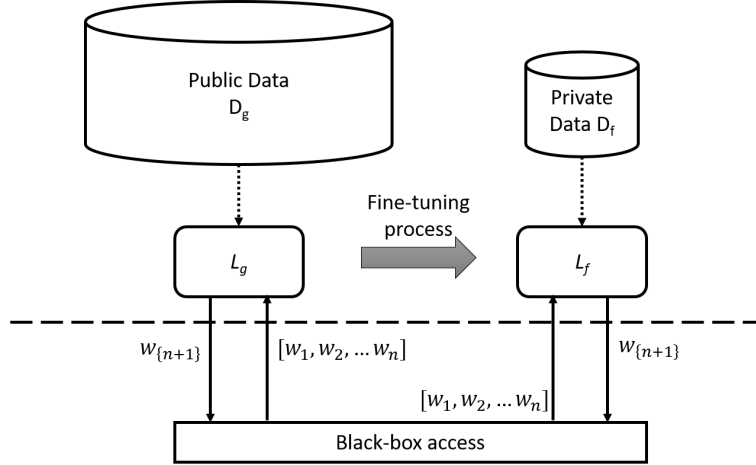
LMs are often used as embedding models to extract the embedding representation, low dimensional vector representation, of words for downstream tasks [23]. Works in [24, 23] design attack against word embedding models. [24] train an attack model against publicly available pre-trained LMs to predict the sensitive information, given embedding vector representation of a word sequence obtained using the LM. [23] performs a similar attack to infer the sensitive attributes of a word sequence using the embedding vector representation along with embedding inversion attack and MIA. They show that with this inversion attack they can recover 50% to 70% of a word sequence from its embedding vector.

Recent works [19, 20] have focused on training data extraction attack against LMs. [19] use perplexity of the generated sequences to choose the top 100 sequences as the extracted training data of publicly available pre-trained LMs. [20] is closest to our work, as it analyses the information leakage in fine-tuned LMs. It adapts the motivation brought in by [21], that the behavioral changes in the snapshots of machine learning models may leak information about the data used to update the model. [20] proposes *differential score* metric to capture the difference between probabilities assigned to a word sequence by public and fine-tuned models. Finally, they rank the sequences to conclude that word sequences with higher *differential score* generally belong to the private data used to update the LM. Both data extraction attacks [19, 20], assume that the attacker knows the vocabulary of the LM and the LM returns the probability distribution over the vocabulary for the prediction of the next word.

Our work differs from existing training data extraction attacks in that, our attack focuses on the reconstruction of the entire private dataset used to fine-tune the LM. In contrast, the existing works only focus on either extraction of targeted information or quantitative analysis of determining sentences that belong to the training data of the LM. Moreover, our attack accommodates the real-world scenario of widespread deployment of LMs to end-user systems, where the attacker does not have any information about the fine-tuned LM including the vocabulary of words present in the private dataset. This enables our threat model to infer informative words present in the private dataset along with the reconstruction of the private dataset.

### 3. General Attack Pipeline

While there are several pre-trained LMs available publicly, applying them in specific downstream tasks requires fine-tuning the model on task-specific datasets, which we refer to as *private dataset*. Our attack accommodates such a realistic scenario, where the adversary with black-box query access to a fine-tuned LM attempts to reconstruct the private dataset used for the fine-tuning process. This section presents the problem setting and the threat model of our attack.



**Figure 1:** Generic and Fine-tuned LMs with Black-box Access

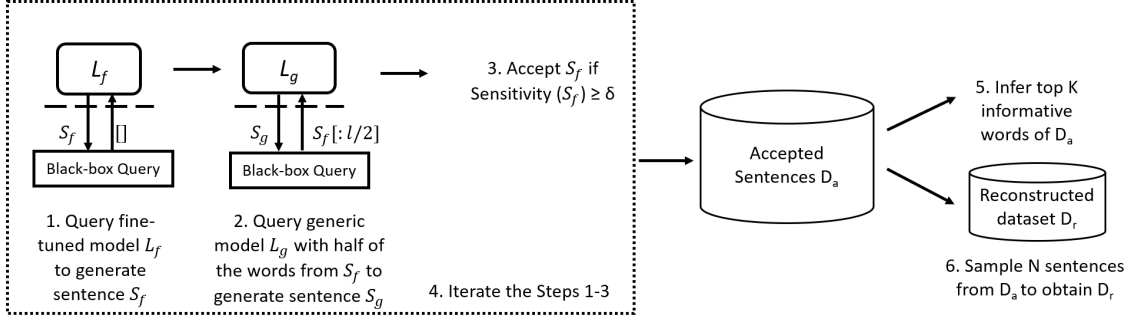
### 3.1. Problem Setting

A LM  $L_g$  is trained on a publicly available dataset  $D_g$  and then fine-tuned on a private dataset  $D_f$  to obtain a fine-tuned LM  $L_f$ . We refer to  $L_g$  and  $L_f$  as generic and fine-tuned LMs respectively. Here, the size of  $D_g$  is extremely large compared to  $D_f$ . Also,  $D_f$  may contain more specific information compared to  $D_g$ . A common vocabulary  $V$  is used to train both LMs, where  $V$  includes a limited number of words along with all the possible characters and a special token  $EOS$  indicating the end-of-sequence. This allows the LM to learn and generate any word as a combination of characters appearing in its vocabulary. Given a sequence of words,  $S = \{w_1, w_2, \dots, w_n\}$ , the generic and fine-tuned model return the next most probable word of the sequence. A complete sentence can be obtained by querying the LMs until  $EOS$  is predicted as the next possible word. Figure 1 illustrates our problem setting.

We design our attack on various scenarios of black-box query access to the generic and fine-tuned model as follows,

- **restrictive** - Given a sequence of input words  $S$ ,  $L_g$  and  $L_f$  return the next word of the sequence  $w_{n+1}$
- **relaxed** - Given  $S$ ,  $L_g$  and  $L_f$  return the next word of the sequence and its probability  $(w_{n+1}, p_{n+1})$
- **relaxed++** - Given  $S$ ,  $L_f$  returns the next word of the sequence and its probability  $(w_{n+1}, p_{n+1})$  while  $L_g$  returns list of probabilities  $P = \{p_1, p_2, \dots, p_n\}$  to generate the sequence  $S$ .

Generic LMs are usually trained using a large amount of publicly available training data (mostly scraped from the internet) and have fewer privacy risks compared to the fine-tuned LMs. Therefore, we include a more relaxed query access to the generic model in the **relaxed++** scenario.



**Figure 2:** Flow of the Dataset Reconstruction Attack in *restrictive* Scenario

**Table 1**

List of Notations

Notation	Description
$D_g$	Public dataset used to train a LM
$D_f$	Private dataset used to fine-tune a LM
$L_g$	Generic LM trained on $D_g$
$L_f$	Fine-tuned LM using $D_f$
$V$	Vocabulary of the LMs $L_g$ and $L_f$
$S_x$	A sentence generated using LM $L_x$
$S_x[i : j]$	Phrase obtained using $i^{th}$ word to $j^{th}$ word
$EOS$	Word token indicating the <i>end of sequence</i>
$P_x$	List of probabilities returned while generating $S_x$
$D_a$	List of accepted reconstructed sentences of $D_f$
$D_r$	Reconstructed dataset of $D_f$
$C(D_a, w)$	#sentences from $D_a$ containing $w$

### 3.2. Threat Model

We consider an adversary who has unlimited concurrent black-box query access to generic and fine-tuned models,  $L_g$  and  $L_f$ . The adversary can query these models to generate word sequences in three different settings, *restrictive*, *relaxed* and *relaxed++*. The adversary does not have any knowledge about the training datasets  $D_g$  and  $D_f$  including the words present and size of the dataset. However, similar to the real-world scenario, where the user is aware of the domain he or she is engaged in, we assume that the adversary is aware of the domain that the private dataset belongs to e.g. medical, financial, product. The goal of the adversary is to reconstruct a dataset  $D_r$  such that it can well represent  $D_f$  and infer the informative words present in the same.

## 4. Dataset Reconstruction Attack

Our dataset reconstruction attack exploits the observation from previous works [21, 20] that the behavioral change in snapshots of machine learning models can leak information about

the training data used to update the model. The objective of the adversary is to model such behavioral changes between the generic and fine-tuned models in order to identify sentences belonging to the private dataset. The adversary’s goal is to construct a representative dataset  $D_r$  of the private dataset by iteratively constructing sentences belonging to the private dataset. Figure 2 shows the flow of the attack in *restrictive* scenario and Table 1 lists key notations. The first two steps for querying the LMs differ with the change in problem setting. The following subsections explain our dataset reconstruction attack.

#### 4.1. Reconstruct a Sentence from Private Dataset

Our attack begins by querying the fine-tuned model  $L_f$  with an empty input sequence. A complete sentence  $S_f$  is constructed by iteratively querying  $L_f$  until it predicts *EOS* token as the next possible word. Due to the fine-tuning of the LM, the generation of sentence  $S_f$  would have resulted due to the words learned from either the public or private dataset or both. We refer to such important words that contribute to the generation of  $S_f$  as *context* of the sentence. Now the goal of the adversary is to determine the context origin of the sentence to either accept the sentence if the context origin is a private dataset or reject it otherwise. To determine the context origin of the sentence  $S_f$ , we exploit the fact that the generic and fine-tuned models may have a behavioral difference for the same input sequence due to the fine-tuning process. We quantify such behavioral differences using a novel metric called *Sensitivity*. *Sensitivity* of a sentence generated by a fine-tuned model for the three different problem settings is computed as follows.

**restrictive.** In *restrictive* scenario, we query the generic model with the word sequence  $S_f[: l/2]$ , where  $l$  indicates the length of the sentence  $S_f$  and  $S[p : q]$  indicates the partial sequence of  $S$  from  $p^{th}$  word to the  $q^{th}$  word. The partial sentence  $S_f[: l/2]$  would represent the partial context of the sentence  $S_f$ . The corresponding sentence  $S_g$  is constructed by querying the generic model with the input sequence  $S_f[: l/2]$  until the model predicts *EOS* as the next probable word. Both  $S_f$  and  $S_g$  share the same partial context of  $l/2$  number of words. Therefore, the information gap between the remaining sentences ( $S_f[l/2 : ]$  and  $S_g[l/2 : ]$ ) would reveal whether the context origin of  $S_f$  is public or private dataset. We develop a novel notion of *Sensitivity* score to quantify the information gap between  $S_f$  and  $S_g$ . *Sensitivity* of a sentence  $S_f$  is computed as a fraction of new words in the second half of  $S_f$  with respect to the word sequence predicted by the generic model given  $S_f[: l/2]$  as follows,

$$Sensitivity(S_f) = \frac{|S_f[l/2 : ] - S_g[l/2 : ]|}{|S_f[l/2 : ]|} \in [0, 1] \quad (1)$$

where  $|S|$  denotes the number of words in a sequence  $S$  and  $S_f - S_g$  denotes the word sequence obtained by removing the words appearing in  $S_g$  from  $S_f$ . A higher *Sensitivity* score indicates that there is a high chance that  $S_f$  is generated based on the context seen in the private dataset.

**relaxed.** In the *relaxed* scenario, both the fine-tuned and generic models return the next word and it’s probability, indicating the confidence of the model. We generate  $S_f$  and  $S_g$  similar to *restrictive* scenario along with their corresponding list of probabilities  $P_f$  and  $P_g$ . The *Sensitivity* of  $S_f$  in *relaxed* scenario is computed as a multiplication of fraction of new words present in

$S_f[l/2 : ]$  and the log-likelihood of the new words as follows,

$$Sensitivity(S_f) = \frac{|S_f[l/2 : ] - S_g[l/2 : ]|}{|S_f[l/2 : ]|} * \log\left(\prod_{w \in S_f[l/2 : ], w \notin S_g[l/2 : ]} P_f(w)\right) \in [-\infty, 0] \quad (2)$$

**relaxed++**. In the final problem setting of our attack, the adversary can query the generic model with a word sequence to obtain a list of probabilities. Hence, once we generate  $S_f$  and  $P_f$  using the fine-tuned model and we query the generic model with  $S_f$  to obtain  $P_g$ . In this problem setting, the *Sensitivity* of  $S_f$  can be computed as a log likelihood ratio of  $P_f$  and  $P_g$  as follows,

$$Sensitivity(S_f) = \log\left(\prod_{w \in S_f} \frac{P_f(w)}{P_g(w)}\right) \in [-\infty, +\infty] \quad (3)$$

In all three problem settings,  $S_f$  is accepted if it's *Sensitivity* score is higher than a threshold  $\delta$  and rejected otherwise. By iteratively constructing a list of accepted sentences,  $D_a$  can be obtained for further inference of informative words and reconstruction of the private dataset. Note that  $D_a$  may contain duplicate or similar sentences as we accept all the sentence with *Sensitivity* above  $\delta$ .

## 4.2. Inference of Informative Words of Private Data

The private dataset may contain words that convey useful information about the dataset. Among such informative words, we attempt to infer top K informative words which can be revealed by the fine-tuned model. We quantify the information revealed by a word by defining a *word-info* (*WI*) score for each word present in  $D_a$ . We assume that a word appearing in most sensitive sentences are more informative, hence we define the *word-info* score using the *Sensitivity* of the sentences as follows,

$$WI(w) = \frac{\sum_i Sensitivity(D_{a_i})_{norm}}{C(D_a, w)} * \Theta_{D_a}(w) \quad (4)$$

where,  $D_{a_i}$  indicates the  $i^{th}$  sentence in  $D_a$ ,  $Sensitivity(S)_{norm}$  indicates normalized *Sensitivity* score of a sentence,  $C(D_a, w)$  indicates the count function to obtain the number of sentences from  $D_a$  containing the word  $w$  and  $\Theta_{D_a}$  indicates the word distribution of  $D_a$ . We choose top K words with highest *word-info* score as the most informative words of the private dataset.

## 4.3. Reconstruction of Private Data.

The accepted sentences  $D_a$  may contain duplicate information as we iteratively query the fine-tuned model with an empty sequence. Moreover, larger the size of  $D_a$ , more accurate will be the informative words we infer. Therefore, it is necessary to carefully choose the representative sentences from  $D_a$  to reconstruct the private data. In order to choose the sentences, we use the existing technique called submodular optimization [25] which greedily chooses the content sequentially to maximize an objective function. In addition to the *Sensitivity* score of a sentence, we define the following criteria to develop the final objective function to choose sentences for the reconstruction.



**Table 2**  
Statistics of the Datasets

	<b>Public</b>	<b>Private</b>	<b>Shadow</b>
<b>Dataset</b>	PubMed	MIMIC-III	Kaggle
<b>#Sentences</b>	1445679	7452	10000
<b>#Unique Words</b>	209099	6098	21585
<b>Avg. Sentence Length</b>	22.12	10.48	23.18

*Coverage* - Let the function  $\Theta$  denotes word distribution of a sentence or a collection of sentences, defined over a set of words, then the *Coverage* of  $S_f$  is computed as follows,

$$Coverage(S_f) = 1 - Divergence(\Theta_{D_r \cup S_f} || \Theta_{D_a})_{norm} \quad (5)$$

where  $D_r \cup S_f$  indicates the state of the reconstructed dataset  $D_r$  after adding the sentence  $S_f$  to the dataset.  $Divergence_{norm}$  indicates normalized value of Kullback-Leibler (KL) divergence [26] between two distributions  $\Theta_1$  and  $\Theta_2$  as follows,

$$Divergence(\Theta_1 || \Theta_2) = \sum_w p(w|\Theta_1) \log \frac{p(w|\Theta_1)}{p(w|\Theta_2)} \quad (6)$$

A lower KL divergence score indicates a higher similarity between the two distributions  $\Theta_1$  and  $\Theta_2$ .

*Novelty* - *Novelty* of  $S_f$  indicates how much novel information it can add to  $D_r$  during the reconstruction process. We define *Novelty* of a sentence as follows,

$$Novelty(S_f) = Divergence(\Theta_{D_r} || \Theta_{S_f})_{norm} \quad (7)$$

Finally, the overall objective function to choose members of  $D_r$  is obtained as a weighted combination of *Sensitivity*, *Coverage* and *Novelty* as follows,

$$Score_r(S_f) = \omega_1 * Sensitivity(S_f)_{norm} + \omega_2 * Coverage(S_f) + \omega_3 * Novelty(S_f) \quad (8)$$

where  $\omega_1$ ,  $\omega_2$ , and  $\omega_3$  are the weights controlling the contribution of each criteria with  $\sum_i \omega_i = 1$ . Using the above objection function (Equation-8),  $N$  sentences from  $D_a$  are iteratively chosen to reconstruct the private data.

## 5. Experimental Set-up

In this section, we explain the experimental setup used for the execution of our attack. Our attack simulates the real-world scenario of the application of LMs for downstream tasks in specific domains. As such, we choose the medical domain for our experiment and attempt to reconstruct the patient notes. The following subsections explain the datasets we used, training, and fine-tuning of the LM, and parameter selection.

## 5.1. Dataset

We use PubMed [27] which has 200,000 abstracts of medical journals as the public dataset for training a LM. To obtain the fine-tuned LM, we use MIMIC-III [28] dataset as our private dataset which contains doctors’ notes pertaining to over 40,000 patients. Along with these two datasets, we use clinical notes published in Kaggle<sup>1</sup> as a shadow dataset to choose the *Sensitivity* threshold,  $\delta$ . The following subsections explain how an adversary can use a shadow dataset to determine  $\delta$ . Table 2 summarizes the statistics of all three datasets. We drop all the sentences with a word count less than 3 before the inference.

## 5.2. Model

We use GPT-2 [3] as our model, which is one of the largest LMs with 1.5 billion parameters. We train the GPT-2 model on the public dataset for 2000 epochs and then fine-tune it on the private data for 100 epochs to obtain the generic and fine-tuned LMs respectively.

**Table 3**  
Parameter Setting

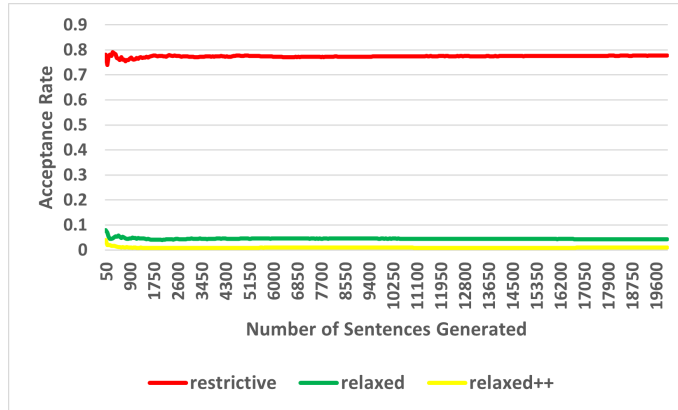
Parameter	Value
Training epochs - $L_g, L_f$	2000, 200
$\delta$ - <i>restrictive</i>	0.736
$\delta$ - <i>relaxed</i>	-10.225
$\delta$ - <i>relaxed++</i>	150.22
$\omega_1, \omega_2, \omega_3$	1/3

## 5.3. Parameter Setting

The key parameter of our reconstruction attack is the *Sensitivity* threshold  $\delta$ , which is used to determine whether we accept or reject a sentence as a member or non-member of the private dataset respectively. While the adversary can accept all the sentences ( $\delta = 0$ ) or choose an arbitrary value for  $\delta$ , carefully determining the value of  $\delta$  is necessary for an efficient and successful attack. In-order to determine  $\delta$  we execute the attack with a shadow dataset that is accessible to the adversary. Since the adversary is aware of the domain, we choose a shadow dataset from the same domain. We train a shadow model  $L_s$  by fine-tuning our generic model  $L_g$ .

We generate 10,000 sentences by querying the shadow model  $L_s$  and compute their *Sensitivity* scores in all three problem settings. Now the goal of the adversary is to determine whether the generated sentences are present in the shadow dataset. We use the ROUGE metric [29], which is a widely used metric to compute the similarity between two text sequences to determine the presence of a model-generated sentence in the shadow dataset. ROUGE metric is available in different variations and we use ROUGE-L for the similarity computation. We consider a generated sentence as a member of the shadow dataset if there exists a sentence in the shadow dataset with a similarity above 0.5. We compute the average *Sensitivity* scores of the successfully

<sup>1</sup><https://www.kaggle.com/rsnayak/hackathon-disease-extraction-saving-lives-with-ai>



**Figure 3:** Acceptance Rate of Generated Sentences in all Three Problem Settings

reconstructed sentences as the threshold  $\delta$ . Table 3 lists all the parameters of our experiment setup.

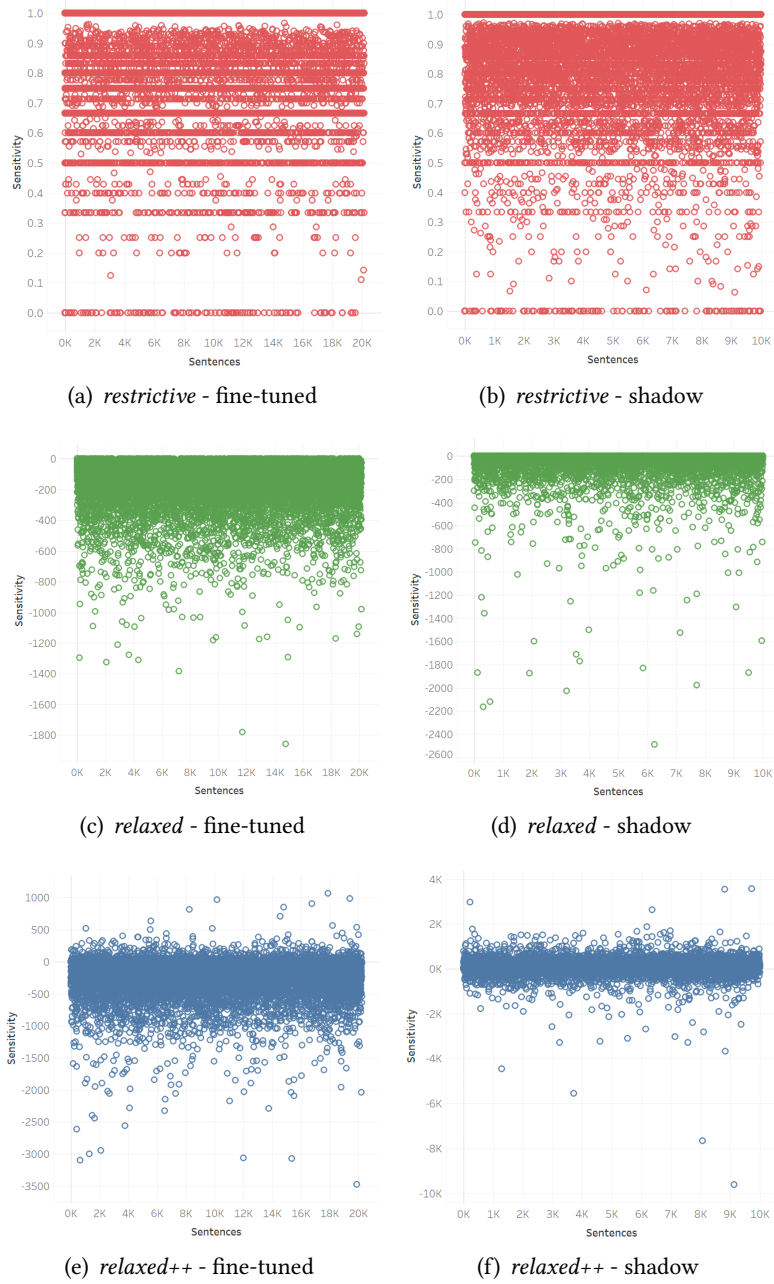
## 6. Results

In this section, we present the results of our experiments to evaluate the effectiveness of the proposed attack.

### 6.1. Acceptance of Generated Sentence

The rate of acceptance of sentences generated by the fine-tuned model over time is a good indicator of the probability of  $L_f$  generating  $S_f$  with the context from the private dataset. We generate 20,000 sentences in each problem setting and report the acceptance rate of generated sentences after every 50 sentence generation. Figure 3 shows the acceptance rate for the three different settings. We observe that the acceptance rate converges with time in all three settings. For example, in the *restrictive* scenario the acceptance rate converges to 0.777. This indicates that the fine-tuned model is generating sentences with the *Sensitivity* level of 0.736 ( $\delta$ ) from the private data with a probability of 0.777.

We can see that the acceptance rate in *relaxed* and *relaxed++* are almost close to zero. This could be mainly due to the different distribution of *Sensitivity* values obtained using the likelihood of the fine-tuned and shadow models. Figure 4 shows the *Sensitivity* distribution of generated sentences using both fine-tuned and shadow models. We observe that in the *restrictive* scenario, the *Sensitivity* distribution of sentences generated using the fine-tuned and shadow models are more similar and the majority of the sentences' *Sensitivity* value is above 0.7. However, for the other two settings, the *Sensitivity* distributions vary a lot for fine-tuned and shadow models. This results in a *Sensitivity* threshold value which does not well represent the sentences to be accepted while querying the fine-tuned model. Therefore, following the *restrictive* setting, we accept 77.7% of the sentences with the highest *Sensitivity* values in the



**Figure 4:** Sensitivity distribution of sentences generated using fine-tuned and shadow LMs

*relaxed* and *relaxed++* settings instead of using the threshold values obtained using the shadow training.

**Table 4**  
Performance of Vocabulary Inference

Setting	Vocabulary Size	Precision	Recall	F1-Score
<i>restrictive</i> ( $\delta = 0$ )	5919	0.549	0.533	0.541
<i>restrictive</i> ( $\delta = 0.736$ )	5321	0.579	0.505	0.54
<i>relaxed</i>	4285	0.634	0.446	0.5234
<i>relaxed++</i>	4309	0.633	0.447	0.524

## 6.2. Inference of Private Dataset Vocabulary

As the adversary is not aware of any word present in the private dataset,  $D_a$  reveals the possible set of words present in the private dataset. In this section, we analyze the percentage of words of the private dataset that has been successfully inferred by the adversary. We use NLTK<sup>2</sup> library for tokenizing the sentences and removing stopwords to obtain the final vocabulary of the private dataset and accepted sentences  $D_a$ . Table 4 shows the performance of the inference in all three settings. We can observe that the *restrictive* setting performs better compared to the other two and we were able to infer 50.5% of the vocabulary of the private dataset. Moreover, the smaller vocabulary of accepted sentences in the *relaxed* and *relaxed++* indicates that both settings accept sentences with similar words which result in smaller vocabulary, whereas *restrictive* setting accepts sentences with decisive information which results in larger vocabulary.

We also report the performance for the *restrictive* setting when all the sentences generated are accepted ( $\delta = 0$ ). However, there is no significant increase in the F1-Score observed when all the sentences are accepted. This indicates that the *Sensitivity* value of the sentences enables the adversary to correctly classify the context origin of the sentences.



**Figure 5:** Accuracy of Top K Informative Words Inference

<sup>2</sup><https://www.nltk.org/>

### 6.3. Inference of Informative Words

Inference of informative words poses more privacy threat as a small amount of informative words can leak sensitive information about the dataset and may enable further targeted attacks. In this section, we analyze how accurately the informative words can be inferred as explained in section 4.2. As there is no ground truth, we use TF-IDF [30] score of the words present in the private dataset to obtain the ground truth informative words. TF-IDF is a widely used statistical measure in the field of information retrieval to determine how important a word is to a document in a collection of documents. Figure 5 shows the accuracy of up to top 1000 informative words inferred in all three problem settings. We observe that for  $K$  value above 450, the *restrictive* setting outperforms the other two in a small margin and we are able to infer the informative words with nearly 75% accuracy.

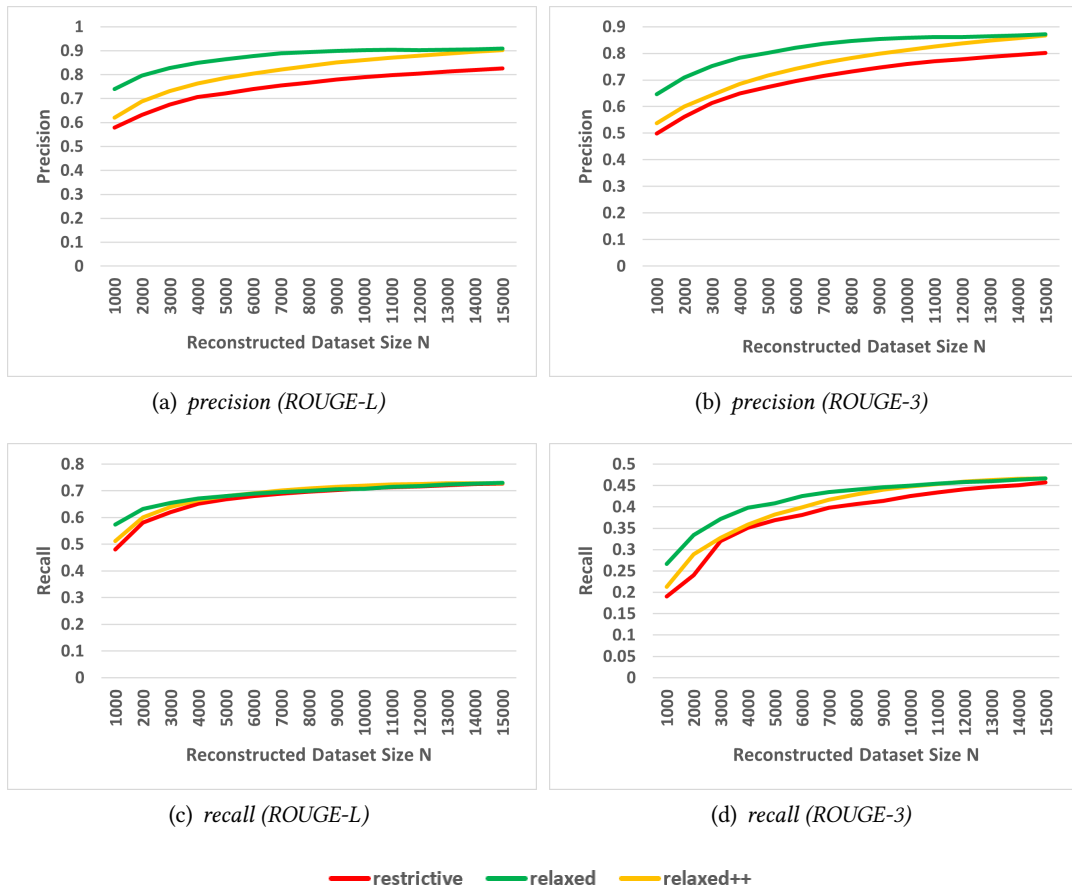


Figure 6: Reconstruction Performance for a Range of N

## 6.4. Dataset Reconstruction

We sample  $N$  number of sentences from the accepted sentences  $D_a$  to obtain the reconstructed dataset  $D_r$ . To evaluate the dataset reconstruction, we use ROUGE [29] score similar to the shadow training explained in 5.3. We use both ROUGE-L and ROUGE-3 scores with the threshold of  $\gamma_L = 0.5$  and  $\gamma_3 = 0.4$  respectively to determine whether a reconstructed sentence is present in the private dataset. Figure 6 shows the performance for a range of  $N$  value from 1000 to 15000 in a step size of 1000. We observe that unlike the performance of vocabulary and informative words inference, *relaxed* and *relaxed++* settings are obtaining a slightly higher precision compare to *restrictive* setting. This could be due to the *relaxed* and *relaxed++* settings generating similar sentences that are appearing in the private dataset, hence it results in relatively smaller vocabulary and higher precision.

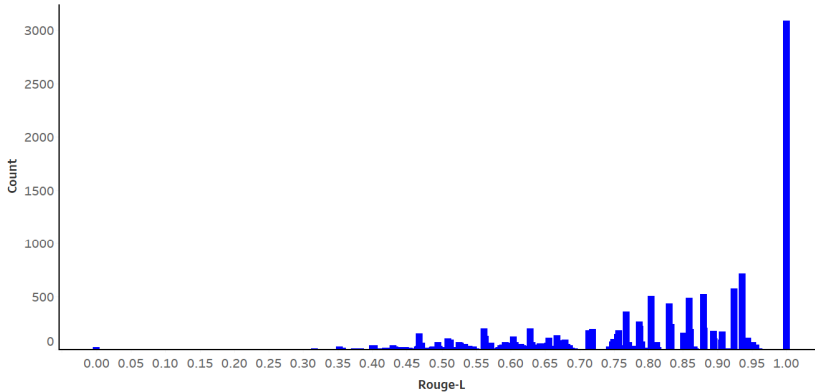


Figure 7: Distribution of Rouge-L Values of Accepted Sentences

Moreover, there is a significant drop in recall observed when the ROUGE version used for similarity computation is changed from ROUGE-L to ROUGE-3. This is mainly due to the shorter sentences appearing in the private dataset compared to the public or shadow datasets (refer Table 2), which result in higher ROUGE-L. However, even with ROUGE-3 metric, the recall obtained is 46.68% which indicates the adversary can reconstruct nearly 46.68% of the dataset without knowing any information of the model or private dataset, except the domain of the private dataset. Figure 7 shows the distribution of ROUGE-L values obtained for accepted sentences. We observe that majority of the sentences' ROUGE-L values are higher than 0.3 which indicates that most of the generated sentences contain nearly 30% of consecutive terms appearing in the private dataset. Moreover, 3090 sentences from  $D_a$  are the same (ROUGE-L = 1) as the sentences appearing in the private dataset which shows that the fine-tuned LMs pose a realistic threat, which needs to be considered during its deployment.

## 7. Case-Study

The informative words that are easily leaked by the model can be used either by the adversary for further targeted attacks or by the data owner for the execution of prevention mechanisms

**Table 5**

Sample informative sentences from the private data with informative words and their corresponding word-info score.

Top Informative Sentences	Informative Words	Word-info Score
The <b>patient</b> is a <b>NUM year old right</b> -handed african american man with past <b>medical history</b> of stroke in <b>date</b> with <b>right</b> arm weakness, <b>treated</b> at <b>doctor hospital</b> , with complete resolution and no residual symptoms, iddm, tobacco abuse, obesity, who <b>presented to hospital ed</b> on <b>date</b> at <b>NUM</b> as a code stroke.	date, ed, history, hospital, medical, NUM, old, patient, presented, right, treated, year.	0.00558, 0.00342, 0.02613, 0.00939, 0.00482, 0.06515, 0.00484, 0.10477, 0.00501, 0.00690, 0.00287, 0.00447
<b>History</b> of present illness: <b>patient</b> is an <b>NUM</b> who is a retired former gm worker who has <b>cardiac history</b> that dates back to <b>information</b> when he was <b>admitted</b> to an outside <b>hospital</b> with chest <b>pain</b> , shortness of breath and diaphoresis after shoveling snow at which time ecg was consistent with an <b>acute</b> inferior mi with a ck peaking at <b>NUM</b> with <b>NUM</b> mb.	acute, admitted, cardiac, history, hospital, information, NUM, pain, patient.	0.00673, 0.00351, 0.00393, 0.02613, 0.00939, 0.00417, 0.06515, 0.00472, 0.10477

to minimize the information leakage. In this case study, we determine the most informative sentences of the private dataset which are more vulnerable to the targeted attacks due to the presence of highly informative words. Determining the informative sentences will enable the data owner to come up with prevention mechanisms such as masking sensitive terms appearing in the highly informative sentences or removing the entire sentence to minimize the information leakage. To determine the highly informative sentences, we compute *Info-Leak* (IL) score for each sentence present in the private dataset based on the Word-info (WI) score (Equation 4). Let  $I_k$  be the top  $K$  informative words inferred and  $D_{f_i} \cap I_k$  denotes the top  $K$  informative words present in the  $i^{th}$  sentence of the private dataset  $D_f$ . Then the *Info-Leak* score of a sentence  $D_{f_i}$  is given by,

$$IL(D_{f_i}) = \sum_{w \in D_{f_i} \cap I_k} WI(w) \quad (9)$$

Table 5 portrays two highly informative sentences detected in the *restrictive* setting for  $K$  value of 50 along with the informative words and their WI score. Here, *NUM* is the token used to mask numerical values in the private dataset. We observe that both sentences are more informative as they contain details regarding the patient history or patient description. For example, the first sentence describes a patient along with the medical conditions. Although the age, hospital name and date are masked out, the detailed description about the patient (right-handed african american man) with the detailed medical diagnosis (stroke with right arm weakness, etc) leads to information leakage. This is especially useful for targeted attacks where an adversary is aiming to gain access to the medical details of a particular person.



## 8. Conclusion

In this study, we design a novel attack, i.e., dataset reconstruction attack and inference of informative words against LMs, to demonstrate the possible information leakage from fine-tuned LM with black-box query access. We have shown that the behavioral difference between LM updates can reveal the context origin of a word sequence generated. We quantify this behavioral difference using a novel metric, *Sensitivity* for three different problem settings. Our experiment shows, we are able to successfully infer 50.5% of the vocabulary of the private dataset and the informative words can be predicted with an accuracy of nearly 75%. Moreover, our experiments validate that only with the black-box query access to a fine-tuned LM, an adversary can reconstruct the private dataset with sentences that are similar to 46.67% of the sentences available in the private data. Future works include reconstruction attack with white-box query access to LMs and targeted attack using the informative words inferred.

## References

- [1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
- [2] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, *Bioinformatics* 36 (2020) 1234–1240.
- [3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, *OpenAI blog* 1 (2019) 9.
- [4] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual bert?, arXiv preprint arXiv:1906.01502 (2019).
- [5] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, I. Androutsopoulos, Large-scale multi-label text classification on eu legislation, arXiv preprint arXiv:1906.02192 (2019).
- [6] P. Nayak, Understanding searches better than ever before, 2019.
- [7] O. Giles, A. Karlsson, S. Masiala, S. White, G. Cesareni, L. Perfetto, J. Mullen, M. Hughes, L. Harland, J. Malone, Optimising biomedical relationship extraction with biobert: Best practices for data creation, *bioRxiv* (2020).
- [8] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: *SP, IEEE*, 2017, pp. 3–18.
- [9] M. Rigaki, S. Garcia, A survey of privacy attacks in machine learning, arXiv preprint arXiv:2007.07646 (2020).
- [10] M. Nasr, R. Shokri, A. Houmansadr, Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning, in: *SP, IEEE*, 2019, pp. 739–753.
- [11] B. Hilprecht, M. Härterich, D. Bernau, Monte carlo and reconstruction membership inference attacks against generative models, *PoPETS 2019* (2019) 232–249.
- [12] J. Hayes, L. Melis, G. Danezis, E. De Cristofaro, Logan: Membership inference attacks against generative models, *PoPETS 2019* (2019) 133–152.

- [13] C. Song, V. Shmatikov, Auditing data provenance in text-generation models, in: Proceedings of the 25th ACM SIGKDD, 2019, pp. 196–206.
- [14] S. Hisamoto, M. Post, K. Duh, Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system?, *TACL* 8 (2020) 49–63.
- [15] V. Misra, Black box attacks on transformer language models, 2019.
- [16] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, D. Song, The secret sharer: Measuring unintended neural network memorization & extracting secrets (2018).
- [17] A. Thomas, D. I. Adelani, A. Davody, A. Mogadala, D. Klakow, Investigating the impact of pre-trained word embeddings on memorization in neural networks, in: *TSD*, Springer, 2020, pp. 273–281.
- [18] Y. Nakamura, S. Hanaoka, Y. Nomura, N. Hayashi, O. Abe, S. Yada, S. Wakamiya, E. Aramaki, Kart: Privacy leakage framework of language models pre-trained with clinical records, *arXiv preprint arXiv:2101.00036* (2020).
- [19] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al., Extracting training data from large language models, *arXiv preprint arXiv:2012.07805* (2020).
- [20] S. Zanella-Béguelin, L. Wutschitz, S. Tople, V. Rühle, A. Paverd, O. Ohrimenko, B. Köpf, M. Brockschmidt, Analyzing information leakage of updates to natural language models, in: *ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 363–375.
- [21] A. Salem, A. Bhattacharya, M. Backes, M. Fritz, Y. Zhang, Updates-leak: Data set inference and reconstruction attacks in online learning, in: *{USENIX}*), 2020, pp. 1291–1308.
- [22] S. Gehman, S. Gururangan, M. Sap, Y. Choi, N. A. Smith, Realtoxicityprompts: Evaluating neural toxic degeneration in language models, *arXiv preprint arXiv:2009.11462* (2020).
- [23] C. Song, A. Raghunathan, Information leakage in embedding models, *arXiv preprint arXiv:2004.00053* (2020).
- [24] X. Pan, M. Zhang, S. Ji, M. Yang, Privacy risks of general-purpose language models, in: *SP*, IEEE, 2020, pp. 1314–1331.
- [25] G. L. Nemhauser, L. A. Wolsey, M. L. Fisher, An analysis of approximations for maximizing submodular set functions—i, *Mathematical programming* 14 (1978) 265–294.
- [26] S. Kullback, R. A. Leibler, On information and sufficiency, *The annals of mathematical statistics* 22 (1951) 79–86.
- [27] F. Dernoncourt, J. Y. Lee, Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts, *arXiv preprint arXiv:1710.06071* (2017).
- [28] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, R. G. Mark, Mimic-iii, a freely accessible critical care database, *Scientific data* 3 (2016) 1–9.
- [29] C.-Y. Lin, F. Och, Looking for a few good metrics: Rouge and its evaluation, in: *Ntcir Workshop*, 2004.
- [30] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, in: *Proceedings of the first instructional conference on machine learning*, volume 242, Citeseer, 2003, pp. 29–48.